Lecture Notes in Economics and Mathematical Systems 647

Jiuping Xu
Liming Yao

Random-Like Multiple Objective Decision Making

Springer

Lecture Notes in Economics and Mathematical Systems 647

For further volumes:
http://www.springer.com/series/300

Jiuping Xu • Liming Yao

Random-Like Multiple Objective Decision Making

Springer

Prof. Dr. Dr. Jiuping Xu
Uncertainty Decision Making Laboratory
Sichuan University
610064 Chengdu
P.R. China
xujiuping@scu.edu.cn

Assis. Prof. Liming Yao
Uncertainty Decision Making Laboratory
Sichuan University
610064 Chengdu
P.R. China
yaoliming1984@163.com

ISSN 0075-8442
ISBN 978-3-642-17999-0 e-ISBN 978-3-642-18000-2
DOI 10.1007/978-3-642-18000-2
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011924492

Cover design: WMXDesign GmbH, Heidelberg, Germany

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

In real-life world, decision makers usually face multiple objectives and need to make the decision in a state of uncertainty. How do we obtain the optimal strategy in uncertain environments? The purpose of the book is to provide random-like multiple objective decision making to solve the question. This book aims at discussing a class of uncertain phenomenon, that is, random-like uncertainty including random phenomenon, bi-random phenomenon, random fuzzy phenomenon and random rough phenomenon. Then random multiple objective decision making, bi-random multiple objective decision making, random fuzzy multiple objective decision making and random rough multiple objective decision making will be introduced one by one.

It was generally believed that the study of probability theory was started by Pascal and Fermat in 1654 when they succeeded in deriving the exact probabilities for certain gambling problems. From then on, people have broadly paid close attention to the random phenomenon. Afterwards, probability theory was widely applied to many social and technology problems, such as, vital statistics, premium theory, astro observation, the theory of errors, quality control and so on. From the seventeenth to nineteenth century, many distinguished scholars such as Bernoulli, De-Moivre, Laplace, Gauss, Poisson, Tchebychev, and Markov made contributions to the development of the probability theory. As the probability theory was applied to more and more real-life problems in many fields, the basic definition proposed by Laplace proved to be limiting, and unable to be used to deal with the usual random events. Great progress was achieved when Von Mises initialized the concept of sample space, and filled the gape between probability theory and measure theory in 1931. Strict theoretical principles, however, did not exist until 1933, when the outstanding mathematician Kolmogorov from former Soviet Union published the famous paper "The basic concept of probability theory", in which he put forward the axiomatization structure which is considered the milestone and foundation of the development of probability theory. Since then, probability theory has been the foundation axioms system and has been widely applied to many fields. Soon after, Zadeh and Pawlak initialized the fuzzy set theory in 1965 and rough set theory in 1982, respectively. Generally, fuzzy events are regarded as an uncertainty that people subjectively know for example, the temperature of water, 'cold' and 'hot', which isn't marked by a crisp number. Rough sets are generally regarded as a tool to distinguish something which is not easily discriminated. However, many scholars believed that

the traditional single-fold uncertain variables(random variables, fuzzy variables and rough variables) have some difficulties in clearly describing complicated, changeable realistic problems. In 1978, Kwakernaak combined randomness with fuzziness and initialized the concept of the fuzzy random variable, then introduced its basic definition and property. This viewpoint combining two different uncertain variables to describe complicated events received approval from many scholars and helped people move forward a further step in understanding uncertain events. Since then, the concept of the random fuzzy variable, bi-random variable and random rough variable have been proposed one by one by many scholars. Many papers and books about the two-fold uncertain theory were presented, which consequently promoted the development of two-fold uncertain theory. For example, the following description regarding the useful time of the spares in a factory proved the existence of a two-fold uncertain phenomenon. It might be known that the lifetime ξ of a modern engine is an exponentially distributed variable with an unknown mean $\tilde{\theta}$,

$$\phi(x) = \begin{cases} \frac{1}{\tilde{\theta}} e^{-x/\tilde{\theta}}, \text{ if } 0 \leq x < \infty \\ 0, \qquad \text{otherwise} \end{cases} \tag{1}$$

Generally, there is some relevant information in practice. It is thus possible to specify an interval in which the value of $\tilde{\theta}$ is likely to lie, or give an approximate estimation of the value of ξ. Here may be a random variable, or a fuzzy variable, or a rough variable, thus it is regarded as a random-like two-fold uncertain variable. Our research concentrates on random-like two-fold uncertain variables including bi-random variables, random fuzzy variables and random rough variables, and then deduces their properties and application to real-life world.

Multiobjective decision making problems are always a primary concern that many scholars pay attention to as it mainly provides decision makers with the help to find an optimal solution for many objectives with limited resources. Traditional multi-objective decision making usually consider problems with certain parameters. However, it is a usual phenomena that many decision making problems have abundant imprecise information. Hence, research into multiobjective decision making with random-like parameters is very necessary. To trace the origin of multi-objective decision making with certain parameters, we have to go to the eighteenth century. Franklin introduced how to coordinate multiple objectives in 1772. Cournot proposed the multi-objective decision making model from the standpoint of the economics in 1836. Pareto firstly presented an optimal solution to the multi-objective decision making model from the standpoint of the mathematics in 1896. The seeds of what is a strong branch of operations research can be traced to the early work of Kunh and Tucker and Koopmans in 1951. Later, Arrow proposed the concept of efficient points in 1953. Danzig claimed in the paper 'Review the origin of the linear programming' that stochastic programming would the most promising areas for future research in 1955. Then the single objective and multi-objective stochastic programming was widely researched and developed. As the fuzzy set theory was gradually perfected, it was rapidly and widely applied into the fileds of operations, management science, control theory and so on. In 1970, Bellman

and Zadeh collaborated to propose a fuzzy decision making model based on multi-objective programming. Similarly, after rough set theory was founded in 1982, it was also applied to decision making problems. From then, it became an area that attracted an enormous amount of attention because it is so useful for real-world decision making. The monographs of Chankong and Hamies, Cohon, Hwang and Masud, Osyczka, Sawaragi et al., and Steuer provide an extensive overview of the area of multiobjective optimization. Theory and methods for multiobjective optimization have been developed chiefly during the last century. Here we do not go into the history as the orgin, the achievements and development can be found in the literature. As the theory of two-fold uncertainty was rapidly developed, it was also applied to the multiobjective decision making in recent years. Above all, the multi-objective decision making model with uncertain parameters can be summarized as follows:

$$\begin{aligned} &\max[f_1(\boldsymbol{x}, \boldsymbol{\xi}), f_2(\boldsymbol{x}, \boldsymbol{\xi}), \ldots, f_m(\boldsymbol{x}, \boldsymbol{\xi})] \\ &\text{s.t. } \boldsymbol{x} \in X \end{aligned} \tag{2}$$

where $\boldsymbol{x} \in X \subset \mathbf{R}^n$ is an n-dimension decision variable, X is the constraint set, f_i is the objective function, and $\boldsymbol{\xi}$ is a random or fuzzy or rough variable or a two-fold uncertain variable.

In this book, real-life problems are considered as the background, and we present the random-like multi-objective decision making research. The basic theory, model and algorithm are proposed and applied to solve realistic problems. This book consists of 6 chapters. Chapter 1 reviews some preliminary knowledge such as measure theory, probability theory, central limit theorem and the Monte Carlo simulation and so on. Chapter 2 introduces the multi-objective decision making with random parameters and its application to DCs location problem. This chapter first reviews the literature of DCs location problem, then proposes the random multi-objective decision making model. Three sections introduce the random expected value model, the random chance-constrained model and the random dependent-chance model. In each section, we deduce the equivalent model of those problems in which the random parameters have crisp distribution and linear relationships. We also propose the Monte Carlo simulation-based simulated annealing algorithm to deal with those problems with random parameters and the nonlinear relationship. Finally, in the last section, the proposed models and algorithms have been applied to solve a realistic problem to show the efficiency of the proposed models and algorithms. Chapters 3, 4 and 5 have the same structure as Chap. 2. Chapter 3 proposes the multi-objective decision making model with bi-random parameters and introduces its application to flow shop scheduling problems. The equivalent models of the bi-random expected value model, bi-random chance-constrained model and bi-random dependent-chance model with crisp distribution are deduced and the interactive fuzzy programming technique is proposed to solve them. Then Ra-Ra simulation-based genetic algorithm is presented to solve the Ra-Ra multi-objective decision making problems with unknown distributions. Finally, the proposed models and algorithms are used to deal with a realistic flow

shop scheduling problem with bi-random parameters. Chapter 4 proposes the multi-objective decision making model with random fuzzy parameters and introduced its application to the supply chain problems. The equivalent models of the random fuzzy expected value model, random fuzzy chance-constrained model and random fuzzy dependent-chance model with crisp distribution are deduced and the interactive fuzzy programming technique is proposed to solve them. Then a random fuzzy simulation-based genetic algorithm is presented to solve the random fuzzy multi-objective decision making problems with unknown distribution. Finally, the proposed models and algorithms are used to deal with realistic supply chain problems with random fuzzy parameters. Chapter 5 proposes the multi-objective decision making model with random rough parameters and introduces its application to the inventory problems. The equivalent models of the random rough expected value model, random rough chance-constrained model and random rough dependent-chance model with crisp distributions are deduced and the interactive fuzzy programming technique is proposed to solve them. Then the random rough simulation-based genetic algorithm is presented to solve the random rough multi-objective decision making problems with unknown distribution. Finally, the proposed models and algorithms are used to deal with realistic inventory problems with random rough parameters. Chapter 6 gives the methodological system of the whole book including the motivation for researching random-like multi-objective decision making, physics-based model system, mathematical model system, model analysis system, algorithm systems, and research ideas and paradigm: 5MRP. Readers can understand the current research of random-like multiobjective decision making and prepare for future research.

Authors wish to thank the support of the National Science Foundation for Distinguished Young Scholars, P. R. China (Grant No. 70425005) and the National Natural Science Foundation of China (Grant No. 79760060, and No. 70171021). Authors also wish to give special acknowledgment to Prof. Dr. Dr. Günter Fandel and Prof. Walter Trockel for their helpful advice and their recommendation to Springer. Authors are also grateful to the senior editor of Economics/Management Science of Springer, Ms. Katharina Wetzel-Vandai, for her help in publication process. Authors would also like to take this opportunity to thank researchers from the Uncertainty Decision-Making Laboratory of Sichuan University, particularly, Z. Tao, X. Zhou, J. Li, Q. Liu, C. Ding, Y. He and so on, who have done much work in this field and made a number of corrections. This book has benefited from many references, and the authors would also like to thank all of these authors here. Finally, the authors would like to express their deep gratitude to the editorial staff of Spinger for their wonderful cooperation and helpful comments.

Sichuan University, *Jiuping Xu*
October 2010 *Liming Yao*

Contents

List of Tables

List of Figures

Acronyms

APSO	Adaptive particle swarm optimization algorithm
ASA	Adaptive simulated annealing algorithm
aw-GA	Adaptive weight-based genetic algorithm
CCM	Chance-constrained model
CCGM	Chance-constrained goal model
CEM	Chance-constrained model with expectation constraints
CLT	Central limit theorem
DC	Distribution center
DCGM	Dependent-chance goal model
DCM	Dependent-chance model
DEM	Dependent-chance model with expectation constraints
DM	Decision maker
ECM	Expectation model with chance constraints
ECTS	Enhanced continuous tabu search algorithm
EVGM	Expected value goal model
EVM	Expected value model
GA	Genetic algorithm
MODM	Multiple objective decision making
PSA	Parallel simulated annealing algorithm
PSO	Particle swarm optimization algorithm
PTS	Parallel tabu search algorithm
RLMODM	Random-like multiple objective decision making
Ra-Fu CCM	Bi-random chance-constrained model
Ra-Fu DCM	Bi-random dependent-chance model
Ra-Fu EVM	Bi-random expected value model
Ra-Ra CCM	Random random chance-constrained model
Ra-Ra DCM	Random random dependent-chance model
Ra-Ra EVM	Random random expected value model
Ra-Ro CCM	Random rough chance-constrained model
Ra-Ro DCM	Random rough dependent-chance model

rw-GA	Random weight-based genetic algorithm
Ra-Ro EVM	Random rough expected value model
RTS	Reactive tabu search algorithm
SA	Simulated annealing algorithm
SCN	Supply chain network
st-GA	Spanning tree-based genetic algorithm
Tribe-PSO	Tribe particle swarm optimization algorithm
TS	Tabu search algorithm
5MPR	5 kinds of relationship among research, model and problem

Chapter 1
Random Set Theory

The last decades witnessed a deeper thrust of science into the production of material goods. Thus, the successful management of research in such domains and the basic functions of management and scientific organization can be expressed almost entirely through quantitative methods. This explains the increasing use of mathematical methods in applied work, as well as the present efforts to improve them.

1.1 Algebra and σ-Algebra

This section is only the prerequisite for reading this book about some definitions by measure theory and Lebesgue integral. In this section we concentrate on measure, Borel set, measurable function, Lebesgue integral, measure continuity theorem, monotone convergence theorem and so on. Since these results are well-known, we only introduce results and readers can refer to correlative literatures such as [21, 26, 62, 95, 127, 128, 147, 175, 185, 223, 334].

Definition 1.1. (Srivastava [299]) Assume that Ω is a nonempty set. A collection $\mathscr{A}$ is called an algebra of subsets of Ω if the following conditions hold:

(a) $\Omega \in \mathscr{A}$
(b) if $A \in \mathscr{A}$, then $A^c \in \mathscr{A}$
(c) if $A_i \in \mathscr{A}$, for $i = 1, 2, \ldots$, then $\bigcup_{i=1}^{\infty} A_i \in \mathscr{A}$

If (c) is replaced with closure under countable union, then $\mathscr{A}$ is called a σ-algebra over Ω. Thus a σ-algebra on Ω is a family of subsets of Ω that contains Ω and is closed under complement, under the formation of countable unions, and under the formation of countable intersections. Apparently, each σ-algebra on Ω is an algebra on Ω. Let's consider the following examples.

Example 1.1. Assume that Ω is a finite nonempty set. Then the smallest σ-algebra over Ω is $\{\Phi, \Omega\}$, and the largest σ-algebra over Ω the power set $\mathscr{P}(\Omega)$which is combined with all subsets of Ω.

J. Xu and L. Yao, *Random-Like Multiple Objective Decision Making*, Lecture Notes in Economics and Mathematical Systems 647, DOI 10.1007/978-3-642-18000-2_1,

Example 1.2. Let B be a nonempty subset of X. Then the smallest σ-algebra including B is $\{\Phi, \Omega, B, B^c\}$.

Example 1.3. Assume that Ω is an infinite set, and $\mathscr{A}$ is combined with all finite subsets of Ω. Since $\Omega \notin \mathscr{A}$ and $\mathscr{A}$ is not closed under complement, it is not an algebra (or a σ-algebra) on Ω.

Example 1.4. Assume that Ω is an infinite set, and $\mathscr{A}$ is combined with all subsets B of Ω such that either B or B^c is finite. Then $\mathscr{A}$ is an algebra on Ω. But since it is not closed under the formation of countable unions, and so is not a σ-algebra.

Example 1.5. Let $\mathscr{A}$ be the collection of all subsets of R that are unions of finitely many intervals of the form $(a, b]$, $(a, +\infty)$, or $(-\infty, b]$. It is easy to check that each set that belongs to $\mathscr{A}$ is the union of a finite disjoint collection of intervals of the types listed above, and then to check that $\mathscr{A}$ is an algebra on $\mathbf{R}$ (the empty set belongs to $\mathscr{A}$, since it is the union of the empty, and hence finite, collection of intervals). The algebra $\mathscr{A}$ is not a σ-algebra; for example, the bounded open subintervals of $\mathbf{R}$ are unions of sequences of sets in $\mathscr{A}$, but do not themselves belong to $\mathscr{A}$.

Remark 1.1. Let Ω be a nonempty set. Then the intersection of an arbitrary nonempty collection of σ-algebras on Ω is a σ-algebra on Ω.

This remark implies the following result, which is an useful tool to construct a σ-algebra.

Remark 1.2. Assume that Ω is a nonempty set and $\mathscr{F}$ is a family of subsets of Ω. Then there is a smallest σ-algebra on Ω that includes $\mathscr{F}$.

Assume that $\mathbf{R}$ is combined with all real numbers, and $\mathbf{R}^n$ is the set of n-dimensional real vectors. Suppose that $\boldsymbol{a} = (a_1, a_2, \ldots, a_n)$ and $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ are vectors in $\mathbf{R}^n$ with $a_i < b_i$ for $i = 1, 2, \ldots, n$. Kelley and Srinivasan [155] defined the concepts of the open interval, closed interval, left-semiclosed interval and right-semiclosed. The open interval of $\mathbf{R}^n$ is defined as

$$(\boldsymbol{a}, \boldsymbol{b}) = \{(x_1, x_2, \ldots, x_n) | a_i < x_i < b_i, i = 1, 2, \ldots, n\} \tag{1.1}$$

The closed interval, left-semiclosed interval and right-semiclosed interval are defined as

$$[\boldsymbol{a}, \boldsymbol{b}] = \{(x_1, x_2, \ldots, x_n) | a_i \leq x_i \leq b_i, i = 1, 2, \ldots, n\} \tag{1.2}$$

$$[\boldsymbol{a}, \boldsymbol{b}) = \{(x_1, x_2, \ldots, x_n) | a_i \leq x_i < b_i, i = 1, 2, \ldots, n\} \tag{1.3}$$

$$(\boldsymbol{a}, \boldsymbol{b}] = \{(x_1, x_2, \ldots, x_n) | a_i < x_i \leq b_i, i = 1, 2, \ldots, n\} \tag{1.4}$$

Now let's use the preceding basic properties to define an important family of σ-algebra.

Definition 1.2. (Srivastava [299]) A Borel algebra on $\mathbf{R}^n$ is the σ-algebra on $\mathbf{R}^n$ generated by the collection of all open intervals of $\mathbf{R}^n$ and is denoted by $\mathscr{B}(\mathbf{R}^n)$. Any element in $\mathscr{B}(\mathbf{R}^n)$ is called a Borel set.

Especially, in case $n = 1$, one generally writes $\mathscr{B}(\mathbf{R})$ in place of $\mathscr{B}(\mathbf{R}^1)$. We may also replace the open intervals in Definition 1.2 with other classes of intervals, for example, closed intervals, left-semiclosed intervals , right-semiclosed intervals, or all intervals. It can be referred to the following two theorems.

Theorem 1.1. *(Cohn [62]) The σ-algebra $\mathscr{B}(\mathbf{R})$ of Borel subsets of $\mathbf{R}$ is generated by each of the following collections of sets:*

(a) The collection of all closed subsets of $\mathbf{R}$
(b) The collection of all closed subintervals of $\mathbf{R}$ of the form $(-\infty, b]$
(c) The collection of all subintervals of $\mathbf{R}$ of the form $(a, b]$

Theorem 1.2. *(Cohn [62]) The σ-algebra $\mathscr{B}(\mathbf{R}^n)$ of Borel subsets of $\mathbf{R}^n$ is generated by each of the following collection of sets:*

(a) The collection of all closed subsets of $\mathbf{R}^n$
(b) The collection of all closed half-spaces in $\mathbf{R}^n$ that have the form $\{(x_1, x_2, \ldots, x_n)|x_i \leq b_i, i = 1, 2, \ldots, n\}$ for some b_i in $\mathbf{R}$
(c) The collection of all rectangles in $\mathbf{R}^n$ that have the form $\{(x_1, x_2, \ldots, x_n)|a_i < x_i \leq b_i, i = 1, 2, \ldots, n\}$ for some a_i, b_i

Let's have a look at some examples generated by these sets.

Remark 1.3. For a set $O \subset \mathbf{R}^n$, if for any $x \in O$, there exists a small positive number δ such that $\{\boldsymbol{y} \in \mathbf{R}^n | ||\boldsymbol{y} - \boldsymbol{x}|| < \delta\} \subset O$, it is said to be open. The set Φ and $\mathbf{R}^n$ are open sets. Then each open set is a Borel set.

Remark 1.4. The complement of an open set is called a closed set. Each closed set is a Borel set.

Remark 1.5. The set of rational numbers, the set of irrational numbers, and countable set or real numbers are all Borel sets.

Next, we will introduce an important Borel set, that is, Cantor set.

Example 1.6. Make the interval $[0, 1]$ divided into three equal open intervals from which we throw off the middle open interval, i.e. $(\frac{1}{3}, \frac{2}{3})$. Then we go on dividing each of the remaining two intervals into three equal open intervals, and throw the middle open interval in each case, i.e. $(\frac{1}{9}, \frac{2}{9})$ and $(\frac{7}{9}, \frac{8}{9})$. We carry out this process and obtain a set C. Define the set

$$D = \bigcup_{i=1}^{\infty} \bigcup_{j=1}^{2^{i-1}} D_{ij} \tag{1.5}$$

where D_{ij} is a sequence of mutually disjoint open intervals and $D_{i1} < D_{i2} < \cdots < D_{i,2^{i-1}}$ for $j = 1, 2, \ldots, 2^{j-1}$ and $i = 1, 2, \ldots, \infty$. Then $C = [0, 1]$d is called the

Cantor set. In other words, $x \in C$ if and only if x can be expressed in ternary form using only digits 0 and 2, i.e.,

$$x = \sum_{i=1}^{\infty} \frac{a_i}{3^i} \tag{1.6}$$

where $a_i = 0$ or 2 for $i = 1, 2, \ldots, \infty$. The Cantor set is closed, uncountable, and a Borel set.

Theorem 1.3. *(Cohn [62]) Let Ω be a set, and let $\mathscr{A}$ be an algebra on Ω. Then $\mathscr{A}$ is a σ-algebra if either*

(a) $\mathscr{A}$ is closed under the formation of unions of increasing sequences of sets, or
(b) $\mathscr{A}$ is closed under the formation of intersections of decreasing sequences of sets.

1.2 Measurable Set and Measure Space

In this section, the concept of measurable set and measure space will be introduced and some properties will also be exhibited.

Definition 1.3. (Halmos [127]) Let Ω be a nonempty set , and $\mathscr{A}$ a σ-algebra over Ω. Then $(\Omega, \mathscr{A})$ is called a measurable space, and the sets in $\mathscr{A}$ are called measurable sets.

Definition 1.4. (Cohn [62]) Let Ω be a nonempty set , and $\mathscr{A}$ a σ-algebra over Ω. A function π whose domain is the σ-algebra $\mathscr{A}$ and whose values belong to the extended half-line $[0, +\infty)$ is said to be countably additive if it satisfies

$$\pi\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} \pi(A_i) \tag{1.7}$$

for each infinite sequence $\{A_i\}$ of disjoint sets that belong to $\mathscr{A}$.

Definition 1.5. (Cohn [62]) Let $(\Omega, \mathscr{A})$ be a measurable space. A function π is a measure if it satisfies

(a) $\pi(\Phi) = 0$.
(b) π is countably additive.

Another related concept should be noted since it is sometimes of interest. Assume that $\mathscr{A}$ is an algebra (not necessarily a σ-algebra) on the set Ω. A function π whose domain is $\mathscr{A}$ and whose values belong to $[0, +\infty)$ is finitely additive if it satisfies

$$\pi\left(\bigcup_{i=1}^{n} A_i\right) = \sum_{i=1}^{n} \pi(A_i) \tag{1.8}$$

for each finite sequence $A_1, A_2, \ldots, A_n$ of disjoint sets that belong to $\mathscr{A}$. A finitely additive measure on the algebra $\mathscr{A}$ is a function $\pi : \mathscr{A} \to [0, \infty)$ that satisfies $\pi(\Phi) = 0$ and is finitely additive.

It seems that finitely additivity is a more natural property than countable additivity. However countably additive measures on the one hand seem to be sufficient for almost all applications, and on the other hand support a much more powerful theory of integration than do finitely additive measures. We should emphasize that a measure will always be a countably additive measure. The expression "finitely additive measure" will always be written out in full.

Definition 1.6. (Cohn [62]) Let Ω be a nonempty set, and $\mathscr{A}$ a σ-algebra over Ω. If π is a measure on $\mathscr{A}$, then the triple $(\Omega, \mathscr{A}, \pi)$ is called a measure space.

Usually, if $(\Omega, \mathscr{A}, \pi)$ is a measure space, π is said to be a measure on measurable space $(\Omega, \mathscr{A})$, or if the σ-algebra $\mathscr{A}$ is clear from context, a measure on Ω. Let's turn to some examples.

Example 1.7. Let $(\Omega, \mathscr{A}, \pi)$ be a measure space, where π is defined as a function $\pi : \mathscr{A} \to [0, +\infty)$ by letting $\pi(A)$ be n if A is a finite set with n elements, and letting $\pi(A)$ be $+\infty$ if A is an infinite set. Then π is a measure, it is often called *counting measure* on $(\Omega, \mathscr{A})$.

Example 1.8. Let Ω be a nonempty set, $\mathscr{A}$ be a σ-algebra on Ω and x be an member of Ω. Define a function $\delta_x : \mathscr{A} \to [0, +\infty)$ as follows

$$\delta_x = \begin{cases} 1, \text{ if } x \in A \\ 0, \text{ if } x \notin A \end{cases}$$

Then δ_x is a measure, it is called a *point mass* concentrated at x.

Example 1.9. Consider the set $\mathbf{R}$ of all real numbers, and the σ-algebra $\mathscr{B}(\mathbf{R})$ of Borel subsets of $\mathbf{R}$. In the following part we shall construct a measure on $\mathscr{B}(\mathbf{R})$ that assigns to each subinterval of $\mathbf{R}$ its length. This measure is known as *Lebesgue measure*.

Next, an important result will be listed here with proof and interested readers can also consult the original book related to measure theory such as Halmos [127, 128] and so on.

Theorem 1.4. *(Halmos [128]) There is a unique measure π on the Borel algebra of $\mathbf{R}$ such that $\pi\{(a, b]\} = b - a$ for any interval $(a, b]$ of $\mathbf{R}$.*

Example 1.10. Let Ω be the set of all positive integers, and let $\mathscr{A}$ be the collection of all subsets A of Ω such that either A or A^c is finite. Then $\mathscr{A}$ is an algebra, but not a σ-algebra (see Example 1.12). Define a function $\pi : \mathscr{A} \to [0, +\infty)$ by

$$\pi(A) = \begin{cases} 1, \text{ if } A \text{ is infinite} \\ 0, \text{ if } A \text{ is finite} \end{cases}$$

It is easy to check that π is a finitely additive measure, however, it is impossible to extend π to a countably additive measure on the σ-algebra generated by $\mathscr{A}$ (if $A_k = k$) for each k, then $\pi(\bigcup_{k=1}^{\infty}) = \pi(\Omega) = 1$, while $\sum_{i=1}^{\infty} \pi(A_k) = 0$.

Example 1.11. Let Ω be an arbitrary set, and let $\mathscr{A}$ be an arbitrary σ-algebra on Ω. Define a function $\pi : \mathscr{A} \to [0, +\infty)$ by

$$\pi(A) = \begin{cases} +\infty, & \text{if } A \neq \Phi \\ 0, & \text{if } A = \Phi \end{cases}$$

Then π is a measure.

Example 1.12. Let Ω be a set that has at least two members, and let $\mathscr{A}$ be an σ-algebra consisting of all subsets of Ω. Define a function $\pi : \mathscr{A} \to [0, +\infty)$ by

$$\pi(A) = \begin{cases} 1, & \text{if } A \neq \Phi \\ 0, & \text{if } A = \Phi \end{cases}$$

Then π is not a measure, nor even a finitely additive measure, for if A_1 and A_2 are disjoint nonempty subsets of Ω, then $\pi(A_1 \bigcup A_2) = 1$, while $\pi(A_1) + \pi(A_2) = 2$.

The monotone class theorem, Carathéodory extension theorem, and approximation theorem will be listed here without proof. The interested reader may consult books related to measure theory such as Halmos [127, 128], Lang [185], Berberian [21], Jacobs [147] and so on.

Theorem 1.5. *(Monotone Class Theorem)(Srivastava [299]) Assume that $\mathscr{A}_0$ is an algebra over Ω, and $\mathscr{C}$ is a monotone class of subsets of Ω (if $\mathscr{A}_i \in \mathscr{C}$ and $A_i \uparrow A$ or $A_i \downarrow A$, then $A \in \mathscr{C}$). If $\mathscr{A}_0 \subset \mathscr{C}$ and $\sigma(\mathscr{A}_0)$ is the smallest σ-algebra containing $\mathscr{A}_0$, then $\sigma(\mathscr{A}_0) \subset \mathscr{C}$.*

Theorem 1.6. *(Carathéodory Extension Theorem)(Kelley and Srinivasan [155]) A σ-finite measure π on the algebra $\mathscr{A}_0$ has a unique extension to a measure on the smallest σ-algebra $\mathscr{A}$ containing $\mathscr{A}_0$.*

Theorem 1.7. *(Approximation Theorem)(Jacobs [147]) Let $(\Omega, \mathscr{A}, \pi)$ be a measure space, and let $\mathscr{A}_0$ be an algebra over Ω such that $\mathscr{A}$ is the smallest σ-algebra containing $\mathscr{A}_0$. If π is σ-finite and $A \in \mathscr{A}$ has finite measure, then for any given $\varepsilon > 0$, there exists a set $A_0 \in \mathscr{A}_0$ such that $\pi\{A/A_0\} < \varepsilon$.*

Theorem 1.8. *Let $(\Omega, \mathscr{A}, \pi)$ be a measure space, and let A and B be subsets of Ω that belong to $\mathscr{A}$ such that $A \subset B$. Then $\pi(A) \leq \pi(B)$. If $\pi(A) < +\infty$, then $\pi(B) = \pi(B - A) + \pi(A)$.*

Proof. Since $B = A \bigcup B - A$ satisfying that A and $B - A$ are disjoint. Since $A \subset B$, it means $\pi(B - A) \geq 0$. Thus it follows from the additivity of π that

$$\pi(B) = \pi(B - A) + \pi(A) \tag{1.9}$$

□

Let π be a measure on a measurable space $(\Omega, \mathscr{A})$. If $\pi(\Omega) < +\infty$, π is said to be a *finite* measure. Further it is said to be a σ-*finite* measure if Ω is the union of a sequence $A_1, A_2, \ldots$ of sets that belong to $\mathscr{A}$ and $\pi(A_i) < +\infty$ for each i. More generally, a set in $\mathscr{A}$ is σ-*finite* under π if it is the union of a sequence of sets that belong to $\mathscr{A}$ and have finite measure under π. The measure space $(\Omega, \mathscr{A}, \pi)$ is also called *finite* or σ-*finite* if π is finite or σ-finite. Most of the constructions and basic properties that we shall consider are valid for all measures. For a few important theorems, however, we shall need to assume that the measures involved are finite or σ-finite.

Theorem 1.9. *(Halmos [128]) Let $(\Omega, \mathscr{A}, \pi)$ be a measure space. If $\{A_k\}$Cohn [62] is an arbitrary sequence of sets that belong to $\mathscr{A}$, then*

$$\pi\left(\bigcup_{k=1}^{\infty} A_k\right) \leq \sum_{k=1}^{\infty} \pi(A_k) \tag{1.10}$$

Theorem 1.10. *(Cohn [62]) Let $(\Omega, \mathscr{A}, \pi)$ be a measure space.*

(a) If $\{A_k\}$ is an increasing sequence of sets that belong to $\mathscr{A}$, then $\pi(\bigcup_k A_k) = \lim_k \pi(A_k)$.
(b) If $\{A_k\}$ is an decreasing sequence of sets that belong to $\mathscr{A}$, and if $\pi(A_n) < +\infty$ holds for some n, then $\pi(\cap_k A_k) = \lim_k \pi(A_k)$.

Theorem 1.11. *(Cohn [62]) Let $(\Omega, \mathscr{A})$ be a measurable space, and let π be a finitely additive measure on $(\Omega, \mathscr{A})$. Then π is a measure if either*

(a) $\lim_k \pi(A_k) = \pi(\bigcup_k A_k)$ holds for each increasing sequence $\{A_k\}$ of sets that belong to $\mathscr{A}$.
(b) $\lim_k \pi(A_k) = 0$ holds for each decreasing sequence $\{A_k\}$ of sets that belong to $\mathscr{A}$ and satisfy $\cap_k A_k = \Phi$.

The preceding theorem will give the continuity of the measure π.

Theorem 1.12. *(Berberian [21]) Let $(\Omega, \mathscr{A}, \pi)$ be a measure space, and $A_1, A_2, \ldots, \in \mathscr{A}$.*

(a) If $\{A_i\}$ is an increasing sequence, then

$$\lim_{i\to\infty} \pi\{A_i\} = \pi\left\{\lim_{i\to\infty} A_i\right\} \tag{1.11}$$

(b) If $\{A_i\}$ is a decreasing sequence, and $\pi\{A_1\}$ is finite, then

$$\lim_{i\to\infty} \pi\{A_i\} = \pi\left\{\lim_{i\to\infty} A_i\right\} \tag{1.12}$$

Example 1.13. If $\pi\{A_i\}$ are not finite for any i, then the part (b) of Theorem 1.12 does not hold. For example, let $A_i = [i, +\infty)$ for $i = 1, 2, \ldots$, and let π be the length of intervals. Then $A_i \downarrow \Phi$ as $i \to \infty$. However, $\pi\{A_i\} \equiv +\infty \neq 0 = \pi\{\Phi\}$.

Let $\Omega_1, \Omega_2, \ldots, \Omega_n$ be any sets (not necessarily subsets of the same space). The product $\Omega = \Omega_1 \times \Omega_2 \times \ldots \times \Omega_n$ is the set of all ordered n-tuples of the form $(x_1, x_2, \ldots, x_n)$, where $\xi_i \in \Omega_i$ for $i = 1, 2, \ldots, n$.

Definition 1.7. (Evans and Gariepy [95]) Let $\mathscr{A}_i$ be σ-algebras over $\Omega_i, i = 1, 2, \ldots, n$, respectively. Write $\Omega = \Omega_1 \times \Omega_2 \times \cdots \times \Omega_n$. A measurable rectangle in Ω is a set $A = A_1 \times A_2 \times \cdots \times A_n$, where $A_i \in \mathscr{A}_i$ for $i = 1, 2, \ldots, n$. The smallest σ-algebra containing all measurable rectangles of Ω is called the product σ-algebra, denoted by $\mathscr{A} = \mathscr{A}_1 \times \mathscr{A}_2 \times \cdots \times \mathscr{A}_n$.

Note that the product σ-algebra $\mathscr{A}$ is the smallest σ-algebra containing measurable rectangles, rather than the product of $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n$.

Remark 1.6. Let $(\Omega_i, \mathscr{A}_i, \pi_i), i = 1, 2, \ldots, n$ be measure spaces. Assume that $\pi_i, i = 1, 2, \ldots, n$ are σ-finite, $\Omega = \Omega_1 \times \Omega_2 \times \cdots \times \Omega_n$, $\mathscr{A} = \mathscr{A}_1 \times \mathscr{A}_2 \times \cdots \times \mathscr{A}_n$. Then there is a unique measure π on $\mathscr{A}$ such that

$$\pi\{A_1 \times A_2 \times \cdots \times A_n\} = \pi_1\{A_1\} \times \pi_2\{A_2\} \times \cdots \times \pi_n\{A_n\} \tag{1.13}$$

for every measurable rectangle $A_1 \times A_2 \times \cdots \times A_n$. The measure π is called the product of $\pi_1, \pi_2, \ldots, \pi_n$, denoted by $\pi = \pi_1 \times \pi_2 \times \cdots \times \pi_n$. The triplet $(\Omega, \mathscr{A}, \pi)$ is called the product measure space.

If the sequence $(\Omega_i, \mathscr{A}_i, \pi_i), i = 1, 2, \ldots$, is an finite measure spaces such that $\pi_i(\Omega_i) = 1$ for $i = 1, 2, \ldots$. The product $\Omega = \Omega_1 \times \Omega_2 \times \cdots$ is defined as the set of all ordered tuples of the form $(x_1, x_2, \ldots)$, where $x_i \in \Omega_i$ for $i = 1, 2, \ldots$. For this case, we define a *measurable rectangle* as a set of the form $A_1 \times A_2 \times \cdots$, where $A_i \in \mathscr{A}_i$, where $A_i \in \mathscr{A}_i$ for all i and $A_i = \Omega_i$ for all but finitely many i. The smallest σ-algebra containing all measurable rectangles of Ω is called the product σ-algebra, denoted by $\mathscr{A}_1 \times \mathscr{A}_2 \times \cdots$.

Remark 1.7. Assume that $(\Omega_i, \mathscr{A}_i, \pi_i)$ are measure spaces such that $\pi_i\{\Omega_i\} = 1$ for $i = 1, 2, \ldots$. Let $\Omega = \Omega_1 \times \Omega_2 \times \cdots$ and $\mathscr{A}_1 \times \mathscr{A}_2 \times \cdots$. Then there is a unique measure π on $\mathscr{A}$ such that

$$\pi\{A_1 \times \cdots \times A_n \times \Omega_{n+1} \times \Omega_{n+2} \times \cdots\} = \pi_1\{A_1\} \times \pi_2\{A_2\} \times \cdots \tag{1.14}$$

for any measurable rectangle $A_1 \times \cdots \times A_n \times \Omega_{n+1} \times \Omega_{n+2} \times \cdots$ and all $n = 1, 2, \ldots$. The measure π is called the infinite product, denoted by $\pi = \pi_1 \times \pi_2 \times \cdots$. The triplet $(\Omega, \mathscr{A}, \pi)$ is called the infinite product measure space.

1.3 Outer Measures

In this section we develop one of the standard techniques for constructing measures, then we use it to construct Lebesgue measure on $\mathbf{R}^n$.

Definition 1.8. (Cohn [62]) Let Ω be a set, and let $\mathscr{P}(\Omega)$ be the collection of all subsets of Ω. An outer measure on Ω is a function $\pi^* : \mathscr{P}(\Omega) \to [0, +\infty)$ such that

(a) $\pi^*(\Phi) = 0$
(b) if $A \subset B \subset \Omega$, then $\pi^*(A) \leq \pi^*(A)$
(c) if $\{A_n\}$ is an infinite sequence of subsets of Ω, then

$$\pi^*(\cup_n A_n) \leq \sum_n \pi^*(A_n).$$

Thus an outer measure on Ω is a *monotone* and *countably subadditive* function from $\mathscr{P}(\Omega)$ to $[0, +\infty)$ whose value at Φ is 0.

Note that a measure can fail to be an outer measure, in fact, a measure on Ω is an outer measure if and only if its domain is $\mathscr{P}(\Omega)$ (see Theorem 1.8 and 1.9). On the other hand, an outer measure generally fails to be countably additive, and so fails to be a measure.

In Theorem 1.13 we shall prove that for each outer measure π^* on Ω there is a relatively natural σ-algebra Π_{π^*} on Ω for which the restriction of π^* to Π_{π^*} is countably additive, and hence a measure. Many important measures can be derived from outer measures in this way.

Let's turn to some examples.

Example 1.14. Let Ω be an arbitrary set, and define π^* on $\mathscr{P}(\Omega)$ by $\pi^*(A) = 0$ if $A = \Phi$, and $\pi^*(A) = 1$ otherwise. Then π^* is an outer measure.

Example 1.15. Let Ω be an arbitrary set, and define π^* on $\mathscr{P}(\Omega)$ by $\pi^*(A) = 0$ if A is countable, and $\pi^*(A) = 1$ if A is uncountable. Then π^* is an outer measure.

Example 1.16. Let Ω be an infinite set, and define π^* on $\mathscr{P}(\Omega)$ by $\pi^*(A) = 0$ if A is finite, and $\pi^*(A) = 1$ if A is infinite. Then π^* fails to be countably subadditive, and so is not an outer measure.

Example 1.17. *Lebesgue outer measure* on $\mathbf{R}$, which we shall denote by λ^*, is defined as follows. For each subset A of $\mathbf{R}$ let ϕ_A be the set of all infinite sequences $\{(a_i, b_i)\}$ of bounded open intervals such that $A \subset \cup_{i=1}^{\infty}(a_i, b_i)$. Then $\lambda^* : \mathscr{P}(\mathbf{R}) \to [0, +\infty)$ is defined by letting $\lambda^*(A)$ be the infimum of the set

$$\left\{ \sum_i (b_i - a_i) | \{(a_i, b_i)\} \in \phi_A \right\}.$$

Note that this set of sums is non-empty, and that the infimum of the set consisting of $+\infty$ alone is $+\infty$.

Lemma 1.1. *(Cohn [62]) Lebesgue outer measure on* $\mathbf{R}$ *is an outer measure, and it assigns to each subinterval of* $\mathbf{R}$ *its length.*

Example 1.18. Let us turn to Lebesgue outer measure on $\mathbf{R}^n$, which we shall denote by λ^* or, if necessary in order to avoid ambiguity, by λ_n^*. An *n-dimensional interval* is a subset of $\mathbf{R}^n$ of the form $I_1 \times \cdots \times I_n$, where $I_1, \ldots, I_n$ are subintervals of $\mathbf{R}$ and $I_1 \times \cdots \times I_n$ is given by

$$I_1 \times \cdots \times I_n = \{(x_1, \ldots, x_n) | x_i \in I_i, i = 1, 2, \ldots, n\}.$$

Note that the intervals $I_1, \ldots, I_n$, and hence the n-dimensional interval $I_1 \times \cdots \times I_n$, can be open, closed, or neither open nor closed. The *volume* of the lengths of the intervals $I_1, \ldots, I_n$, and will be denoted by $vol(I_1 \times \cdots \times I_n)$. For each subset A of $\mathbf{R}^n$ let ϕ_A be the set of all sequences $\{R_i\}$ of bounded and open n-dimensional intervals for which $A \subset \cup_{i=1}^{\infty} R_i$. Then $\lambda^*(A)$, the outer measure of A, is the infimum of the set

$$\left\{ \sum_{i=1}^{\infty} vol(R_i) | \{R_i\} \in \phi_A \right\}.$$

We note the following analogue of Lemma 1.1.

Lemma 1.2. *(Cohn [62]) Lebesgue outer measure on $\mathbf{R}^n$ is an outer measure, and it assigns to each n-dimensional interval its volume.*

Let Ω be a set, and let π^* be an outer measure on Ω. A subset B of Ω is *π^*-measure* (or *measurable with respect to π^**) if

$$\pi^*(A) = \pi^*(A \cap B) + \pi^*(A \cap B^c)$$

holds for each subset A of Ω. Thus a π^*-measurable subset of Ω is one that divides each subset of Ω in such a way that the sizes (as measured by π^*) of the pieces add properly.

Definition 1.9. (Cohn [62]) A *Lebesgue measurable* subset of $\mathbf{R}$ or $\mathbf{R}^n$ is of course one that is measurable with respect to Lebesgue outer measure.

Note that the subadditivity of the outer measure π^* implies that

$$\pi^*(A) \leq \pi^*(A \cap B) + \pi^*(A \cap B^c)$$

holds for all subsets A and B of Ω. Thus to check that a subset B of Ω is π^*-measurable we need only check that

$$\pi^*(A) \geq \pi^*(A \cap B) + \pi^*(A \cap B^c) \tag{1.15}$$

holds for each subset A of Ω. Note also that inequality (1.15) certainly holds if $\pi^*(A) = +\infty$. Thus the π^*-measurability of B can be verified by checking that (1.15) holds for each A that satisfies $\pi^*(A) < +\infty$.

Lemma 1.3. *(Cohn [62]) Let Ω be a set, and let π^* be an outer measure on Ω. Then each subset B of Ω that satisfies $\pi^*(B) = 0$ or that satisfies $\pi^*(B^c) = 0$ is π^*-measurable.*

It follows that the sets Φ and Ω are measurable for every outer measure on Ω.

The following theorem is very the fundamental fact about outer measures, it will be the key to many of out constructions of measures, then we listed the theorem and the proving process as follows. Interested readers can consult books related to the measure.

Theorem 1.13. *(Cohn [62]) Let Ω be a set, and let π^* be an outer measure on Ω, and let Π_{π^*} be the collection of all π^*-measurable subsets of Ω. Then*

(a) Π_{π^*} is a σ-algebra.
(b) The restriction of π^* to Π_{π^*} is a measure on Π_{π^*}.

Next, we first turn to some applications of Theorem 1.13, and then begin with Lebesgue measure and deduce some properties of it on $\mathbf{R}^n$.

Remark 1.8. Every Borel subset of $\mathbf{R}$ is Lebesgue measurable.

Remark 1.9. Every Borel subset of $\mathbf{R}^n$ is Lebesgue measurable.

Definition 1.10. (Cohn [62]) The restriction of Lebesgue outer measure on $\mathbf{R}$ (or on $\mathbf{R}^n$) to the collection Π_{π^*} of Lebesgue measurable subsets of $\mathbf{R}$ (or of $\mathbf{R}^n$) is called Lebesgue measure, and will be denoted by λ or by λ_n.

Lemma 1.4. *(Cohn [62]) Let A be a Lebesgue measurable subset of $\mathbf{R}^n$. Then*

(a) $\lambda(A) = \inf\{\lambda(U) | U \; is \; open \; and \; A \subset U\}$.
(b) $\lambda(A) = \sup\{\lambda(K) | K \; is \; compact \; and \; K \subset U\}$.

The following lemma will be needed for the proof of Theorem 1.4. In this lemma we shall be dealing with a certain collection of half-open cubes, namely with those that have the form

$$\{(x_1, \ldots, x_n) | j_i 2^{-k} \leq x_i < (j_i + 1)2^{-k}, i = 1, 2, \ldots, n\} \tag{1.16}$$

for some integers $j_1, \ldots, j_n$ and some positive integer k.

Lemma 1.5. *(Cohn [62]) Each open subset of $\mathbf{R}^n$ is the union of a countable disjoint collection of half-open cubes, each of which is of the form given in expression (1.16).*

Since the following theorem is very important for us to know about the Lebesgue measure, then we listed its original proof as follows.

Theorem 1.14. *(Cohn [62]) Lebesgue measure is the only measure on $(\mathbf{R}^n, \mathscr{B}(\mathbf{R}^n))$ that assigns to each n-dimensional interval, or even to each half-open cube of the form given in expression (1.16), its volume.*

For each element $\boldsymbol{x}$ and subset A of $\mathbf{R}^n$ we shall denote by $A + \boldsymbol{x}$ the subset of $\mathbf{R}^n$ defined by

$$A + \boldsymbol{x} = \{\boldsymbol{y} \in \mathbf{R}^n | \boldsymbol{y} = \boldsymbol{a} + \boldsymbol{x} \text{ for some } \boldsymbol{a} \text{ in } A\},$$

the set $A + \boldsymbol{x}$ is called the translate of A by $\boldsymbol{x}$. We turn to the invariance of Lebesgue measure under such translations.

1.4 Probability Space

The fundamental concepts of probability theory have rooted in measure theory. Like any branch of mathematics, probability theory has its own terminology and its own tools. The probability theory has been widely pushed forward by [11,26,91,97,167, 175, 184, 252, 288, 327] many scholars. In this section, we will introduce some of this terminology and study some basic concepts of probability theory.

Let Ω be a nonempty set, and $\mathscr{A}$ a σ-algebra over Ω. If Ω is countable, usually $\mathscr{A}$ is the power set of Ω. If Ω is uncountable, for example $\Omega = [0, 1]$, usually $\mathscr{A}$ is the Borel algebra of Ω. Each element in $\mathscr{A}$ is called an event. In order to present an axiomatic definition of probability, it is necessary to assign to each event A a number $Pr\{A\}$ which indicates the probability that A will occur. In order to ensure that the number $Pr\{A\}$ has certain mathematical properties which we intuitively expect a probability to have, the following three axioms must be satisfied [26]:

Axiom 1. (*Normality*) $Pr\{\Omega\} = 1$.
Axiom 2. (*Nonnegativity*) $Pr\{A\} \geq 0$ for any $A \in \mathscr{A}$.
Axiom 3. (*Countable Additivity*) For every countable sequence of mutually disjoint events $\{A_i\}$, we have

$$Pr\left\{\bigcup_{i=1}^{\infty} A_i\right\} = \sum_{i=1}^{\infty} Pr\{A_i\} \tag{1.17}$$

Definition 1.11. (Billingsley [26]) The set function Pr is called a probability measure if it satisfies the three axioms. And the triple $(\Omega, \mathscr{A}, Pr)$ is called a probability space.

We note that, if $A_n \in \mathscr{A}, n = 1, 2, \ldots,$ then A_n^c, $\bigcup_{n=1}^{\infty} A_n$, $\bigcap_{n=1}^{\infty} A_n$, $\liminf_{n\to+\infty} A_n$, $\limsup_{n\to+\infty} A_n$ and $\lim_{n\to+\infty} A_n$ (if exists) are events. Also, the probability measure Pr is defined on $\mathscr{A}$, and for all events A, A_n

$$Pr\{A\} \geq 0,\ Pr\left\{\bigcup_{i=1}^{\infty} A_i\right\} = \sum_{n=1}^{\infty} Pr\{A_n\} (A_n \text{ 's disjoint}),\ Pr\{\Omega\} = 1.$$

It follows that

$$Pr\{\Phi\} = 0,\ Pr\{A\} \le Pr\{B\} \text{ for } A \subset B,\ Pr\left\{\bigcup_{n=1}^{\infty} A_n\right\} \le \sum_{n=1}^{\infty} Pr(A_n).$$

Moreover,

$$Pr\left(\liminf_{n\to\infty} A_n\right) \le \liminf_{n\to\infty} Pr(A_n) \le \limsup_{n\to\infty} Pr(A_n) \le Pr\left(\limsup_{n\to\infty} A_n\right),$$

and if $\lim_{n\to\infty} A_n$ exists, then

$$Pr\left(\lim_{n\to\infty} A_n\right) = \lim_{n\to\infty} Pr(A_n) \tag{1.18}$$

The last result is known as the continuity property of probability measures.

Example 1.19. Let $\Omega = \{\omega_j : j \ge 1\}$, and let $\mathscr{A}$ be the σ-algebra of all subsets of Ω. Let $\{p_j : j \ge 1\}$ be any sequence of nonnegative real numbers satisfying $\sum_{j=1}^{\infty} p_j = 1$. Define Pr on $\mathscr{A}$ by

$$Pr\{E\} = \sum_{\omega\in E} p_j, \quad E \in \mathscr{A} \tag{1.19}$$

Then Pr defines a probability measure on $(\Omega, \mathscr{A})$, and $(\Omega, \mathscr{A}, Pr)$ is a probability space.

Example 1.20. Let $\Omega = (0, 1]$ and $\mathscr{A} = \mathscr{B}$ be the σ-algebra of Borel sets on Ω. Let λ be the Lebesgue measure on $\mathscr{B}$. Then $(\Omega, \mathscr{A}, \lambda)$ is a probability space.

Lemma 1.6. *(Krickeberg [175]) Let Ω be a nonempty set, $\mathscr{A}$ be a σ-algebra over Ω, and Pr a probability measure. Then we have*

(a) $Pr\{\Phi\} = 0$.
(b) Pr is self-dual, i.e., $Pr\{A\} + Pr\{A^c\} = 1$ *for any* $A \in \mathscr{A}$.
(c) Pr is increasing, i.e., $Pr\{A\} \le Pr\{B\}$ *whenever* $A \subset B$.
(d) $0 \le Pr\{A\} \le 1$ *for any* $A \in \mathscr{A}$.

The usual starting point in the construction of probability measure is that probabilities are assigned to a restricted class of sets. The probability extension theorem gives a method to construct probability measure.

Theorem 1.15. *(Probability Extension Theorem)(Blackwell [27]) Let Ω be a nonempty set, $\mathscr{A}_0$ be an algebra over Ω, and Pr a measure on $\mathscr{A}_0$ such that $Pr\{\Omega\} = 1$. Then Pr has a unique extension to a probability measure on the smallest σ-algebra $\mathscr{A}$ containing $\mathscr{A}_0$.*

1.5 Product Probability Space

Let $(\Omega_i, \mathscr{A}_i, Pr_i), i=1,2,\ldots,n$ be probability spaces, and $\Omega = \Omega_1 \times \Omega_2 \times \cdots \times \Omega_n$, $\mathscr{A} = \mathscr{A}_1 \times \mathscr{A}_2 \times \cdots \times \mathscr{A}_n$, $Pr = Pr_1 \times Pr_2 \times \cdots Pr_n$. Note that the probability measures $Pr_i, i = 1,2,\ldots,n$ are finite. It follows from the product measure theorem that there is a unique measure Pr on $\mathscr{A}$ such that

$$Pr\{A_1 \times A_2 \times \cdots \times A_n\} = Pr_1\{A_1\} \times Pr_2\{A_2\} \times \cdots \times Pr_n\{A_n\} \tag{1.20}$$

for any $A_i \in \mathscr{A}_i, i = 1,2,\ldots,n$. This conclusion is called the product probability theorem. The measure Pr is also a probability measure since

$$Pr\{\Omega\} = Pr_1\{\Omega_1\} \times Pr_2\{\Omega_2\} \times \cdots \times Pr_n\{\Omega_n\} = 1 \tag{1.21}$$

Such a probability measure is called the product probability measure, denoted by $Pr = Pr_1 \times Pr_2 \times \cdots \times Pr_n$.

Definition 1.12. (Krickeberg [175]) Let $(\Omega_i, \mathscr{A}_i, Pr_i), i = 1,2,\ldots,n$ be probability spaces, and $\Omega = \Omega_1 \times \Omega_2 \times \cdots \times \Omega_n$, $\mathscr{A} = \mathscr{A}_1 \times \mathscr{A}_2 \times \cdots \times \mathscr{A}_n$, $Pr = Pr_1 \times Pr_2 \times \cdots \times Pr_n$. Then the triplet $(\Omega, \mathscr{A}, Pr)$ is called the product probability space.

If $(\Omega_i, \mathscr{A}_i, Pr_i), i = 1,2,\ldots,$ be an arbitrary sequence of probability spaces, and

$$\Omega = \Omega_1 \times \Omega_2 \times \cdots, \quad \mathscr{A} = \mathscr{A}_1 \times \mathscr{A}_2 \times \cdots \tag{1.22}$$

It follows from the infinite product measure theorem that there is a unique probability measure Pr on $\mathscr{A}$ such that

$$Pr\{A_1 \times \cdots \times A_n \times \Omega_{n+1} \times \Omega_{n+2} \times \cdots\} = Pr_1\{A_1\} \times \cdots \times Pr_n\{A_n\} \tag{1.23}$$

for any measurable rectangle $A_1 \times \cdots A_n \times \Omega_{n+1} \times \Omega_{n+2} \times \cdots$ and all $n = 1,2,\ldots$ The probability measure Pr is called the infinite product of $Pr_i, i = 1,2,\ldots$ and is denoted by

$$Pr = Pr_1 \times Pr_2 \times \cdots \tag{1.24}$$

Definition 1.13. (Krickeberg [175]) Let $(\Omega_i, \mathscr{A}_i, Pr_i), i = 1,2,\ldots$ be probability spaces, and $\Omega = \Omega_1 \times \Omega_2 \times \cdots$, $\mathscr{A} = \mathscr{A}_1 \times \mathscr{A}_2 \times \cdots$, $Pr = Pr_1 \times Pr_2 \times \cdots$. Then the triplet $(\Omega, \mathscr{A}, Pr)$ is called the infinite product probability space.

1.6 Conditional Probability and Independence

Consider an experiment that consists of flipping a coin twice, nothing each time whether the result was heads or tails. The sample space of this experiment can be taken to be the following set of four outcomes:

$$\Omega = \{(H, H), (H, T), (T, H), (T, T)\},$$

where (H, T) means, for example, that the first flip lands heads and the second tails. Suppose now that each of the four possible outcomes is equally likely to occur and thus has probability 1/4. Suppose further that we observe that the first flip lands on heads. Then, given this information, what is the probability that both flips land on heads? To calculate this probability we reason as follows: Given that the initial flip lands heads, there can be at most two possible outcomes of our experiment, namely, (H, H) or (H, T). In addition, as each of these outcomes originally had the same probability of occurring, they should still have equal probabilities. That is, given that the first flip lands heads, the (conditional) probability of each of the outcomes (H, H) and (H, T) is 1/2, whereas the (conditional) probability of the other two outcomes is 0. Hence the desired probability is 1/2.

If we let A and B denote, respectively, the event that both flips land on heads and the event that the first flip lands on heads, then the probability obtained above is called the conditional probability of A given that B has occurred and is denoted by

$$Pr(A|B).$$

A general formula for $Pr(A|B)$ that is valid for all experiments and events A and B can be obtained in the same manner as given previously. Namely, if the event B occurs, then in order for A to occur it is necessary that the actual occurrence be a point in both A and B, that is, it must be in AB. Now since we know that B has occurred, it follows that B becomes our new sample space and hence the probability that the event AB occurs will equal the probability of AB relative to the probability of B. That is,

$$Pr(A|B) = \frac{Pr(AB)}{Pr(B)}.$$

As indicated by the coin flip example, $Pr(A|B)$, the conditional probability of A, given that B occurred, is not generally equal to $Pr(A)$, the unconditional probability of A. In other words, know that B has occurred generally changes the probability that A occurs (what if they were mutually exclusive?). In the special case where $Pr(A|B)$ is equal $Pr(A)$, we say that A and B are independent. Since $Pr(A|B) = Pr(AB)/Pr(B)$, we see that A is independent of B if

$$Pr(AB) = Pr(A)Pr(B).$$

Since this relation is symmetric in A and B, it follows that whenever A is independent of B, B is independent of A.

1.7 Random Variable

Considering the above foundation, a random variable on the probability space is defined as follows.

Definition 1.14. (Feller [97]) Let $(\Omega, \mathscr{A}, Pr)$ be a probability space. A real-valued function ξ defined on Ω is said to be a random variable if

$$\xi^{-1}(B) = \{\omega \in \Omega : \xi(\omega) \in B\} \in \mathscr{A}, \text{ for all } B \in \mathscr{B} \tag{1.25}$$

where $\mathscr{B}$ is the σ-algebra of Borel sets in $\mathbf{R} = (-\infty, +\infty)$, that is, a random variable ξ is a measurable transform of $(\Omega, \mathscr{A}, Pr)$ into $(\mathbf{R}, \mathscr{B})$. We note that it suffices to require that $\xi^{-1}(I) \in \mathscr{A}$ for all intervals I in $\mathbf{R}$, or for all semiclosed intervals $I = (a, b]$, or for all intervals $I = (-\infty, b]$, and so on. We also note that a random variable ξ defined on $(\mathbf{R}, \mathscr{B})$ includes a measure Pr_ξ on $\mathscr{B}$ defined by the relation

$$Pr_\xi(B) = Pr\{\xi^{-1}(B)\}, \quad B \in \mathscr{B} \tag{1.26}$$

Clearly Pr_ξ is a probability measure on $\mathscr{B}$ and is called the *probability distribution* or, simply the *distribution* of ξ. (Fig. 1.1).

Example 1.21. Assume that the probability space $(\Omega, \mathscr{A}, Pr)$ is $\{\omega_1, \omega_2\}$ and $Pr\{\omega_1\} = 0.2$, $Pr\{\omega_2\} = 0.8$. The function ξ is defined by

$$\xi(\omega) = \begin{cases} 0, \text{ if } \omega = \omega_1 \\ 1, \text{ if } \omega = \omega_2 \end{cases} \tag{1.27}$$

Then ξ is a random variable.

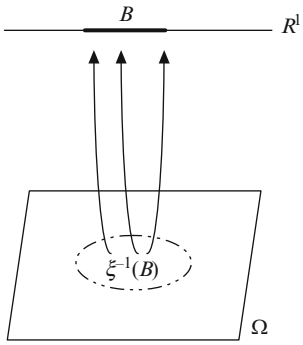


Fig. 1.1 Random variables

Example 1.22. A deterministic number c may be regarded as a special random variable. In fact, it is the constant function $\xi(\omega) \equiv c$ on the probability space $(\Omega, \mathscr{A}, Pr)$.

Example 1.23. In the model of two tosses of a coin with sample space $\Omega = \{HH, HT, TH, TT\}$, define a random variable $\xi = \xi(\omega)$ by the following table

ω	HH	HT	TH	TT
$\xi(\omega)$	2	1	1	0

Here, from its very definition, $\xi(\omega)$ is nothing but the number of heads in the outcome ω.

Definition 1.15. (Krickeberg [175]) A random variable ξ is said to be

(a) Nonnegative if $Pr\{\xi < 0\} = 0$
(b) Positive if $Pr\{\xi \leq 0\} = 0$
(c) Simple if there exists a finite sequence $\{x_1, x_2, \ldots, x_m\}$ such that

$$Pr\{\xi \neq x_1, \xi \neq x_2, \ldots, \xi \neq x_m\} = 0 \tag{1.28}$$

(d) Discrete if there exists a countable sequence $\{x_1, x_2, \ldots\}$ such that

$$Pr\{\xi \neq x_1, \xi \neq x_2, \ldots\} = 0 \tag{1.29}$$

Definition 1.16. (Krickeberg [175]) Let ξ_1 and ξ_2 be random variables defined on the probability space $(\Omega, \mathscr{A}, Pr)$. Then $\xi_1 = \xi_2$ if $\xi_1(\omega) = \xi_2(\omega)$ for almost all $\omega \in \Omega$.

1.8 Random Vector

Naturally, an idea extending the real space to n-dimensional real space to define a random vector is considered.

Definition 1.17. (Krickeberg [175]) An n-dimensional random vector is a measurable function from a probability space $(\Omega, \mathscr{A}, Pr)$ to the set of n-dimensional real vectors.

Since a random vector $\boldsymbol{\xi}$ is a function from Ω to $\mathbf{R}^n$, we can write $\boldsymbol{\xi}(\omega) = (\xi_1(\omega), \xi_2(\omega), \ldots, \xi_n(\omega))$ for every $\omega \in \Omega$, where $\xi_1, \xi_2, \ldots, \xi_n$ are functions from Ω to $\mathbf{R}$. Are $\xi_1, \xi_2, \ldots, \xi_n$ random variables in the sense of Definition 1.14? Conversely, we assume that $\xi_1, \xi_2, \ldots, \xi_n$ are random variables. Is $(\xi_1, \xi_2, \ldots, \xi_n)$ a random vector in the sense of Definition 1.17? The answer is in the affirmative. In fact, we have the following theorem and list the proving process since it used the important measure theory and set theory.

Theorem 1.16. *(Krickeberg [175]) The vector $(\xi_1, \xi_2, \ldots, \xi_n)$ is a random vector if and only if $\xi_1, \xi_2, \ldots, \xi_n$ are random variables.*

1.9 Probability Distribution

The probability distribution of a random variable is given as follows.

Definition 1.18. (Durrett [91]) For every $x \in \mathbf{R}$ such that

$$F(x) = Pr\{-\infty \leq \xi \leq x\} = Pr\{\omega \in \Omega | \xi(\omega) \leq x\}.$$

We call $F(x)$ the distribution function that the random variable ξ takes a value less than or equal to x.

In the following we write $\{\xi \leq x\}$ for the event $\{\omega \in \Omega | \xi(\omega) \leq x\}$. We first recall the following elementary property of a distribution function. In order to let readers easily understand them, we listed all the proving process and interested readers can consult the related books [26, 175, 206, 207, 288].

Theorem 1.17. *(Krickeberg [175]) The distribution function F of random variable ξ is nondecreasing, right-continuous function on $\mathbf{R}$ which satisfies*

$$F(-\infty) = \lim_{x \to -\infty} F(x) = 0$$

and

$$F(+\infty) = \lim_{x \to +\infty} F(x) = 1.$$

Theorem 1.18. *(Krickeberg [175]) A distribution function F is continuous at $x \in \mathbf{R}$ if and only if $Pr\{\omega \in \Omega | \xi(\omega) = x\} = 0$.*

Remark 1.10. Let ξ be a random variable, and let g be a Borel-measurable function defined on $\mathbf{R}$. Then $g(\xi)$ also a random variable whose distribution is determined by that of ξ.

We now show that a function F on $\mathbf{R}$ with the properties stated in Theorem 1.17 determines uniquely a probability measure Pr_F on $\mathscr{B}$.

Theorem 1.19. *(Krickeberg [175]) Let F be a nondecreasing, right-continuous function defined on $\mathbf{R}$ and satisfying*

$$F(-\infty) = 0 \quad \textit{and} \quad F(+\infty) = 0.$$

Then there exists a probability measure $Pr = Pr_F$ on $\mathscr{B}$ determined uniquely by the relation

$$Pr_F(-\infty, x] = F(x) \quad \textit{for every} \quad x \in \mathbf{R}.$$

Remark 1.11. Let F be bounded nondecreasing, right-continuous function defined on $\mathbf{R}$ satisfying $F(-\infty) = 0$. Then it is the clear from the proof of Theorem 1.19 that there exists a finite measure $\mu = \mu_F$ on $\mathscr{A}$ determined uniquely by $\mu_F(-\infty, x] = F(x), x \in \mathbf{R}$.

Remark 1.12. Let F on $\mathbf{R}$ satisfy the conditions of Theorem 1.19. Then there exists a random variable ξ on some probability space such that F is the distribution function of ξ. In fact, consider the probability space $(\mathbf{R}, \mathscr{A}, Pr)$, where Pr is the probability measure as constructed in Theorem 1.19. Let $\xi(\omega) = \omega$, for $\omega \in \mathbf{R}$. It is easy to see that F is the distribution function of the random variable ξ.

Let F be a distribution function, and let $x \in \mathbf{R}$ be a discontinuity point of F. Then $p(x) = F(x) - F(x-0)$ is called the *jump* of F at x. A point is said to be a *point of increase* of F if, for every $\epsilon > 0$, $F(x+\epsilon) - F(x-\epsilon) > 0$.

1.10 Central Limit Theorem

Many of the fundamental results in probability theory are formulated as *limit theorems*. Bernoulli's law of large numbers was formulated as a limit theorem, so was the De Moivre–Laplace theorem, which can fairly be called the origin of a genuine theory of probability and, in particular, which led the way to numerous investigations that clarified the conditions for the validity of the central limit theorem. Poisson's theorem on the approximation of the binomial distribution by the "Poisson" distribution in the case of rare events was formulated as a limit theorem. After the example of these propositions, and of results on the rapidity of convergence in the De Moivre–Laplace and Poisson theorems, it became clear that in probability it is necessary to deal with various kinds of convergence of distributions, and to establish the rapidity of convergence connected with the introduction of various "natural" measures of the distance between distributions. From then, many results about the central limit theorem has been obtained, and readers can refer to the related literatures [8, 31, 80, 89, 90, 148, 216, 266]. Let's begin by recalling the statement of the law of large numbers for the Bernoulli scheme.

Just as in analysis, in probability theory we need to use various kinds of convergence of random variables. Four of these are particularly important: *in probability, with probability one, in mean of order p, in distribution.*

First some definitions. Let $\xi, \xi_1, \xi_2, \ldots$ be random variables defined on a probability space $(\Omega, \mathscr{A}, Pr)$.

Definition 1.19. (Shiryaev [288]) The sequence $\xi_1, \xi_2, \ldots$ of random variables converges *in probability* to the random variable ξ (notation: $\xi_n \to^{Pr} \xi$) if for every $\xi > 0$

$$Pr\{|\xi_n - \xi| > \varepsilon\} \to 0, \; n \to \infty \tag{1.30}$$

We have already encountered this convergence in connection with the law of large numbers for a Bernoulli scheme, which stated that

$$Pr\left(\left|\frac{S_n}{n} - p\right| > \varepsilon\right) \to 0, \; n \to \infty \tag{1.31}$$

In analysis this is known as *convergence in measure*.

Definition 1.20. (Shiryaev [288]) The sequence $\xi_1, \xi_2, \ldots$ of random variables converges *with probability one (almost surely, almost everywhere)* to the random variable ξ if

$$Pr\{\omega : \xi_n \not\to \xi\} = 0 \tag{1.32}$$

i.e. if the set of sample points ω for which $\xi_n(\omega)$ does not converge to ξ has probability zero.

This convergence is denoted by $\xi_n \to \xi$ (Pr-a.s.), or $\xi_n \xrightarrow{a.s.} \xi$ or $\xi_n \xrightarrow{a.e.} \xi$.

Definition 1.21. (Shiryaev [288]) The sequence $\xi_1, \xi_2, \ldots$ of random variables converges *in mean of order* p, $0 < p < \infty$, to the random variable ξ if

$$E[|\xi_n - \xi|^p] \to 0, \quad n \to \infty \tag{1.33}$$

In analysis this is known as *convergence in* L^p, and denoted by $\xi_n \xrightarrow{L^p} \xi$. In the special case $p = 2$ it is called *mean square convergence* and denoted by $\xi = l.i.m.\xi_n$ (for "limit in the mean").

Definition 1.22. (Shiryaev [288]) The sequence $\xi_1, \xi_2, \ldots$ of random variables converges *in distribution* to the random variable ξ (notation: $\xi_n \xrightarrow{d} \xi$) if

$$E[f(\xi_n)] \to E[f(\xi)], \quad n \to \infty \tag{1.34}$$

for every bounded continuous function $f = f(x)$. The reason for the terminology is that, according to what will be proved later condition (1.34) is equivalent to the convergence of the distribution $F_{\xi_n}(x)$ to $F_\xi(x)$ at each point x of continuity of $F_\xi(x)$. This convergence is denoted by $F_{\xi_n} \Rightarrow F_\xi$.

We emphasize that the convergence of random variables in distribution is defined only in terms of the convergence of their distribution functions. Therefore it makes sense to discuss this mode of convergence even when the random variables are defined on different probability spaces.

Lemma 1.7. *(Shiryaev [288]) (a) A necessary and sufficient condition that* $\xi_n \to \xi (Pr - a.s.)$ *is that*

$$Pr\left\{\sup_{k \geq n} |\xi_k - \xi| \geq \varepsilon\right\} \to 0, \quad n \to \infty \tag{1.35}$$

for every $\varepsilon > 0$.

(b) The sequence $\{\xi_n\}_{n\geq 1}$ is the fundamental with probability 1 if and only if

$$Pr\left\{\begin{array}{l}\sup\limits_{\substack{k\geq n\\ l\geq n}}|\xi_k-\xi_l|\geq\varepsilon\end{array}\right\}\to 0,\quad n\to\infty \tag{1.36}$$

for every $\varepsilon>0$; or equivalently

$$Pr\left\{\sup_{k\geq n}|\xi_{n+k}-\xi_n|\geq\varepsilon\right\}\to 0,\quad n\to\infty \tag{1.37}$$

Lemma 1.8. *(Shiryaev [288]) We have the following implications:*

$$\begin{array}{l}\xi_n\xrightarrow{a.s.}\xi\Rightarrow\xi_n\xrightarrow{Pr}\xi,\\ \xi_n\xrightarrow{L^p}\xi\Rightarrow\xi_n\xrightarrow{Pr}\xi,\\ \xi_n\xrightarrow{Pr}\xi\Rightarrow\xi_n\xrightarrow{d}\xi.\end{array}$$

Definition 1.23. (Shiryaev [288]) Let $\xi_1,\xi_2,\ldots$ be a sequence of independent identically distributed random variables with $Pr(\xi_i=1)=p$, $Pr(\xi_i=0)=q$, $p+q=1$. In terms of the concept of convergence in probability, Bernoulli's law of large numbers can be stated as follows:

$$\frac{S_n}{n}\xrightarrow{P}p,\quad n\to\infty \tag{1.38}$$

where $S_n=\xi_1+\xi_2+\cdots+\xi_n$.

We put

$$F_n(x)=Pr\left\{\frac{S_n}{n}\leq x\right\} \tag{1.39}$$

$$F(x)=\left\{\begin{array}{l}1,\ x\geq p\\ 0,\ x\leq p\end{array}\right. \tag{1.40}$$

where $F(x)$ is the distribution function of the degenerate random variable $\xi\equiv p$. Also let Pr_n and Pr be the probability measures on $(\mathbf{R},\mathscr{B}(\mathbf{R}))$ corresponding to the distributions F_n and F.

In accordance with Lemma 1.8, convergence in probability, $S_n/n\xrightarrow{Pr}p$, implies convergence in distribution, $S_n/n\xrightarrow{d}p$, which means that

$$E\left[f\left(\frac{S_n}{n}\right)\right]\to E[f(p)],\quad n\to\infty \tag{1.41}$$

for every function $f = f(x)$ belonging to the class $\mathscr{C}(\mathbf{R})$ of bounded continuous functions on $\mathbf{R}$. By the definition of expected value of random variables, (1.41) can be rewritten in the form

$$\int_{\mathbf{R}} f(x)dF_n(x) \to \int_{\mathbf{R}} f(x)dF(x),\ f \in \mathscr{C}(\mathbf{R}). \tag{1.42}$$

In analysis, (1.42) is called weak convergence of F_n to F and denote it by $F_n \xrightarrow{w} F$. Thus we may say that in a Bernoulli scheme

$$\frac{S_n}{n} \xrightarrow{Pr} p \Rightarrow F_n \xrightarrow{w} F \tag{1.43}$$

It is also easy to see from (1.38) that, for the distribution functions defined in (1.39),

$$F_n(x) \to F(x),\ \ n \to \infty \tag{1.44}$$

for all points $x \in \mathbf{R}$ except for the single point $x = p$, where $F(x)$ has a discontinuity.

This shows that weak convergence $F_n \to F$ does not imply pointwise convergence for $F_n(x)$ to $F(x)$, $n \to \infty$, for all points $x \in \mathbf{R}$. However, it turns out that, both for Bernoulli schemes and for all arbitrary distribution functions, weak convergence is equivalent to "convergence in general" in the sense of the following definition.

Definition 1.24. (Shiryaev [288]) A sequence of distribution function $\{F_n(x)\}$ defined on the real line, converges *in general* to the distribution function $F(x)$ (notation: $F_n \Rightarrow F$) if as $n \to \infty$

$$F_n(x) \to F(x),\ \ x \in P_C(F), \tag{1.45}$$

where $P_C(F)$ is the set of points of continuity of $F = F(x)$.

For Bernoulli schemes, $F = F(x)$ is degenerate, and it is easy to see that

$$(F_n \Rightarrow F) \Rightarrow \left(\frac{S_n}{n} \xrightarrow{Pr} p\right).$$

Therefore,

$$\left(\frac{S_n}{n} \xrightarrow{Pr} p\right) \Rightarrow \left(F_n \xrightarrow{w} F\right) \Leftrightarrow (F_n \Rightarrow F) \Rightarrow \left(\frac{S_n}{n} \xrightarrow{Pr} p\right).$$

In this section we summarize, usually without proof, some important mathematical facts about the characteristic function. For detailed development, one may consult any of a number of advanced texts on probability theory or mathematical statistics, such as [26, 175, 252, 288] and so on.

1. In integral form, $\phi_\xi(-u) = \int e^{-iut} F_\xi dt$. This is the Fourier–Stieltjes transform of F_ξ. Thus, the characteristic function is determined by the distribution function rather than the random variable. Two quite different random variables may have the same distribution and hence the same characteristic function. If the distribution has a density function $p(t)$, then

$$\phi_\xi(-u) = \int e^{-iut} p(t)dt,$$

which is the Fourier transform of $p(x)$. The Fourier and the Fourier–Stieltjes transforms have been studied extensively, and the pertinent literature provides a powerful body of theory. These transforms have been used widely in physics and engineering, so there are important resources for application.
2. Since $\left|e^{iu\xi\omega}\right| = 1$ for any ω for which $\xi(\omega)$ is finite, $\phi_\xi(u)$ is defined for any real u for any probability distribution function F_ξ. This condition ensures a great deal of regularity for the characteristic function. For example, it may be used to show that ϕ_ξ is uniformly continuous on the entire real line.
3. There is a well known inversion integral for the Fourier integral. This shows that every distribution function is determined by its characteristic function. The form of the characteristic function often serves to establish the form of the distribution function. We use this in establishing a special case of the central limit theorem in the following part.
4. Since for a complex random variable $E[\bar{\xi}] = \overline{E[\xi]}$, we must have $\phi_\xi(u) = \overline{\phi_\xi(-u)}$. The overbar indicates the complex conjugate. If the distribution of ξ is symmetric with respect to the origin, so that ξ and $-\xi$ have the same distribution, then

$$\phi_\xi(u) = \phi_{-\xi}(u) = \phi_\xi(-u) = \overline{\phi_\xi(u)},$$

which ensures that ϕ_ξ is real-valued and even.

Theorem 1.20. *(Fundamental Convergence Theorem) (Shiryaev [288]) Consider a sequence $F_n (n \geq 1)$ of probability distribution functions, with ϕ_n the characteristic function for F_n for each n.*

1. *If F is a distribution function such that $F_n(t) \to F(t)$ at every point of continuity for F and ϕ is the characteristic function for F, then $\phi_n(u) \to \phi(u)$ for all u.*
2. *If $\phi_n(u) \to \phi(u)$ for all u and ϕ is continuous at 0, then ϕ is the characteristic function for a distribution function F such that $F_n(t) \to F(t)$ at each point of continuity of F.*

The central limit theorem (abbr. CLT) asserts that if random variables X is the sum of a large number of independent random variables, each with reasonable distributions, then ξ is approximately normally distributed. This celebrated theorem has been the object of prodigious research effort directed toward the discovery of the most general conditions under which it is valid. On the other hand, this theorem

serves as the basis of an extraordinary amount of applied work. In the statistics of large samples, the sample average is a constant times the sum the random variables in the sampling process. Thus, for large samples, the sample average is approximately normal-whether or not the population distribution is normal. In much of the theory of errors of measurement, the observed error is the sum of a large number of independent random quantities that contribute to the result. Similarly, in the theory of noise, the noise signal is the sum of a large number of random components, independently produced. In such situations, the assumption of a normal population distribution may be quite appropriate.

We consider a form of the CLT under hypotheses that are reasonable assumptions in many practical situations. The proof of this theorem, known as the Lindeberg–Lévy theorem [25], uses characteristic functions. The proof is not difficult; it illustrates the kind of argument used in more sophisticated proofs required for more general cases of the CLT.

Suppose ξ is a standardized normal random variable. That is, $\xi \sim \mathcal{N}(0, 1)$, with distribution function Φ. Consider an independent sequence $\{\xi_n : 1 \leq n\}$ of random variables. Form the sequence $\{S_n : 1 \leq n\}$ of partial sums

$$S_n = \sum_{i=1}^{n} \xi_i, \text{ for all } n \geq 1.$$

Then, by properties of expectation and of variance,

$$E[S_n] = \sum_{i=1}^{n} E[\xi], \text{ and } V[S_n] = \sum_{i=1}^{n} V[\xi], \text{ for all } n \geq 1.$$

Let S_n^* be the standardized sum obtained by subtracting the mean and dividing by the standard deviation for S_n. Let F_n be the distribution function for S_n^*. The CLT asserts that under appropriate conditions $F_n(t) \to \Phi(t)$ as $n \to \infty$ for any real t.

We list the simple case referred to earlier, using properties of characteristic functions and a simple lemma from the theory of complex variables. Readers can also consult the book related the probability theory.

Lemma 1.9. *(Shiryaev [288]) If ξ_n is independently and identically distributed random variables, with*

$$E[\xi_n] = \mu, \quad V[\xi_n] = \sigma^2, \text{ and } S_n^* = \frac{S_n - n\mu}{\sigma\sqrt{n}}$$

then

$$F_n(t) \to \Phi(t), \text{ as } n \to \infty \text{ for all } t.$$

Example 1.24. Suppose ζ_i $(i = 1, 2, \ldots)$ is independently and identically distributed random variables, with $E[\zeta_i] = 0$ and $V[\zeta_i] = \sigma^2$. For each $n \geq 1$, let

$$\xi_n = \sum_{i=1}^{n} \zeta_i \tag{1.46}$$

The sequence $\{\xi_n\}_{n=1}^{\infty}$ forms a *random walk*. This designation comes from the fact such a sequence may be used to model the following behavioral system. At discrete time $t = i$, a particle moves a distance ζ_i along a line. The net distance moved after the nth displacement is ξ_n. Individual movements are independent, with the same distribution, and the average movement is zero.

Show that as time increases, the probability of being within distance c of the starting position goes to zero, no matter how large c is.

Apparently, for large n, $\xi_n^* = \xi_n/\sigma\sqrt{n}$ is approximately $\mathcal{N}(0, 1)$. Thus,

$$Pr\{|\xi_n| \leq c\} = Pr\left(|\xi_n^*| \leq \frac{c}{\sigma\sqrt{n}}\right) \approx 2\Phi\left(\frac{c}{\sigma\sqrt{n}}\right) - 1.$$

Since $n \to \infty, c/\sigma\sqrt{n} \to 0$ and $\Phi(c/\sigma\sqrt{n}) \to 0.5$. Hence, $Pr\{|\xi_n| \leq c\} \to 0$ as $n \to \infty$.

1.11 Monte Carlo Simulation

Monte Carlo simulation has been applied to numerous areas, and is defined as a technique of performing sampling experiments on the models of stochastic systems. Although simulation is an imprecise technique which provides only statistical estimates rather than exact results and is also a slow and costly way to study problems, it is indeed a powerful tool dealing with complex problems without analytic techniques.

The basis of stochastic simulation is random number generation. Generally, let x be a random variable with a probability distribution $F(\cdot)$. Since $F(\cdot)$ is a nondecreasing function, the inverse function $F^{-1}(\cdot)$ is defined on $[0, 1]$. Assume that u is a uniformly distributed variable on the interval $[0, 1]$. Then we have

$$Pr\{F^{-1}(u) \leq y\} = Pr\{u \leq F(y)\} = F(y) \tag{1.47}$$

which proves that the variable $x = F^{-1}(u)$ has the probability distribution $F(\cdot)$. In order to get a random variable x with probability distribution $F(\cdot)$, we can produce a uniformly distributed variable u from the interval $[0, 1]$, and x is assigned to be $F^{-1}(u)$. The above process is called the *inverse transform method*. But for the main known distributions, instead of using the inverse transform method, we have direct generating processes. For detailed expositions, interested readers may consult

Fishman [101, 102], Law and Kelton [188], Bratley et al. [30], Rubinstein [269], Ross [267] and many other scholars [72, 77, 165, 194, 206, 226, 262, 263, 269, 279]. Here we give some generating methods for probability distributions frequently used in this book. We also use the inverse transform method to generate a discrete random variable. Suppose we want to generate the value of a discrete random variable ξ having probability mass function

$$P\{\xi = x_j\} = p_j, \quad j = 0, 1, \ldots, \quad \sum_j p_j = 1.$$

To accomplish this, we generate a random number U-that is, U is uniformly distributed over (0,1)-and set

$$\xi = \begin{cases} x_0 \text{ if } U < p_0 \\ x_1 \text{ if } p_0 \le U < p_0 + p_1 \\ \vdots \\ x_j \text{ if } \sum_{i=1}^{j-1} p_i \le U < \sum_{i=1}^{j} p_i \\ \vdots \end{cases}$$

Since, for $0 < a < b < 1$, $P\{a \le U < b\} = b - a$, we have that

$$P\{\xi = x_j\} = p\left\{\sum_{i=1}^{j-1} p_i \le U < \sum_{i=1}^{j} p_i\right\} = p_j$$

and so ξ has the desired distribution.

Remark 1.13. The preceding can be written algorithmically as

Generate a random number U
If $U < p_0$, set $\xi = x_0$ and stop
If $U < p_0 + p_1$, set $\xi = x_1$ and stop
If $U < p_0 + p_1 + p_2$, set $\xi = x_2$ and stop
$\vdots$

Remark 1.14. If the x_i, $i \ge 0$, are ordered so that $x_0 < x_1 < x_2 < \cdots$ and if we let F denote the distribution function of ξ, then $F(x_k) = \sum_{i=0}^{k} p_i$ and so

$$\xi \text{ will equal } x_j, \text{ if } F(x_{j-1} \le U < F(x_j)).$$

In other words, after generating a random number U we determine the value of ξ by finding the interval $[F(x_{j-1}), F(x_j))$ in which U lies (or, equivalently, by finding the inverse of $F(U)$). It is for this reason that the above is called the discrete inverse transform method for generating ξ.

The amount of time it takes to generate a discrete random variable by the above method is proportional to the number of intervals one must search. For this reason it is sometimes worthwhile to consider the possible values x_j of ξ in decreasing order of the p_j.

Example 1.25. If we wanted to simulate a random variable ξ such that

$$p_1 = 0.35, \quad p_2 = 0.25, \quad p_3 = 0.4, \quad \text{where } p_j = Pr\{\xi = j\},$$

then we could generate U and do the following:

Generate a random number U
If $U < 0.30$, set $\xi = 1$ and stop,
If $U < 0.60$, set $\xi = 2$ and stop,
Otherwise set $\xi = 3$.

However, a more efficient procedure is the following:

Generate a random number U
If $U < 0.40$, set $\xi = 3$ and stop,
If $U < 0.65$, set $\xi = 2$ and stop,
Otherwise set $\xi = 1$.

One case where it is not necessary to search for the appropriate interval in which the random number lies is when the desired random variable is the discrete uniform random variable. That is, suppose we want to generate the value of ξ which is equally likely to taken on any of the values $1, \ldots, n$. That is, $Pr\{\xi = j\} = 1/n$, $j = 1, \ldots, n$. Using the preceding results it follows that we can accomplish this by generating U and then setting

$$\xi = j, \quad \text{if } \frac{j-1}{n} \leq U < \frac{j}{n}$$

Therefore, ξ will equal j if $j - 1 \leq nU < j$; or, in other words,

$$\xi = \text{Int}(nU) + 1,$$

where $\text{Int}(x)$-sometimes written as $[x]$-is the integer part of x (i.e., the largest integer less than or equal to x). Next, let's discuss how to generate the two special discrete random variables.

Binomial Distribution: Suppose we want to generate the value of a binomial (n, p) random variable ξ-that is, ξ is such that

$$Pr\{\xi = i\} = \frac{n!}{i!(n-i)!} p^i (1-p)^{n-i}, \quad i = 0, 1, \ldots, n.$$

To do so, we employ the inverse transform method by making use of recursive identity

$$Pr\{\xi = i+1\} = \frac{n-i}{i+1}\frac{p}{1-p}Pr\{\xi = i\} \tag{1.48}$$

With i denoting the value currently under consideration, $Pr = Pr\{\xi = i\}$ the probability that ξ is equal to i, and $F = F(i)$ the probability that ξ is less than or equal to i, the algorithm can be expressed as follows:

Step 1. Generate a random number U.
Step 2. $c = p/(1-p), i = 0, Pr = (1-p)^n, F = Pr$.
Step 3. If $U < F$, set $\xi = i$ and stop.
Step 4. $Pr = [c(n-i)/(i+1)]Pr, F = F + Pr, i = i+1$.
Step 5. Go to Step 3.

The preceding algorithm first checks whether $\xi = 0$, then whether $\xi = 1$, and so on. Hence, the number of searches it makes is 1 more than the value of ξ. Therefore, on average, it will take $1 + np$ searches to generate ξ. Since a binomial (n, p) random variable represents the number of successes in n independent trials when each is a success with probability p, it follows that such a random variable can also be generated by subtracting from n the value of a binomial $(n, 1-p)$ random variable. Hence, when $p > 1/2$, we can generate a binomial $(n, 1-p)$ random variable by the above method and subtract its value from n to obtain the desired generation.

Remark 1.15. Another way of generating a binomial (n, p) random variable is by utilizing its interpretation as the number of successes in n independent trials by generating n random numbers $U_1, \ldots, U_n$ and then setting ξ equal to the number of the U_i that are less than or equal to p. By regarding the ith trial as a success if $U_1 < p$ and noting that the probability of this event is equal to p, it is easy to see that this results in a binomial (n, p) random variable. However, this approach requires n random numbers and makes n comparisons, whereas the inverse transform algorithm only requires one random number and makes, on average, $1 + np$ comparisons (along with an equal number of divisions).

Poisson Distribution: The random variable ξ is Poisson with mean λ if

$$p_i = Pr\{\xi = i\} = e^{-\lambda}\frac{\lambda^i}{i!}, \quad i = 0, 1, \ldots$$

The key to using the inverse transform method to generate such a random variable is the following identity:

$$p_{i+1} = \frac{\lambda}{i+1}p_i, \quad i \geq 0 \tag{1.49}$$

Upon using the above recursion to compute the Poisson probabilities as they become needed, the inverse transform algorithm for generating a Poisson random variable with mean λ can be expressed as follows. (The quantity i refers to the value presently under consideration; $p = p_i$ is the probability that ξ equals i, and $F = F(i)$ is the probability that ξ is less than or equal to i.)

Step 1. Generate a random number U.
Step 2. $i = 0$, $p = e^{-\lambda}$, $F = p$.
Step 3. If $U < F$, set $\xi = i$ and stop.
Step 4. $p = \lambda p/(i + 1)$, $F = F + p$, $i = i + 1$.
Step 5. Go to Step 3.

(In the above it should be noted that when we write, for example, $i = i + 1$, we do not mean that i is equal to $i + 1$ but rather that the value of i should be increased by 1.) To see that the above algorithm does indeed generate a Poisson random variable with mean λ, note that it first generates a random number U and then checks whether or not $U < e^{-\lambda} = p_0$. If so, it sets $\xi = 0$. If not, then it computes (in Step 4) p_1 by using the recursion (1.49). It now checks whether $U < p_0 + p_1$ (where the right-hand side is the new value of F), and if so it sets $\xi = 1$ and so on.

The above algorithm successively checks whether the Poisson value is 0, then whether it is 1, then 2, and so on. Thus, the number of comparisons needed will be 1 greater than the generated value of the Poisson. Hence, on average, the above will improved upon when λ is large. Indeed, since a Poisson random variable with mean λ is most likely to take on one of the two integral values, rather than starting at 0 and working upward. For instance, let $I = Int(\lambda)$ and use (1.49) to recursively determine $F(I)$. Now generate a Poisson random variable ξ with mean λ by generating a random number U, noting whether or not $\xi \leq I$ by seeing whether or not $U \leq F(I)$. Then search downward starting from I in the case where $\xi \leq I$ and upward starting from $I + 1$ otherwise.

The number of searches needed by this algorithm is roughly 1 more than the absolute difference between the random variable ξ and its mean λ. Since for λ large a Poisson is (by the central limit theorem) approximately normal with mean and variance both equal to λ, it follows that

$$\begin{aligned}\text{Average number of searches} &\simeq 1 + E[|\xi - \lambda|] \\ &= 1 + \sqrt{\lambda} E\left[\frac{|\xi - \lambda|}{\sqrt{\lambda}}\right] \\ &= 1 + \sqrt{\lambda} E[|Z|] \\ &= 1 + 0.798\sqrt{\lambda}\end{aligned}$$

where $\xi \sim \mathcal{N}(\lambda, \lambda)$ and $Z \sim \mathcal{N}(0, 1)$. That is, the average number of searches grows with the square root of λ rather than with λ as λ becomes larger and larger.

Consider a continuous random variable having distribution function F. A general method for generating such a random variable-called the inverse transformation method-is based on the following lemma.

Lemma 1.10. *Let U be a uniform (0,1) random variable. For any continuous distribution function F the random variable ξ defined by*

$$\xi = F^{-1}(U)$$

has distribution F. $F^{-1}(u)$ is defined to be that value of x such that $F(x) = u$.

Proof. Let F_ξ denote the distribution function of $\xi = F^{-1}(U)$. Then

$$\begin{aligned} F_\xi(x) &= Pr\{\xi \le x\} \\ &= Pr\{F^{-1}(U) \le x\} \end{aligned} \tag{1.50}$$

Now since F is a distribution function it follows that $F(x)$ is a monotone increasing function of x and so the inequality "$a \le b$" is equivalent to the inequality "$F(a) \le F(b)$". Hence, from (1.50), we see that

$$\begin{aligned} F_\xi(x) &= Pr\{F(F^{-1}(U)) \le F(x)\} \\ &= Pr\{U \le F(x)\} \\ &= F(x) \end{aligned}$$

This completes the proof. □

The above lemma thus shows that we can generate a random variable ξ from the continuous distribution function F by generating a random number U and then setting $\xi = F^{-1}(U)$.

Example 1.26. Suppose we wanted to generate a random variable ξ having distribution function

$$F(x) = x^n, \quad 0 < x < 1.$$

If we let $x = F^{-1}(u)$, then

$$u = F(x) = x^n \text{ or, equivalently, } x = u^{1/n}$$

Hence we can generate such a random variable ξ by generating a random number U and then setting $\xi = U^{1/n}$.

The inverse transform method yields a powerful approach to generating exponential random variables, as is indicated in the next example.

Example 1.27. If ξ is an exponential random variable with rate 1, then its distribution function is given by

$$F(x) = 1 - e^{-x}$$

If we let $x = F^{-1}(u)$, then

$$u = F(x) = 1 - e^{-x}$$

or

$$1 - u = e^{-x}$$

or, taking logarithms

$$x = -\log(1 - u).$$

Hence we can generate an exponential with parameter 1 by generating a random number U and then setting

$$\xi = F^{-1}(U) = -\log(1 - U).$$

A small saving in time can be obtained by noting that $1 - U$ is also uniform on (0,1) and thus $-\log(1 - U)$ has the same distribution as $-\log U$. That is, the negative logarithm of a random number is exponentially distributed with rate 1.

In addition, note that if ξ is exponential with mean 1 then, for any positive constant c, $c\xi$ is exponential with mean c. Hence, an exponential random variable ξ with rate λ (mean $1/\lambda$) can be generated by generating a random number U and setting

$$\xi = -\frac{1}{\lambda}\log U.$$

Remark 1.16. The above also provides us with another algorithm for generating a Poisson random variable. To begin, recall that a Poisson process with rate λ results when the times between successive events are independent exponentials with rate time 1 is Poisson distributed with mean λ. For such a process $N(1)$, the number of events by time 1 is Poisson distributed with mean λ. However, if we let $\xi_i, i = 1, 2, \ldots$, denote the successive interarrival times, then the nth event will occur at time $\sum_{i=1}^{n} \xi_i$, and so the number of events by time 1 can be expressed as

$$N(1) = \max\left\{n | \sum_{i=1}^{n} \xi_i \leq 1\right\}.$$

That is, the number of events by time 1 is equal to the largest n for which the nth event has occurred by time 1. (For example, if the fourth event occurred by time 1 but the fifth event did not, then clearly there would have been a total of four events by time 1.) Hence, using the results of Example 1.27, we can generate $N = N(1)$, a Poisson random variable with mean λ, by generating random numbers $U_1, \ldots, U_n, \ldots$ and setting

$$\begin{aligned} N &= \max\left\{n|\sum_{i=1}^{n} -\tfrac{1}{\lambda}\log U_i \le 1\right\} \\ &= \max\left\{n|\sum_{i=1}^{n} \log U_i \ge -\lambda\right\} \\ &= \max\left\{n|\log(U_1,\ldots,U_n) \ge -\lambda\right\} \\ &= \max\left\{n|U_1\ldots U_n \ge e^{-\lambda}\right\} \end{aligned}$$

Hence, a Poisson random variable N with mean λ can be generated by successively generating random numbers until their product falls below $e^{-\lambda}$, and then setting N equal to 1 less than the number of random numbers required. That is,

$$N = \min\left\{n|U_1\ldots U_n < e^{-\lambda}\right\} - 1.$$

The results of Example 1.27 along with the relationship between the gamma and the exponential distribution can be used to efficiently generate a gamma (n,λ) random variable. Next, let's introduce how to simulate the three special distributed random variables.

Uniform Distribution: A random variable ξ has a uniform distribution if its probability density function is defined by

$$f(x) = \begin{cases} \dfrac{1}{b-a}, & a \le x \le b \\ 0, & \text{otherwise} \end{cases} \tag{1.51}$$

denoted by $U(a,b)$, where a and b are given real numbers with $a < b$. The subfunction of generating pseudorandom numbers has been provided by the C library for any type of computer, defined as

include hstdlib.hi
int rand(void)

which produces a pseudorandom integer between 0 and $RAND - MAX$, where $RAND-MAX$ is defined in *stdlib.h* as $2^{15}-1$. Thus a uniformly distributed variable on an interval $[a,b]$ can be produced as follows:

Algorithm (Uniform Distribution)
Step 1. $u = rand()$;
Step 2. $u \leftarrow u/RAND - MAX$;
Step 3. Return $a + u(b-a)$.

Exponential Distribution : A random variable ξ has an exponential distribution with expected value $\beta(\beta > 0)$ if its probability density function is defined by

$$f(x) = \begin{cases} \dfrac{1}{\beta}e^{-x/\beta}, & \text{if } 0 \le x < \infty \\ 0, & \text{otherwise} \end{cases} \tag{1.52}$$

denoted by $exp(\beta)$. An exponentially distributed variable can be generated by the following way:

Algorithm (Exponential Distribution)

Step 1. Generate u from $U(0, 1)$;

Step 2. Return $-\beta \ln(u)$.

Normal Distribution: A random variable ξ has a normal distribution if its probability density function is defined as:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right], -\infty < x < +\infty \tag{1.53}$$

denoted by $\mathcal{N}(\mu, \sigma^2)$, where μ is the expected value and σ^2 is the variance.

Algorithm (Normal Distribution)

Step 1. Generate μ_1 and μ_2 from $U(0, 1)$;

Step 2. $y = [-2\ln(\mu_1)]^{\frac{1}{2}} sin(2\pi\mu_2)$;

Step 3. Return $\mu + \sigma y$.

Chapter 2
Random Multiple Objective Decision Making

Mathematical programming with multiple objective functions (also called vector programming) is a recent development in mathematical programming, and emerged from an attempt to tackle the problems raised by the present development of science, engineering, industry, economics, etc. Due to the complexity of these problems, several objectives had to be incorporated in the optimization process. Basically, the problem consists of optimizing several objectives functions (some of which must be maximized, some minimized) provided the variables satisfy the linear and nonlinear constraints.

Nowadays, mathematical programming problems with uncertain parameters are extremely important and many scholars have begun to pay attention to these. In this section, we mainly discuss a random multiple objective decision making model. Many mathematical programming problems are stochastic because some of the data involved are random variables. This may be due to

1. Errors and variations in the problem parameters, which are often associated with probabilities.
2. Risk and uncertainty, which may sometimes allow a significant numerical representation of the utility function of the decision-maker (e.g. the von Neumann axiomatic system).
3. The need for optimum decision rules connected with some statistical decision functions, etc.

Hence, one has to study vector programming under the assumption that some of the problem parameters are random variables.

2.1 Distribution Center Location Problem with Random Phenomena

With the appearance of the importance of logistics in global economics. The importance of distribution location problems has emerged as being one of the most critical issues for logistics managers reducing cost and improving service. Manufactorers,

J. Xu and L. Yao, *Random-Like Multiple Objective Decision Making*, Lecture Notes in Economics and Mathematical Systems 647, DOI 10.1007/978-3-642-18000-2_2,

customers and suppliers are important members of the supply chain. To some extent, the success of an enterprise depends on its ability to link these members seamlessly. Distribution centers (abbr. DC) have the competency to connect manufactorers with their customers. The number, size and locations of distribution centers along with the decision on which customers to serve from each center (i.e. the allocation of customers to distribution centers) significantly affects the cost of the distribution system and the level of customer service provided by the system. In real logistics systems, the sources (factories, vendors, etc.) supply the distribution centers, which in turn supply the demand locations or consumers. Thus, it is necessary for manufactorers to evaluate and select suitable DC locations from a potential set.

In recent years, increasing production economies of scale and reducing transport costs have focused attention on distribution centers and many methods for location selection have been developed [17, 132, 164, 308, 320, 362]. An integer programming model for the plant location was presented by Barahona and Jensen [17]. They considered not only the fixed costs and transport costs, but also inventory costs, which had been solved by the Dantzig–Wolfe (D-W) decomposition method. Tragantalerngsak et al. [320] considered a two-echelon facility location problem in which the facilities in the first echelon are uncapacitated and the facilities in the second echelon are capacitated. The goal is to determine the number and locations of facilities in both echelons in order to satisfy customer demand for the product. Zhou et al. [362] investigated the balanced allocation of customers to multiple distribution centers with a genetic algorithm approach. Syam [308] proposed a model and methodologies for a location problem with logistical components. Harkness [132] investigated a new type of facility location model: one in which production costs per unit increase once a certain scale of output is reached. Klose and Drexl [164] reviewed some of the contributions to the current state of facility location models for distribution systems.

However, although the facility location problem has been studied widely, the majority of these papers assume that the design parameters of the DC location problem are deterministic. For real decision making, the situation is often not deterministic, and some factors such as demand, and costs of transport are usually changing, hence we must consider the DC location problem under an uncertain environment. In practice, many parameters for location models are subject to uncertainty. For example, queuing location models are discussed by Berman and Larson [22]. They give certain distribution functions for the customer arrival process, waiting, and approximate service times. Waiting times are a function of demand allocation and, hence, of facility location. Further stochastic location models are proposed by Laporte et al. [186] and Listes and Dekker [202]. Laporte et al. [186] considers customer demand as a stochastic variable and proposes the branch-and-cut algorithm to solve location problems with stochastic parameters. Listes and Dekker [202] use stochastic models with recourse for the purposes of locating facilities in product recovery networks. More recently, since Zadeh's pioneering work [356], a lot of research on fuzzy parameters have been done [35, 197, 225, 238, 340, 345]. Many successful applications of fuzzy optimization theory in the area of facility the location can be found in literature. Chen [48] proposes a fuzzy multiple

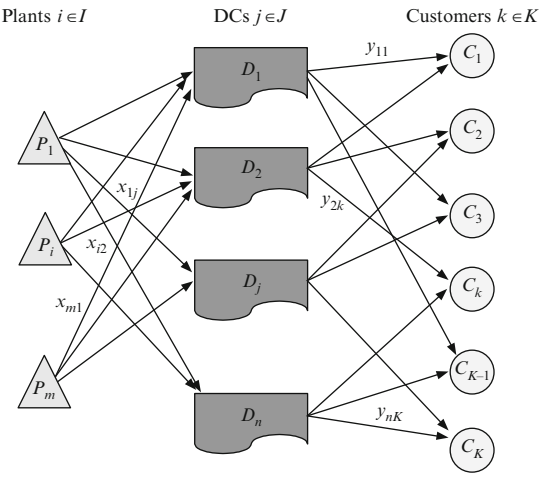


Fig. 2.1 Network model of logistics DCs location problem

criteria group decision-making method to solve distribution center location selection problems. Yang [351] investigates the logistics distribution centers location problem under a fuzzy environment and fuzzy chance-constrained programming is constructed as a decision model for the problem.

This chapter mainly concentrates on discussing a class of distribution center location problems with random parameters, i.e., customer demand, transportation costs and so on are considered as random variables. In fact, in real-life world, this case usually occurs. Let's recall a simple example proposed by Mirchandani et al. [220] (Fig. 2.1). They introduces a stochastic variant of the p-median problem. In particular, they consideres the demand and arc weights to be random variables. Under certain assumptions a finite number of states $i \in I$ of the graph with known probabilities can be enumerated. The model provided by them is as follows:

$$\begin{cases} \min \sum\limits_{i\in I}\sum\limits_{k\in K}\sum\limits_{j\in J} \pi_i c_{ikj} z_{ikj} \\ \text{s.t.} \begin{cases} \sum\limits_{j\in J} z_{ikj} = 1 \ \ \forall i \in I, j \in J \\ z_{ikj} - y_j \le 0 \ \ \forall i \in I, k \in K, j \in J \\ \sum\limits_{j\in J} y_j = p \\ z_{ikj}, y_j \in \mathbf{B} \ \ \forall i \in I, k \in K, j \in J \end{cases} \end{cases}$$

The symbol c_{ikj} denotes the demand weighted distance between nodes $k \in K$ and $j \in J$ in state $i \in I$. z_{ikj} are decision variables denoting demand allocation. y_j denotes location decisions. In this chapter, we will extend this model to be closer to the realistic problems and use the expected value operator and chance operator to resolve them. Readers can find these in the last section of this chapter.

2.2 Two Kinds of Random Variables

In the first chapter, we have introduced some basic properties about random variables. Here, we just research two special types of random variables which frequently appear in some realistic problems. One is the discrete, and the other is the continuous. Interested readers can refer to those related books such as [11, 76, 151, 152, 184, 281].

2.2.1 Discrete Random Variable

In this part, we consider certain types of discrete random variables.

Definition 2.1. Let ξ be a random variable on the probability space $(\Omega, \mathscr{A}, Pr)$. If $\Omega = \{\omega_1, \omega_2, \ldots\}$ is a set combined with finite or infinite discrete elements, where $Pr\{\omega = \omega_i\} = p_i$ and $\sum_{i=1}^{\infty} p_i = 1$, then ξ is called the discrete random variable.

From the above definition, we know that a discrete random variable ξ is a mapping from the discrete probability space Ω to the real space $\mathbf{R}$.

Example 2.1. Let $\Omega = \{1, 2, 3, 4\}$ be the probability space and $Pr\{\omega_i = i\} = 0.25$, $i = 1, \ldots, 4$. If $\xi(\omega_i) = 1/\omega_i$, then ξ is a discrete random variable.

Intuitively, we want to know what is the distribution of a discrete random variable. The following equation is usually used to describe the distribution,

$$\begin{pmatrix} \xi(\omega_1) & \xi(\omega_2) & \cdots & \xi(\omega_n) \\ p_1 & p_2 & \cdots & p_n \end{pmatrix}$$

In the following part, three special discrete random variables will be introduced.

As we know, in the trial of tossing a coin, the probabilities of the front and the back are all 0.5. Then we denote that $\omega_1 =$ 'Front' and $\omega_2 =$ 'Back' and let ξ be a mapping from $\{\omega_1, \omega_2\}$ to $\{0, 1\}$ satisfying $\xi(\omega_1) = 1$ and $\xi(\omega_2) = 0$.

Definition 2.2. (Binomial random variable) Suppose that n independent trials, each of which results in a "success" with probability p, are to be performed. If ξ represents the number of successes that occur in the n trials, then ξ is said to be a binomial random variable with parameters (n, p). Its probability mass function is given by

$$Pr\{\xi = i\} = \binom{n}{i} p^i (1-p)^{n-i}, \ i = 0, 1, \ldots, n \tag{2.1}$$

where

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}$$

is the binomial coefficient, equal to the number of different subsets of i elements that can be chosen from a set of n elements. Obviously, in this example, Ω has n elements combined with the natural number $i = 1, 2, \ldots, n$.

Since we assume that all trials are independent with each other, then the probability of any particular sequence of outcomes results in i successes and $n - i$ failures. Furthermore, it can be seen that (2.1) is valid since there are $\binom{n}{i}$ different sequences of the n outcomes that result in i successes and $n - i$ failures, which can be seen by noting that there are $\binom{n}{i}$ different choices of the i trials that result in successes.

Definition 2.3. A binomial random variable $(1, p)$ is called a Bernoulli random variable.

Since a binomial (n, p) random variable ξ represents the number of successes in n independent trials, each of which results in a success with probability p, we can represent it as follows:

$$\xi = \sum_{i=1}^{n} \xi_i \tag{2.2}$$

where

$$\xi_i = \begin{cases} 1, \text{ if the } i\text{th trial is a success} \\ 0, \text{ otherwise} \end{cases}$$

The following recursive formula expressing p_{i+1} in terms of p_i is useful when computing the binomial probabilities:

$$\begin{aligned} p_{i+1} &= \frac{n!}{(n-i-1)!(i+1)!} p^{i+1}(1-p)^{n-i-1} \\ &= \frac{n!(n-i)}{(n-i)!i!(i+1)} p^i (1-p)^{n-i} \frac{p}{1-p} \\ &= \frac{n-i}{i+1} \frac{p}{1-p} p_i . \end{aligned}$$

Definition 2.4. (Poisson random variable) A random variable ξ that takes on one of the values $0, 1, 2, \ldots$ is said to be a Poisson random variable with parameter λ, $\lambda > 0$, if its probability mass function is given by

$$p_i = Pr\{\xi = i\} = e^{-\lambda} \frac{\lambda^i}{i!}, \; i = 1, 2, \ldots \tag{2.3}$$

The symbol e, defined by $e = \lim_{n\to\infty}(1+1/n)^n$, is a famous constant in mathematics that is roughly equal to 2.7183.

Poisson random variables have a wide range of applications. One reason for this is that such random variables may be used to approximate the distribution of the number of successes in a large number of trials (which are either independent or at most "weakly dependent") when each trial has a small probability of being a success. To see why this is so, suppose that ξ is a binomial random variable with parameters (n, p), and so represents the number of successes in n independent trials when each trial is a success with probability p, and let $\lambda = np$. Then

$$\begin{aligned} Pr\{\xi = i\} &= \frac{n!}{(n-i)!\cdot i!}p^i(1-p)^{n-i} \\ &= \frac{n!}{(n-i)!\cdot i!}\left(\frac{\lambda}{n}\right)^i\left(1-\frac{\lambda}{n}\right)^{n-i} \\ &= \frac{n(n-1)\cdots(n-i+1)}{n^i}\cdot\frac{\lambda^i}{i!}\cdot\frac{(1-\lambda/n)^n}{(1-\lambda/n)^i} \end{aligned}$$

Now for n large and p small,

$$\begin{aligned} &\lim_{n\to\infty}\left(1-\frac{\lambda}{n}\right)^n \to e^{-\lambda}, \\ &\lim_{n\to\infty}\frac{n(n-1)\cdots(n-i+1)}{n^i} \to 1, \\ &\lim_{n\to\infty}\left(1-\frac{\lambda}{n}\right)^i \to 1. \end{aligned}$$

Hence, for n large and p small,

$$Pr\{\xi = i\} \approx e^{-\lambda}\frac{\lambda^i}{i!}.$$

To compute the Poisson probabilities we make use of the following recursive formula:

$$\frac{p_{i+1}}{p_i} = \frac{\frac{e^{-\lambda}\lambda^{(i+1)}}{(i+1)!}}{\frac{e^{-\lambda}\lambda^i}{i!}} = \frac{\lambda}{i+1}$$

or equivalently,

$$p_{i+1} = \frac{\lambda}{i+1}p_i, \quad i \geq 0.$$

Consider independent trials, each of which is a success with probability p. If ξ represents the number of the first trial that is a success, then

$$Pr\{\xi = n\} = p(1-p)^{n-1} \tag{2.4}$$

which is easily obtained by noting that in order for the first success to occur on the nth trial, the first $n-1$ must all be failures and the nth success. Equation (2.4) is said to be a geometric random variable with parameter p.

If we let ξ denote the number of trials needed to amass a total of r successes when each trial is independently a success with probability p, then ξ is said to be a negative binomial, sometimes called a Pascal random variable with parameters p and r. The probability mass function of such a random variable is given as follows,

$$Pr\{\xi = n\} = \binom{n-1}{r-1} p^r (1-p)^{n-r}, \quad n \geq r \tag{2.5}$$

To see why (2.5) is valid note that in order for it to take exactly n trials to amass r successes, the first $n-1$ trials must result in exactly $r-1$ successes, and the probability of this is $\binom{n-1}{r-1} p^r (1-p)^{n-r}$, and then the nth trial must be a success, and the probability of this is p.

Consider an urn containing $N+M$ balls, of which N are light colored and M are dark colored. If a sample of size n is randomly chosen (in the sense that each of the $\binom{N+M}{n}$ subsets of size n is equally likely to be chosen) then ξ, the number of light colored balls selected, has probability mass function,

$$Pr\{\xi = i\} = \frac{\binom{N}{i}\binom{M}{n-i}}{\binom{N+M}{n}}.$$

A random variable ξ whose probability mass function is given by the preceding equation is called a hypergeometric random variable.

2.2.2 *Continuous Random Variable*

In this part, we consider certain types of continuous random variables.

Definition 2.5. (Uniform random variable) A random variable ξ is said to be uniformly distributed over the interval (a, b), $a < b$, if its probability density function is given by

$$f(x) = \begin{cases} \dfrac{1}{b-a}, & \text{if } a < x < b, \\ 0, & \text{otherwise.} \end{cases}$$

In other words, ξ is uniformly distributed over (a, b) if it puts all its mass on that interval and it is equally likely to be "near" any point on that interval.

The distribution function of ξ is given, for $a < x < b$, by

$$F(x) = Pr\{\xi \leq x\} = \int_a^x (b-a)^{-1} dx = \frac{x-a}{b-a}.$$

Definition 2.6. (Normal random variable) A random variable ξ is said to be normally distributed with mean μ and variance σ^2 if its probability density function is given by

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2}, \quad -\infty < x < \infty.$$

The normal density is a bell-shaped curve that is symmetric about μ.

An important fact about normal random variables is that if ξ is normal with mean μ and variance σ^2, then for any constants a and b, $a\xi + b$ is normally distributed with mean $a\mu + b$ and variance $a^2\sigma^2$. It follows from this that if ξ is normal with mean μ and variance σ^2, then

$$\zeta = \frac{\xi - \mu}{\sigma}$$

is normal with mean 0 and variance 1. Such a random variable ζ is said to have a standard (or unit) normal distribution. Let Φ denote the distribution function of a standard normal random variable, that is

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-x^2/2} dx, \quad -\infty < x < \infty.$$

The result that $\zeta = (\xi - \mu)/\sigma$ has a standard normal distribution when ξ is normal with mean μ and variance σ^2 is quite useful because it allows us to evaluate all probabilities concerning ξ in terms of Φ. For example, the distribution function of ξ can be expressed as

$$\begin{aligned} F(x) &= Pr\{\xi \leq x\} \\ &= Pr\left\{\frac{\xi - \mu}{\sigma} \leq \frac{x-\mu}{\sigma}\right\} \\ &= Pr\left\{\zeta \leq \frac{x-\mu}{\sigma}\right\} \\ &= \Phi\left(\frac{x-\mu}{\sigma}\right) \end{aligned}$$

The value of $\Phi(x)$ can be determined either by looking it up in a table or by writing a computer program to approximate it. For a in the interval $(0, 1)$, let ζ_a be such that

$$Pr\{\zeta > \zeta_a\} = 1 - \Phi(\zeta_a) = a.$$

That is, a standard normal will exceed ζ_a with probability a. The value of ζ_a can be obtained from a table of the values of Φ. For example, since

$$\Phi(1.64) = 0.95, \ \ \Phi(1.96) = 0.975, \ \ \Phi(2.33) = 0.99,$$

we see that

$$\zeta_{0.05} = 1.64, \ \ \zeta_{0.025} = 1.96, \ \ \zeta_{0.01} = 2.33.$$

The wide applicability of normal random variables results from one of the most important theorems of probability theory – the central limit theorem, which asserts that the sum of a large number of independent random variables has approximately a normal distribution.

Definition 2.7. (Exponential random variable) A continuous random variable having probability density function,

$$f(x) = \begin{cases} \lambda e^{-\lambda x}, & 0 \le x < \infty, \\ 0, & \text{otherwise.} \end{cases}$$

for some $\lambda > 0$ is said to be an exponential random variable with parameter λ.

Its cumulative distribution is given by

$$F(x) = \int_0^x \lambda e^{-\lambda x} dx = 1 - e^{-\lambda x}, \ 0 < x < \infty.$$

The key property of exponential random variables is that they possess the "memoryless property", where we say that the nonnegative random variable ξ is memoryless if

$$Pr\{\xi > s + t | \xi > s\} = Pr\{\xi > t\}, \text{ for all } s, t \ge 0. \tag{2.6}$$

To understand why the above is called the memoryless property, imagine that ξ represents the lifetime of some unit, and consider the probability that a unit of age s will survive an additional time t. Since this will occur if the lifetime of the unit exceeds $t + s$ given that it is still alive at time s, we see that

$$Pr\{\text{additional life of an item of age } s \text{ exceeds } t\} = Pr\{\xi > s + t | \xi > s\}.$$

Thus, (2.6) is a statement of fact that the distribution of the remaining life of an item of age s does not depend on s. That is, it is not necessary to remember the age of the unit to know its distribution of remaining life. Equation (2.6) is equivalent to

$$Pr\{\xi > s + t\} = Pr\{\xi > s\}Pr\{\xi > t\}.$$

As the above equation is satisfied whenever ξ is an exponential random variable – since, in this case, $Pr\{\xi > x\} = e^{-\lambda x}$ – we see that exponential random variables are memoryless (and indeed it is not difficult to show that they are the only memoryless random variables).

Another useful property of exponential random variables is that they remain exponential when multiplied by a positive constant. To see this suppose that ξ is exponential with parameter λ, and let c be a positive number. Then

$$Pr\{c\xi \leq x\} = Pr\left\{\xi \leq \frac{x}{c}\right\} = 1 - e^{-\lambda x/c},$$

which shows that $c\xi$ is exponential with parameter λ/c.

Let $\xi_1, \xi_2, \ldots, \xi_n$ be independent exponential random variables with respective rates $\lambda_1, \lambda_2, \ldots, \lambda_n$. A useful result is that $\min\{\xi_1, \xi_2, \ldots, \xi_n\}$ is exponential with rate $\sum_i \lambda_i$ and is independent of which one of the ξ_i is the smallest. To verify this, let $M = \min\{\xi_1, \xi_2, \ldots, \xi_n\}$. Then

$$\begin{aligned} Pr\left\{\xi_j = \min_i \xi_i | M > t\right\} &= Pr\{\xi_j - t = \min_i (\xi_i - t) | M > t\} \\ &= Pr\{\xi_j - t = \min_i (\xi_i - t) | \xi_i > t, i = 1, 2, \ldots, n\} \\ &= Pr\{\xi_j = \min_i \xi_i\}. \end{aligned}$$

The final equality follows because, by the lack of memory property of exponential random variables, given that ξ_i exceeds t, the amount by which it exceeds it is exponential with rate λ_i. Consequently, the conditional distribution of $\xi_1 - t, \ldots, \xi_n - t$ given that all the ξ_i exceed t is the same as the unconditional distribution of $\xi_1, \ldots, \xi_n$. Thus, M is independent of which of the ξ_i is the smallest.

The result that the distribution of M is exponential with rate $\sum_i \lambda_i$ follows from

$$Pr\{M > t\} = Pr\{\xi_i > t, i = 1, 2, \ldots, n\} = \prod_{i=1}^{n} Pr\{\xi_i > t\} = e^{-\sum_{i=1}^{n} \lambda_i t}.$$

The probability that ξ_j is the smallest is obtained from

$$\begin{aligned} Pr\{\xi_j = M\} &= \int Pr\{\xi_j = M | \xi_j = t\} \lambda_j e^{-\lambda_j t} dt \\ &= \int Pr\{\xi_j > t, i \neq j | \xi_j = t\} \lambda_j e^{-\lambda_j t} dt \\ &= \int Pr\{\xi_j > t, i \neq j\} \lambda_j e^{-\lambda_j t} dt \end{aligned}$$

$$= \int \left(\prod_{i \neq j} e^{-\lambda_i t} \right) e^{-\lambda_j t} dt$$
$$= \lambda_j \int e^{-\sum_i \lambda_i t} dt$$
$$= \frac{\lambda_j}{\sum_i \lambda_i}$$

There are also other special random variables following other distribution, readers can refer to the related literatures and we don't introduce here.

2.3 Random EVM

The *expectation* of a random variable is a central concept in the study of probability. It is the average of all possible values of a random variable, where a value is weighted according to the probability that it will appear. The expectation is sometimes also called *average*. It is also called the *expected value* or the *mean* of the random variable. These terms are all synonymous. The so-called *expected value model* (EVM) means to optimize some expected objective functions subject to some expected constraints, for example, minimizing expected cost, maximizing expected profit, and so on.

2.3.1 General Model for Random EVM

Now let us recall the well-known newsboy problem [206] in which a boy operating a news stall has to determine the number x of newspapers to order in advance from the publisher at a cost of c per one newspaper every day. It is known that the selling price is a per one newspaper. However, if the newspapers are not sold at the end of the day, then the newspapers have a small value of b per one newspaper at the recycling center. Assume that the demand for newspapers is denoted by ξ in a day, then the number of newspapers at the end of the day is clearly $x - \xi$ if $x > \xi$ or 0 if $x < \xi$. Thus the profit of the newsboy should be

$$f(x, \xi) = \begin{cases} (a-c)x, & \text{if } x \leq \xi \\ (b-c)x + (a-b)\xi, & \text{if } x > \xi \end{cases} \tag{2.7}$$

In practice, the demand ξ for newspapers is usually a stochastic variable, so is the profit function $f(x, \xi)$. Since we cannot predict how profitable the decision of ordering x newspapers will actually be, a natural idea is to employ the expected

profit, shown as follows,

$$E[f(x,\xi)] = \int_0^x [(b-c)x + (a-b)r]dF(r) + \int_x^{+\infty} (a-c)xdF(r) \quad (2.8)$$

where E denotes the expected value operator and $F(\cdot)$ is the distribution function of demand ξ. The newsboy problem is related to determining the optimal integer number x of newspapers such that the expected profit $E[f(x,\xi)]$ achieves the maximal value, i.e.,

$$\begin{cases} \max E[f(x,\xi)] \\ \text{s.t.} x \geq 0, \text{ integers} \end{cases} \quad (2.9)$$

This is a typical example of an expected value model.

Then we firstly should give the basic definition of the expected value. For the discrete random variable, we can define its expected value as follows.

Definition 2.8. Let ξ be a discrete random variable on the probability $(\Omega, \mathscr{A}, Pr)$ as follow,

$$\xi(\omega) = \begin{cases} x_1 \text{ if } \omega = \omega_1 \\ x_2 \text{ if } \omega = \omega_2 \\ \cdots \quad \cdots \end{cases} \quad (2.10)$$

where the probability of $\omega = \omega_i (i = 1, 2, \ldots)$ is p_i. If the series $\sum_{\omega\in\Omega} \xi(\omega_i) Pr\{\omega = \omega_i\}$ is absolutely convergent, then we call it the expected value of ξ, denoted by $E[\xi]$.

For the continuous random variable, its expected value can be defined as follows.

Definition 2.9. (Durrett [91]) Let ξ be a random variable on the probability space $(\Omega, \mathscr{A}, Pr)$. Then the expected value of ξ is defined by

$$E[\xi] = \int_0^{+\infty} Pr\{\xi \geq r\}dr - \int_{-\infty}^0 Pr\{\xi \leq r\}dr \quad (2.11)$$

There is another equivalent definition by the density function.

Definition 2.10. (Durrett [91]) The expected value of a random variable ξ with probability density function $f(x)$ is

$$E[\xi] = \int_{-\infty}^{+\infty} xf(x)dx \quad (2.12)$$

Expected value, average and mean are the same thing, but median is entirely different. The median is defined below, but only to make the distinction clear. After this, we won't make further use of the median.

Definition 2.11. (Durrett [91]) The median of a random variable ξ is the unique value r in the range of ξ such that $Pr\{\xi < r\} \leq 1/2$ and $Pr\{\xi > r\} < 1/2$.

For example, with an ordinary die, the median thrown value is 4, which not the same as the mean 3.5. The median and the mean can be very far apart. For example, consider a $2n$-side die, with n 0s and 100s. The mean is 50, and the median is 100.

To deeply understand random variables, the variance of a random variable is given as follows.

Definition 2.12. (Durrett [91]) The variance of a random variable ξ is defined by

$$V[\xi] = E[(\xi - E[\xi])^2].$$

The following properties about the expected value and variance of a random variable are very useful to the decision making problems with random parameters [91].

Lemma 2.1. *Let ξ and η be random variables with finite expected values. Then for any numbers a and b, we have*

$$E[a\xi + b\eta] = aE[\xi] + bE[\eta] \tag{2.13}$$

Lemma 2.2. *For two independent random variables ξ and η, we have*

$$E[\xi\eta] = E[\xi]E[\eta] \tag{2.14}$$

Lemma 2.3. *For the random variable ξ, we have*

$$V[\xi] = E[\xi^2] - (E[\xi])^2 \tag{2.15}$$

and for $a, b \in \mathbf{R}$, we have

$$V[a\xi + b] = a^2 V[\xi] \tag{2.16}$$

Let's consider the typical single objective with random parameters,

$$\begin{cases} \max f(\boldsymbol{x}, \boldsymbol{\xi}) \\ \text{s.t.} \begin{cases} g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, \ j = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{2.17}$$

where $f(\boldsymbol{x}, \boldsymbol{\xi})$ and $g_j(\boldsymbol{x}, \boldsymbol{\xi}), j = 1, 2 \ldots, p$ are continuous functions in X and $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_n)$ is a random vector on the probability space $(\Omega, \mathscr{A}, Pr)$. Then it follows from the expected operator that,

$$\begin{cases} \max E[f(\boldsymbol{x}, \boldsymbol{\xi})] \\ \text{s.t.} \begin{cases} E[g_j(\boldsymbol{x}, \boldsymbol{\xi})] \leq 0, \ j = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{2.18}$$

After being dealt with by expected value operator, the problem (2.17) has been converted into a certain programming and then decision makers can easily obtain the optimal solution. However, whether the problem (2.18) has optimal solutions is a spot which decision makers pay more attention to, then its convexity is the focus we will discuss in the following part.

Definition 2.13. $\boldsymbol{x}$ is said to be a *feasible solution* of problem (2.18) if and only if $E[g_j(\boldsymbol{x}, \boldsymbol{\xi})] \leq 0\,(j = 1, 2, \ldots, p)$. For any feasible solution $\boldsymbol{x}$, if $E[f(\boldsymbol{x}^*, \boldsymbol{\xi})] \geq E[f(\boldsymbol{x}, \boldsymbol{\xi})]$, then $\boldsymbol{x}^*$ is an optimal solution of problem (2.18).

A mathematical programming model is called convex if both the objective function and the feasible set are convex. For the expected value model (2.18), we have the following result on convexity.

Theorem 2.1. *Assume that, for each $\boldsymbol{\xi}$, the functions $f(\boldsymbol{x}, \boldsymbol{\xi})$ and $g_j(\boldsymbol{x}, \boldsymbol{\xi})$, $j = 1, 2, \ldots, p$ are convex in X. Then the expected value model (2.18) is a convex programming.*

Proof. For each $\boldsymbol{\xi}$, since the function $f(\boldsymbol{x}, \boldsymbol{\xi})$ is convex in X, we have

$$f(\lambda \boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2, \boldsymbol{\xi}) \leq \lambda f(\boldsymbol{x}_1, \boldsymbol{\xi}) + (1-\lambda) f(\boldsymbol{x}_2, \boldsymbol{\xi}),$$

for any given solutions $\boldsymbol{x}_1, \boldsymbol{x}_2$ and any scalar $\lambda \in [0, 1]$. It follows from the expected value operator that

$$E[f(\lambda \boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2, \boldsymbol{\xi})] \leq \lambda E[f(\boldsymbol{x}_1, \boldsymbol{\xi})] + (1-\lambda) E[f(\boldsymbol{x}_2, \boldsymbol{\xi})],$$

which proves the convexity of the objective function $E[f(\boldsymbol{x}, \boldsymbol{\xi})]$ in X.

Let us prove the convexity of the feasible set by verifying that $\lambda \boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2$ is feasible for any feasible solutions $\boldsymbol{x}_1$, and $\boldsymbol{x}_2$ constrained by $E[g_j(\boldsymbol{x}, \boldsymbol{\xi})] \leq 0$, $j = 1, 2, \ldots, p$ and any scalar $\lambda \in [0, 1]$. By the convexity of the functions $g_j(\boldsymbol{x}, \boldsymbol{\xi})$, $j = 1, 2, \ldots, p$, we know that

$$g_j(\lambda \boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2, \boldsymbol{\xi}) \leq \lambda g_j(\boldsymbol{x}_1, \boldsymbol{\xi}) + (1-\lambda) g_j(\boldsymbol{x}_2, \boldsymbol{\xi}),$$

which yields that

$$E[g_j(\lambda \boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2, \boldsymbol{\xi})] \leq \lambda E[g_j(\boldsymbol{x}_1, \boldsymbol{\xi})] + (1-\lambda) E[g_j(\boldsymbol{x}_2, \boldsymbol{\xi})],$$

for $j = 1, 2, \ldots, p$. It follows that $\lambda \boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2$ is a feasible solution. Hence the feasible set is convex. This completes the proof. □

In many cases, there are usually multiple objectives which decision makers must consider. Thus we have to employ the following expected value model (EVM),

$$\begin{cases} \max[E[f_1(\boldsymbol{x}, \boldsymbol{\xi})], E[f_2(\boldsymbol{x}, \boldsymbol{\xi})], \ldots, E[f_m(\boldsymbol{x}, \boldsymbol{\xi})]] \\ \text{s.t.} \begin{cases} E[g_j(\boldsymbol{x}, \boldsymbol{\xi})] \leq 0, \ j = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{2.19}$$

where $f_i(\boldsymbol{x}, \boldsymbol{\xi})$ are return functions for $i = 1, 2, \ldots, m$. $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_n)$ is a random vector on probability space $(\Omega, \mathscr{A}, Pr)$.

Definition 2.14. $\boldsymbol{x}^*$ is the Pareto solution of problem (2.19), if there doesn't exist feasible solutions $\boldsymbol{x}$ such that

$$E[f_i(\boldsymbol{x}, \boldsymbol{\xi})] \geq E[f_i(\boldsymbol{x}^*, \boldsymbol{\xi})], i = 1, 2, \ldots, m$$

and there exists at least one $j (j = 1, 2, \ldots, m)$ such that $E[f_j(\boldsymbol{x}, \boldsymbol{\xi})] > E[f_j(\boldsymbol{x}^*, \boldsymbol{\xi})]$.

We can also formulate a stochastic decision system as an expected value goal model (EVGM) according to the priority structure and target levels set by the decision maker:

$$\begin{cases} \min \sum_{j=1}^{l} P_j \sum_{i=1}^{m} (u_{ij} d_i^+ + v_{ij} d_i^-) \\ \text{s.t.} \begin{cases} E[f_i(\boldsymbol{x}, \boldsymbol{\xi})] + d_i^- - d_i^+ = b_i, & i = 1, 2, \ldots, m \\ E[g_j(\boldsymbol{x}, \boldsymbol{\xi})] \leq 0, & j = 1, 2, \ldots, p \\ d_i^-, d_i^+ \geq 0, & i = 1, 2, \ldots, m \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{2.20}$$

where P_j is the preemptive priority factor which expresses the relative importance of various goals, $P_j >> P_{j+1}$, for all j, u_{ij} is the weighting factor corresponding to positive deviation for goal i with priority j assigned, v_{ij} is the weighting factor corresponding to negative deviation for goal i with priority j assigned, d_i^+ is the positive deviation from the target of goal i, defined as

$$d_i^+ = [E[f_i(\boldsymbol{x}, \boldsymbol{\xi})] - b_i] \vee 0,$$

d_i^- is the negative deviation from the target of goal i, defined as

$$d_i^- = [b_i - E[f_i(\boldsymbol{x}, \boldsymbol{\xi})]] \vee 0,$$

f_i is a function in goal constraints, g_j is a function in real constraints, b_i is the target value according to goal i, l is the number of priorities, m is the number of goal constraints, and p is the number of real constraints.

2.3.2 Linear Random EVM and the Weight Sum Method

Generally, many uncertain problems cannot be directly converted into crisp ones unless they have favorable properties and their random parameters have crisp distribution. For those which cannot be directly transformed, Monte Carlo simulation

is a useful tool to deal with them. Next, we will exhibit some examples which can be converted into crisp models. Let's consider the following linear random multi-objective programming (hereby, the linear relation is the relation among the coefficients $\bar{c}_i$ or $\bar{e}_r$, and the linearity and non-linearity in the following parts aims at this relationship),

$$\begin{cases} \max\left[\bar{c}_1^T\boldsymbol{x}, \bar{c}_2^T\boldsymbol{x}, \ldots, \bar{c}_m^T\boldsymbol{x}\right] \\ \text{s.t.} \begin{cases} \bar{e}_r^T\boldsymbol{x} \leq \bar{b}_r,\ r = 1, 2, \ldots, p \\ \boldsymbol{x} \geq 0 \end{cases} \end{cases} \tag{2.21}$$

where $\boldsymbol{x} \in X \subset \mathbf{R}^n$, $\bar{c}_i = (\bar{c}_{i1}, \bar{c}_{i2}, \ldots, \bar{c}_{in})^T$, $\bar{e}_r = (\bar{e}_{r1}, \bar{e}_{r2}, \ldots, \bar{e}_{rn})^T$ are random vectors, and $\bar{b}_r$ are random variables, $i = 1, 2, \ldots, m$, $r = 1, 2, \ldots, p$.

Because of the existence of some random parameters $\bar{c}_i$, $\bar{e}_r$ and $\bar{b}_r$, we cannot easily obtain its optimal solutions. Then we can obtain the following expected value model of problem (2.21),

$$\begin{cases} \max\left[E[\bar{c}_1^T\boldsymbol{x}], E[\bar{c}_2^T\boldsymbol{x}], \ldots, E[\bar{c}_m^T\boldsymbol{x}]\right] \\ \text{s.t.} \begin{cases} E[\bar{e}_r^T\boldsymbol{x}] \leq E[\bar{b}_r],\ r = 1, 2, \ldots, p \\ \boldsymbol{x} \geq 0 \end{cases} \end{cases} \tag{2.22}$$

2.3.2.1 Crisp Equivalent Model

One way to solve the expected value model is to convert it into crisp equivalent model, and then deal with it by the multi-objective programming technique.

Theorem 2.2. *Assume that random vectors $\bar{c}_i = (\bar{c}_{i1}, \bar{c}_{i2}, \ldots, \bar{c}_{in})^T$ is normally distributed with mean vectors $\mu_i^c = (\mu_{i1}^c, \mu_{i2}^c, \ldots, \mu_{in}^c)^T$ and positive definite covariance matrix V_i^c, written as $\bar{c}_i \sim \mathcal{N}(\mu_i^c, V_i^c)(i = 1, 2, \ldots, m)$ and random vectors $\bar{e}_r \sim \mathcal{N}(\mu_r^e, V_i^e)$, $\bar{b}_r \sim \mathcal{N}(\mu_r^b, (\sigma_r^b)^2)$ $(r = 1, 2, \ldots, p)$. Assume that for any $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$ and $r = 1, 2, \ldots, p$, $\bar{c}_{ij}$, $\bar{e}_{ij}$ and $\bar{b}_{ij}$ are independently random variables. Then problem (2.22) is equivalent to*

$$\begin{cases} \max\left[\mu_1^{cT}\boldsymbol{x}, \mu_2^{cT}\boldsymbol{x}, \ldots, \mu_m^{cT}\boldsymbol{x}\right] \\ s.t. \begin{cases} \mu_r^{eT}\boldsymbol{x} \leq \mu_r^b,\ r = 1, 2, \ldots, p \\ \boldsymbol{x} \geq 0 \end{cases} \end{cases} \tag{2.23}$$

Proof. Since random vector $\bar{c}_i$ is normally distributed on $(\Omega, \mathscr{A}, Pr)$ and $\bar{c}_i \sim \mathcal{N}(\mu_i^c, V_i^c)(i = 1, 2, \ldots, m)$, then

$$\bar{c}_i^T\boldsymbol{x} = \sum_{j=1}^{n} x_j\bar{c}_{ij} \sim \mathcal{N}\left(\sum_{j=1}^{n} x_j\mu_{ij}^c, \boldsymbol{x}^T V_i^c\boldsymbol{x}\right) = \mathcal{N}(\mu_i^{cT}\boldsymbol{x}, \boldsymbol{x}^T V_i^c\boldsymbol{x}),$$

so $E[\bar{c}_i^T \boldsymbol{x}] = \mu_i^{cT} \boldsymbol{x}$. Similarly, we obtain

$$E[\bar{e}_r^T \boldsymbol{x}] = \mu_r^{eT} \boldsymbol{x}, E[\bar{b}_r] = \mu_r^b, r = 1, 2, \ldots, p,$$

then it follows that problem (2.22) is equivalent to

$$\begin{cases} \max\left[\mu_1^{cT} \boldsymbol{x}, \mu_2^{cT} \boldsymbol{x}, \ldots, \mu_m^{cT} \boldsymbol{x}\right] \\ \text{s.t.} \begin{cases} \mu_r^{eT} \boldsymbol{x} \leq \mu_r^b, r = 1, 2, \ldots, p \\ \boldsymbol{x} \geq 0 \end{cases} \end{cases}$$

This completes the proof. □

Theorem 2.3. *Assume that random vector* $\bar{c}_i = (\bar{c}_{i1}, \bar{c}_{i2}, \ldots, \bar{c}_{in})^T$ *is exponentially distributed, written as* $\bar{c}_{ij} \sim exp(\lambda_{ij}^c)$ $(i = 1, 2, \ldots, m, j = 1, 2, \ldots, n)$ *and random variable* $\bar{e}_{rj} \sim exp(\lambda_{rj}^e)$, $\bar{b}_r \sim exp(\lambda_r^b)$ $(r = 1, 2, \ldots, p, j = 1, 2, \ldots, n)$. *Then problem (2.22) is equivalent to*

$$\begin{cases} \max\left[\lambda_1^{cT} \boldsymbol{x}, \lambda_2^{cT} \boldsymbol{x}, \ldots, \lambda_m^{cT} \boldsymbol{x}\right] \\ s.t. \begin{cases} \lambda_r^{eT} \boldsymbol{x} \leq \dfrac{1}{\lambda_r^b}, r = 1, 2, \ldots, p \\ \boldsymbol{x} \geq 0 \end{cases} \end{cases} \tag{2.24}$$

where $\lambda_i^c = \left(\frac{1}{\lambda_{i1}^c}, \frac{1}{\lambda_{i2}^c}, \ldots, \frac{1}{\lambda_{in}^c}\right)^T$ *and* $\lambda_i^e = \left(\frac{1}{\lambda_{r1}^e}, \frac{1}{\lambda_{r2}^e}, \ldots, \frac{1}{\lambda_{rn}^e}\right)^T$.

Proof. Since random variables $\bar{c}_{ij}$ are exponentially distributed on the probability space $(\Omega, \mathscr{A}, Pr)$ and $\bar{c}_{ij} \sim exp(\lambda_{ij}^c)$ $(i = 1, 2, \ldots, m, j = 1, 2, \ldots, n)$, then it follows from Theorem 2.1 that

$$E[\bar{c}_i^T \boldsymbol{x}] = E\left[\sum_{j=1}^n x_j \bar{c}_{ij}\right] = \sum_{j=1}^n x_j E[\bar{c}_{ij}] = \sum_{j=1}^n x_j / \lambda_{ij}^c = \lambda_i^{cT} \boldsymbol{x},$$

where $\lambda_i^c = (\frac{1}{\lambda_{i1}^c}, \frac{1}{\lambda_{i2}^c}, \ldots, \frac{1}{\lambda_{in}^c})^T, i = 1, 2, \ldots, m$. Similarly, we have

$$E[\bar{e}_r^T \boldsymbol{x}] = \lambda_r^{eT} \boldsymbol{x}, \quad E[\bar{b}_r] = \frac{1}{\lambda_r^b}$$

where $\lambda_r^e = (\frac{1}{\lambda_{r1}^e}, \frac{1}{\lambda_{r2}^e}, \ldots, \frac{1}{\lambda_{rn}^e})^T$, $r = 1, 2, \ldots, p$. Then problem (2.22) is equivalent to

$$\begin{cases} \max\left[\lambda_1^{cT} \boldsymbol{x}, \lambda_2^{cT} \boldsymbol{x}, \ldots, \lambda_m^{cT} \boldsymbol{x}\right] \\ \text{s.t.} \begin{cases} \lambda_r^{eT} \boldsymbol{x} \leq \dfrac{1}{\lambda_r^b}, r = 1, 2, \ldots, p \\ \boldsymbol{x} \geq 0 \end{cases} \end{cases}$$

where $\lambda_i^c = \left(\frac{1}{\lambda_{i1}^c}, \frac{1}{\lambda_{i2}^c}, \ldots, \frac{1}{\lambda_{in}^c}\right)^T$ and $\lambda_r^e = \left(\frac{1}{\lambda_{r1}^e}, \frac{1}{\lambda_{r2}^e}, \ldots, \frac{1}{\lambda_{rn}^e}\right)^T$. This completes the proof. □

We just take the normal distribution and the exponential distribution as examples, and readers can get the similar results when random parameters are subject to other distributions. If there are more than two different distributions in the same problem, readers can also deal with it by the expected value operator and convert it into the crisp one.

2.3.2.2 The Weight Sum Method

In this section, we will introduce the weight sum method to solve the multi-objective problem (2.23) and (2.24). Take the problem (2.23) as an example. Let $H_i(\boldsymbol{x}) = \mu_1^{cT}\boldsymbol{x}, i = 1, 2, \ldots, m$ and $X = \{\boldsymbol{x} | \mu_r^{eT}\boldsymbol{x} \leq \mu_r^b, \boldsymbol{x} \geq 0, r = 1, 2, \ldots, p\}$, then the problem (2.77) can be rewritten as,

$$\begin{cases} \max\ [H_1(\boldsymbol{x}), H_2(\boldsymbol{x}), \ldots, H_m(\boldsymbol{x})] \\ \text{s.t. } \boldsymbol{x} \in X \end{cases} \tag{2.25}$$

The weight sum method is one of the techniques which are broadly applied to solve the multi-objective programming problem. Assume that the related weight of the objective function $H_i(\boldsymbol{x})$ is w_i such that $\sum_{i=1}^m w_i = 1$ and $w_i \geq 0$. Construct the evaluation function as follows,

$$u(\boldsymbol{H}(\boldsymbol{x})) = \sum_{i=1}^{m} w_i H_i(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{H}(\boldsymbol{x}),$$

where w_i expresses the importance of the object $H_i(\boldsymbol{x})$ for DM. Then we get the following weight problem,

$$\max_{x \in X} u(\boldsymbol{H}(\boldsymbol{x})) = \max_{x \in X} \sum_{i=1}^{m} w_i H_i(\boldsymbol{x}) = \max_{x \in X} \boldsymbol{w}^T \boldsymbol{H}(\boldsymbol{x}) \tag{2.26}$$

Let $\bar{\boldsymbol{x}}$ be an optimal solution of the problem (2.26), we can easily deduce that if $\boldsymbol{w} > 0$, $\bar{\boldsymbol{x}}$ is an efficient solution of the problem (2.25). By changing $\boldsymbol{w}$, we can obtain a set composed of the efficient solutions of the problem (2.25) by solving the problem (2.26).

2.3.3 Nonlinear Random EVM and Random Simulation-Based SA

For some complex problems, it is usually difficult to convert them into crisp ones and obtain their expected values. For example, let's consider the problem:

$\max_{x \in X} E[\sqrt{(x_1 - \xi_1)^2 + (x_2 - \xi_2)^2}]$, where ξ_1 is a uniformly distributed random variable and ξ_2 is a normally distributed random variable. As we know, it is almost impossible to convert it into a crisp one. Thus, an intelligent algorithm should be provided to solve it. The technique of stochastic simulation-based SA is a useful and efficient tool when dealing with them. Let's consider the following random multi-objective problem,

$$\begin{cases} \max[E[f_1(\boldsymbol{x}, \boldsymbol{\xi})], E[f_2(\boldsymbol{x}, \boldsymbol{\xi})], \ldots, E[f_m(\boldsymbol{x}, \boldsymbol{\xi})]] \\ \text{s.t.} \begin{cases} E[g_j(\boldsymbol{x}, \boldsymbol{\xi})] \leq 0, j = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases}$$

where $f_i(\boldsymbol{x}, \boldsymbol{\xi})$ or $g_j(\boldsymbol{x}, \boldsymbol{\xi})$ or both of them are nonlinear with respect to $\boldsymbol{\xi}$, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, p$. $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_n)$ is a random vector on probability space $(\Omega, \mathscr{A}, Pr)$. Because of the existence of the nonlinear functions, we cannot usually convert it into the crisp one. Then we have to apply the technique of the random simulation (Monte Carlo Simulation) to compute its expected value.

2.3.3.1 Random Simulation for EVM

Let $\boldsymbol{\xi}$ be an n-dimensional random vector defined on the probability space $(\Omega, \mathscr{A}, Pr)$ (equivalently, it is characterized by a probability distribution $F(\cdot)$), and $f : \mathbf{R}^n \to \mathbf{R}$ a measurable function. Then $f(\boldsymbol{x}, \boldsymbol{\xi})$ is also a random variable. In order to calculate the expected value $E[f(\boldsymbol{x}, \boldsymbol{\xi})]$ for given $\{\boldsymbol{x}\}$, we generate ω_k from Ω according to the probability measure Pr, and write $\boldsymbol{\xi}_k = \boldsymbol{\xi}(\omega_k)$ for $k = 1, 2, \ldots, N$, where $\omega_k = (\omega_k^1, \ldots, \omega_k^n)$ is an n-dimensional vector and ω_k^j is generated according to the random variable ξ_k. Equivalently, we generate random vectors $\boldsymbol{\xi}_k$, $k = 1, 2, \ldots, N$ according to the probability distribution $F(\cdot)$. It follows from the strong law of large numbers that

$$\frac{\sum_{k=1}^{N} f(\boldsymbol{x}, \boldsymbol{\xi}_k)}{N} \to E[f(\boldsymbol{x}, \boldsymbol{\xi})],$$

as $N \to \infty$. Therefore, the value $E[f(\boldsymbol{x}, \boldsymbol{\xi})]$ can be estimated by $\frac{1}{N} \sum_{k=1}^{N} f(\boldsymbol{x}, \boldsymbol{\xi}_k)$ provided that N is sufficiently large. Then the procedure simulating the expected value of the function $f(\boldsymbol{x}, \boldsymbol{\xi})$ can be summarized as follows:

Procedure `Random simulation for EVM`

Input: The decision vector $\boldsymbol{x}$
Output: The expected value $E[f(\boldsymbol{x}, \boldsymbol{\xi})]$
Step 1. Set $L = 0$;
Step 2. Generate ω_k from Ω according to the probability measure Pr, $k = 1, 2, \ldots, N$;
Step 3. $L \leftarrow L + f(\boldsymbol{x}, \boldsymbol{\xi}_k)$;
Step 4. Repeat the second and third steps N times;
Step 5. Return $E[f(\boldsymbol{x}, \boldsymbol{\xi})] = L/N$.

Example 2.2. Let ξ_1 be an exponentially distributed variable $exp(1)$, ξ_2 a normally distributed variable $\mathcal{N}(3, 1)$, and ξ_3 a uniformly distributed variable $\mathcal{U}(0, 1)$. A run of stochastic simulation with 10000 cycles shows that $E[\sqrt{\xi_1^2 + \xi_2^2 + \xi_3^2}] = 3.3566$.

2.3.3.2 Simulated Annealing Algorithm

Simulated annealing algorithm (abbr. SA) are proposed by Kirkpatrick et al. [160, 161] for the problem of finding, numerically, a point of the global minimum of a function defined on a subset of a n-dimensional Euclidean space. The name of the algorithm derived from an analogy between the simulation of the annealing of solid first proposed by Metropolis et al. [218] and the strategy of solving combinatorial optimization problems. The motivation of the methods lies in the physical process of annealing, in which a solid is heated to a liquid state and, when cooled sufficiently slowly, takes up the configuration with minimal inner energy. Metropolis et al. [218] described this process mathematically. Simulating annealing uses this mathematical description for the minimization of other functions than the energy. The first results have been published by Černý [39], Kirkpatrick et al. [160,161] and Geman and Geman [107]. For a related earlier result, see Hasminskij [232]. Most of the early considerations concern minimization of functions defined on a finite set. Kushner [179] and Gelfand and Mitter [106] obtained results for functions with infinite domains. Laarhoven and Aarts [183], and Laarhoven [182] are monographs on simulated annealing. Steel [302], in a review of [182], calls simulated annealing the most exciting algorithmic development of the decade.

Annealing, physically, refers to the process of heating up a solid to a high temperature followed by slow cooling achieved by decreasing the temperature of the environment in steps. At each step the temperature is maintained constant for a period of time sufficient for the solid to reach thermal equilibrium. At equilibrium, the solid could have many configurations, each corresponding to different spins of the electrons and to specific energy level. Simulated annealing is a computational stochastic technique for obtaining near global optimum solutions to combinatorial and function optimization problems. The method is inspired from the thermodynamic process of cooling (annealing) of molten metals to attain the lowest free energy state. When molten metal is cooled slowly enough it tends to solidify in a structure of minimum energy. This annealing process is mimicked by a search strategy. The key principle of the method is to allow occasional worsening moves so that these can eventually help locate the neighborhood to the true (global) minimum. The associated mechanism is given by the Boltzmann probability, namely,

$$\text{probability } (p) = \exp\left(\frac{-\triangle E}{K_B T}\right), \tag{2.27}$$

where $\triangle E$ is the change in the energy value from one point to the next, K_B is the Boltzmanns constant and T is the temperature (control parameter). For the purpose

of optimization the energy term, $\triangle E$ refers to the value of the objective function and the temperature, T, is a control parameter that regulates the process of annealing. The consideration of such a probability distribution leads to the generation of a Markov chain of points in the problem domain. The acceptance criterion given by (2.27) is popularly referred to as the Metropolis criterion [218]. Another variant of this acceptance criterion (for both improving and deteriorating moves) has been proposed by Galuber [113] and can be written as

$$\text{probability } (p) = \frac{\exp(-\triangle E/T)}{1 + \exp(-\triangle E/T)} \tag{2.28}$$

In simulated annealing search strategy: at the start any move is accepted. This allows us to explore solution space. Then, gradually the temperature is reduced which means that one becomes more and more selective in accepting new solution. By the end, only the improving moves are accepted in practice. The temperature is systematically lowered using a problem-dependent schedule characterized by a set of decreasing temperatures.

Next, we introduce the general framework for the simulated annealing algorithm. The standard SA technique makes the analogy between the state of each molecule that determines the energy function and the value of each parameter that affects the objective functions. It then uses the statistical mechanics principle for energy minimization to minimize the objective function and optimize the parameter estimates. Starting with a high temperature, it randomly perturbs the parameter values and calculates the resulting objective function. The new state of objective function after perturbation is then accepted by a probability determined by the Metropolis criterion. The system temperature is then gradually reduced as the random perturbation proceeds, until the objective function reaches its global or nearly global minimum. A typical SA algorithm is described as follows (Fig. 2.2):

Step 1. Specify initial temperature $T_k = T_0$ for $k = 0$; randomly initialize the parameter set estimate $\theta^* = \theta_0$.

Step 2. Under kth temperature, if the inner loop break condition is met, go to step 3; otherwise, for $(j + 1)$th perturbation, randomly produce a new parameter set θ_{j+1}, compute the change in objective function $\triangle f = f(\theta^*) - f(\theta_{j+1})$. If $\triangle f \leq 0$, accept $\theta_{j+1}(\theta^* = \theta_j)$; if not, follow the Metropolis criterion to accept θ_{j+1} with a probability of $\min(1, e^{-\triangle f/T_k})$ and step 2 continues.

Step 3. Reduce T_k to T_{k+1} following a specified cooling schedule. If outer loop break condition is met, computation stops and optimal parameter set is reached; if not, return back to step 2.

The steps outlined above consist of one inner loop (step 2) and one outer loop (step 3). The proceeding of SA is mainly controlled by (1) the choice of T_0; (2) the way a new perturbation is generated; (3) the inner loop break conditions; (4) the choice of cooling schedule; and (5) the outer loop break conditions. The pseudocode can be seen in Table 2.1.

For the multiobjective optimization problem, many researchers have proposed many kinds of SA algorithms to obtain the Pareto-optimal solutions.

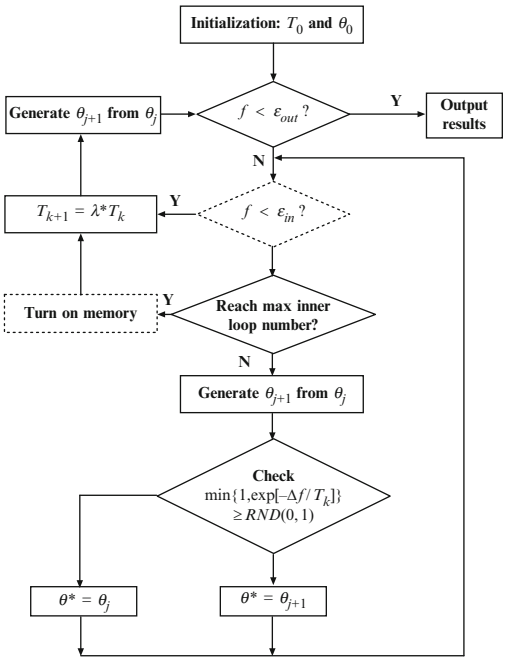


Fig. 2.2 The flow chart of modified SA algorithm

Table 2.1 Simulated annealing algorithm in pseudo-code

Select an initial temperature $T_0 > 0$;
Select an initial solution, S_0, and make it the current solution, S, and the current best solution, S^*;
repeat
  set repetition counter $n = 1$;
    repeat
      generates solution S_n in the neighborhood of S;
      calculate $\Delta = f(S_n) - f(S)$
      if $(\Delta \leq 0)$, then $S = S_n$
        else $S = S_n$ with probability of $p = e^{-\Delta/T}$
      if $(f(S_n) \leq f(S^*))$, then $S^* = S_n$
      n = n + 1;
    until $n >$ number of repetitions allowed at each temperature level (L);
  reduce the temperature T;
until stop criterion is true.

Suppapitnarm et al. [307] has used the concept of archiving the Pareto-optimal solutions coupled with return to base strategy to solve multiobjective problems with simulated annealing. They use the objective function values but not the weight vector as an acceptance criterion after penalizing them and annealing temperature and consider multiple annealing temperatures (usually one per objective). Therefore, the key probability step can be given as:

$$\text{probability } (p) = \min\left(1, \prod_{i=1}^{m} \exp\left\{\frac{-\triangle s_i}{T_i}\right\}\right), \tag{2.29}$$

where $\triangle s_i = f_i(\boldsymbol{y}) - f_i(\boldsymbol{x})$, $\boldsymbol{x}$ is the current solution, $\boldsymbol{y}$ is the generated solution, f_i is the ith objective function, T_i is the ith annealing temperature and m is the number of objective functions. On the other hand, the penalty function approach can help us to convert the constrained problem to an unconstrained one.

Above all, the SA algorithm which is proposed to solve the multiobjective programming problem(m objective functions and n decision variables) by Suppapitnarm et al. [307] (abbr. SMOSA) can be summarized as follows:

Procedure `The SMOSA algorithm`

Input: The initial temperature T_0
Output: The Pareto-solution $\boldsymbol{x}^*$
Step 1. Randomly generate a feasible $\boldsymbol{x}$ by random simulation and put $\boldsymbol{x}$ into a Pareto set of solutions. Compute all objective values;
Step 2. Generate a new solution $\boldsymbol{y}$ in the neighborhood of $\boldsymbol{x}$ by the random perturbation. Compute the objective values and apply the penalty function approach to the corresponding objective functions, if necessary;
Step 3. Compare the generated solution with all solutions in the Pareto set and update the Pareto set if necessary;
Step 4. Replace the current solution $\boldsymbol{x}$ with the generated solution $\boldsymbol{y}$ if $\boldsymbol{y}$ is archived and go to Step 7;
Step 5. Accept the generated solution $\boldsymbol{y}$ as the current solution if it is not archived with the probability:

$$\text{probability } (p) = \min\left(1, \prod_{i=1}^{m} \exp\left\{\frac{-\triangle s_i}{T_i}\right\}\right),$$

where $\triangle s_i = f_i(\boldsymbol{y}) - f_i(\boldsymbol{x})$. If the generated solution is accepted, replace $\boldsymbol{x}$ with $\boldsymbol{y}$ and go to Step 7;
Step 6. If the generated solution as current solution vector by $\boldsymbol{x} = \boldsymbol{x}$ and go to Step 7;
Step 7. Periodically, restart with a randomly selected solution from the Pareto set. While periodically restarting with the archived solutions, Suppapitnarm et al. [307] have recommended biasing towards the extreme ends of the trade-off surface;
Step 8. Periodically reduce the temperature using a problem-dependent annealing schedule;
Step 9. Repeat steps 2–8, until a predefined number of iterations is carried out.

There are also many other SA algorithms designed to solve multiobjective programming problems by many scholars. For example, Ulungu et al. [323, 324]

proposed UMOSA to project the multidimensional objective space into a mono dimensional space using weighed sum-scalarizing technique. Czyżak et al. [70] and Czyżak and Jaszkiewicz [71] proposed the PSA algorithm which modified the procedure Ulungu et al. [323] introduced by using the concept of interacting efficient solutions which are obtained by combining unicriterion simulated annealing and genetic algorithm. Suman [304, 305] proposed the WMOSA algorithm to handle constraints with its main algorithm by using weight vector in the acceptance criterion. Suman [306] also proposed the PDMOSA algorithm which uses the fitness value in the acceptance criteria to handle the multiobjective optimization problems. We consider only the random simulation-based SMOSA in this book and readers can find more detail about the multiobjective optimization problem solved by SA in [306].

2.3.4 Numerical Examples

Example 2.3. Let's consider the following programming problem,

$$\begin{cases} \max \ f_1(\boldsymbol{x}, \boldsymbol{\xi}) = 3\xi_1 x_1 - 2\xi_2 x_2 + 1.3\xi_3 x_3 \\ \max \ f_2(\boldsymbol{x}, \boldsymbol{\xi}) = -2.5\xi_1 x_1 + 3\xi_2 x_2 + 5\xi_3 x_3 \\ \text{s.t.} \begin{cases} x_1 + x_2 + x_3 \leq 10 \\ 3x_1 + 5x_2 + 3x_3 \geq 4 \\ x_1, x_2, x_3 \geq 0 \end{cases} \end{cases} \tag{2.30}$$

where ξ_1 is a random variable with 0–1 distribution, $Pr\{\xi_1 = 1\} = 0.8$, $\xi_2 \sim B(20, 0.6)$ is a binomially distributed random variable and $\xi_3 \sim P(5)$ is a poisson distributed random variable. Since there are uncertain variables in objective functions, we usually aims at the expected values of objective functions, then problem (2.30) can be rewritten as

$$\begin{cases} \max \ E[f_1(\boldsymbol{x}, \boldsymbol{\xi})] = E[3\xi_1 x_1 - 2\xi_2 x_2 + 1.3\xi_3 x_3] \\ \max \ E[f_2(\boldsymbol{x}, \boldsymbol{\xi})] = E[-2.5\xi_1 x_1 + 3\xi_2 x_2 + 5\xi_3 x_3] \\ \text{s.t.} \begin{cases} x_1 + x_2 + x_3 \leq 10 \\ 3x_1 + 5x_2 + 3x_3 \geq 4 \\ x_1, x_2, x_3 \geq 0 \end{cases} \end{cases} \tag{2.31}$$

Then we have

$$E[\xi_1] = 0.8, \ E[\xi_2] = 12, \ E[\xi_3] = 5$$

It follows from the linearity of expected value operator that, (2.31) can be rewritten as

Table 2.2 The optimal solution of expected value model

w_1	w_2	x_1	x_2	x_3	$H_1(\boldsymbol{x}^*)$	$H_2(\boldsymbol{x}^*)$
0.1	0.9	0	10	0	−240	360
0.2	0.8	0	10	0	−240	360
0.3	0.7	0	0	10	65	250
0.4	0.6	0	0	10	65	250
0.5	0.6	0	0	10	65	250

$$\begin{cases} \max H_1(\boldsymbol{x}) = 2.4x_1 - 24x_2 + 6.5x_3 \\ \max H_2(\boldsymbol{x}) = -2x_1 + 36x_2 + 25x_3 \\ \text{s.t.} \begin{cases} x_1 + x_2 + x_3 \leq 10 \\ 3x_1 + 5x_2 + 3x_3 \geq 4 \\ x_1, x_2, x_3 \geq 0 \end{cases} \end{cases} \tag{2.32}$$

Let's firstly obtain the weight by solving $H_i^0 = \max_{\boldsymbol{x} \in X} H_i(\boldsymbol{x})$ and $w_i = H_i^0 / (H_1^0 + H_2^0)$. Then we get

$$\begin{aligned} &H_1^0 = 65.00,\ H_2^0 = 360.00 \\ &w_1 = 0.153,\ \ w_2 = 0.847 \end{aligned}$$

Then we obtain the optimal solution $\boldsymbol{x}^* = (0, 10.00, 0)^T$ and $H_1(\boldsymbol{x}^*) = -240$ and $H_2(\boldsymbol{x}^*) = 360$.

By changing the weight, we can get different solutions under different weights as shown in Table 2.2.

Example 2.4. Consider the following programming problem,

$$\begin{cases} \max f_1(\boldsymbol{x}, \boldsymbol{\xi}) = \sqrt{(\xi_1 - x_1)^2 + (\xi_2 - x_2)^2} \\ \max f_2(\boldsymbol{x}, \boldsymbol{\xi}) = \sqrt{(\xi_1 + x_1)^2 + (\xi_2 + x_2)^2} \\ \text{s.t.} \begin{cases} x_1 + x_2 \leq 5 \\ x_1 \geq 0, x_2 \geq 0 \end{cases} \end{cases} \tag{2.33}$$

where $\xi_1 \sim \mathscr{N}(2, 0.5)$ is a normally distributed random variable and $\xi_2 \sim \mathscr{U}(1, 2)$ is also a normally distributed random variable. Then by the expected value operator, we have

$$\begin{cases} \max H_1(\boldsymbol{x}, \boldsymbol{\xi}) = E\left[\sqrt{(\xi_1 - x_1)^2 + (\xi_2 - x_2)^2}\right] \\ \max H_2(\boldsymbol{x}, \boldsymbol{\xi}) = E\left[\sqrt{(\xi_1 + x_1)^2 + (\xi_2 + x_2)^2}\right] \\ \text{s.t.} \begin{cases} x_1 + x_2 \leq 5 \\ x_1 \geq 0, x_2 \geq 0 \end{cases} \end{cases} \tag{2.34}$$

Next, we will use the random simulation-based SA to solve the above problem. Set the initial temperature $T_0 = 500$, the last temperature be 0 and the cooling method be 1 decrement once. The neighborhood can be constructed as follows,

$$x_1^1 = x_1^0 + rh, \quad x_2^1 = x_2^0 + rh,$$

where r is a random number in (0,1) and h is the step length (here $h = 2.0$). After the simulation with many cycles, we get the optimal solution under different weights as shown in Table 2.3. Figure 2.3 shows the cooling process when the weight is 0.5. The real line expresses the weight sum of two objective functions, and it shows that it gradually converges from T = 360. Figure 2.4 shows the changes of two objective values when the temperature decreases.

Table 2.3 The optimal solution by random simulation-based SA

w_1	w_2	x_1	x_2	$H_1(\boldsymbol{x})$	$H_2(\boldsymbol{x})$	$H(\boldsymbol{x})$	T_0
0.1	0.9	2.2448	2.7160	1.4333	6.1385	5.6680	500
0.2	0.8	2.2823	0.0925	1.7342	6.4436	5.5017	500
0.3	0.7	0.1658	2.8159	2.5864	6.3471	5.2189	500
0.4	0.6	1.0663	1.1634	1.3053	6.1609	4.2187	500
0.5	0.5	0.0307	0.7439	1.3862	5.9708	3.6785	500

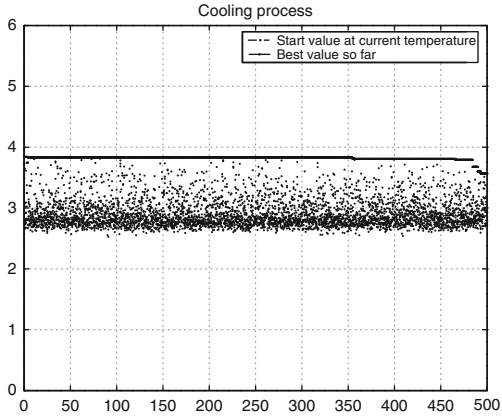


Fig. 2.3 The simulation process of random simulation-based SA

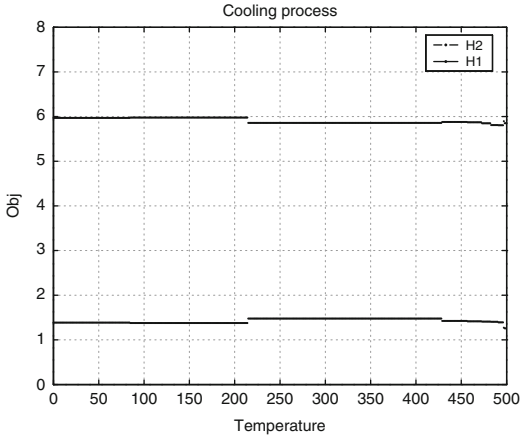


Fig. 2.4 Two objective values by random simulation-based SA

2.4 Random CCM

In 1959, Charnes and Cooper [45] developed another technique to deal with random programming problems, that is chance-constrained model (CCM). It is a powerful tool to help decision makers to make the decision in stochastic decision systems with assumption that the random constrains will hold at least α level value, where α is referred to as the confidence level provided as an appropriate safety margin by the decision maker.

This section will introduce some basic theories of chance-constrained model including chance-constrained operator, basic model, chance-constrained multi-objective model, and crisp equivalent model. Finally, the random simulation for CCM and some numerical examples will be provided.

2.4.1 General Model for Random CCM

In practice, the goal of decision makers is to maximize the objective value on the condition of probability α, where α is predetermined confidence level. Next, we will introduce the concept of the chance measure of random variables. The chance measure of a random event is considered as the probability of the event $f(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}$. Then the chance constraint is considered as $Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}\} \geq \alpha$, where α is the predetermined confidence level, and $\bar{f}$ is called the *critical value*. A natural idea is to provide a confidence level α at which it is desired that random constrains hold. Let's still consider the following model,

$$\begin{cases} \max f(\boldsymbol{x}, \boldsymbol{\xi}) \\ \text{s.t.} \begin{cases} g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, \ j = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases}$$

where $f(\boldsymbol{x}, \boldsymbol{\xi})$ and $g_j(\boldsymbol{x}, \boldsymbol{\xi})$, $j = 1, 2 \ldots, p$ are continuous functions in X and $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_n)$ is a random vector on probability space $(\Omega, \mathscr{A}, Pr)$. Based on the chance-constraint operator, the random chance-constrained model (CCM):

$$\begin{cases} \max \bar{f} \\ \text{s.t.} \begin{cases} Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}\} \geq \beta \\ Pr\{g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \geq \alpha_j, \ j = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{2.35}$$

where β and α_j are the predetermined confidence levels, $\bar{f}$ is the critical value which needs to determine.

Definition 2.15. A solution $\boldsymbol{x} \in X$ is said to be the feasible solution of problem (2.35) if and only if $Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}\} \geq \beta$ and $Pr\{g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \geq \alpha_j$ hold for

all j. For any feasible $\boldsymbol{x}$, if there is a solution $\boldsymbol{x}^*$ such $\bar{f}^* > \bar{f}$, then $\boldsymbol{x}^*$ is called the optimal solution.

If the objective is to be minimized (for example, the objective is a cost function), the CCM should be as follows,

$$\begin{cases} \min \bar{f} \\ \text{s.t.} \begin{cases} Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}) \leq \bar{f}\} \geq \beta \\ Pr\{g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \geq \alpha_j, \ j = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{2.36}$$

where β and α_j are the predetermined confidence levels. Similarly, we have the following definition.

Definition 2.16. A solution $\boldsymbol{x} \in X$ is called the feasible solution of problem (2.36) if and only if $Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}) \leq \bar{f}\} \geq \beta$ and $Pr\{g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \geq \alpha_j$ holds for all j. For any feasible $\boldsymbol{x}$, if there is a solution $\boldsymbol{x}^*$ such $\bar{f}^* < \bar{f}$, then $\boldsymbol{x}^*$ is called the optimal solution.

In practice, the real-life problems are more complex, and there usually exist multiple objectives which need decision makers to decide. Thus we have to employ the following multi-objective CCM:

$$\begin{cases} \max[\bar{f_1}, \bar{f_2}, \ldots, \bar{f_m}] \\ \text{s.t.} \begin{cases} Pr\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f_i}\} \geq \beta_i, \ i = 1, 2, \ldots, m \\ Pr\{g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \geq \alpha_j, \ j = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{2.37}$$

where β_i and α_j are the predetermined confidence levels, $\bar{f_i}$ are critical values which need to be determined.

Definition 2.17. $\boldsymbol{x} \in X$ is called the Pareto solution of problem (2.37) if there doesn't exist a feasible $\boldsymbol{x}$ such that

$$\bar{f_i} \geq \bar{f_i}^*, \ i = 1, 2, \ldots, m \tag{2.38}$$

and there at least exists one $j (j = 1, 2, \ldots, m)$ such that $\bar{f_j} > \bar{f_j}^*$.

Sometimes, we may formulate a stochastic decision system as a chance-constrained goal model (CCGM) according to the priority structure and target levels set by the decision-maker:

$$\begin{cases} \min \sum_{j=1}^{l} P_j \sum_{i=1}^{m} (u_{ij} d_i^+ + v_{ij} d_i^-) \\ \text{s.t.} \begin{cases} Pr\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) - b_i \leq d_i^+\} \geq \beta_i^+, \ i = 1, 2, \ldots, m \\ Pr\{b_i - f_i(\boldsymbol{x}, \boldsymbol{\xi}) \leq d_i^-\} \geq \beta_i^-, \ i = 1, 2, \ldots, m \\ Pr\{g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \geq \alpha_j, \qquad j = 1, 2, \ldots, p \\ d_i^-, d_i^- \geq 0, \qquad\qquad\qquad i = 1, 2, \ldots, m \end{cases} \end{cases} \tag{2.39}$$

where P_j is the preemptive priority factor which express the relative importance of various goals, $P_j >> P_{j+1}$, for all j, u_{ij} is the weighting factor corresponding to positive deviation for goal i with priority j assigned, v_{ij} is the weighting factor corresponding to negative deviation for goal i with priority j assigned, d_i^+ is the β_i^+-optimistic positive deviation from the target of goal i, defined as

$$\min\{d \vee 0 | Pr\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) - b_i \leq d_i^+\} \geq \beta_i^+\} \tag{2.40}$$

d_i^- is the β_i^--optimistic positive deviation from the target of goal i, defined as

$$\min\{d \vee 0 | Pr\{b_i - f_i(\boldsymbol{x}, \boldsymbol{\xi}) \leq d_i^-\} \geq \beta_i^-\} \tag{2.41}$$

f_i is a function in goal constraints, g_j is a function in system constraints, b_i is the target value according to goal i, l is the number of priorities, m is the number of goal constraints,and p is the number of system constraints.

Remark 2.1. If the random vector $\boldsymbol{\xi}$ degenerates to the deterministic case, then the two probabilities $Pr\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) - b_i \leq d_i^+\}$ and $Pr\{b_i - f_i(\boldsymbol{x}, \boldsymbol{\xi}) \leq d_i^-\}$ should be always 1 provided that $\beta_i^+, \beta_i^- > 0$, and

$$Pr\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) - b_i \leq d_i^+\} \geq \beta_i^+, d_i^+ \geq 0,$$
$$Pr\{b_i - f_i(\boldsymbol{x}, \boldsymbol{\xi}) \leq d_i^-\} \geq \beta_i^-, d_i^- \geq 0$$

imply that

$$d_i^+ = [f_i(\boldsymbol{x}, \boldsymbol{\xi}) - b_i] \vee 0, d_i^- = [b_i - f_i(\boldsymbol{x}, \boldsymbol{\xi})] \vee 0.$$

This coincides with the deterministic goal programming.

2.4.2 Linear Random CCM and the Lexicographic Method

As we know, traditional solution methods need the conversion of the chance constraints to their respective deterministic equivalents. However, this process is usually hard to perform and only successful for some special cases. Many scholars has developed it a lot and made great achievements.

Theorem 2.4. *Assume that the random vector $\boldsymbol{\xi}$ degenerates to a random variable ξ with distribution function Φ, and the function $g(\boldsymbol{x}, \xi)$ has the form $g(\boldsymbol{x}, \xi) = h(\boldsymbol{x}) - \xi$. Then $Pr\{g(\boldsymbol{x}, \xi) \leq 0\} \geq \alpha$ if and only if $h(\boldsymbol{x}) \leq K_\alpha$, where $K_\alpha = sup\{K | K = \Phi^{-1}(1-\alpha)\}$.*

Proof. The assumption implies that $Pr\{g(\boldsymbol{x}, \xi) \leq 0\} \geq \alpha$ can be written in the following form,

$$Pr\{h(\boldsymbol{x}) \leq \xi\} \geq \alpha \tag{2.42}$$

It is clear that, for each given confidence level $\alpha(0 < \alpha < 1)$, there exists a number K_α (may be multiple or ∞) such that

$$Pr\{K_\alpha \leq \xi\} = \alpha \tag{2.43}$$

and the probability $Pr\{\alpha \leq \xi\}$ will increase if K_α is replaced with a smaller number. Hence $Pr\{h(\boldsymbol{x}) \leq \xi\} \geq \alpha$ if and only if $h(\boldsymbol{x}) \leq K_\alpha$.

Notice that the equation $Pr\{\alpha \leq \xi\} = 1 - \Phi(K_\alpha)$ always holds, and we have, by (2.43),

$$K_\alpha = \Phi^{-1}(1-\alpha),$$

where Φ^{-1} is the inverse function of Φ. Sometimes, the solution of (2.43) is not unique. Equivalently, the function Φ^{-1} is multi-valued. For this case, we should choose it as the largest one, i.e.,

$$K_\alpha = sup\{K | K = \Phi^{-1}(1-\alpha)\}.$$

Thus the deterministic equivalent is $h(\boldsymbol{x}) \leq K_\alpha$. The theorem is proved. □

Theorem 2.5. *Assume that the random vector* $\boldsymbol{\xi} = (a_1, a_2, \ldots, a_n, b)$ *and the function* $g(\boldsymbol{x}, \boldsymbol{\xi})$ *has the form* $g(\boldsymbol{x}, \boldsymbol{\xi}) = a_1x_1 + a_2x_2 + \cdots + a_nx_n - b$. *If* a_i *and* b *are assumed to be independently normally distributed variables, then* $Pr\{g(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \geq \alpha$ *if and only if*

$$\sum_{i=1}^{n} E[a_i]x_i + \Phi^{-1}(\alpha)\sqrt{\sum_{i=1}^{n} V[a_i]x_i^2 + V[b]} \leq E[b] \tag{2.44}$$

where Φ *is the standardized normal distribution.*

Proof. The chance constraint $Pr\{g(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \geq \alpha$ can be written in the following form,

$$Pr\left\{\sum_{i=1}^{n} a_i x_i \leq b\right\} \geq \alpha \tag{2.45}$$

Since a_i and b are assumed to be independently normally distributed variables, the function

$$y(\boldsymbol{x}) = \sum_{i=1}^{n} a_i x_i - b$$

is also normally distributed with the following expected value and variance,

$$E[y(\boldsymbol{x})] = \sum_{i=1}^{n} E[a_i]x_i - E[b],$$
$$V[y(\boldsymbol{x})] = \sum_{i=1}^{n} V[a_i]x_i^2 + V[b].$$

We note that

$$\frac{\sum_{i=1}^{n} a_i x_i - b - (\sum_{i=1}^{n} E[a_i]x_i - E[b])}{\sqrt{\sum_{i=1}^{n}\sum_{i=1}^{n} V[a_i]x_i^2 + V[b]}}$$

must be standardized normally distributed. Since the inequality $\sum_{i=1}^{n} a_i x_i \leq b$ is equivalent to

$$\frac{\sum_{i=1}^{n} a_i x_i - b - \left(\sum_{i=1}^{n} E[a_i]x_i - E[b]\right)}{\sqrt{\sum_{i=1}^{n}\sum_{i=1}^{n} V[a_i]x_i^2 + V[b]}} \leq -\frac{\sum_{i=1}^{n} E[a_i]x_i - E[b]}{\sqrt{\sum_{i=1}^{n}\sum_{i=1}^{n} V[a_i]x_i^2 + V[b]}}.$$

The chance constraint (2.45) is equivalent to

$$Pr\left\{\eta \leq -\frac{\sum_{i=1}^{n} E[a_i]x_i - E[b]}{\sqrt{\sum_{i=1}^{n}\sum_{i=1}^{n} V[a_i]x_i^2 + V[b]}}\right\} \geq \alpha \tag{2.46}$$

where η is the standardized normally distributed variable. Then the chance constraint (2.46) holds if and only if

$$\Phi^{-1}(\alpha) \leq -\frac{\sum_{i=1}^{n} E[a_i]x_i - E[b]}{\sqrt{\sum_{i=1}^{n}\sum_{i=1}^{n} V[a_i]x_i^2 + V[b]}}\alpha$$

That is, the deterministic equivalent of chance constraint is (2.44). The theorem is proved. □

2.4.2.1 Crisp Equivalent Model

Then let's still consider a class of multi-objective linear models as follows,

$$\begin{cases} \max\left[\bar{c}_1^T\boldsymbol{x}, \bar{c}_2^T\boldsymbol{x}, \ldots, \bar{c}_m^T\boldsymbol{x}\right] \\ \text{s.t.} \begin{cases} \bar{e}_r^T\boldsymbol{x} \le \bar{b}_r, \ r = 1, 2, \ldots, p \\ \boldsymbol{x} \ge 0 \end{cases} \end{cases} \tag{2.47}$$

where $\boldsymbol{x} \in X$, $\bar{c}_i = (\bar{c}_{i1}, \bar{c}_{i2}, \ldots, \bar{c}_{in})^T$, $\bar{e}_r = (\bar{e}_{r1}, \bar{e}_{r2}, \ldots, \bar{e}_{rn})^T$ are random vectors, and $\bar{b}_r$ are random variables, $i = 1, 2, \ldots, m$, $r = 1, 2, \ldots, p$.

Based on the chance-constrained operator, the CCM of problem (2.47) can be defined as

$$\begin{cases} \max[\bar{f}_1, \bar{f}_2, \ldots, \bar{f}_m] \\ \text{s.t.} \begin{cases} Pr\{\bar{c}_i^T\boldsymbol{x} \ge \bar{f}_i\} \ge \beta_i, & i = 1, 2, \ldots, m \\ Pr\{\bar{e}_r^T\boldsymbol{x} \le \bar{b}_r\} \ge \alpha_i, & r = 1, 2, \ldots, p \\ \boldsymbol{x} \ge 0, 0 \le \alpha_r, \beta_i \le 1 \end{cases} \end{cases} \tag{2.48}$$

where α_i and β_i are predetermined confidence levels decision makers gave.

Theorem 2.6. *Assume that random vector $\bar{c}_i = (\bar{c}_{i1}, \bar{c}_{i2}, \ldots, \bar{c}_{in})^T$ is normally distributed with mean vector $\mu_i^c = (\mu_{i1}^c, \mu_{i2}^c, \ldots, \mu_{in}^c)^T$ and positive definite covariance matrix V_i^c, written as $\bar{c}_i \sim \mathcal{N}(\mu_i^c, V_i^c)(i=1, 2, \ldots, m)$ and random vector $\bar{e}_r \sim \mathcal{N}(\mu_r^e, V_r^e)$, $\bar{b}_r \sim \mathcal{N}(\mu_r^b, (\sigma_r^b)^2)$ $(r = 1, 2, \ldots, p)$. Assume that for any $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$ and $r = 1, 2, \ldots, p$, $\bar{c}_{ij}$, $\bar{e}_{ij}$ and $\bar{b}_{ij}$ are independently random variables. Then problem (2.48) is equivalent to*

$$\begin{cases} \max[H_1(\boldsymbol{x}), H_2(\boldsymbol{x}), \ldots, H_m(\boldsymbol{x})] \\ s.t. \begin{cases} g_r(\boldsymbol{x}) \le 0, r = 1, 2, \ldots, p \\ \boldsymbol{x} \ge 0, 0 \le \alpha_r, \beta_i \le 1 \end{cases} \end{cases}$$

where $H_i(\boldsymbol{x}) = \Phi^{-1}(1-\beta_i)\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}} + \mu_i^{cT}\boldsymbol{x}$, $g_r(\boldsymbol{x}) = \Phi^{-1}(\alpha_r)$ $\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2} + \mu_r^{eT}\boldsymbol{x} - \mu_r^b$ and Φ is the standardized normal distribution.

Proof. Since random vector $\bar{c}_i$ is normally distributed with mean vector μ_i^c and positive definite covariance matrix V_i^c, then we have that

$$\bar{c}_i^T\boldsymbol{x} = \sum_{j=1}^{n} x_j\bar{c}_{ij} \sim \mathcal{N}(\sum_{j=1}^{n} x_j\mu_{ij}^c, \boldsymbol{x}^T V_i^c \boldsymbol{x}) = \mathcal{N}(\mu_i^{cT}\boldsymbol{x}, \boldsymbol{x}^T V_i^c \boldsymbol{x}).$$

It follows that

$$\frac{\bar{c}_i^T\boldsymbol{x} - \mu_i^{cT}\boldsymbol{x}}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}$$

must be standardized normally distributed. For any $i\,(i = 1, 2, \ldots, m)$, we have

$$
\begin{aligned}
Pr\{\bar{c}_i^T \boldsymbol{x} \geq \bar{f}_i\} \geq \beta_i \Leftrightarrow \beta_i &\leq Pr\left\{\frac{\bar{c}_i^T x - \mu_i^{cT} x}{\sqrt{x^T V_i^c x}} \geq \frac{\bar{f}_i - \mu_i^{cT} x}{\sqrt{x^T V_i^c x}}\right\} \\
&\Leftrightarrow \beta_i \leq 1 - Pr\left\{\frac{\bar{c}_i^T x - \mu_i^{cT} x}{\sqrt{x^T V_i^c x}} \leq \frac{\bar{f}_i - \mu_i^{cT} x}{\sqrt{x^T V_i^c x}}\right\} \\
&\Leftrightarrow \beta_i \leq 1 - \Phi\left(\frac{\bar{f}_i - \mu_i^{cT} x}{\sqrt{x^T V_i^c x}}\right) \\
&\Leftrightarrow \bar{f}_i \leq \Phi^{-1}(1 - \beta_i)\sqrt{x^T V_i^c x} + \mu_i^{cT} x
\end{aligned}
$$

where Φ is the standardized normal distribution. Similarly, we know that

$$\bar{e}_r^T \boldsymbol{x} \sim \mathcal{N}(\mu_r^{eT} \boldsymbol{x}, \boldsymbol{x}^T V_r^e \boldsymbol{x}).$$

Since $\bar{b}_r \sim \mathcal{N}(\mu_r^b, (\sigma_r^b)^2)$, it follows from, $\bar{e}_{ij}$ and $\bar{b}_{ij}$ are independently random variables for any r, j, that

$$
\begin{aligned}
E[\bar{e}_r^T \boldsymbol{x} - \bar{b}_r] &= E[\bar{e}_r^T \boldsymbol{x}] - E[\bar{b}_r] = \mu_r^{eT} \boldsymbol{x} - \mu_r^b, \\
V[\bar{e}_r^T \boldsymbol{x} - \bar{b}_r] &= V[\bar{e}_r^T \boldsymbol{x}] + V[\bar{b}_r] = \boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2,
\end{aligned}
$$

then $\bar{e}_r^T \boldsymbol{x} - \bar{b}_r$ is also normally distributed with

$$\bar{e}_r^T \boldsymbol{x} - \bar{b}_r \sim \mathcal{N}(\mu_r^{eT} \boldsymbol{x} - \mu_r^b, \boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2).$$

For any $r\,(r = 1, 2, \ldots, p)$, we have

$$
\begin{aligned}
Pr\{\bar{e}_r^T \boldsymbol{x} \geq \bar{b}_r\} \geq \alpha_r \Leftrightarrow \alpha_r &\leq Pr\left\{\frac{\bar{e}_r^T \boldsymbol{x} - \bar{b}_r - (\mu_r^{eT} \boldsymbol{x} - \mu_r^b)}{\sqrt{x^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2}} \right. \\
&\qquad \left. \leq -\frac{\mu_r^{eT} \boldsymbol{x} - \mu_r^b}{\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2}}\right\} \\
&\Leftrightarrow \alpha_r \leq \Phi\left(-\frac{\mu_r^{eT} \boldsymbol{x} - \mu_r^b}{\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2}}\right) \\
&\Leftrightarrow \Phi^{-1}(\alpha_r)\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2} + \mu_r^{eT} \boldsymbol{x} - \mu_r^b \leq 0
\end{aligned}
$$

where Φ is the standardized normal distribution. Then we have (2.48) is equivalent to

$$\begin{cases} \max[\bar{f}_1, \bar{f}_2, \ldots, \bar{f}_m] \\ \text{s.t.} \begin{cases} \bar{f}_i \le \Phi^{-1}(1-\beta_i)\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}} + \mu_i^{cT}\boldsymbol{x},\ i = 1, 2, \ldots, m \\ \Phi^{-1}(\alpha_r)\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2} + \mu_r^{eT}\boldsymbol{x} - \mu_r^b \le 0,\ r = 1, 2, \ldots, p \\ \boldsymbol{x} \ge 0,\ 0 \le \alpha_r, \beta_i \le 1 \end{cases} \end{cases} \tag{2.49}$$

Let $H_i(\boldsymbol{x}) = \Phi^{-1}(1-\beta_i)\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}} + \mu_i^{cT}\boldsymbol{x}$ and $g_r(\boldsymbol{x}) = \Phi^{-1}(\alpha_r)\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2} + \mu_r^{eT}\boldsymbol{x} - \mu_r^b$. In view of the purpose of maximizing $\bar{f}_i$ in problem (2.49), it also can be converted into

$$\begin{cases} \max[H_1(\boldsymbol{x}), H_2(\boldsymbol{x}), \ldots, H_m(\boldsymbol{x})] \\ \text{s.t.} \begin{cases} g_r(\boldsymbol{x}) \le 0, r = 1, 2, \ldots, p \\ \boldsymbol{x} \ge 0, 0 \le \alpha_r, \beta_i \le 1 \end{cases} \end{cases} \tag{2.50}$$

This completes the proof. □

For the crisp multi-objective programming problem (2.50), whether there are some optimal solutions is the spot we pay more attention to. Sometimes, the convexity of the multi-objective problem provides an useful rule researchers consult. Next, let's introduce the definition of the convexity of the multi-objective problem.

Definition 2.18. For the following multi-objective programming problem,

$$\begin{cases} \min f(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})) \\ s.t.\ g_r(\boldsymbol{x}) \ge 0, r = 1, 2, \ldots, p \end{cases} \tag{2.51}$$

If $f_i(\boldsymbol{x})$ are convex functions for all i, and $g_r(\boldsymbol{x})$ are concave functions for all r, then (2.51) is a convex programming.

Then let's discuss the convexity of the problem (2.50).

Theorem 2.7. *Let* $X = \{\boldsymbol{x} \in R^n | \Phi^{-1}(\alpha_r)\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2} + \mu_r^{eT}\boldsymbol{x} - \mu_r^b \le 0, r = 1, 2, \ldots, p; \boldsymbol{x} \ge 0\}$. *If* α_r *and* β_i *are more than 0.5,* $\bar{c}_{ij}$, $\bar{e}_{rj}$ *and* $\bar{b}_r$ *are respectively independent,* $i = 1, 2, \ldots, m, j = 1, 2, \ldots, n, r = 1, 2, \ldots, p$, *then the problem (2.50) is convex.*

Proof. Since $\bar{c}_{ij}$ are independent random variable with each other, the covariance matrix has the following form,

$$V_i^c = \begin{pmatrix} (\sigma_{i1}^c)^2 & 0 & \cdots & 0 \\ \vdots & & \cdots & \vdots \\ 0 & 0 & \cdots & (\sigma_{in}^c)^2 \end{pmatrix}.$$

It's obvious that V_i^c is a positive definite matrix. Then according to [301], $\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}$ is a convex function. In addition, since $\alpha_r \geq 0.5$ and $\beta_i \geq 0.5$, it follows that $\Phi^{-1}(\alpha_r) \geq 0$ and $\Phi^{-1}(1-\beta_i) \leq 0$, we know $H_i(\boldsymbol{x})$ are concave functions, $i = 1, 2, \ldots, m$. Really, let $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ be any two points in feasible set, and $\lambda \in [0, 1]$, we have that

$$H_i[\lambda \boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2] \geq \lambda H_i(\boldsymbol{x}_1) + (1-\alpha) H_i(\boldsymbol{x}_2).$$

Next, we prove that X is convex. Since

$$X = \{\boldsymbol{x} \in R^n | \Phi^{-1}(\alpha_r)\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2} + \mu_r^{eT}\boldsymbol{x} - \mu_r^b \leq 0, r = 1, 2, \ldots, p; \boldsymbol{x} \geq 0\},$$

and $\Phi^{-1}(\alpha_r) \geq 0$, it follows that $g_r(\boldsymbol{x}) = \Phi^{-1}(\alpha_r)\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2} + \mu_r^{eT}\boldsymbol{x} - \mu_r^b \leq 0$ are convex functions, $r = 1, 2, \ldots, p$. Really, let $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ be two feasible solutions, then

$$g_r(\boldsymbol{x}_1) \leq 0, g_r(\boldsymbol{x}_2) \leq 0,$$

according to g_r's convexity, we have

$$g_r[\lambda \boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2] \leq \lambda g_r(\boldsymbol{x}_1) + (1-\lambda) g_r(\boldsymbol{x}_2) \leq 0,$$

where $0 \leq \lambda \leq 1$, $r = 1, 2, \ldots, p$. This means that $\lambda \boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2$ is also a feasible solution. So X is a convex set. Above all, we can conclude from Definition 2.18 that problem (2.50) is a convex programming and its global optimal solution can be obtained easily. □

2.4.2.2 Lexicographic Method

The basic idea of lexicographic method is to rank the objective function by its importance to decision makers and then resolve the next objective function after resolving the above one. We take the solution of the last programming problem as the final solution.

Consider the following multi-objective programming problem,

$$\begin{cases} \min[f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})] \\ \text{s.t. } \boldsymbol{x} \in X. \end{cases} \tag{2.52}$$

Without loss of generality, assume the rank as $f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})$ according to different importance. Solve the following single objective problem in turn,

$$\min_{x \in X} f_1(\boldsymbol{x}) \tag{2.53}$$

$$\begin{cases} \min f_i(\boldsymbol{x}) \\ \text{s.t.} \begin{cases} f_k(\boldsymbol{x}) = f_k(\boldsymbol{x}^k),\ k = 1, 2, \ldots, i-1 \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{2.54}$$

where $i = 1, 2, \ldots, m$, X is the feasible area and denote the feasible area of problem (2.54) as X^i.

Theorem 2.8. *Let $X \subset \mathbf{R}^n$, $\boldsymbol{f} : X \to \mathbf{R}^m$. If $\boldsymbol{x}^m$ be the optimal solution by the lexicographic method, then $\boldsymbol{x}^m$ is an efficient solution of problem (2.52).*

Proof. If $\boldsymbol{x}^m$ is not an efficient solution of problem (2.52), there exists $\bar{\boldsymbol{x}} \in X$ such that $\boldsymbol{f}(\bar{\boldsymbol{x}}) \leq \boldsymbol{f}(\boldsymbol{x}^m)$. Since $f_1(\boldsymbol{x}^m) = f_1^* = f_1(\boldsymbol{x}^1)$, $f_1(\bar{\boldsymbol{x}}) < f_1(\boldsymbol{x}^m)$ cannot hold. It necessarily follows that $f_1(\bar{\boldsymbol{x}}) = f_1(\boldsymbol{x}^m)$.

If we have proved $f_k(\bar{\boldsymbol{x}}) = f_k(\boldsymbol{x}^m)$ $(k = 1, 2, \ldots, i-1)$, but $f_i(\bar{\boldsymbol{x}}) < f_i(\boldsymbol{x}^m)$. It follows that $\bar{\boldsymbol{x}}$ is a feasible solution of problem (2.54). Since $f_i(\bar{\boldsymbol{x}}) < f_i(\boldsymbol{x}^m) = f_i(\boldsymbol{x}^i)$, this results in the conflict with that $\boldsymbol{x}^i$ the optimal solution of problem (2.54). Thus, $f_k(\bar{\boldsymbol{x}}) = f_k(\boldsymbol{x}^m)(k = 1, 2, \ldots, i)$ necessarily holds. Then we can prove $f_k(\bar{\boldsymbol{x}}) = f_k(\boldsymbol{x}^m)(k = 1, 2, \ldots, m)$ by the mathematical induction. This conflicts with $\boldsymbol{f}(\bar{\boldsymbol{x}}) \leq \boldsymbol{f}(\boldsymbol{x}^m)$. This completes the proof. □

2.4.3 Nonlinear Random CCM and Random Simulation-Based ASA

Let's return to the nonlinear multi-objective programming problem,

$$\begin{cases} \max[\bar{f_1}, \bar{f_2}, \ldots, \bar{f_m}] \\ \text{s.t.} \begin{cases} Pr\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f_i}\} \geq \beta_i,\ i = 1, 2, \ldots, m \\ Pr\{g_r(\boldsymbol{x}, \boldsymbol{\xi}) \leq b_r\} \geq \alpha_r,\ r = 1, 2, \ldots, p \\ \boldsymbol{x} \geq 0, 0 \leq \alpha_r, \beta_i \leq 1 \end{cases} \end{cases} \tag{2.55}$$

where $f_i(\boldsymbol{x}, \boldsymbol{\xi})$ or $g_r(\boldsymbol{x}, \boldsymbol{\xi})$ or both of them are nonlinear functions with respect $\boldsymbol{\xi}$ is a random vector, $i = 1, 2, \ldots, m$, $r = 1, 2, \ldots, p$. If their distributions are all crisp, we can convert it into a crisp programming problem. For the class of problems which cannot be converted into crisp ones, the random simulation is a useful tool to compute the probability of a random variable.

2.4.3.1 Random Simulation for CCM

Suppose that $\boldsymbol{\xi}$ is an n-dimensional random vector defined on the probability space $(\Omega, \mathscr{A}, Pr)$, and $f : \mathbf{R}^n \to \mathbf{R}$ is a measurable function. The problem is to determine the maximal value $\bar{f}$ for given $\boldsymbol{x}$ such that

$$Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}\} \geq \alpha \tag{2.56}$$

where α is a predetermined confidence level with $0 < \alpha < 1$. We generate ω_k from Ω according to the probability measure Pr, and write $\boldsymbol{\xi}_k = \boldsymbol{\xi}(\omega_k)$ for $k = 1, 2, \ldots, N$, where $\boldsymbol{\omega_k} = (\omega_k^1, \ldots, \omega_k^n)$ is an n-dimensional vector and ω_k^j is generated according to the random variable ξ_k. Now we define

$$h(\boldsymbol{x}, \boldsymbol{\xi}_k) = \begin{cases} 1, \text{ if } f(\boldsymbol{x}, \boldsymbol{\xi}_k) \geq \bar{f} \\ 0, \text{ otherwise} \end{cases} \tag{2.57}$$

for $k = 1, 2, \ldots, N$, which are a sequence of random variables, and $E[h(\boldsymbol{x}, \boldsymbol{\xi}_k)] = \alpha$ for all k. By the strong law of large numbers, we obtain

$$\frac{\sum_{k=1}^{N} h(\boldsymbol{x}, \boldsymbol{\xi}_k)}{N} \to \alpha,$$

as $N \to \infty$. Note that the sum $\sum_{k=1}^{N} h(\boldsymbol{x}, \boldsymbol{\xi}_k)$ is just the number of $\boldsymbol{\xi}_k$ satisfying $f(\boldsymbol{x}, \boldsymbol{\xi}_k) \geq \bar{f}$ for $k = 1, 2, \ldots, N$. Thus the value $\bar{f}$ can be taken as the N'th largest element in the sequence $\{f(\boldsymbol{x}, \boldsymbol{\xi}_1), f(\boldsymbol{x}, \boldsymbol{\xi}_2), \ldots, f(\boldsymbol{x}, \boldsymbol{\xi}_N)\}$, where N' is the integer part of αN. Then the procedure simulating the critical value $\bar{f}$ of $Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}_i\}$ can be summarized as follows:

Procedure `Random simulation for CCM`

Input: The decision vector $\boldsymbol{x}$
Output: The critical value $\bar{f}$
Step 1. Set $L = 0$;
Step 2. Generate $\omega_1, \omega_2, \ldots, \omega_N$ from Ω according to the probability measure Pr;
Step 3. Return the N'th largest element in $\{f(\boldsymbol{x}, \boldsymbol{\xi}_1), f(\boldsymbol{x}, \boldsymbol{\xi}_2), \ldots, f(\boldsymbol{x}, \boldsymbol{\xi}_N)\}$.

Example 2.5. Let us employ the random simulation to search for the maximal $\bar{f}$ such that

$$Pr\left\{\sqrt{\xi_1^2 + \xi_2^2 + \xi_3^2} \geq \bar{f}\right\} \geq 0.8,$$

where $\xi_1 \sim exp(1)$ is an exponentially distributed variable, $\xi_2 \sim \mathcal{N}(3, 1)$ is a normally distributed variable, and $\xi_3 \sim \mathcal{U}(0, 1)$ is a uniformly distributed variable. A run of stochastic simulation with 10000 cycles shows that $\bar{f} = 2.0910$.

2.4.3.2 Adaptive Simulated Algorithm

Too often the management of complex systems is ill-served by not utilizing the best tools available. For example, requirements set by decision-makers often are not formulated in the same language as constructs formulated by powerful mathematical

formalisms, and so the products of analyses are not properly or maximally utilized, even if and when they come close to faithfully representing the powerful intuitions they are supposed to model. In turn, even powerful mathematical constructs are ill-served, especially when dealing with approximations to satisfy constraints of numerical algorithms familiar to particular analysts, but which tend to destroy the power of the intuitive constructs developed by decision-makers. In order to deal with fitting parameters or exploring sensitivities of variables, as models of systems have become more sophisticated in describing complex behavior, it has become increasingly important to retain and respect the nonlinearities inherent in these models, as they are indeed present in the complex systems they model. The adaptive simulated algorithm (abbr. ASA) can help to handle these fits of nonlinear models of real-world data.

ASA, also known as the very fast simulated reannealing, is a very efficient version of SA. ASA is a global optimization technique with certain advantages. It is versatile and needs very few parameters to tune. The distinct feature of this method is the temperature change mechanism, which is an important part of the transition probability equation. The conventional can allows a higher chance of transition to a worse solution when beginning with a high temperature. By doing so, the search can move out of the local optimization. However, as the search process develops, the continuously declining temperature will result in a reduced chance of uphill transition. Such an approach could be useful if the local optimization is near the start point, but may not lead to a near optimal solution if some local optimization is encountered at a relatively low temperature toward the end of the search. Therefore, a improved method should be considered to alleviated this difficulty. In conventional method, the cooling schedule is usually monotonically nonincreasing. In ASA, an adaptive cooling schedule based on the profile of the search path to dynamically adjust the temperature. Such adjustments could be in any direction including the possibility of reheating. Some scholars have proposed the idea of reversing the temperature before. Dowsland [83] considered two functions together to control temperature in the application to packing problem. The first function is a function that reduces the temperature, whereas the second function is used as a heating up function that gradually increases the temperature if needed. In ASA, a single function is proposed to maintain the temperature above a minimum level. If there is any upward move, the heating process gradually takes place, but the cooling process may suddenly take place by the first downhill move. Azizi [12] proposed the following temperature control function,

$$T_i = T_{\min} + \lambda ln(1 + r_i), \tag{2.58}$$

where $T_{\min}$ is the minimum value that the temperature can take, λ is a coefficient that controls the rate of temperature rise, and r_i is the number of consecutive upward moves at iteration i. The initial value of r_i is zero, thus the initial temperature $T_0 = T_{\min}$. The purpose of the minimum temperature, $T_{\min}$, is two-fold. Firstly, it prevents the probability function from becoming invalid when r_i is zero. Secondly, it determines the initial value of the temperature. $T_{\min}$ can take any value greater

than zero. The parameter λ controls the rate of temperature rise. The greater value of λ, the faster the temperature rises. The search spends less time looking for good solutions in its current neighborhood when a large value is assigned to λ. Similarly, by assigning a small value to the parameter λ, the search spends more time looking for better solutions in the neighborhood. Choosing a value for the parameter λ could be linked to computation time, which also depends on the size and complexity of the problem. We don't clarify other parts about ASA, readers can find more detailed analysis of the algorithm in [12, 52, 141–143].

Consider the following general optimization problem,

$$\min_{\boldsymbol{x} \in X} f(\boldsymbol{x}) \tag{2.59}$$

where $\boldsymbol{x} = [x_1, \ldots, x_n]^T$ is the n-dimensional decision vector to be optimized, X is the feasible set of $\boldsymbol{x}$. The cost function $f(\boldsymbol{x})$ can be multi-modal and non-smooth. The ASA is a global optimization scheme for solving this kind of constrained optimization problems. Next, let's introduce how to use ASA to solve the above problem.

Although there are many possible realizations of the ASA, an implementation is illustrated in Fig. 2.5, and this algorithm is detailed as follows:

Step 1. An initial $\boldsymbol{x} \in X$ is randomly generated, the initial temperature of the acceptance probability function, $T_{\text{accept}}(0)$, is set to $f(\boldsymbol{x})$, and the initial temperatures of the parameter generating probability functions, $T_{i,gen}(0)$, $1 \leq i \leq n$, are set

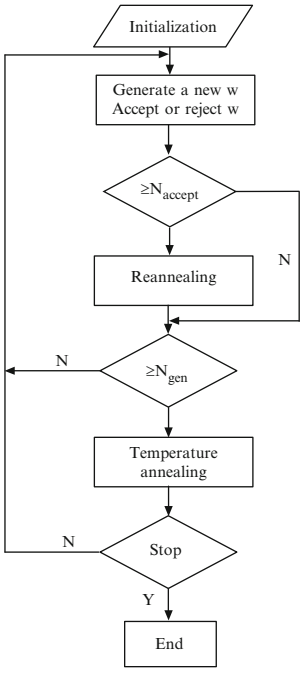


Fig. 2.5 Flow chart of ASA algorithm

to 1.0. A user-defined control parameter c in annealing is given, and the annealing times, k_i for $1 \leq i \leq n$ and k_a, are all set to 0.

Step 2. The algorithm generates a new point in the parameter space with

$$x_i^{\text{new}} = x_i^{\text{old}} + q_i M_i, \ 1 \leq i \leq n \tag{2.60}$$

where M_i is a positive number determined by the difference between the upper and lower boundary of x_i and

$$\boldsymbol{x}^{\text{new}} \in X \tag{2.61}$$

where q_i is calculated as

$$q_i = sgn(\upsilon_i - \frac{1}{2}) T_{i,gen}(k_i) \times \left(\left(1 + \frac{1}{T_{i,gen}(k_i)} \right)^{|2\upsilon_i - 1|} - 1 \right) \tag{2.62}$$

and υ_i a uniformly distributed random variable in [0,1]. Notice that if a generated $\boldsymbol{x}^{\text{new}}$ is not in X, it is simply discarded and a new point is tried again until $\boldsymbol{x}^{\text{new}} \in X$.

The value of the cost function $f(\boldsymbol{x}^{\text{new}})$ is then evaluated and the acceptance probability function of $\boldsymbol{x}^{\text{new}}$ is given by

$$P_{\text{accept}} = \frac{1}{1 + exp((f(\boldsymbol{x}^{\text{new}}) - f(\boldsymbol{x}^{\text{old}}))/T_{\text{accept}}(K_a))} \tag{2.63}$$

A uniform random variable P_{unif} is generated in [0,1]. If $P_{unif} \leq P_{\text{accept}}$, $\boldsymbol{x}^{\text{new}}$ is accepted; otherwise it is rejected.

Step 3. After every N_{accept} acceptance points, reannealing takes place by first calculating the sensitivies

$$s_i = \left| \frac{f(\boldsymbol{x}^{\text{best}} + e_i \delta) - f(\boldsymbol{x}^{\text{best}})}{\delta} \right|, \ 1 \leq i \leq n \tag{2.64}$$

where $\boldsymbol{x}^{\text{best}}$ is the best point found so far, δ is a small step size, the n-dimensional vector e_i has unit ith element and the rest of elements of e_i are all zeros. Let $s_{\max} = \max\{s_i, 1 \leq i \leq n\}$. Each parameter generating temperature $T_{i,gen}$ is scaled by a factor $s_{\max}/s_i$ and the annealing time k_i is reset,

$$T_{i,\,gen}(k_i) = \frac{s_{\max}}{s_i} T_{i,gen}(k_i), \ k_i = \left(-\frac{1}{c} log \left(\frac{T_{i,\,gen}(k_i)}{T_{i,\,gen}(0)} \right) \right)^n.$$

Similarly, $T_{\text{accept}}(0)$ is reset to the value of the last accepted cost function, $T_{\text{accept}}(k_a)$ is reset to $f(\boldsymbol{x}^{\text{best}})$ and the annealing time k_a is rescaled accordingly,

$$k_a = \left(-\frac{1}{c} log \left(\frac{T_{i,\,gen}(k_a)}{T_{i,\,gen}(0)} \right) \right)^n \tag{2.65}$$

Step 4. After every N_{gen} generated points, annealing takes place with

$$k_i = k_i + 1,\ T_{i,\ gen}(k_i) = T_{i,\ gen}(0)exp(-ck_i^{1/n});$$

and

$$k_a = k_a + 1,\ T_{\text{accept}}(k_a) = T_{\text{accept}}(0)exp(-ck_a^{1/n});$$

otherwise, go to step 2.

Step 5. The algorithm is terminated if the parameters have remained unchanged for a few successive reannealing or a preset maximum number of cost function evaluations has been reached; otherwise, go to step 2.

As in a standard SA algorithm, this ASA contains two loops. The inner loop ensures that the parameter space is searched sufficiently at a given temperature, which is necessary to guarantee that the algorithm finds a global optimum. The ASA also uses only the value of the cost function in the optimization process and is very simple to program.

Last, we discuss about the algorithm parameter tuning. For the above ASA algorithm, most of the algorithm parameters are automatically set and "tuneda", and the user only needs to assign a control parameter c and set two values N_{accept} and N_{genera}. Obviously, the optimal values of N_{accept} and N_{genera} are problem dependent, but our experience suggests that an adequate choice for N_{accept} is in the range of tens to hundreds and an appropriate value for N_{gen} is in the range of hundreds to thousands. The annealing rate control parameter c can be determined form the chosen initial temperature, final temperature and predetermined number of annealing steps. We have found out that a choice of c in the range 1.0–10.0 is often adequate.

It should be emphasized that, as the ASA has excellent self adaptation ability, the performance of the algorithm is not critically influenced by the specific chosen values of c, N_{accept} and N_{gen}.

2.4.4 Numerical Examples

Example 2.6. Let's still consider the following problem,

$$\begin{cases} \max f_1(\boldsymbol{x}, \boldsymbol{\xi}) = \xi_1 x_1 + \xi_2 x_2 + \xi_3 x_3 + \xi_4 x_4 + \xi_5 x_5 \\ \max f_2(\boldsymbol{x}, \boldsymbol{\xi}) = c_1\xi_6 x_1 + c_2\xi_7 x_2 + c_3\xi_8 x_3 + c_4\xi_9 x_4 + c_5\xi_{10} x_5 \\ \text{s.t.} \begin{cases} x_1 + x_2 + x_3 + x_4 + x_5 \le 350 \\ x_1 + x_2 + x_3 + x_4 + x_5 \ge 300 \\ 4x_1 + 2x_2 + 1.5x_3 + x_4 + 2x_5 \le 1085 \\ x_1 + 4x_2 + 2x_3 + 5x_4 + 3x_5 \le 660 \\ x_1 \ge 20, x_2 \ge 20, x_3 \ge 20, x_4 \ge 20, x_5 \ge 20 \end{cases} \end{cases} \tag{2.66}$$

where $c = (c_1, c_2, c_3, c_4, c_5) = (1.2, 0.5, 1.3, 0.8, 0.9)$,

$$\xi_1 \sim \mathcal{N}(113, 1),\ \xi_2 \sim \mathcal{N}(241, 4),\ \ \xi_3 \sim \mathcal{N}(87, 1),\ \ \xi_4 \sim \mathcal{N}(56, 2),$$
$$\xi_5 \sim \mathcal{N}(92, 1),\ \ \xi_6 \sim \mathcal{N}(628, 1),\ \ \xi_7 \sim \mathcal{N}(143, 2),\ \xi_8 \sim \mathcal{N}(476, 2),$$
$$\xi_9 \sim \mathcal{N}(324, 2),\ \xi_{10} \sim \mathcal{N}(539, 2).$$

and $\xi_i (i = 1, 2, \ldots, 9)$ are independently random variables. We set $\beta_1 = \beta_2 = 0.9$, from (2.48) we have the following chance-constrained model,

$$\begin{cases} \max[\bar{f_1}, \bar{f_2}] \\ \text{s.t.} \begin{cases} Pr\{\xi_1 x_1 + \xi_2 x_2 + \xi_3 x_3 + \xi_4 x_4 + \xi_5 x_5 \geq \bar{f_1}\} \geq \beta_1 \\ Pr\{c_1\xi_6 x_1 + c_2\xi_7 x_2 + c_3\xi_8 x_3 + c_4\xi_9 x_4 + c_5\xi_{10} x_5 \geq \bar{f_2}\} \geq \beta_2 \\ x_1 + x_2 + x_3 + x_4 + x_5 \leq 350 \\ x_1 + x_2 + x_3 + x_4 + x_5 \geq 300 \\ 4x_1 + 2x_2 + 1.5x_3 + x_4 + 2x_5 \leq 1085 \\ x_1 + 4x_2 + 2x_3 + 5x_4 + 3x_5 \leq 660 \\ x_1 \geq 20, x_2 \geq 20, x_3 \geq 20, x_4 \geq 20, x_5 \geq 20 \end{cases} \end{cases} \tag{2.67}$$

Since $\Phi^{-1}(1-\beta_i) = -1.28, i = 1, 2$, from Theorem 2.6, problem (2.67) is equivalent to

$$\begin{cases} \max H_1(\boldsymbol{x}) = 113x_1 + 241x_2 + 87x_3 + 56x_4 + 92x_5 \\ \qquad\qquad -1.28\sqrt{x_1^2 + 4x_2^2 + x_3^2 + 2x_4^2 + x_5^2} \\ \max H_2(\boldsymbol{x}) = 753.6x_1 + 71.5x_2 + 618.8x_3 + 259.2x_4 + 485.1x_5 \\ \qquad\qquad -1.28\sqrt{x_1^2 + 2x_2^2 + 2x_3^2 + 2x_4^2 + 2x_5^2} \\ \text{s.t.} \begin{cases} x_1 + x_2 + x_3 + x_4 + x_5 \leq 350 \\ x_1 + x_2 + x_3 + x_4 + x_5 \geq 300 \\ 4x_1 + 2x_2 + 1.5x_3 + x_4 + 2x_5 \leq 1085 \\ x_1 + 4x_2 + 2x_3 + 5x_4 + 3x_5 \leq 660 \\ x_1 \geq 20, x_2 \geq 20, x_3 \geq 20, x_4 \geq 20, x_5 \geq 20 \end{cases} \end{cases} \tag{2.68}$$

Then we use the lexicographic method to resolve the above programming problem. Assume that the objective function $H_1(\boldsymbol{x})$ is more important than $H_2(\boldsymbol{x})$ to DM. Let's firstly solve the following problem,

$$\begin{cases} \max H_1(\boldsymbol{x}) = 113x_1 + 241x_2 + 87x_3 + 56x_4 + 92x_5 \\ \qquad\qquad -1.28\sqrt{x_1^2 + 4x_2^2 + x_3^2 + 2x_4^2 + x_5^2} \\ \text{s.t.} \begin{cases} x_1 + x_2 + x_3 + x_4 + x_5 \leq 350 \\ x_1 + x_2 + x_3 + x_4 + x_5 \geq 300 \\ 4x_1 + 2x_2 + 1.5x_3 + x_4 + 2x_5 \leq 1085 \\ x_1 + 4x_2 + 2x_3 + 5x_4 + 3x_5 \leq 660 \\ x_1 \geq 20, x_2 \geq 20, x_3 \geq 20, x_4 \geq 20, x_5 \geq 20 \end{cases} \end{cases} \tag{2.69}$$

Then we get the optimal solution $\boldsymbol{x}^* = (218.18, 59.10, 22.73, 20.00, 20.00)$ and the objective value $H_2(\boldsymbol{x}^*) = 197290.30$.

Secondly, construct the following programming problem,

$$\begin{cases} \max H_2(\boldsymbol{x}) = 753.6x_1 + 71.5x_2 + 618.8x_3 + 259.2x_4 + 485.1x_5 \\ \qquad\qquad -1.28\sqrt{x_1^2 + 2x_2^2 + 2x_3^2 + 2x_4^2 + 2x_5^2} \\ \text{s.t.} \begin{cases} 113x_1 + 241x_2 + 87x_3 + 56x_4 + 92x_5 \\ \quad - 1.28\sqrt{x_1^2 + 4x_2^2 + x_3^2 + 2x_4^2 + x_5^2} = 43510.72 \\ x_1 + x_2 + x_3 + x_4 + x_5 \le 350 \\ x_1 + x_2 + x_3 + x_4 + x_5 \ge 300 \\ 4x_1 + 2x_2 + 1.5x_3 + x_4 + 2x_5 \le 1085 \\ x_1 + 4x_2 + 2x_3 + 5x_4 + 3x_5 \le 660 \\ x_1 \ge 20, x_2 \ge 20, x_3 \ge 20, x_4 \ge 20, x_5 \ge 20 \end{cases} \end{cases} \tag{2.70}$$

Then we get the final optimal solution $\boldsymbol{x}^* = (218.18, 59.10, 22.73, 20.00, 20.00)$ and the objective value $H_1(\boldsymbol{x}^*) = 43510.72$.

Example 2.7. Consider the following multi-objective programming problem with random parameters,

$$\begin{cases} \max[\bar{f_1}, \bar{f_2}] \\ \text{s.t.} \begin{cases} Pr\{\sqrt{(x_1 - \xi_1)^2 + (x_2 - \xi_2)^2} \ge \bar{f_1}\} \ge \beta_1 \\ Pr\{\sqrt{(x_1 + \xi_1)^2 + (x_2 + \xi_2)^2} \ge \bar{f_2}\} \ge \beta_2 \\ x_1 + x_2 \le 5 \\ x_1 \ge 0, x_2 \ge 0 \end{cases} \end{cases} \tag{2.71}$$

where $\xi_1 \sim \mathscr{N}(2, 0.5)$ ia normally distributed random variable and $\xi_2 \sim \mathscr{U}(1, 2)$ is also a normally distributed random variable.

Next we will use the random simulation-based ASA to solve the above problem. Set the initial temperature T_0, the last temperature be 1 and the cooling method be 0.05% decrement once. The neighborhood can be constructed as follows,

$$x_1^1 = x_1^0 + rh, \;\; x_2^1 = x_2^0 + rh,$$

where r is a random number in (0,1) and h is the step length (here $h = 2.0$). After the simulation with many cycles, we get the optimal solution under different weights as shown in Table 2.4. Figure 2.6 shows the cooling process when the weight is 0.5. The real line expresses the weight sum of two objective functions, and it shows that it gradually converges from T = 25. The second figure shows the changes of two objective values when the temperature decreases.

Table 2.4 The optimal solution by random simulation-based ASA

w_1	w_2	x_1	x_2	$\bar{f}_1$	$\bar{f}_2$	$\bar{f}$	T_0
0.1	0.9	2.2823	0.0925	1.1123	6.0069	5.5175	1000
0.2	0.8	1.1542	2.7342	1.4170	5.8217	4.9407	1000
0.3	0.7	0.2140	1.9297	1.5559	5.8935	4.5922	1000
0.4	0.6	1.5625	0.2912	0.9358	5.8060	3.8579	1000
0.5	0.5	2.6555	1.1940	0.8505	5.7968	3.3236	1000

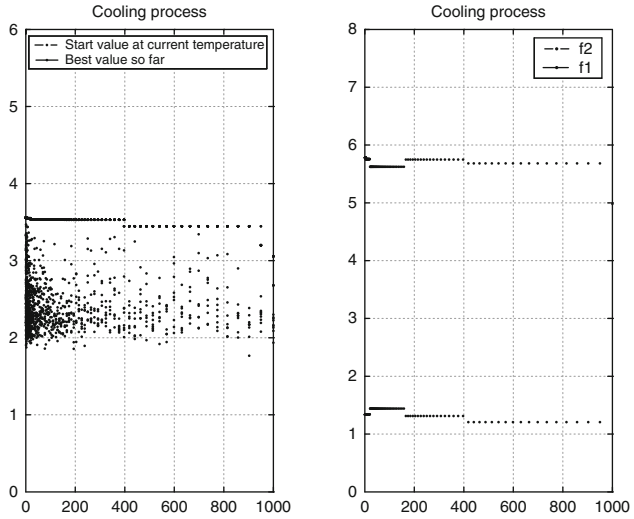


Fig. 2.6 The cooling process of random simulation-based ASA

2.5 Random DCM

In practice, there usually exist multiple events in a complex random decision system. Sometimes, the decision-maker wishes to maximize the chance functions of these events (i.e. the probabilities of satisfying the events). In order to model this type of random decision system, Schneider [280] developed another technique, called probability maximization model (also called dependent-chance model (abbr. DCM) by some scholars, we will use this name in this book), in which the underlying philosophy is based on selecting the decision with maximal chance to meet the event. Then [150] discussed some coverage probability maximization problems. From then on, it was widely used in many fields to solve some realistic problems [133, 146, 272].

DCM theory breaks the concept of feasible set and replaces it with uncertain environment. Roughly speaking, DCM involves maximizing chance functions of events in an uncertain environment. In deterministic model, expected value model (EVM), and chance-constrained programming (CCM), the feasible set is essentially assumed

to be deterministic after the real problem is modeled. That is, an optimal solution is given regardless of whether it can be performed in practice. However, the given solution may be impossible to perform if the realization of uncertain parameter is unfavorable. Thus DCM theory never assumes that the feasible set is deterministic. In fact, DCM is constructed in an uncertain environment. This special feature of DCM is very different from the other existing types of random programming.

In this section, we introduce the concept of chance function and discuss the DCM models. We also give some crisp equivalent models. Finally, some numerical examples are exhibited.

2.5.1 General Model for Random DCM

Let's consider the following typical DCM model,

$$\begin{cases} \max Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}) \geq f\} \\ \text{s.t.} \begin{cases} g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, j = 1, 2, \ldots, p \\ \boldsymbol{x} \geq 0 \end{cases} \end{cases} \tag{2.72}$$

Since a complex decision system usually undertakes multiple events, there undoubtedly exist multiple potential objectives (some of them are chance functions) in a decision process. A typical formulation of dependent-chance multi-objective programming model (DCM) is represented as maximizing multiple chance functions subject to an uncertain environment,

$$\begin{cases} \max \begin{bmatrix} Pr\{h_1(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \\ Pr\{h_2(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \\ \cdots \\ Pr\{h_m(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \end{bmatrix} \\ \text{s.t.} \quad g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, j = 1, 2, \ldots, p \end{cases} \tag{2.73}$$

where $h_i(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0$ are represent events ε_i for $i = 1, 2, \ldots, m$, respectively.

It follows from the principle of uncertainty that we can construct a relationship between decision vectors and chance function, thus calculating the chance functions by stochastic simulations or traditional methods. Then we can solve DCM by utility theory if complete information of the preference function is given by the decision-maker or search for all of the efficient solutions if no information is available. In practice, the decision maker can provide only partial information. In this case, we have to employ the interactive methods.

Sometimes, the objective function may minimize the deviations, positive, negative, or both, with a certain priority structure set by the decision maker. Then we may formulate the stochastic decision system as the following model,

$$\begin{cases} \min \sum_{j=1}^{l} P_j \sum_{i=1}^{m} (u_{ij} d_i^+ + v_{ij} d_i^-) \\ \text{s.t.} \begin{cases} Pr\{h_i(\boldsymbol{x}, \boldsymbol{\xi}) \le 0\} + d_i^- - d_i^+ = b_i, & i = 1, 2, \ldots, m \\ g_j(\boldsymbol{x}, \boldsymbol{\xi}) \le 0, & j = 1, 2, \ldots, p \\ d_i^-, d_i^+ \ge 0, & i = 1, 2, \ldots, m \end{cases} \end{cases} \tag{2.74}$$

where P_j is the preemptive priority factor which express the relative importance of various goals, $P_j >> P_{j+1}$, for all j, u_{ij} is the weighting factor corresponding to positive deviation for goal i with priority j assigned, v_{ij} is the weighting factor corresponding to negative deviation for goal i with priority j assigned, d_i^+ is the positive deviation from the target value according to goal i, d_i^- is the negative deviation from the target of goal i, b_i is the target value according to goal i, l is the number of priorities, and m is the number of goal constraints.

2.5.2 Linear Random DCM and the Fuzzy Programming Method

As we know, traditional solution methods need conversion of the chance constraints to their respective deterministic equivalents. However, this process is usually hard to perform and only successful for some special cases. Next, let's consider a special case in which the model is linear and parameters are subject to normal distribution (see the following linear model)

$$\begin{cases} \max \left[\bar{c}_1^T \boldsymbol{x}, \bar{c}_2^T \boldsymbol{x}, \ldots, \bar{c}_m^T \boldsymbol{x}\right] \\ \text{s.t.} \begin{cases} \bar{e}_r^T \boldsymbol{x} \le \bar{b}_r, r = 1, 2, \ldots, p \\ \boldsymbol{x} \ge 0 \end{cases} \end{cases} \tag{2.75}$$

where $\bar{c}_i = (c_{i1}(\omega), c_{i2}(\omega), \ldots, c_{in}(\omega))$, $\bar{e}_r = (e_{r1}(\omega), e_{r2}(\omega), \ldots, e_{rn}(\omega))$ and $\bar{b}_r$ are independently random vectors on the probability space $(\Omega, \mathscr{A}, Pr)$, f_i is the predetermined objective value. By the definition of chance function, we can get the following model,

$$\begin{cases} \max \left[Pr\{\bar{c}_1^T \boldsymbol{x} \ge f_1\}, Pr\{\bar{c}_2^T \boldsymbol{x} \ge f_2\}, \ldots, Pr\{\bar{c}_m^T \boldsymbol{x} \ge f_m\}\right] \\ \text{s.t.} \begin{cases} Pr\{\bar{e}_r^T \boldsymbol{x} \le \bar{b}_r\} \ge \beta_r, \ r = 1, 2, \ldots, p \\ \boldsymbol{x} \ge 0 \end{cases} \end{cases} \tag{2.76}$$

where f_i is predetermined objective value and β_r is the predetermined level value.

2.5.2.1 Crisp Equivalent Model

By the equivalent transformation, we get the following theorem.

Theorem 2.9. *Assume that random vector* $\bar{c}_i = (\bar{c}_{i1}, \bar{c}_{i2}, \ldots, \bar{c}_{in})^T$ *is normally distributed with mean vector* $\mu_i^c = (\mu_{i1}^c, \mu_{i2}^c, \ldots, \mu_{in}^c)^T$ *and positive definite*

covariance matrix V_i^c, written as $\bar{c}_i \sim \mathcal{N}(\mu_i^c, V_i^c)(i = 1, 2, \ldots, m)$ and random vector $\bar{e}_r \sim \mathcal{N}(\mu_r^e, V_i^e)$, $\bar{b}_r \sim \mathcal{N}(\mu_r^b, (\sigma_r^b)^2)$ $(r = 1, 2, \ldots, p)$. Assume that for any $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$ and $r = 1, 2, \ldots, p$, $\bar{c}_{ij}$, $\bar{e}_{rj}$ and $\bar{b}_r$ are independent with each other, respectively. Then problem (2.76) is equivalent to

$$\begin{cases} \max \left[1 - \Phi \left(\dfrac{f_i - \mu_i^{cT} \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}} \right), i = 1, 2, \ldots, m \right] \\ s.t. \begin{cases} \Phi^{-1}(\beta_r) \sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2} + \mu_r^{eT} \boldsymbol{x} - \mu_r^b \leq 0 \\ \boldsymbol{x} \geq 0, r = 1, 2, \ldots, p \end{cases} \end{cases}$$

Proof. Since the random vector $\bar{c}_i$ is normally distributed with mean vector μ_i^c and positive definite covariance matrix V_i^c, then we have that

$$\bar{c}_i^T \boldsymbol{x} = \sum_{j=1}^{n} x_j \bar{c}_{ij}(\omega) \sim \mathcal{N} \left(\sum_{j=1}^{n} x_j \mu_{ij}^c, \boldsymbol{x}^T V_i^c \boldsymbol{x} \right) = \mathcal{N}(\mu_i^{cT} \boldsymbol{x}, \boldsymbol{x}^T V_i^c \boldsymbol{x}).$$

It follows that

$$\frac{\bar{c}_i^T \boldsymbol{x} - \mu_i^{cT} \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}$$

must be standardized normally distributed. Then

$$\begin{aligned} Pr\{c_i^T \boldsymbol{x} \geq f_i\} &= Pr \left\{ \frac{\bar{c}_i^T \boldsymbol{x} - \mu_i^{cT} \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}} \geq \frac{f_i - \mu_i^{cT} \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}} \right\} \\ &= 1 - Pr \left\{ \frac{\bar{c}_i^T \boldsymbol{x} - \mu_i^{cT} \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}} \leq \frac{f_i - \mu_i^{cT} \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}} \right\} \\ &= 1 - \Phi \left(\frac{f_i - \mu_i^{cT} \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}} \right) \end{aligned}$$

where Φ is the standardized normally distributed function. Similarly, we know that

$$\bar{e}_r^T \boldsymbol{x} \sim \mathcal{N}(\mu_r^{eT} \boldsymbol{x}, \boldsymbol{x}^T V_r^e \boldsymbol{x}).$$

Since $\bar{b}_r \sim \mathcal{N}(\mu_r^b, (\sigma_r^b)^2)$, it follows from, $\bar{e}_{rj}(\omega)$ and $\bar{b}_{rj}(\omega)$ are independently random variables for any r, j, that

$$E[\bar{e}_r^T \boldsymbol{x} - \bar{b}_r] = E[\bar{e}_r^T \boldsymbol{x}] - E[\bar{b}_r] = \mu_r^{eT} \boldsymbol{x} - \mu_r^b,$$
$$V[\bar{e}_r^T \boldsymbol{x} - \bar{b}_r] = V[\bar{e}_r^T \boldsymbol{x}] + V[\bar{b}_r] = \boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2,$$

then $\bar{e}_r^T \boldsymbol{x} - \bar{b}_r$ is also normally distributed with

$$\bar{e}_r^T \boldsymbol{x} - \bar{b}_r \sim \mathcal{N}(\mu_r^{eT} \boldsymbol{x} - \mu_r^b, \boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2).$$

For any $r (r = 1, 2, \ldots, p)$, we have

$$\begin{aligned} Pr\{\bar{e}_r^T \boldsymbol{x} \geq \bar{b}_r\} \geq \beta_r \Leftrightarrow \beta_r \leq Pr \Bigg\{ & \frac{\bar{e}_r^T \boldsymbol{x} - \bar{b}_r - (\mu_r^{eT} \boldsymbol{x} - \mu_r^b)}{\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2}} \\ & \leq -\frac{\mu_r^{eT} \boldsymbol{x} - \mu_r^b}{\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2}} \Bigg\} \\ \Leftrightarrow \beta_r \leq \Phi & \left(-\frac{\mu_r^{eT} \boldsymbol{x} - \mu_r^b}{\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2}} \right) \\ \Leftrightarrow \Phi^{-1}(\beta_r) & \sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2} + \mu_r^{eT} \boldsymbol{x} - \mu_r^b \leq 0 \end{aligned}$$

where Φ is the standardized normal distribution. Then we have that problem (2.76) is equivalent to

$$\begin{cases} \max \left[1 - \Phi \left(\dfrac{f_i - \mu_i^{cT} \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}} \right), i = 1, 2, \ldots, m \right] \\ \text{s.t.} \begin{cases} \Phi^{-1}(\beta_r) \sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2} + \mu_r^{eT} \boldsymbol{x} - \mu_r^b \leq 0 \\ \boldsymbol{x} \geq 0, r = 1, 2, \ldots, p \end{cases} \end{cases} \tag{2.77}$$

This completes the proof. □

2.5.2.2 Fuzzy Programming Method

The fuzzy programming method for multi-objective programming problems was proposed by Zimmermann [364] and has been advanced by Sakawa and colleagues [271]. An interactive fuzzy satisficing method for multiobjective linear programming problems and the interactive fuzzy decision making for multiobjective nonlinear programming using augmented minimax problems have been also introduced [212, 270, 282]. The fuzzy programming method in which fuzziness in the decision making process is represented by using the fuzzy concept has also been studied extensively and many results have been published [265, 317]. This method can be applied to not only the linear multi-objective but also the nonlinear multi-objective programming.

Take the problem (2.77) as an example. Let $H_i(\boldsymbol{x}) = 1 - \Phi\left(\frac{f_i - \mu_i^{cT}\boldsymbol{x}}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right)$, and $X = \{\boldsymbol{x} | \Phi^{-1}(\beta_r)\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2} + \mu_r^{eT}\boldsymbol{x} - \mu_r^b \leq 0, r = 1, 2, \ldots, p; \boldsymbol{x} \geq 0\}$, then problem (2.77) is equivalent to

$$\begin{cases} \max[H_1(\boldsymbol{x}), H_2(\boldsymbol{x}), \ldots, H_m(\boldsymbol{x})] \\ \text{s.t. } \boldsymbol{x} \in X \end{cases} \tag{2.78}$$

Considering the imprecise nature of the decision maker's judgements for each objective function of problem (2.78), a new fuzzy objective goal such as "make $H_i(\boldsymbol{x})$ approximately larger than a certain value" is introduced. Then problem (2.78) is converted into

$$\begin{cases} \max[\rho_1(H_1(\boldsymbol{x})), \rho_2(H_2(\boldsymbol{x})), \ldots, \rho_m(H_m(\boldsymbol{x}))] \\ \text{s.t. } \boldsymbol{x} \in X \end{cases} \tag{2.79}$$

where the random goal is characterized by the following linear membership function,

$$\rho_i(H_i(\boldsymbol{x})) = \begin{cases} 1, & H_i(\boldsymbol{x}) > H_i^1 \\ \dfrac{H_i(\boldsymbol{x}) - H_i^0}{H_i^1 - H_i^0}, & H_i^0 \leq H_i(\boldsymbol{x}) \leq H_i^1 \\ 0, & H_i(\boldsymbol{x}) < H_i^0 \end{cases}$$

where H_i^1 and H_i^0 respectively denote the maximal and minimal values of the objective functions $H_i(\boldsymbol{x})$ as follows,

$$H_i^0 = \min_{\boldsymbol{x} \in X} H_i(\boldsymbol{x}), \ H_i^1 = \max_{\boldsymbol{x} \in X} H_i(\boldsymbol{x}), \ i = 1, 2, \ldots, m.$$

For each objective function $\rho_i(H_i(\boldsymbol{x}))$, assume that the DM can specify the so-called reference probability function value $\bar{\rho_i}$ which reflects the probability function value of $\rho_i(H_i(\boldsymbol{x}))$. The corresponding Pareto optimal solution, which is nearest to the requirements in the minimax sense or better than that if the reference probability function value is attainable, is obtained by solving the following minimax problem,

$$\begin{cases} \min \max\limits_{i=1,2,\ldots,m} \{\bar{\rho_i} - \rho_i(H_i(\boldsymbol{x}))\} \\ \text{s.t. } \boldsymbol{x} \in X \end{cases} \tag{2.80}$$

By introducing auxiliary variable λ, problem (2.80) is equivalent to

$$\begin{cases} \min \lambda \\ \text{s.t.} \begin{cases} \bar{\rho_i} - \rho_i(H_i(\boldsymbol{x})) \leq \lambda, \ i = 1, 2, \ldots, m \\ 0 \leq \lambda \leq 1 \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{2.81}$$

or equivalently

$$\begin{cases} \min \lambda \\ \text{s.t.} \begin{cases} \lambda_i^{cT} \boldsymbol{x} \geq H_i^0 + (\bar{\rho}_i - \lambda)(H_i^1 - H_i^0), i = 1, 2, \ldots, m \\ 0 \leq \lambda \leq 1 \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{2.82}$$

Obviously, it follows that (2.82) is a convex programming problem of which the global optimal solution is easily obtained.

The relationship between the optimal solution of problem (2.81) and the Pareto optimal solution of problem (2.78) can be characterized by the following theorem.

Theorem 2.10. *1. If $\boldsymbol{x}^* \in X$ is a unique optimal solution to problem (2.81) for some $\bar{\rho}_i, i = 1, 2, \ldots, m$, then $\boldsymbol{x}^*$ is a Pareto optimal solution to problem (2.78).*

2. If $\boldsymbol{x}^*$ is a Pareto optimal solution to problem (2.78) with $0 < \rho_i(H_i(\boldsymbol{x}^*)) < 1$ holding for all i, then there exist ε_i such that $\boldsymbol{x}^*$ is an optimal solution to problem (2.81).

Proof. 1. For some $\bar{\rho_k}$, if a unique optimal solution $\boldsymbol{x}^*$ to problem (2.81) is not the Pareto optimal solution of problem (2.78), then there exists $\boldsymbol{x} \in X$ such that $H_k(\boldsymbol{x}) > H_k(\boldsymbol{x}^*)$, where $H_k(\boldsymbol{x}^*) = \max\{H_1(\boldsymbol{x}^*), H_2(\boldsymbol{x}^*), \ldots, H_m(\boldsymbol{x}^*)\}$. Then,

$$\begin{aligned} &\frac{H_k(x) - H_k^0}{H_k^1 - H_k^0} > \frac{H_k(\boldsymbol{x}^*) - H_k^0}{H_k^1 - H_k^0} \\ &\Leftrightarrow \rho(H_k(x)) > \rho(H_k(\boldsymbol{x}^*)) \\ &\Leftrightarrow \bar{\rho_k} - \rho(H_k(\boldsymbol{x})) < \bar{\rho_k} - \rho(H_k(\boldsymbol{x}^*)) \end{aligned}$$

This means that there exits a λ, satisfying $\lambda < \lambda^*$. It follows that $\boldsymbol{x}^*$ is not the optimal solution of problem (2.81), which contradicts the assumption that $\boldsymbol{x}^*$ is a unique optimal solution to problem (2.81).

2. If $\boldsymbol{x}^*$ is not an optimal solution to problem (2.81), then there exists $\boldsymbol{x}^{'} \in X$ such that $\bar{\rho}_i - \rho(H_i(\boldsymbol{x}^{'})) < \bar{\rho}_i - \rho(H_i(\boldsymbol{x}^*)), i = 1, 2, \ldots, m$, because of $0 < \rho_i(H_i(\boldsymbol{x}^*)) < 1$, then

$$\begin{aligned} &\rho(H_i(\boldsymbol{x}^{'})) > \rho(H_i(\boldsymbol{x}^*)) \\ &\quad \Leftrightarrow \frac{H_i(\boldsymbol{x}^{'}) - H_i^0}{H_i^1 - H_i^0} > \frac{H_i(\boldsymbol{x}^*) - H_i^0}{H_i^1 - H_i^0} \\ &\quad \Leftrightarrow H_i(\boldsymbol{x}^{'}) > H_i(\boldsymbol{x}^*) \end{aligned}$$

This means that there exists $\boldsymbol{x}^{'}$ such that $H_i(\boldsymbol{x}^{'}) > H_i(\boldsymbol{x}^*)$, then $\boldsymbol{x}^*$ is not the Pareto optimal solution to problem (2.78), which contradicts the assumption that $\boldsymbol{x}^*$ is a Pareto optimal solution to problem (2.78). This completes the proof. □

If $\boldsymbol{x}^*$, an optimal solution to problem (2.81), is not unique, then the Pareto optimality test for $\boldsymbol{x}^*$ can be performed by solving the following problem

$$\begin{cases} \max \sum_{i=1}^{m} \varepsilon_i \\ \text{s.t.} \begin{cases} \rho_i(H_i(\boldsymbol{x}^*)) + \varepsilon_i = \bar{\rho}_i, i = 1, 2, \ldots, m \\ \boldsymbol{x} \in X, \varepsilon_i \geq 0 \end{cases} \end{cases}$$

From Theorem 2.10, we know that the optimal solution $\boldsymbol{x}^*$ of problem (2.81) is a Pareto optimal solution to problem (2.78). Then the interactive random satisfying method, which is similar to the interactive fuzzy satisfying method, can be constructed to obtain a satisfactory solution of problem (2.78),

Step 1. The DM is required to present reference probability values $\bar{\rho}_i$, $i = 1, 2, \ldots, m$.

Step 2. The optimal solution of problem (2.80), which is also a Pareto optimal solution of problem (2.78), is considered as a satisfactory solution to problem (2.78).

Step 3. If the obtained $\rho_i(H_i(\boldsymbol{x}^*))$ are satisfying, the process stops and $\boldsymbol{x}^*$ is selected as satisfactory solution to problem (2.78); Or else, the DM should update his or her reference random probability values $\bar{\rho}_i$ and return to Step 2.

2.5.3 Nonlinear Random DCM and Random Simulation-Based PSA

Similarly to the EVM and CCM, for the problems which are easily converted into crisp ones, we deal with them by the chance measure, and the random simulation is used to deal with those which cannot be converted into crisp ones. Next, let's introduce the process of the random simulation dealing with the dependent chance models. Consider the nonlinear multi-objective programming problem as follows,

$$\begin{cases} \max[Pr\{f_1(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f_1}\}, \ldots, Pr\{f_m(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f_m}\}] \\ \text{s.t.} \begin{cases} Pr\{g_r(\boldsymbol{x}, \boldsymbol{\xi}) \leq b_r\} \geq \alpha_i, r = 1, 2, \ldots, p \\ \boldsymbol{x} \geq 0, 0 \leq \alpha_r \leq 1 \end{cases} \end{cases} \tag{2.83}$$

where $f_i(\boldsymbol{x}, \boldsymbol{\xi})$ and $g_r(\boldsymbol{x}, \boldsymbol{\xi})$ are nonlinear functions with respect to $\boldsymbol{\xi}$, $i = 1, 2, \ldots, m$, $r = 1, 2, \ldots, p$, $\bar{f_i}$ are predetermined confidence levels, and $\boldsymbol{\xi}$ is a random vector.

2.5.3.1 Random Simulation for DCM

Let $\boldsymbol{\xi}$ be an n-dimensional random vector defined on the probability space $(\Omega, \mathscr{A}, Pr)$, and $f : \mathbf{R}^n \to \mathbf{R}$ a measurable function. In order to obtain the probability for given $\boldsymbol{x}$,

$$L = Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}\},$$

we generate ω_k from Ω according to the probability measure Pr, and write $\boldsymbol{\xi}_k = \boldsymbol{\xi}(\omega_k)$ for $k = 1, 2, \ldots, N$, where $\omega_k = (\omega_k^1, \ldots, \omega_k^n)$ is an n-dimensional vector

and ω_k^j is generated according to the random variable ξ_k. Let $N^{'}$ denote the number of occasions on which $f(\boldsymbol{x}, \boldsymbol{\xi}_k) \geq \bar{f}$ for $k = 1, 2, \ldots, N$ (i.e., the number of random vectors satisfying the system of inequalities). Let us define

$$h(\boldsymbol{x}, \boldsymbol{\xi}_k) = \begin{cases} 1, \text{ if } f(\boldsymbol{x}, \boldsymbol{\xi}_k) \geq \bar{f} \\ 0, \text{ otherwise} \end{cases}$$

Then we have $E[h(\boldsymbol{x}, \boldsymbol{\xi}_k)] = L$ for all k, and $N^{'} = \sum_{k=1}^{N} h(\boldsymbol{x}, \boldsymbol{\xi}_k)$. It follows from the strong law of large numbers that

$$\frac{N^{'}}{N} = \frac{\sum\limits_{k=1}^{N} h(\boldsymbol{x}, \boldsymbol{\xi}_k)}{N}$$

converges a.s. to L. Thus the probability L can be estimated by $N^{'}/N$ provided that N is sufficiently large. Then the procedure simulating the probability of the event $f(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}$ can be summarized as follows:

Procedure `Random simulation for DCM`

Input: The decision vector $\boldsymbol{x}$
Output: The probability $Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}\}$
Step 1. Set $N^{'} = 0$;
Step 2. Generate ω from Ω according to the probability measure Pr;
Step 3. If $f(\boldsymbol{x}, \boldsymbol{\xi}_k) \geq \bar{f}$, then $N^{'}++$;
Step 4. Repeat the second and third steps N times;
Step 5. Return $L = N^{'}/N$.

Example 2.8. Let $\xi_1 \sim exp(2)$ be an exponentially distributed variable, $\xi_2 \sim \mathcal{N}(4, 1)$ a normally distributed variable, and $\xi_3 \sim \mathcal{U}(0, 2)$ a uniformly distributed variable. A run of random simulation with 10000 cycles shows that

$$Pr\left\{\sqrt{\xi_1^2 + \xi_2^2 + \xi_3^2} \leq 8\right\} = 0.9642.$$

2.5.3.2 The Parallel SA Algorithm

The parallel SA algorithm, which is usually shorten as PSA, is a more efficient tool to deal with complex problems. Parallel processing appears to be the only viable way to substantially speed up the method and thus expand its applicability. For fast tailored SAs, parallel implementations may also reduce the loss of robustness. In an effort to increase convergence speed, parallel SA optimization algorithms have been implemented with mixed results in various scientific fields [50]. Czech and Czarnas [69] presented a parallel simulated annealing for the vehicle routing problem. Many scholars [16, 16, 41, 69, 121, 235, 268, 336] introduced the PSA with the feature that multiple processing elements follow a single search path (Markov chain) (to be

referred to as SMC PSA), but Aarts and Korst [1] described the idea of the multiple Markov chains PSA (to be referred to as MMC PSA), namely the divison algorithm and Lee and Lee [190] proposed the MMC PSA and applied it to graph partition and Li, Cha and Lu [195] applied it for 3D engineering layout design.

Although optimization performance can be greatly improved through parallelization of SA [78, 163], there has few applications of this techniques in multiobjective programming problems. Since the "annealing community" has so far not achieved a common agreement with regards to a general approach for the serial SA, the best parallel scheme is still the object of current research. Bevilacqua [23] claimed that Many scholars has done many attempts of classifying methods: that is, if parallelism is used to generated a move or to run independent moves on different processors, if these methods depend on the problem or not, if they are synchronous or not, and others.

In any case, a key issue in developing a PSA is to ensure that it maintains the same convergence property as the sequential version, for which a formal proof to converge to a global minimum exists. For this purpose, it is easy to show that it is not necessary for a parallel implementation to generate exactly the same sequence of solutions as the sequential version. Frequently, it is enough to keep the same ratio between the number of accepted moves and the number of attempted moves (acceptance rate), averaged over a given temperature, as the correspondent sequential version.

Next, let's discuss the detailed algorithm about PSA. This section mainly refers to [23, 195, 217, 352], and readers can also consult the related literatures. Let's firstly introduce the parallel Markov chain in [217]. Let X be a finite set and $U : X \to R^+$ be a non-constant function to be minimized. We denote by $X_{\min}$ the set of global minima of U. Suppose that we have $p > 1$ processors. The optimization algorithm which is the center of our interest is described as follows. Choose any starting point $x_0 \in X$ and let each processor individually run a Metropolis Markov chain of fixed length $L > 1$ at inverse temperature $\beta(0)$ starting in x_0. After L transitions the simulation is stopped and only one state x_1 is selected from the p states according to a selection strategy. Again each processor individually simulates a Metropolis Markov chain of length L at an updated inverse temperature $\beta(1)$ starting in x_1. Again at the end of the simulation a state x2 is selected from the p states and the next simulation starts from x_2, etc. Fig. 2.7. This, algorithm is closely related to the so-called parallel chain algorithm. However, the main difference is that the number of parallel chains and the length of the Markov chains L is kept fixed in our model. In the parallel chain algorithm the length L is usually increased and the number of parallel Markov chains is decreased during the run of the algorithm so that the parallel chain

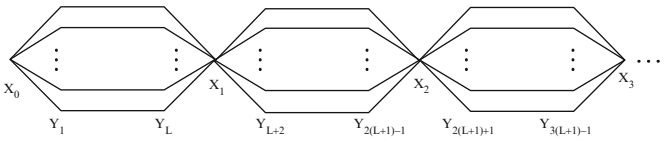


Fig. 2.7 Illustration of the algorithm's working method

algorithm asymptotically behaves like the so-called one-chain algorithm. Secondly, let's introduce the process of PSA. PSA starts with a high temperature. After generating an initial solution, it attempts to move randomly from the current solution to one of its neighborhood solutions. The changes in the objective function values Δf are computed(we usually consider a weighted sum of all objective functions in a multiobjective programming problem). If the new solution results in a better objective value, it is accepted. However, if the new solution yields a worse value, it can still be accepted according to the acceptance probability function, $P(T_j)$. The PSA algorithm repeats this Metropolis algorithm L times at each temperature to reach thermal equilibrium, where L is a control parameter, usually called the Markov Chain Length. After some temperature annealing, solutions migrate from the neighboring processors at each interval with migration probability, P_m. Parameter T_j is gradually decreased by a cooling function as parallel SA proceeds until the

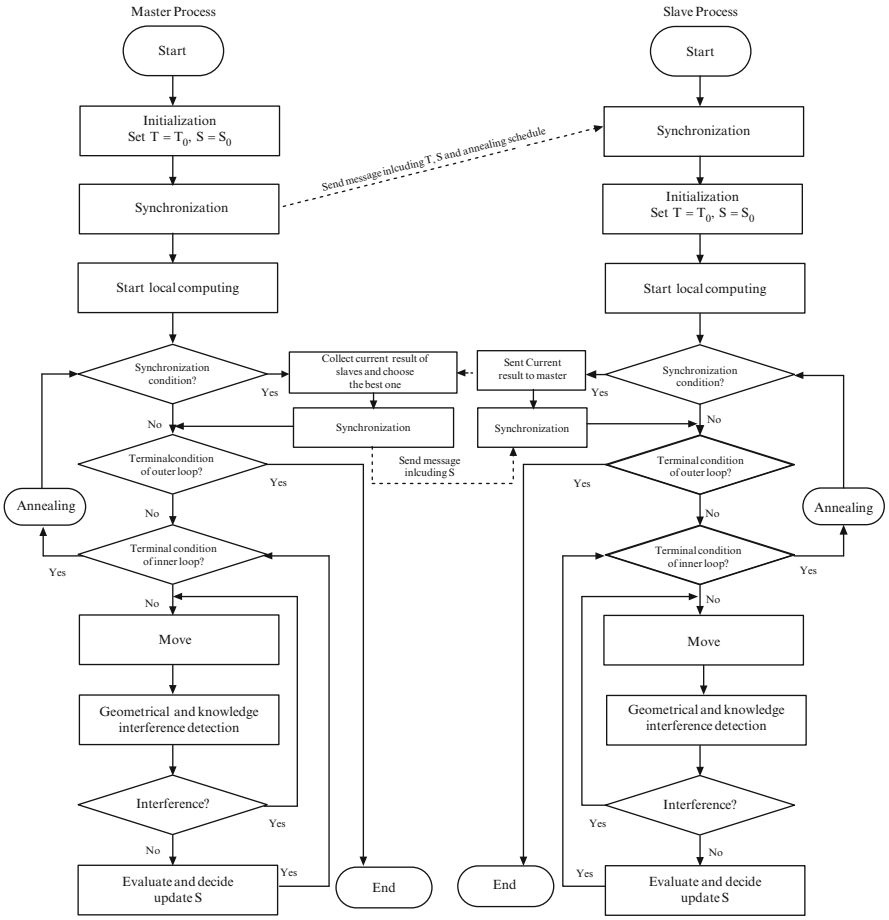


Fig. 2.8 Flow chart of the PSA

terminating condition is met. Readers can refer to Fig. 2.8 to know the work flow of PSA. The parallel SA algorithms proposed can be summarized as follows:

Procedure `The PSA algorithm`

Input: The initial parameters: T_0, T_f, P_m, M_i
Output: The Pareto-solution $\boldsymbol{x}^*$
Step 1. Initialize PSA parameters: starting temperature T_0, final temperature T_f, Markov chain length L, migration rate P_m, and migration interval M_i;
Step 2. Randomly generate an initial feasible $\boldsymbol{x}$ and assign to;
Step 3. Repeat the following L times: (a) Generate a neighborhood solution $\boldsymbol{x}_j^{new}$ through a random number generator; (b) Compute $\Delta f = f(\boldsymbol{x}_j^{new}) - f(\boldsymbol{x}_j^{old})$; (c) If $\Delta f \leq 0$(when we minimize the objective function), set $\boldsymbol{x}_j^{old} = \boldsymbol{x}_j^{new}$; (d) If $\Delta f > 0$, generate a random number X in (0,1), and compute $P(T_j)$. If $P(T_j) > X$, set $\boldsymbol{x}_j^{old} = \boldsymbol{x}_j^{new}$;
Step 4. the migration interval (M_i) is reached, then solutions migrate from the neighboring processors with migration rate P_m;
Step 5. If the terminating condition is met, stop; otherwise, let T_j decrease by the cooling schedule and go to Step 3.

2.5.4 Numerical Examples

Example 2.9. Let's consider the following problem,

$$\begin{cases} \max f_1(\boldsymbol{x}, \boldsymbol{\xi}) = \xi_1 x_1 + \xi_2 x_2 + \xi_3 x_3 + \xi_4 x_4 + \xi_5 x_5 \\ \max f_2(\boldsymbol{x}, \boldsymbol{\xi}) = c_1 \xi_6 x_1 + c_2 \xi_7 x_2 + c_3 \xi_8 x_3 + c_4 \xi_9 x_4 + c_5 \xi_{10} x_5 \\ \text{s.t.} \begin{cases} x_1 + x_2 + x_3 + x_4 + x_5 \leq 10 \\ x_1 + x_2 + x_3 - x_4 - x_5 \geq 0 \\ x_1 + 4x_2 + 2x_3 - x_4 + 5x_5 \leq 20 \\ x_i \geq 0, i = 1, 2, \ldots, 5 \end{cases} \end{cases} \tag{2.84}$$

where $c = (c_1, c_2, c_3, c_4, c_5) = (1.2, -0.5, 1.3, -0.1, 2)$, and $\xi_i (i = 1, 2, \ldots, 9)$ are independently random variables as follows,

$$\begin{aligned} &\xi_1 \sim \mathcal{N}(2, 1), \quad \xi_2 \sim \mathcal{N}(3, 0.5), \quad \xi_3 \sim \mathcal{N}(1, 0.2), \xi_4 \sim \mathcal{N}(5, 1), \\ &\xi_5 \sim \mathcal{N}(2, 0.1), \xi_6 \sim \mathcal{N}(7, 0.5), \quad \xi_7 \sim \mathcal{N}(3, 0.2), \xi_8 \sim \mathcal{N}(4, 1), \\ &\xi_9 \sim \mathcal{N}(5, 1), \quad \xi_{10} \sim \mathcal{N}(1, 0.1). \end{aligned}$$

If DM predetermines two level values $\bar{f_1}$ and $\bar{f_2}$, and aim at obtaining the maximum probability based on the predetermined level value $\bar{f_1}$ and $\bar{f_2}$. Set $\bar{f_1} = 14$ and $\bar{f_2} = 22$, then we get the following dependent-chance model,

$$\begin{cases} \max g_1(\boldsymbol{x}, \boldsymbol{\xi}) = Pr\{\xi_1 x_1 + \xi_2 x_2 + \xi_3 x_3 + \xi_4 x_4 + \xi_5 x_5 \geq 14\} \\ \max g_2(\boldsymbol{x}, \boldsymbol{\xi}) = Pr\{c_1\xi_6 x_1 + c_2\xi_7 x_2 + c_3\xi_8 x_3 + c_4\xi_9 x_4 + c_5\xi_{10} x_5 \geq 22\} \\ \text{s.t.} \begin{cases} x_1 + x_2 + x_3 + x_4 + x_5 \leq 10 \\ x_1 + x_2 + x_3 - x_4 - x_5 \geq 0 \\ x_1 + 4x_2 + 2x_3 - x_4 + 5x_5 \leq 20 \\ x_i \geq 0, i = 1, 2, \ldots, 5 \end{cases} \end{cases} \tag{2.85}$$

It follows from Theorem 2.9 that, (2.85) is equivalent to

$$\begin{cases} \max H_1(\boldsymbol{x}) = 1 - \Phi\left(\dfrac{14 - (2x_1 + 3x_2 + x_3 + 5x_4 + 2x_5)}{\sqrt{x_1^2 + 0.5x_2^2 + 0.2x_3^2 + x_4^2 + 0.1x_5^2}}\right) \\ \max H_2(\boldsymbol{x}) = 1 - \Phi\left(\dfrac{22 - (8.4x_1 - 1.5x_2 + 5.2x_3 - 0.5x_4 + 2x_5)}{\sqrt{0.5x_1^2 + 0.2x_2^2 + x_3^2 + x_4^2 + 0.1x_5^2}}\right) \\ \text{s.t.} \begin{cases} x_1 + x_2 + x_3 + x_4 + x_5 \leq 10 \\ x_1 + x_2 + x_3 - x_4 - x_5 \geq 0 \\ x_1 + 4x_2 + 2x_3 - x_4 + 5x_5 \leq 20 \\ x_i \geq 0, i = 1, 2, \ldots, 5 \end{cases} \end{cases} \tag{2.86}$$

Next, we will use the fuzzy programming method to solve the above problem. Then we have that

$$H_1^1 = 43944.64,\ H_1^0 = 30220.00,\ H_2^1 = 225444.00,\ H_2^0 = 158178.70$$

By the fuzzy programming method, we have the satisfactory solutions to problem (2.86) as shown in Table 2.5.

Example 2.10. Consider the following multi-objective programming problem with random parameters,

Table 2.5 Interactive process of expected value model

$\bar{\mu_1}$	$\bar{\mu_2}$	H_1	H_2	$\mu_1(H_1)$	$\mu_2(H_2)$	x_1	x_2	x_3	x_4	x_5	λ
1	1	41649.56	214197.2	0.8328	0.8328	216.08	39.62	54.30	20.00	20.00	0.1672
1	0.95	42033.36	212717.6	0.8607	0.8108	215.56	42.20	52.24	20.00	20.00	0.1392
0.95	1	41265.50	215675.4	0.8048	0.8548	216.59	37.04	56.37	20.00	20.00	0.1452
1	0.90	42418.70	211232.6	0.8888	0.7887	215.04	44.79	50.17	20.00	20.00	0.1112
1	1	41649.56	21419.72	0.8328	0.8328	216.08	39.62	54.30	20.00	20.00	0.1672
0.90	1	40881.7	217154.9	0.7768	0.8768	217.11	34.46	58.43	20.00	20.00	0.1232
1	0.85	42796.17	209722.0	0.9163	0.7663	214.57	47.36	48.00	20.00	20.00	0.0837
0.85	1	38756.36	206263.9	0.6220	0.7149	217.63	31.87	60.50	20.00	20.00	0.1012
1	0.8	42999.18	207354.5	0.9311	0.7311	215.28	49.66	43.04	20.00	20.00	0.0689
0.8	1	40112.30	220118.1	0.7208	0.9208	218.14	29.29	62.57	20.00	20.00	0.0792

$$\begin{cases} \max H_1(\boldsymbol{x}, \boldsymbol{\xi}) = Pr\{\sqrt{(x_1 - \xi_1)^2 + (x_2 - \xi_2)^2} \geq 1.75\} \\ \max H_2(\boldsymbol{x}, \boldsymbol{\xi}) = Pr\{\sqrt{(x_1 + \xi_1)^2 + (x_2 + \xi_2)^2} \geq 6.2\} \\ \text{s.t.} \begin{cases} x_1 + x_2 \leq 5 \\ x_1 \geq 0, x_2 \geq 0 \end{cases} \end{cases} \tag{2.87}$$

where $\xi_1 \sim \mathcal{N}(2, 0.5)$ ia normally distributed random variable and $\xi_2 \sim \mathscr{U}(1, 2)$ is also a normally distributed random variable.

Next we will use the random simulation-based PSA algorithm to solve the above problem. Set the initial temperature T_0, the last temperature be 1 and the cooling method be 1 decrement once. The neighborhood can be constructed as follows,

$$x_1^1 = x_1^0 + rh, \;\; x_2^1 = x_2^0 + rh,$$

where r is a random number in (0,1) and h is the step length (here $h = 2.0$). After the simulation with many cycles, we get the optimal solution under different weights as shown in Table 2.6. Figure 2.9 shows the cooling process when the weight is 0.5. The real line expresses the weight sum of two objective functions, and it shows that it gradually converges from T = 440. Figure 2.10 shows the changes of two objective values when the temperature decreases.

Table 2.6 The optimal solution by random simulation-based PSA

w_1	w_2	x_1	x_2	$\bar{f}_1$	$\bar{f}_2$	$\bar{f}$	T_0
0.1	0.9	0.7407	0.7124	0.0400	0.6500	0.5890	500
0.2	0.8	1.0209	1.8018	0.1800	0.5500	0.4760	500
0.3	0.7	0.5342	0.5561	0.0200	0.6000	0.4260	500
0.4	0.6	3.5738	1.4076	0.4400	0.8500	0.6860	500
0.5	0.5	0.2657	2.0287	0.7400	0.6000	0.6700	500

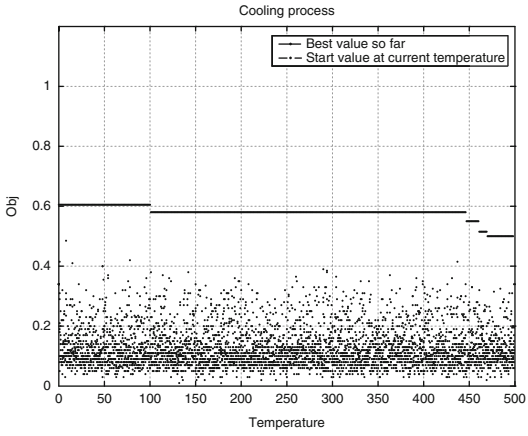


Fig. 2.9 The cooling process of random simulation-based PSA

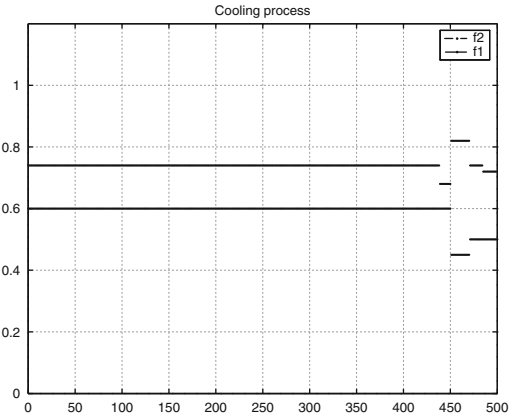


Fig. 2.10 Two objective values by random simulation-based PSA

2.6 Application to Chinese Fruit Beverage

2.6.1 Background Statement

As we know, the DC location problem involves how to select locations of distribution centers from a potential set and how to transport products from the manufactorers to each selected distribution center and also from the selected distribution center to each customer so that the total relevant cost is minimized and customer satisfaction is maximized. However, in practice, it is hard to describe the problem parameters as known due to the complexity of the social and economic environment as well as unpredictable factors such as weather. For instance, since the changing gasoline price often results in a scarcity of precise data, the transport cost from one facility to one DC (or from one DC to a customer) is usually a normal distributed variable with an expected value μ. For some seasonal products, their sales also vary along with the changing season, and usually follow a normal stochastic distribution. Therefore, there does exist a situation where shipping costs and the customer demand where may be random variables. In this situation, we can use random variables to deal with these uncertain parameters. Considering company managers' objectives, we minimize the total relevant costs (including transport costs and fixed costs) of logistic distribution centers, and maximize customer satisfaction in terms of an acceptable delivery time. Hence, we formulate a single-product, multi-objective logistic distribution centers location problem with consumers' random demands and transport costs, and the following assumptions are considered:

1. The number of customers and suppliers are known.
2. The number of potential DCs and their maximum capacities are known.
3. Customers are supplied products from a single DC.
4. The amount of demand on the products and the transport costs are regarded as random variables.

5. Since the manager of this company wants to reduce costs and find an optimal network strategy, inventory is considered as zero in this example.

For understanding this problem easily, we can consult Fig. 2.1.

2.6.2 Notations

The mathematic notation and formulations are as follows:

1. Indices

Suppose that there are I plants, J DCs and K customers. The task is to transfer the products from the plants to the DCs and from DCs to customers to satisfy customer demand.

$i \in \{1, 2, \ldots, I\}$ is the index for plants;

$j \in \{1, 2, \ldots, J\}$ is the index for DCs;

$k \in \{1, 2, \ldots, K\}$ is the index for customers.

2. Parameters

$\widetilde{b_k}$ which is a random variable denoting the demand of customer k;

$\widetilde{pd}_{ij}$ which is a random variable denoting the transport cost of a unit product transported from plant i to distribution center j;

$\widetilde{dc}_{jk}$ which is a random variable denoting the transport cost of a unit product transported from distribution center j to customer k;

t_{jk} denotes the transport time from j to k;

a_i is the capacity of plant i;

f_j denotes the fixed cost of opening distribution center j

m_j denotes the capacity of distribution center j;

3. Decision variables

x_{ij} denotes the quantity transported from plant i to distribution center j.

y_{jk} denotes the quantity transported from distribution center j to customer k.

2.6.3 Modelling and Analysis

For this problem, we need to select the distribution centers from the potential set J. We use the binary z_j to denote whether the distribution center j is selected or not:

$$z_j = \begin{cases} 1, \text{ if distribution center } j \text{ is open} \\ 0, \text{ otherwise} \end{cases}$$

And in this section, we assume that customers are supplied products from a single DC, and we use v_{jk} to denote whether DC j serves customer k.

$$v_{jk} = \begin{cases} 1, \text{ if distribution center } j \text{ servers customer } k \\ 0, \text{ otherwise} \end{cases}$$

3. Objective functions.

The objective F_1 is to minimize the total costs comprised of the fixed costs of DCs $\sum_{j\in J} f_j z_j$, the transport costs from plants to DCs $\sum_{i\in I}\sum_{j\in J} \widetilde{pd}_{ij} x_{ij}$ and from DCs to customers $\sum_{j\in J}\sum_{k\in K} \widetilde{dc}_{jk} y_{jk}$. Further more, the objective F_2 is to maximize customer satisfaction. To measure customer satisfaction, we employ the membership function of fuzzy due time $\zeta_{jk}(t_{jk})$ to specify it, which is described as follows,

$$\zeta_{jk}(t_{jk}) = \begin{cases} 0 & t_{jk} \le ET_k \\ \dfrac{(t_{jk} - ET_k)}{(ET_k^d - ET_k)} & ET_k < t_{jk} < ET_k^d \\ 1 & ET_k^d \le t_{jk} \le LT_k^d \\ \dfrac{LT_k - t_{jk}}{LT_k - LT_k^d} & LT_k^d < t_{jk} < LT_k \\ 0 & t_{jk} \ge LT_k \end{cases} \tag{2.88}$$

where (ET_k, LT_k): the max time range for customer k to tolerant; (ET_k^d, LT_k^d): the expected time range of customer k. Figure 2.11 presents the description of this function.

Thus, we can develop mathematical formulations of objectives as follows,

$$\min F_1 = \sum_{j\in J} f_j z_j + \sum_{i\in I}\sum_{j\in J} \widetilde{pd}_{ij} x_{ij} + \sum_{j\in J}\sum_{k\in K} \widetilde{dc}_{jk} y_{jk},$$
$$\max F_2 = \sum_{j\in J}\sum_{k\in K} \zeta_{jk}(t_{jk}) v_{jk}.$$

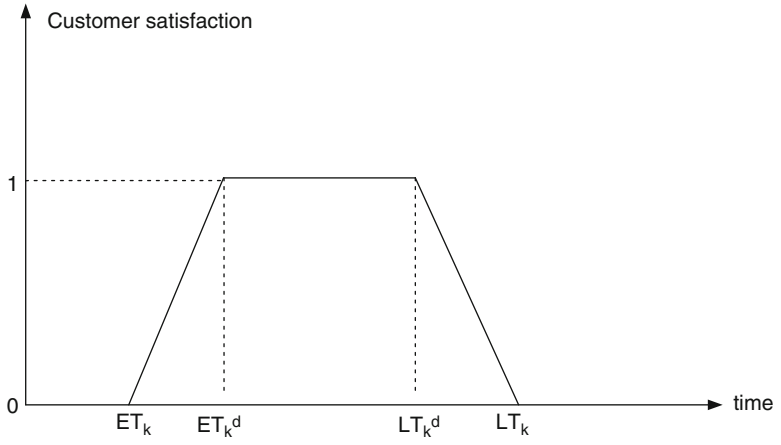


Fig. 2.11 The membership function of $\zeta_{jk}(t_{jk})$

5. Constraints

Since one customer can be serviced by only one DC, thus we employ the constraint

$$\sum_{j \in J} v_{jk} = 1, \qquad \forall k \tag{2.89}$$

With a similar process of dealing with binary variables, we can obtain

$$z_j \in \{0, 1\}, \qquad \forall k \tag{2.90}$$

$$v_{jk} \in \{0, 1\}, \qquad \forall j, k \tag{2.91}$$

And, the number of opened DCs must be not larger than the maximum number M for the DCs. Thus, the constraint

$$\sum_{j \in J} z_j \leq M \tag{2.92}$$

In addition, products transported from the plant i can not exceed its supply capacity a_i, thus we employ the following constraint

$$\sum_{j \in J} x_{ij} \leq a_i, \qquad \forall i \tag{2.93}$$

to specify it. Since there is an assumption of no inventory, the quantity of input is equal to the output at the DC and the quantity of products transported from the DC to the customer is equal to the number of products customers demand. Thus

$$\sum_{i \in I} x_{ij} = \sum_{k \in K} y_{jk}, \qquad \forall j \tag{2.94}$$

$$y_{jk} = \widetilde{b}_k v_{jk}, \qquad \forall j \tag{2.95}$$

Capacity constraints for opened DCs are that the quantity of customers demand on products must be not larger than the quantity of annual throughput of the opened DCs and the quantity of products transported from plants to DCs should be not larger than the DCs' capacity constraint.

$$\sum_{k \in K} \widetilde{b}_k v_{jk} \leq m_j z_j, \qquad \forall j \tag{2.96}$$

$$\sum_{i \in I} x_{ij} \leq m_j z_j, \qquad \forall j \tag{2.97}$$

From the above discussion, by integration of (2.89)–(2.97), we can formulate a random multi-objective programming model as follows:

$$\begin{cases} \min F_1 = \sum\limits_{j\in J} f_j z_j + \sum\limits_{i\in I}\sum\limits_{j\in J} \widetilde{pd}_{ij} x_{ij} + \sum\limits_{j\in J}\sum\limits_{k\in K} \widetilde{dc}_{jk} y_{jk} \\ \max F_2 = \sum\limits_{j\in J}\sum\limits_{k\in K} \zeta_{jk}(t_{jk}) v_{jk} \\ \text{s.t.} \begin{cases} \sum\limits_{j\in J} v_{jk} = 1, \quad \forall k \\ \sum\limits_{j\in J} z_j \le M \\ \sum\limits_{j\in J} x_{ij} \le a_i, \quad \forall i \\ \sum\limits_{i\in I} x_{ij} = \sum\limits_{k\in K} y_{jk}, \quad \forall j \\ y_{jk} = \widetilde{b}_k v_{jk}, \quad \forall j \\ \sum\limits_{k\in K} \widetilde{b}_k v_{jk} \le m_j z_j, \quad \forall j \\ \sum\limits_{i\in I} x_{ij} \le m_j z_j, \quad \forall j \\ z_j = \{0,1\}, \quad \forall k \\ v_{jk} = \{0,1\}, \quad \forall j,k \\ x_{ij} \ge 0, \quad \forall i,j \\ y_{jk} \ge 0, \quad \forall j,k \end{cases} \end{cases} \tag{2.98}$$

Then using the proposed technique in the above sections, let's consider the following real-life problem. Luzhou Xin orchards Co., Ltd. is one of the producers of Chinese fruit beverages in China. Now, the task for a decision-maker is to establish a distribution network at four DC locations which are Shanghai, Beijing, Chengdu and Guangzhou. The objectives, as given in the mathematical model, are the minimization of overall transport costs and the maximization of customer satisfaction. At the beginning of this task, the decision-maker needs to obtain basic data, such as demand quantity, and the per unit transport cost. In fact, since the transport plan is made in advance, we generally cannot exact data. In this condition, the usual way is to obtain random data by means of an experienced evaluation or expert advice. In this example, the notations $\widetilde{b}_k$, $\widetilde{pd}_{ij}$ and $\widetilde{dc}_{jk}$ are employed to denote the demand quantity, transport cost from plant i to DC j and from DC j to the customer respectively. Since it will take more space, we can not give all customers and we give an example of a small sized problem of 5 customers to specify the efficiency of the random simulation-based SA. The information regarding the capacity of the plants and DCs, fixed cost of DCs is given in Table 2.7, and corresponding random data are listed from Tables 2.8 to 2.10.

Transport time from DC j to customer k is in list Table 2.11, the maximum time range for customer tolerance, and the expected time range for the customer can not be directly gained. We assume transport time t_k required by the customer k are independent random variables with a normal distribution: $\mathcal{N}(\mu_k, \sigma_k^2)$, the max

Table 2.7 Capacities and fixed costs for plants and distribution centers

Plants	Capacity(ton/year)	DCs	Capacity (ton/year)	Fixed cost (ten thousand yuan/year)
Jiangy	10,000	Shangh	7,604	48
Longm	10,000	Beij	8,751	44
Luxian	20,000	Chengd	17,600	49
		Wuh	15,800	51

Table 2.8 Shipping cost from plants to DCs (yuan/ton)

Plants	Shipping cost	DCs				σ_i
		Beij	Shangh	Chengd	Wuh	
Jiangy	μ_{1j}^d	550	650	200	350	3
Longm	μ_{2j}^d	550	550	200	350	4
Luxian	μ_{1j}^d	500	500	200	300	3.5

Shipping cost from plants to DCs $\widetilde{pd}_{ij} \sim \mathcal{N}(\mu_{ij}^d, \sigma_i^2)$

Table 2.9 Shipping cost value from DCs to customers (yuan/ton)

DCs	Shipping cost	Customers					σ_i
		1	2	3	4	5	
Shangh	μ_{1j}^c	450	550	200	500	550	4
Beij	μ_{2j}^c	200	400	600	200	550	5
Chengd	μ_{3j}^c	500	200	600	500	400	4
Wuh	μ_{4j}^c	500	310	300	500	550	3

Shipping cost value from DCs to customers $\widetilde{dc}_{jk} \sim \mathcal{N}(\mu_{ij}^c, \sigma_i^2)$

Table 2.10 Customer demand (ton/year)

Customer demand	Customers					σ_j
	1	2	3	4	5	
μ_{1k}	5,950	5,860	5,288	5,520	7,310	15
μ_{2k}	6,000	6,280	7,094	7,760	7,755	20
μ_{3k}	6,050	6,700	8,900	10,000	8,200	16

Customer demand $\widetilde{b}_k \sim \mathcal{N}(\mu_{jk}, \sigma_j^2)$

Table 2.11 Transport time from DCs to customers (h)

Customers DCs	Shangh	Beij	Chengd	Wuh
1	23	32	29	28
2	24	30	27.5	27
3	24.5	30.5	32	30
4	24	33	28.5	30
5	26.5	27	25.5	31.5

time range for customer tolerance is $[\mu_k - \sigma_k, \mu_k + \sigma_k]$ and the expected time range of the customer is $[\mu_k - 0.5\sigma_k, \mu_k + 0.5\sigma_k]$, where μ_k is deemed equal to t_k in practice and σ_k is random from range [1, 5]. Then through (2.88), we have customer satisfaction shown in Table 2.12.

Table 2.12 Customer satisfaction

Customers DCs	μ_k	σ_k	(ET_{jk}, LT_{jk})	(ET^d_{jk}, LT^d_{jk})	Shangh	Beij	Chengd	Wuh
1	28	1.5	[26.5, 29.5]	[27.3, 28.8]	0	0	0.71	1
2	30	4.0	[26.0, 34.0]	[28.0, 32.0]	0	1	0.75	0.5
3	32	1.2	[30.8, 33.2]	[31.4, 32.6]	0	0	1	0
4	28	3.2	[24.8, 31.2]	[26.4, 29.6]	0	0	1	0.75
5	30	4.4	[25.6, 34.4]	[27.8, 32.2]	0.41	0.64	0	1

It is usually difficult for decision makers to justify where to set a distribution and how many products to provide under the condition that there are many uncertain coefficients due to the randomness. So we must apply the methods introduce in the above sections to deal with the uncertainty, i.e., expected value model, chance constraint model and dependent chance model.

Firstly, taking all the random coefficients into the model (2.98), we have the equivalent model by Theorem 2.2 as follows,

$$\begin{cases} \min F_1 = \sum\limits_{j=1}^{4} f_j z_j + \sum\limits_{i=1}^{3}\sum\limits_{j=1}^{4} \mu^d_{ij} x_{ij} + \sum\limits_{j=1}^{4}\sum\limits_{k=1}^{5} \mu^c_{jk} y_{jk} \\ \max F_2 = \sum\limits_{j=1}^{4}\sum\limits_{k=1}^{5} \frac{1}{4}(ET_{jk} + LT_{jk} + ET^d_{jk} + LT^d_{jk}) v_{jk} \\ \text{s.t.} \begin{cases} \sum\limits_{j=1}^{4} v_{jk} = 1, \quad \forall k \\ \sum\limits_{j=1}^{4} z_j \le M \\ \sum\limits_{j=1}^{4} x_{ij} \le a_i, \quad \forall i \\ \sum\limits_{i=1}^{3} x_{ij} = \sum\limits_{k=1}^{5} y_{jk}, \quad \forall j \\ y_{jk} = \mu_{jk} v_{jk}, \quad \forall j \\ \sum\limits_{k=1}^{5} \mu_{jk} v_{jk} \le m_j z_j, \quad \forall j \\ \sum\limits_{i=1}^{3} x_{ij} \le m_j z_j, \quad \forall j \\ z_j = \{0, 1\}, \quad \forall k \\ v_{jk} = \{0, 1\}, \quad \forall j, k \\ x_{ij} \ge 0, \quad \forall i, j \\ y_{jk} \ge 0, \quad \forall j, k \end{cases} \end{cases} \tag{2.99}$$

If the system requires the number of distribution centers cannot be more than 3, i.e., $M = 3$. Then we have the optimal solution as follows: $z* = (1, 1, 0, 1)^T$.

Secondly, if decision makers want to control the cost budget and satisfaction levels to a certain extent, for example, under probability $\alpha = 0.9$ and with credibility $\beta = 0.95$, we can apply the chance constraint operator to deal with it. Taking all the

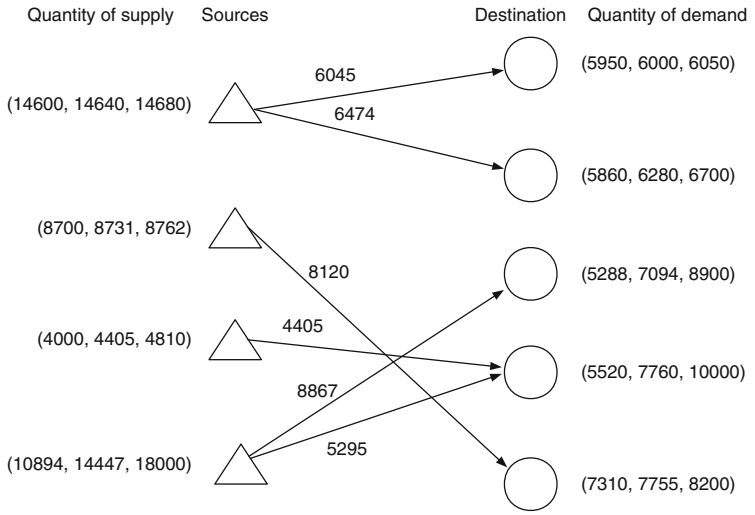


Fig. 2.12 Illustration of the optimal transportation plan

numbers into the model (2.98), the chance constraint model is as follows,

$$
\begin{cases}
\min[\bar{f_1}, -\bar{f_2}] \\
\text{s.t.} \begin{cases}
Pr\left\{ \sum\limits_{j=1}^{4} f_j z_j + \sum\limits_{i=1}^{3} \sum\limits_{j=1}^{4} \widetilde{pd}_{ij} x_{ij} + \sum\limits_{j=1}^{4} \sum\limits_{k=1}^{5} \widetilde{dc}_{jk} y_{jk} \le \bar{f_1} \right\} \ge 0.9 \\
\sum\limits_{j=1}^{4} v_{jk} = 1, \quad \forall k \\
\sum\limits_{j=1}^{4} z_j \le M \\
\sum\limits_{j=1}^{4} x_{ij} \le a_i, \quad \forall i \\
\sum\limits_{i=1}^{3} x_{ij} = \sum\limits_{k=1}^{5} y_{jk}, \quad \forall j \\
y_{jk} = \mu_{jk} v_{jk}, \quad \forall j \\
\sum\limits_{k=1}^{5} \mu_{jk} v_{jk} \le m_j z_j, \quad \forall j \\
\sum\limits_{i=1}^{3} x_{ij} \le m_j z_j, \quad \forall j \\
z_j = \{0, 1\}, \quad \forall k \\
v_{jk} = \{0, 1\}, \quad \forall j, k \\
x_{ij} \ge 0, \quad \forall i, j \\
y_{jk} \ge 0, \quad \forall j, k
\end{cases}
\end{cases}
\tag{2.100}
$$

Then by Theorem 2.6, we can get the equivalent of (2.100) and by the proposed method in the above section we get the optimal transport plan as shown in Fig. 2.12.

Chapter 3
Bi-Random Multiple Objective Decision Making

Random variables have been applied in many fields. The two-fold uncertain variable has also been presented by many scholars. Since the novel concept of birandom variable (abbr. Ra-Ra variable) was proposed by Peng and Liu [251], it has been widely extended to many fields. Xu and Zhou [348] discussed a class of flow shop scheduling problems with Ra-Ra parameters and made use of the expected value operator to deal with it and get the optimal solution. Xu and Ding [338] considered a class of vendors selection problems under the Ra-Ra environment and gave a linear multiobjective programming model with Ra-Ra coefficients.

This chapter mainly introduces multi-objective decision making problems with Ra-Ra parameters. We further propose two kinds of Ra-Ra variables, i.e., discrete Ra-Ra variables and continuous Ra-Ra variables. After introducing the basic concepts and properties, four parts are presented respectively.

1. Ra-Ra expected value model (abbr. Ra-Ra EVM). Usually, decision makers are difficult to make the decision when they encounter the uncertain parameter. A clear criteria must be brought award to help the decision. The expected value operator of Ra-Ra variables is introduced and the crisp equivalent model is deduced when the distribution is clear.
2. Ra-Ra chance-constraint model (abbr. Ra-Ra CCM). Sometimes, decision makers don't strictly require the objective value to be maximal benefit but only obtain maximum benefit under predetermined confidence levels. Then the chance constrained model is proposed and the crisp equivalent model is deduced when the distribution is clear.
3. Ra-Ra dependent-chance model (abbr. Ra-Ra DCM). When decision makers predetermine an objective value and require the maximal probability that the objective values exceed the predetermined one.

Finally, an application to a flow shop scheduling problem are detailed presented to show the effectiveness of the above models. Readers can refer to the following content to know more details.

J. Xu and L. Yao, *Random-Like Multiple Objective Decision Making*, Lecture Notes in Economics and Mathematical Systems 647, DOI 10.1007/978-3-642-18000-2_3,

3.1 Flow Shop Scheduling Problem with Ra-Ra Phenomena

Recently, flow shop production has been widely used in many industrial areas. For this reason, the flow shop scheduling has been attentively studied over the last 50 years [122]. A flow shop scheduling problem addresses determination of sequencing N jobs needed to be processed on M machines so as to optimize performance measures such as the makespan, tardiness, work in process, number of tardy jobs, idle time, etc. The first research concerned with the flow shop scheduling problem was done by Johnson [149], where he described an exact algorithm to minimize the makespan for a n-jobs and two-machines flow shop scheduling problem (see Fig. 3.1). The flow shop scheduling problem includes many jobs and machines. Hence it is classified as a combinatorial optimization problem.

In the flow shop scheduling literature, there are numerous papers that have investigated these problems [228]. Most research is dedicated to single-criterion problems. For example, Pan et al. [243] consider a two-machine flow shop scheduling problem with minimizing total tardiness as the objective. Bulfin and Hallah [33] propose an exact algorithm to solve the two-machine flow shop scheduling problem with the objective of a weighted number of tardy jobs. Choi et al. [54] investigate a proportionate flow shop scheduling problem where only one machine is different and job processing times are inversely proportional to machine speeds. The objective is to reduce as much as possible the maximum completion time. Grabowski and Pempera [118] address the no-wait flow shop problem with a makespan criterion, then develop and compare various local search algorithms for solving this problem. Wang et al. [330] deal with a two-machine flow shop scheduling problem with deteriorating jobs, minimizing total completion time. We can also find that several objectives are considered to establish the best sequence in flow shop production. For example, Gangadhran and Rajendran [105] apply a simulated annealing technique to a heuristic that minimizes the makespan and the total flow time. Kondakci et al. [169] utilize the shortest processing time and the earliest due date rules to minimize the total flow time and the maximum tardiness penalties. Tavakkoli-Moghaddam et al. [316] proposes a multi-objective model for a flow shop scheduling problem

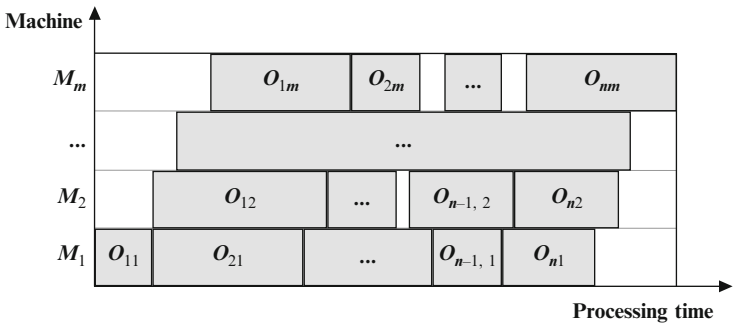


Fig. 3.1 Flow shop scheduling problem with m machines and n jobs

that minimizes both the weighted mean completion time and weighted mean tardiness. We notice that other algorithms have also been developed to deal with the multi-objective scheduling problem, for example, De et al. [73] and Loukil et al. [211] have adapted a multi-objective simulated annealing metaheuristic to solve multi-objective flow shop scheduling problems.

Above all, we notice that the objectives of flow shop scheduling problems mostly focus on minimizing the total completion time, and the makespan. Additionally, objectives such as total flow time, tardiness, and idle time are also considered. But there is little literature which considers that earliness time. In fact the decision maker (DM) often wants to minimize completion time and earliness, and often the objectives conflict with another. Each of these objectives is valid from a general point of view. In this chapter, we focus on this flow shop scheduling problem by considering both completion time and earliness.

Further more, for realistic flow shop scheduling problems we may face uncertainty. In fact, uncertainty exists everywhere in the real world. In order to deal with that, many scholars utilize uncertain variables to describe the uncertainty, like random variables, fuzzy variables, and rough variables. Sometimes people have also used hybrid uncertain variables to be more precise, like the random fuzzy variable, the random rough variable, the fuzzy random variable, and so on.

Because a lot of information is uncertain, the deterministic mathematical model is not suitable to the practical problems, so the uncertain mathematical model is widely used. Scholars introduce uncertain variables into the mathematical model to set up uncertain models and many kinds of uncertain models are proposed, like [24, 57, 152, 192, 251].

3.2 Ra-Ra Variable

Ra-Ra variables are initialize by Peng and Liu [251] in 2007. It is a useful tool to deal with some uncertain real-life problems. There do exist many scenes of such bi-random phenomena in our real world. As a general mathematical description for this kind of stochastic phenomenon with incomplete statistical information, Ra-Ra variables is defined as a mapping with some kind of measurability from a probability space to a collection of random variables. In this section, we firstly recall some basic definitions and properties of Ra-Ra variables and give some deeper deduction.

Roughly speaking, a Ra-Ra variable is a "random variable" taking random variable values. In other words, a Ra-Ra variable is a mapping from a probability space to a collection of random variables. In this section, we will define the new concept which is more universal.

Definition 3.1. A Ra-Ra variable ξ is a random variable with a random parameter.

For each given Borel subset B of the real $\mathbf{R}$, the function $Pr\{\xi(\omega) \in B\}$ is a random variable defined on the probability space $(\Omega, \mathscr{A}, Pr)$.

Example 3.1. Let $\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\}$, and $Pr\{\omega_i\} = p_i$, where $\sum_{i=1}^{n} p_i = 1$. Then $(\Omega, \mathscr{A}, Pr)$ is a probability space on which we define a function as

$$\xi(\omega) = \begin{cases} \xi_1, \text{ if } \omega = \omega_1 \\ \xi_2, \text{ if } \omega = \omega_2 \\ \cdots \\ \xi_n, \text{ if } \omega = \omega_n \end{cases}$$

where ξ_i is a normally distributed random variable. Then the function ξ is a Ra-Ra variable.

3.2.1 Discrete Ra-Ra Variable

Having recalled these basic definitions and properties, we will respectively define two kinds of Ra-Ra variables, i.e. discrete Ra-Ra variables and continuous variables, and then six special Ra-Ra variables are presented and their basic properties are respectively exhibited. Based on Definition 3.1, we know that they can be divided into two kinds.

Definition 3.2. Let $\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\}$, and $Pr\{\omega_i\} = p_i, i = 1, 2, \ldots, n$ and $\sum_{i=n}^{n} Pr\{\omega_i\} = 1$, then $(\Omega, \mathscr{A}, Pr)$ is a probability space. A discrete Ra-Ra variable is defined as a function on $(\Omega, \mathscr{A}, Pr)$ as follows,

$$\xi(\omega) = \begin{cases} \xi_1, & \text{if } \omega = \omega_1 \\ \xi_2, & \text{if } \omega = \omega_2 \\ \quad \vdots & \\ \xi_n, & \text{if } \omega = \omega_n \end{cases}$$

where $\xi_i (i = 1, 2, \ldots, n)$ are random variables defined on the probability space $(\Omega_i, \mathscr{A}_i, Pr_i)$, respectively.

Obviously, ξ is a mapping from a probability space to a set combined with random variables. The set can be combined with discrete random variables or continuous random variables or both of them defined on the probability space $(\Omega_i, \mathscr{A}_i, Pr_i)$, respectively. If the probability space $(\Omega, \mathscr{A}, Pr)$ is infinite dimension, then we get the following definition.

Definition 3.3. Let $\Omega = \{\omega_1, \omega_2, \ldots\}$, and $Pr\{\omega_i\} = p_i$, and $\sum_{i=1}^{\infty} Pr\{\omega_i\} = 1$, then $(\Omega, \mathscr{A}, Pr)$ is an infinite dimensional probability space. An infinite dimensional discrete Ra-Ra variable is defined as a function on $(\Omega, \mathscr{A}, Pr)$ as follows,

$$\xi(\omega) = \begin{cases} \xi_1, & \text{if } \omega = \omega_1 \\ \xi_2, & \text{if } \omega = \omega_2 \\ \quad \vdots & \end{cases}$$

where ξ_i are random variables defined on the probability space $(\Omega_i, \mathscr{A}_i, Pr_i)$, respectively, $i = 1, 2, \ldots$.

In general, the following formula is used to describe the distribution of discrete Ra-Ra variables,

$$\begin{pmatrix} \xi_1, \xi_2, \ldots, \xi_n, \ldots \\ p_1, p_2, \ldots, p_n, \ldots \end{pmatrix} \tag{3.1}$$

then (3.1) is called *distribution sequence* of Ra-Ra variable ξ.

Example 3.2. Let $\Omega = \{\omega_1, \omega_2\}$, and $Pr\{\omega_1\} = 0.6$, $Pr\{\omega_2\} = 0.4$. Then $(\Omega, \mathscr{A}, Pr)$ is a probability space on which we define a function as

$$\xi(\omega) = \begin{cases} \xi_1, \text{ if } \omega = \omega_1 \\ \xi_2, \text{ if } \omega = \omega_2 \end{cases}$$

where ξ_1 is a uniformly distributed random variable on $[0, 1]$, and ξ_2 is a normally distributed random variable. Then the function ξ is a Ra-Ra variable.

Example 3.3. Let $\Omega = \{\omega_1, \omega_2, \ldots\}$, and $Pr\{\omega_i\} = \frac{\lambda^{i-1}}{(i-1)!} e^{-\lambda}$. Then $(\Omega, \mathscr{A}, Pr)$ is a probability space on which we define a function as

$$\xi(\omega) = \xi_i, \quad \text{if } \omega = \omega_i$$

where $\xi_i (i = 1, 2, \ldots)$ is a binomially distributed random variable defined on the probability space $(\Omega_i, \mathscr{A}_i, Pr_i)$, respectively. Then the function ξ is an infinite Ra-Ra variable.

Next, we will introduce some special Ra-Ra variables and induce their properties. In many problems, our interest is usually gathered on whether the event occurs in one trial. For example, when we sample to examine products, whether the product is good or bad greatly attracts us. When throwing the coin, we usually pay more attention on that its front side or back side is above. However, in some trials, the probability that the event occurs is not clear and we only know some historical data, then Ra-Ra event is used to describe them.

Definition 3.4. (Bi-random 0–1 distribution) Let $\Omega = \{0, 1\}$ and $Pr\{\omega = 1\} = \bar{p}$, $Pr\{\omega = 0\} = 1 - \bar{p}$, where $\bar{p}$ is a random variable defined on the probability space $(\Omega^*, \mathscr{A}^*, Pr^*)$ such that $0 \leq \bar{p}(\omega^*) \leq 1$ for $\omega^* \in \Omega^*$. Then $(\Omega, \mathscr{A}, Pr)$ is a probability space on which we define a Ra-Ra variable subject to 0–1 distribution as follows

$$\xi(\omega) = \begin{cases} \bar{p} & \text{if } \omega = 1 \\ \bar{q} & \text{if } \omega = 0 \end{cases} \tag{3.2}$$

where $\bar{q} = 1 - \bar{p}$ is also a random variable. Then ξ is a Ra-Ra variable subject to 0–1 distribution on the probability space $(\Omega, \mathscr{A}, Pr)$, denoted by $\xi \sim B(0, 1)$.

Obviously, Bi-random 0–1 distribution is a special discrete Ra-Ra variables, and its *distribution sequence* is as follows,

$$\begin{pmatrix} 0, & 1 \\ \bar{q}, & \bar{p} \end{pmatrix} \tag{3.3}$$

where $\bar{q} = 1-\bar{p}$ and $\bar{p}$ is a random variable on the probability space $(\Omega^*, \mathscr{A}^*, Pr^*)$. In (3.3), if $\bar{p}$ is a fixed number, then $Pr\{\omega = 1\}(0 < Pr\{\omega = k\} < 1)$ is also a fixed number and ξ degenerates a random variable subject 0–1 distribution from $(\Omega, \mathscr{A}, Pr)$ to $\mathbf{R}$.

Example 3.4. Let $\bar{p} \sim \mathscr{N}(\mu, \sigma^2)$ on the probability space $(\Omega^*, \mathscr{A}^*, Pr^*)$, then $\bar{q}$ is a normally distributed random variable with the mean value $1 - \mu$ and the variance σ^2 on the probability space $(\Omega^*, \mathscr{A}^*, Pr^*)$. Then the function ξ defined as Definition 3.4 from $(\Omega, \mathscr{A}, Pr)$ to $(\Omega^*, \mathscr{A}^*, Pr^*)$ is a Ra-Ra variable subject to 0–1 distribution.

As shown in the probability theory, when we do the Bernoulli experiment n times, how much is the probability that the event A occurs k times? Then the binomial distribution of random variable is brought forward. Similarly, in this section we will introduce the binomial distribution of Ra-Ra variable.

In the Bernoulli trial, if the probability $\bar{p}$ that the event A occurs is stochastic, then the event that Bernoulli experiment done n times is subject to Ra-Ra binomial distribution. Now let's give the definition of Ra-Ra binomial distribution.

Definition 3.5. (Bi-random binomial distribution) Let $\Omega = \{0, 1, \ldots, n\}$ and $Pr\{\omega = k\} = \binom{n}{k} \bar{p}^k \bar{q}^{n-k}$, $k = 0, 1, \ldots, n$, where $\bar{p}$ is a random variable defined on the probability space $(\Omega^*, \mathscr{A}^*, Pr^*)$ such that $0 \leq \bar{p}(\omega^*) \leq 1$ for $\omega^* \in \Omega^*$. Then $(\Omega, \mathscr{A}, Pr)$ is a probability space on which we define a Ra-Ra variable subject to 0–1 distribution as follows

$$\xi(\omega) = \binom{n}{k} \bar{p}^k \bar{q}^{n-k}, \quad \text{if } \omega = k \tag{3.4}$$

where $\bar{q} = 1 - \bar{p}$ is also a random variable. Then ξ is a Ra-Ra variable subject to binomial distribution on the probability space $(\Omega, \mathscr{A}, Pr)$, denoted by $\xi \sim B(k, n, \bar{p})$.

Obviously, $\bar{p}$ and $\bar{q}$ are two mappings from the probability space $(\Omega^*, \mathscr{A}^*, Pr^*)$ to $\mathbf{R}$. Then for any $\omega^* \in \Omega^*$, $\bar{p}(\omega^*)$ and $\bar{q}(\omega^*)$ become two certain probability value. It follows that

$$\sum_{i=1}^{n} \binom{n}{k} \bar{p}^k(\omega^*) \bar{q}^{n-k}(\omega^*) = (\bar{p}(\omega^*) + \bar{q}(\omega^*))^n = 1$$

Table 3.1 The numerical result of $\xi \sim B(k, 20, \bar{\lambda})$

k	$b(k, 20, \bar{p})$			k	$b(k, 20, \bar{p})$		
	$\bar{p}(\omega_1) = 0.1$	$\bar{p}(\omega_2) = 0.3$	$\bar{p}(\omega_3) = 0.5$		$\bar{p}(\omega_1) = 0.1$	$\bar{p}(\omega_2) = 0.3$	$\bar{p}(\omega_3) = 0.5$
0	0.1216	0.0008	–	11	–	0.0120	0.1602
1	0.2702	0.0068	–	12	–	0.0039	0.1201
2	0.2852	0.278	0.0002	13	–	0.0010	0.0739
3	0.1901	0.0716	0.0011	14	–	0.0002	0.0370
4	0.898	0.1304	0.0046	15	–	–	0.0148
5	0.0319	0.1789	0.0148	16	–	–	0.0046
6	0.0089	0.1916	0.0370	17	–	–	0.0011
7	0.0020	0.1643	0.0739	18	–	–	0.0002
8	0.0004	0.1144	0.1201	19	–	–	–
9	0.0001	0.0654	0.1602	20	–	–	–
10	–	0.0308	0.1762				

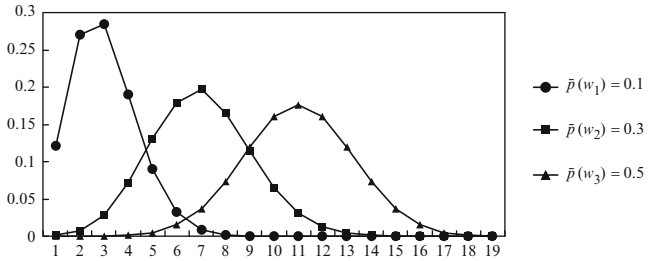


Fig. 3.2 Bi-random binomial distribution

then Definition 3.5 is well defined. In (3.4), if $\bar{p}$ is a fixed number, then $Pr\{\omega = k\}(0 < Pr\{\omega = k\} < 1)$ is also a fixed number and ξ degenerates a random variable subject binomial distribution from $(\Omega, \mathscr{A}, Pr)$ to $\mathbf{R}$.

Example 3.5. In Definition 3.5, assume that $\bar{p}$ has a uniform distribution in $[0.1, 0.5]$, i.e. $\bar{p} \sim \mathscr{U}(0.1, 0.5)$ and $n = 20$, then we can get the numerical result as shown in Table 3.1.

In order to help readers understand easily, we give the intuitive figure of the data of Table 3.1 and Fig. 3.2.

For many real-life problems, we usually encounter many Bernoulli experiments, but n is great and $E[\bar{p}]$ is small. In this case, it is difficult to deal with it by the Ra-Ra binomial distribution. Then Ra-Ra poisson distribution is proposed to deal with some real problems.

Theorem 3.1. *On the probability space $(\Omega, \mathscr{A}, Pr)$ there is a Ra-Ra variable ξ such that $\xi \sim B(n, k, \bar{p})$, where $\bar{p}$ is a random variable defined on the probability space $(\Omega^*, \mathscr{A}^*, Pr^*)$. Assume that $\bar{\lambda}$ is also a random variable on the probability space $(\Omega^*, \mathscr{A}^*, Pr^*)$ such that $\bar{\lambda}(\omega^*) > 0$ for $\omega^* \in \Omega^*$ and satisfies*

$$\lim_{n \to \infty} n \bar{p}_n \to^d \bar{\lambda} \tag{3.5}$$

then for any nonnegative integer number k,

$$\lim_{n\to\infty}\binom{n}{k}\bar{p}_n^k\bar{q}_n^{n-k} \to^d e^{-\bar{\lambda}}\frac{\bar{\lambda}^k}{k!} \tag{3.6}$$

Proof. Since $\bar{p}_n$ and $\bar{\lambda}$ are both random variables on $(\Omega^*, \mathscr{A}^*, Pr^*)$, and satisfy that

$$\lim_{n\to\infty} n\bar{p}_n \to^d \bar{\lambda}.$$

This means that $n\bar{p}$ is convergent in distribution, and for any ω^* the following formula holds,

$$\lim_{n\to\infty} n\bar{p}_n(\omega^*) = \bar{\lambda}(\omega^*).$$

Denote $\bar{\lambda}_n = n\bar{p}_n$. It follows that, for any $\omega^* \in \Omega^*$,

$$\begin{aligned} B(k,n,\bar{p}_n(\omega^*)) &= \binom{n}{k}\bar{p}_n^k(\omega^*)\bar{q}_n^{n-k}(\omega^*) \\ &= \frac{(n)(n-1)\cdots(n-k+1)}{k!}\left(\frac{\lambda(\omega^*)}{n}\right)^k\left(1-\frac{\lambda(\omega^*)}{n}\right)^{n-k} \\ &= \frac{\lambda^k(\omega^*)}{k!}\left(1-\frac{1}{n}\right)\left(1-\frac{2}{n}\right)\cdots\left(1-\frac{k-1}{n}\right)\left(1-\frac{\lambda(\omega^*)}{n}\right)^{n-k} \end{aligned}$$

For the fixed k, we have

$$\lim_{n\to\infty}\lambda_n^k(\omega^*) = \lambda^k(\omega^*), \lim_{n\to\infty}\left(1-\frac{k-1}{n}\right)\left(1-\frac{\lambda(\omega^*)}{n}\right)^{n-k} = e^{-\lambda(\omega^*)}$$

and

$$\lim_{n\to\infty}\left(1-\frac{1}{n}\right)\left(1-\frac{2}{n}\right)\cdots\left(1-\frac{k-1}{n}\right) = 1$$

thus,

$$\lim_{n\to\infty} B(k,n,\bar{p}_n(\omega^*)) = \frac{\lambda^k(\omega^*)}{k!}e^{-\lambda(\omega^*)} \tag{3.7}$$

Because (3.7) holds for any $\omega^* \in \Omega^*$, $B(k,n,\bar{p}_n)$ is convergent to $\frac{\lambda^k}{k!}e^{-\lambda}$ in distribution, i.e.,

$$\lim_{n\to\infty}\binom{n}{k}\bar{p}_n^k\bar{q}_n^{n-k} \to^d e^{-\bar{\lambda}}\frac{\bar{\lambda}^k}{k!}$$

The theorem is proved. □

Table 3.2 The numerical result of $\xi \sim B(k, 4\bar{\lambda})$

$\bar{\lambda} \sim exp(4), \xi \sim P(k, 4\bar{\lambda}), e = 2.718$							
k	$P(k,4)$	$P(k,3)$	$P(k,2)$	$P(k,1)$	$P(k,0.8)$	$P(k,0.5)$	$P(k,0.2)$
0	0.018	0.049	0.135	0.368	0.449	0.607	0.819
1	0.073	0.015	0.271	0.368	0.359	0.304	0.164
2	0.147	0.224	0.271	0.184	0.144	0.076	0.016
3	0.195	0.224	0.180	0.061	0.038	0.013	0.001
4	0.195	0.168	0.090	0.113	0.008	0.002	–
5	0.156	0.101	0.036	0.003	0.001	–	–

Obviously, $P(k, \bar{\lambda}(\omega^*)) > 0, \sum_{k=0}^{\infty} P(k, \bar{\lambda}(\omega^*)) = e^{-\bar{\lambda}(\omega^*)}, \sum_{k=0}^{\infty} \frac{\bar{\lambda}^k(\omega^*)}{k!} = 1.$

Definition 3.6. (Bi-random poisson distribution) Let $P(k, \bar{\lambda}) = e^{-\bar{\lambda}} \frac{\bar{\lambda}^k}{k!}$, $k = 0, 1, 2, \ldots$, where $\bar{\lambda}$ is a random variable on $(\Omega^*, \mathscr{A}^*, Pr^*)$. Let $Pr\{\xi = \omega_k\} = P(k, \bar{\lambda})$ where $\omega_k \in \{\omega_0, \omega_1, \omega_2, \ldots\}$. Then ξ is called the Ra-Ra variable subject to poisson distribution on the probability space $(\Omega, \mathscr{A}, Pr)$, denoted by $\xi \sim P(k, \bar{\lambda})$.

Example 3.6. In the above definition, let $\bar{\lambda} \sim \mathscr{N}(1, 4)$, then $\xi \sim B(k, 3.5\bar{\lambda})$ is a Ra-Ra variable with normally distributed parameter.

As we know, the random variable with poisson distribution is widely applied to many fields, such as the times how many the accident occurs in a period or the number how many radioactive materials produce particles and so on. However, the historical data may present uncertainty, thus, Ra-Ra variable is an useful to deal with it and help decision makers make the decision. Assume the $\lambda \sim exp(4)$, Table 3.2 shows some results of the Ra-Ra variable $\xi \sim B(k, 4\bar{\lambda})$.

3.2.2 Continuous Ra-Ra Variable

There is another kind of Ra-Ra variables other than discrete variables, that is, continuous Ra-Ra variables. They are also mappings from the probability space $(\Omega, \mathscr{A}, Pr)$ to the set combined with random variables on one or many probability spaces, but it is different from the set Ω when ξ is a discrete Ra-Ra variable. Ω is a continuous set combined with all intervals of the form $(-\infty, a]$, $(a, b]$, (b, ∞) and $\mathbf{R}^n$. Then let's give the definition of continuous variables.

Definition 3.7. Let Ω is a continuous set combined with the interval of the form $(-\infty, a]$, $(a, b]$, (b, ∞) and $\mathbf{R}^n$. $Pr\{-\infty < \xi \leq x\} = \bar{F}(x)$, where $\bar{F}(x)$ is a continuous random function with the random density function $\bar{p}(x)$, i.e.,

$$\bar{F}(x) = \int_{-\infty}^{x} \bar{p}(y)dy \tag{3.8}$$

where $\bar{p}(y)$ is a random variable on another probability space $(\Omega^*, \mathscr{A}^*, Pr^*)$. Then we call ξ a continuous random variable.

By the property of the distribution function, we have that, for any $\omega^* \in \Omega^*$,

$$\bar{p}(x)(\omega^*) \geq 0 \tag{3.9}$$

and

$$\int_{-\infty}^{+\infty} \bar{p}(x)(\omega^*)dx = 1 \tag{3.10}$$

Conversely, for any random function $\bar{p}(x)$ on $(\Omega^*, \mathscr{A}^*, Pr^*)$, they all satisfy (3.9) and (3.10), then the random function $\bar{F}(x)$ defined by (3.8) could be a distribution function when $\omega^* \in \Omega^*$.

Similarly, for any fixed ω, by random function $\bar{F}(x)$ satisfies that for $a, b \in \mathbf{R}$,

$$Pr\{a \leq \xi(\omega) \leq b\} = \bar{F}^b(\omega^*) - \bar{F}^a(\omega^*) = \int_a^b \bar{p}(\omega^*)(x)dx \tag{3.11}$$

Similarly with random variables, for any fixed ω, we have

$$Pr\{\xi(\omega) = c\} = 0 \tag{3.12}$$

Example 3.7. Assume that $\bar{a}$ and $\bar{b}$ are two random variables defined on $(\Omega^*, \mathscr{A}^*, Pr^*)$ and for any $\omega^* \in \Omega^*$, $\bar{b}(\omega^*) \geq \bar{a}(\omega^*)$ holds. Define the following density function,

$$\bar{p}(\omega^*)(x) = \begin{cases} \dfrac{1}{\bar{b}(\omega^*) - \bar{a}(\omega^*)}, & \text{if } \bar{a}(\omega^*) \leq x \leq \bar{b}(\omega^*) \\ 0, & \text{others} \end{cases} \tag{3.13}$$

Then ξ is a continuous Ra-Ra variable.

Example 3.8. Let ξ be a Ra-Ra variable defined on the probability space $(\Omega, \mathscr{A}, Pr)$ satisfying $\xi \sim \mathscr{N}(\bar{\mu}, \sigma^2)$, where $\bar{\mu}$ is also a normally distributed random variable on $(\Omega^*, \mathscr{A}^*, Pr^*)$ with the mean μ and variance σ^{*2}. Then ξ is a continuous Ra-Ra variable.

Next, let's discuss some special continuous Ra-Ra variables and their properties. In Examples 3.7 and 3.8, we have simply introduced the uniformly distributed Ra-Ra variables. In this section, we will give its detailed definition.

Definition 3.8. (Bi-random uniform distribution) Le ξ be a Ra-Ra variable on the probability space $(\Omega, \mathscr{A}, Pr)$, where $Pr = \bar{F}(x)$ with the following density function,

$$\bar{p}(x) = \begin{cases} \dfrac{1}{\bar{b} - \bar{a}}, & \text{if } \bar{a}(\omega^*) \leq x \leq \bar{b}(\omega^*) \\ 0, & \text{others} \end{cases} \tag{3.14}$$

where $\bar{a}$ and $\bar{b}$ are both random variables on $(\Omega^*, \mathscr{A}^*, Pr^*)$ such that $\bar{b}(\omega^*) > \bar{a}(\omega^*)$ for any $\omega^* \in \Omega^*$. Then ξ is a uniformly distributed Ra-Ra variable, denoted by $\xi \sim \mathscr{U}(\bar{a}, \bar{b})$.

As we know, a Ra-Ra variable is a mapping from a probability space to a set combined with random variables or random functions, so does the uniform Ra-Ra variable. From Definition 3.8, we know that $\xi(\omega) = \bar{F}(x)(\omega \in \Omega)$ is a random function. It holds that

$$\begin{aligned} Pr\{-\infty < \xi(\omega^*) < +\infty\} &= \int_{-\infty}^{+\infty} \bar{p}(\omega^*)(x)dx \\ &= \int_{\bar{a}(\omega^*)}^{\bar{b}(\omega^*)} \frac{1}{\bar{b}(\omega^*) - \bar{a}(\omega^*)} dx = 1 \end{aligned} \tag{3.15}$$

Example 3.9. Let $\Omega = \Omega^* = \mathbf{R}$, $\bar{a} \sim \mathscr{U}(1, 4)$ and $\bar{b} \sim \mathscr{U}(5, 6)$. For $\omega^* \in \Omega^*$, we have

$$F^b(\omega^*) = \begin{cases} 0, & \text{if } \omega^* \leq 1 \\ \dfrac{\omega^* - 1}{3}, & \text{if } 1 < \omega^* \leq 4 \\ 1, & \text{if } \omega^* > 4 \end{cases}$$

and

$$F^a(\omega^*) = \begin{cases} 0, & \text{if } \omega^* \leq 5 \\ \omega^* - 5, & \text{if } 5 < \omega^* \leq 6 \\ 1, & \text{if } \omega^* > 6 \end{cases}$$

Obviously, for any $\omega^* \in \Omega^*$, $\bar{b}(\omega^*) \geq \bar{a}(\omega^*)$ holds. Define the density function as Definition 3.8, then ξ is a uniformly distributed Ra-Ra variable.

Example 3.10. Let $\bar{\xi}$ be a random variable on $(\Omega^*, \mathscr{A}^*, Pr^*)$, then $\xi \sim \mathscr{U}(\bar{\xi} + 2, \bar{\xi} + 4)$ is a uniformly distributed Ra-Ra variable.

Similarly, we can also define the normally distributed Ra-Ra variable.

Definition 3.9. (Bi-random normal distribution) If ξ is a Ra-Ra variable on $(\Omega, \mathscr{A}, Pr)$, where $Pr = \bar{F}(x)$ with the following density function,

$$\bar{p}(x) = \frac{1}{\sqrt{2\pi}\bar{\sigma}} e^{-\frac{(x-\bar{\mu})^2}{2\bar{\sigma}^2}}, -\infty < x < +\infty \tag{3.16}$$

where $\bar{\mu}$ or $\bar{\sigma}$ or both of them are random variables on $(\Omega^*, \mathscr{A}^*, Pr^*)$. Then ξ is a normally distributed Ra-Ra variable, denoted by $\xi \sim \mathscr{N}(\bar{\mu}, \bar{\sigma}^2)$.

This definition is well defined and still satisfies the condition of density function and distribution function. Take $\xi \sim \mathscr{N}(\bar{\mu}, \sigma^2)$ as an example. For any $\omega^* \in \Omega^*$, we have

$$\bar{p}(\omega^*)(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\bar{\mu}(\omega^*))^2}{2\sigma^2}} \geq 0 \tag{3.17}$$

and

$$\int_{-\infty}^{+\infty} \bar{p}(\omega^*)(x)dx = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{+\infty} e^{-\frac{(x-\bar{\mu}(\omega^*))^2}{2\sigma^2}} dx = 1 \tag{3.18}$$

Its distribution function can be got as follows,

$$\bar{F}(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}\bar{\sigma}} e^{-\frac{(y-\bar{\mu})^2}{2\bar{\sigma}^2}} dy \tag{3.19}$$

Obviously, $\bar{F}(x)$ is a random function. If $\bar{\mu}$ and $\bar{\sigma}$ degenerates to be a certain number, ξ degenerates to be a random variable following the normal distribution.

Example 3.11. Let $\bar{\mu} \sim \mathscr{U}(0, 1)$ be a uniformly distributed random variable on $(\Omega^*, \mathscr{A}^*, Pr^*)$. Then $\xi \sim \mathscr{N}(\bar{\mu}, 1)$ is a normally distributed Ra-Ra variable.

In order to help readers understand it easily, we list the numerical table as shown in Table 3.3.

Example 3.12. Let $\bar{\sigma} \sim exp(4)$ be an exponentially distributed random variable on $(\Omega^*, \mathscr{A}^*, Pr^*)$. Then $\xi \sim \mathscr{N}(2, \bar{\sigma}^2)$ is a normally distributed Ra-Ra variable.

The Ra-Ra exponential distribution can be defined as follows.

Table 3.3 The numerical result of $\xi \sim \mathscr{N}(\bar{\mu}, \bar{\sigma}^2)$

x	$\xi \sim \mathscr{N}(\bar{\mu}, 1), \bar{\mu} \sim \mathscr{U}(0, 1)$			x	$\xi \sim \mathscr{N}(0, \bar{\sigma}^2), \bar{\sigma} \sim exp(2)$		
	$\bar{\mu} = 0$	$\bar{\mu} = 0.5$	$\bar{\mu} = 1$		$\bar{\sigma} = 0.5$	$\bar{\sigma} = 5$	$\bar{\sigma} = 20$
0	0.5000	0.3085	0.1578	0	0.5000	0.5000	0.5000
0.2	0.5793	0.3821	0.2119	0.2	0.9773	0.5398	0.4602
0.4	0.6554	0.4602	0.2743	0.4	0.9999	0.9773	0.6179
0.6	0.7257	0.5398	0.3446	0.6	0.9999	0.9987	0.8849
0.8	0.7881	0.6179	0.4207	0.8	0.9999	0.9999	0.9452
1.0	0.8413	0.6915	0.5000	1.0	0.9999	0.9999	0.9773
1.2	0.8849	0.7580	0.5793	1.2	0.9999	0.9999	0.9918
1.4	0.9192	0.8159	0.6554	1.4	0.9999	0.9999	0.9974
1.6	0.9452	0.8643	0.7257	1.6	0.9999	0.9999	0.9993
1.8	0.9641	0.9032	0.7881	1.2	0.9999	0.9999	0.9998
2.0	0.9773	0.9332	0.8413	2.0	0.9999	0.9999	0.9999
2.2	0.9861	0.9554	0.8849	2.2	0.9999	0.9999	0.9999
2.4	0.9918	0.9713	0.9192	2.4	0.9999	0.9999	0.9999
2.6	0.9953	0.9821	0.9452	2.6	0.9999	0.9999	0.9999
2.8	0.9974	0.9893	0.9641	2.8	0.9999	0.9999	0.9999
3.0	0.9987	0.9938	0.9773	3.0	0.9999	0.9999	0.9999
3.2	0.9993	0.9965	0.9861	3.2	0.9999	0.9999	0.9999
3.4	0.9997	0.9981	0.9918	3.4	0.9999	0.9999	0.9999

Definition 3.10. (Bi-random exponential distribution) If ξ is a Ra-Ra variable on $(\Omega, \mathscr{A}, Pr)$, where $Pr = \bar{F}(x)$ with the following density function,

$$\bar{p}(x) = \begin{cases} \bar{\lambda} e^{-\bar{\lambda} x}, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{3.20}$$

where $\bar{\lambda}$ is a random variable on $(\Omega^*, \mathscr{A}^*, Pr^*)$ and $\bar{\lambda}(\omega^*) > 0$ for any $\omega^* \in \Omega^*$. Then ξ is an exponentially distributed Ra-Ra variable, denoted by $\xi \sim exp(\bar{\lambda})$.

For any $\omega^* \in \Omega^*$, $\bar{\lambda}(\omega^*)$ is a fixed number, then we have

$$\bar{p}(\omega^*)(x) \geq 0$$

and

$$\int_{-\infty}^{+\infty} \bar{p}(\omega^*)(x)dx = \int_{0}^{+\infty} \bar{\lambda}(\omega^*) e^{-\bar{\lambda}(\omega^*)x} = 1.$$

Its distribution function can be got as follows,

$$\bar{F}(x) = \begin{cases} 1 - e^{-\bar{\lambda} x}, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{3.21}$$

It's obvious that $\bar{F}(x)$ is a random function about the random parameter $\bar{\lambda}$. If $\bar{\lambda}$ degenerates to be a certain number, ξ degenerates to be a random variable.

Example 3.13. Let $\bar{\lambda} \sim \mathscr{U}(2, 4)$ is a random variable on $(\Omega^*, \mathscr{A}^*, Pr^*)$. Then $\xi \sim exp(\bar{\lambda})$ is an exponentially distributed Ra-Ra variable.

Example 3.14. Let $\bar{\lambda}$ is a random variable on $(\Omega^*, \mathscr{A}^*, Pr^*)$ with the following distribution,

$$\bar{\lambda} \sim \begin{pmatrix} 4 & 5 \\ 0.4 & 0.6 \end{pmatrix} \tag{3.22}$$

Then $\xi \sim exp(\bar{\lambda})$ is an exponentially distributed Ra-Ra variable. Its numerical result is listed in Table 3.4.

3.3 Ra-Ra EVM

Because of the existence of random parameters, we usually cannot find the precise decision for a complicated real-life problem. Hence, an efficient tool should be provided to convert the random parameter into a crisp one. The expected value

Table 3.4 The numerical result of $\xi \sim exp(\bar{\lambda})$

$\xi \sim exp(\bar{\lambda})$					
x	$\lambda = 4$	$\lambda = 5$	x	$\lambda = 4$	$\lambda = 5$
0	0	0	0.8	0.9592	0.9817
0.1	0.3297	0.3934	1.0	0.9817	0.9933
0.2	0.5506	0.6321	1.5	0.9975	0.9994
0.3	0.6988	0.7768	2.0	0.9997	0.9999
0.4	0.7981	0.8646	2.5	0.9999	0.9999
0.5	0.8646	0.9179	3.0	0.9999	0.9999
0.6	0.9093	0.9502	3.5	0.9999	0.9999
0.7	0.9392	0.9698	4.0	0.9999	0.9999

of uncertain variable serves as a powerful tool for a wide variety of applications. It is naturally desirable to introduce the concept of the expected value of a Ra-Ra variable.

3.3.1 General Model for Ra-Ra EVM

First of all, let us recall that the expected value of a random variable ξ on $(\Omega, \mathscr{A}, Pr)$ can be defined as follows,

$$E[\xi] = \int_0^{+\infty} Pr\{\xi \geq t\}dt - \int_{-\infty}^0 Pr\{\xi \leq t\}dt.$$

As we can see in the following part, the expected value of a Ra-Ra variable is somewhat similar to this expression in form. The expected value operator and variance of the Ra-Ra variable are defined as follows,

Definition 3.11. Let ξ be a Ra-Ra variable defined on the probability space $(\Omega, \mathscr{A}, Pr)$. Then the expected value of Ra-Ra variable ξ is defined as

$$E[\xi] = \int_0^{\infty} Pr\{\omega \in \Omega | E[\xi(\omega)] \geq t\}dt - \int_{-\infty}^0 Pr\{\omega \in \Omega | E[\xi(\omega)] \leq t\}dt$$

provided that at least one of the above two integrals is finite.

Similarly to the variance of random variables, the variance of the Ra-Ra variable is given as follows.

Definition 3.12. Let ξ be a Ra-Ra variable with finite expected value $E[\xi]$. Then the variance of Ra-Ra variable ξ is defined as

$$V[\xi] = E[(\xi - E[\xi])^2] \tag{3.23}$$

The following lemmas gives the related properties of the expected value and variance of Ra-Ra variable, and readers can refer to [251] to find the detail.

Lemma 3.1. *Let ξ be a Ra-Ra variable on the probability space $(\Omega, \mathscr{A}, Pr)$. If the expected value $E[\xi(\omega)]$ of random variable $\xi(\omega)$ is finite for each ω, then $E[\xi(\omega)]$ is a random variable on (Ω, A, Pr).*

Lemma 3.2. *Assume that n and g are Ra-Ra variables with finite expected values. Then for any real numbers a and b, we have*

$$E[a\xi + b\eta] = aE[\xi] + bE[\eta].$$

Lemma 3.3. *Assume that ξ is a Ra-Ra variable, a and b are real numbers. Then we have*

$$V[\xi] = E[\xi^2] - (E[\xi])^2$$
$$V[a\xi + b] = a^2 V[\xi]$$

Lemma 3.4. *Assume that ξ is a Ra-Ra variable whose expected value exists. Then we have*

$$V[E[\xi(\omega)]] \leq V[\xi] \tag{3.24}$$

Remark 3.1. We can note that there is the expected value operator E in both sides of the above definition of $E[\xi]$. In fact, the symbol E represents different meanings. That is to say, the overloading allows us to use the same symbol E for different expected value operators, because we can deduce the meaning from the type of argument.

By the definition of expected value operator of Ra-Ra variables, we can compute the expected value of some Ra-Ra variables with special distribution. Next, let's define the expected value of discrete Ra-Ra variables in another way.

Definition 3.13. (Expected value of discrete Ra-Ra variables) Let ξ be a discrete Ra-Ra variable on $(\Omega, \mathscr{A}, Pr)$, its expected value can be defined as follows,

$$E[\xi] = \sum_{\omega_i \in \Omega} Pr\{\omega = \omega_i\} E[\xi_i] \tag{3.25}$$

where $\xi_i (i = 1, 2, \ldots)$ are random variables on the probability space $(\Omega_i, \mathscr{A}_i, Pr_i)$.

By the above definition, we know that if ξ_i degenerates to be a certain number or a certain function, then ξ degenerates to be a random variable on $(\Omega, \mathscr{A}, Pr)$. Next, let's restrict our attention to three kinds of special discrete Ra-Ra variables and discuss their expected value.

Theorem 3.2. *Let ξ be a 0–1 distributed Ra-Ra variable and $\bar{p} \sim \mathscr{U}(a, b)$, where $0 < a < b < 1$. Then we have*

$$E[\xi] = \frac{a+b}{2} \tag{3.26}$$

Proof. Since $\bar{p} \sim \mathscr{U}(a, b)$, then $1 - \bar{p} = \bar{q} \sim \mathscr{U}(1 - b, 1 - a)$. Then by Definition 3.4 and Definition 3.13, we have

$$E[\xi] = 1 \cdot E[\bar{p}] + 0 \cdot E[\bar{q}] = E[\bar{p}] = \frac{a + b}{2} \qquad \square$$

Theorem 3.3. *Let ξ be a 0–1 distributed Ra-Ra variable and $\bar{p}$ holds that*

$$\begin{pmatrix} a_1 & a_2 & \dots & a_m \\ p_1 & p_2 & \dots & p_m \end{pmatrix} \tag{3.27}$$

where $0 < a_i < 1$ and $\sum_{i=1}^{m} p_i = 1$. Then we have

$$E[\xi] = \sum_{i=1}^{m} a_i \, p_i \tag{3.28}$$

Proof. By Definition 3.13, we have

$$E[\xi] = 1 \cdot E[\bar{p}] + 0 \cdot E[\bar{q}] = E[\bar{p}] = \sum_{i=1}^{m} a_i \, p_i \qquad \square$$

From Theorem 3.2 and 3.3, we know that the probability $\bar{p}$ of $Pr\{\omega = 1\}$ and only varies between 0 and 1. Then $0 < E[\xi] < 1$ must hold. The following theorem will present the variance of a 0–1 distributed Ra-Ra variable.

Theorem 3.4. *Let ξ be a 0–1 distributed Ra-Ra variable defined in Theorem 3.2. Then we have*

$$V[\xi] = \frac{a + b}{2} - \left(\frac{a + b}{2} \right)^2 \tag{3.29}$$

Proof. Since $\xi \sim \begin{pmatrix} 0 & 1 \\ \bar{q} & \bar{p} \end{pmatrix}$, then

$$\xi^2 \sim \begin{pmatrix} 0 & 1 \\ 1 - \bar{p}^2 & \bar{p}^2 \end{pmatrix}$$

By Theorem 3.2, we have $E[\xi^2] = \bar{p}^2$. It follows that

$$V[\xi] = E[\xi^2] - (E[\xi])^2 = \frac{a + b}{2} - \left(\frac{a + b}{2} \right)^2 \qquad \square$$

Theorem 3.5. *Let ξ be a 0–1 distributed Ra-Ra variable defined in Theorem 3.3. Then we have*

$$V[\xi] = \sum_{i=1}^{m} a_i^2 p_i - \left(\sum_{i=1}^{m} a_i p_i \right)^2 \tag{3.30}$$

Proof. It is the similar proved process with Theorem 3.4. □

Of course, the probability $\bar{p}$ may have many other distributions, readers could deduce the expected value and variance similarly. Next, we introduce the expected value of binomially distributed Ra-Ra variables.

Theorem 3.6. *Let $\xi \sim B(n, k, \bar{p})$ be a binomially distributed Ra-Ra variable and $\bar{p} \sim \mathscr{U}(a, b)$, where $0 < a < b < 1$. Then we have*

$$E[\xi] = \frac{n(a+b)}{2} \tag{3.31}$$

$$V[\xi] = \frac{(a+b)n}{2} + \frac{(b-a)^2 n^2}{12} - \frac{(b^2+ab+a^2)n}{3} \tag{3.32}$$

Proof. By the definition of the expected value operator of Ra-Ra variables, we have

$$\begin{aligned} E[\xi] &= \sum_{k=0}^{n} kE[\bar{p}_k] = E\left[\sum_{k=0}^{n} k\bar{p}_k \right] \\ &= E\left[\sum_{k=1}^{n} k \binom{n}{k} \bar{p}^k \bar{q}^{n-k} \right] \\ &= E\left[n\bar{p} \sum_{k=1}^{n} \binom{n-1}{k-1} \bar{p}^{k-1} \bar{q}^{(n-1)-(k-1)} \right] \\ &= E[n\bar{p}(\bar{p}+\bar{q})^{n-1}] = \frac{n(a+b)}{2} \end{aligned}$$

Furthermore, we have $\xi^2 \sim B(n, k^2, \bar{p})$. Then

$$\begin{aligned} E[\xi^2] &= \sum_{k=0}^{n} k^2 E[\bar{p}_k] = E\left[\sum_{k=0}^{n} k^2 \bar{p}_k \right] \\ &= E\left[\sum_{k=1}^{n} k^2 \binom{n}{k} \bar{p}^k \bar{q}^{n-k} \right] \\ &= E[n\bar{p}\bar{q} + n^2\bar{p}^2] \\ &= nE[\bar{p}] + (n^2 - n)E[\bar{p}^2] \end{aligned}$$

Since $\bar{p} \sim \mathscr{U}(a, b)$, then

$$E[\bar{p}^2] = \int_a^b \frac{x^2}{b-a} dx = \frac{b^2+ab+a^2}{3}$$

It follows that

$$\begin{aligned} V[\xi] &= E[\xi^2] - (E[\xi])^2 \\ &= \frac{(a+b)n}{2} + \frac{(b^2+ab+a^2)(n^2-n)}{3} - \left[\frac{(a+b)n}{2}\right]^2 \\ &= \frac{(a+b)n}{2} + \frac{(b-a)^2n^2}{12} - \frac{(b^2+ab+a^2)n}{3} \end{aligned}$$

The proof is completed. □

Theorem 3.7. *Let ξ be a binomially distributed Ra-Ra variable and $\bar{p}$ holds that*

$$\begin{pmatrix} a_1 & a_2 & \dots & a_m \\ p_1 & p_2 & \dots & p_m \end{pmatrix} \tag{3.33}$$

where $0 < a_i < 1$ and $\sum_{i=1}^{m} p_i = 1$. Then we have

$$E[\xi] = n \sum_{i=1}^{m} a_i p_i \tag{3.34}$$

$$V[\xi] = n \sum_{i=1}^{m} a_i p_i + (n^2 - n) \sum_{i=1}^{m} a_i^2 p_i - n^2 \left(\sum_{i=1}^{m} a_i p_i \right)^2 \tag{3.35}$$

Proof. We can prove it as the proved process of Theorem 3.6.

$$E[\xi] = nE[\bar{p}] = n \sum_{i=1}^{m} a_i p_i$$

and

$$E[\xi^2] = nE[\bar{p}] + (n^2 - n)E[\bar{p}^2] = n \sum_{i=1}^{m} a_i p_i + (n^2 - n) \sum_{i=1}^{m} a_i^2 p_i.$$

Then

$$V[\xi] = E[\xi^2] - (E[\xi])^2 = n \sum_{i=1}^{m} a_i p_i + (n^2 - n) \sum_{i=1}^{m} a_i^2 p_i - n^2 \left(\sum_{i=1}^{m} a_i p_i \right)^2.$$

This completes the proof. □

For the poisson distributed Ra-Ra variables, we also obtain the following results.

Theorem 3.8. *Let $\xi \sim B(k, \bar{\lambda})$ be a Ra-Ra variable with $\bar{\lambda} \sim \mathscr{U}(a, b)$, where $b > a > 0$. Then we have*

$$E[\xi] = \frac{a+b}{2} \tag{3.36}$$

$$V[\xi] = E[\xi^2] - (E[\xi])^2 = \frac{a+b}{2} + \frac{(b-a)^2}{12} \tag{3.37}$$

Proof. By the definition of the expected value of discrete Ra-Ra variables, we have

$$\begin{aligned} E[\xi] &= \sum_{k=0}^{\infty} kE[p_k] = E\left[\sum_{k=0}^{\infty} k \cdot \frac{\bar{\lambda}^k}{k!} e^{-\bar{\lambda}}\right] \\ &= E\left[\bar{\lambda} e^{-\bar{\lambda}} \sum_{k=1}^{\infty} \frac{\bar{\lambda}^{k-1}}{(k-1)!}\right] \\ &= E[\bar{\lambda}] = \frac{a+b}{2} \end{aligned}$$

Since $\xi \sim B(k, \bar{\lambda})$, then $\xi^2 \sim B(k^2, \bar{\lambda})$. It follows that

$$\begin{aligned} E[\xi^2] &= \sum_{k=0}^{\infty} k^2 E[p_k] = E\left[\sum_{k=0}^{\infty} k^2 \cdot \frac{\bar{\lambda}^k}{k!} e^{-\bar{\lambda}}\right] \\ &= E\left[\sum_{k=1}^{\infty} k \cdot \frac{\bar{\lambda}^k}{(k-1)!} e^{-\bar{\lambda}}\right] = E\left[\bar{\lambda} \sum_{k=0}^{\infty} (k+1) \frac{\bar{\lambda}^k}{k!} e^{-\bar{\lambda}}\right] \\ &= E[\bar{\lambda}^2 + \bar{\lambda}] = E[\bar{\lambda}^2] + E[\bar{\lambda}] \\ &= \frac{b^2 + ab + a^2}{3} + \frac{a+b}{2} \end{aligned}$$

Then

$$V[\xi] = E[\xi^2] - (E[\xi])^2 = \frac{a+b}{2} + \frac{(b-a)^2}{12}.$$

This completes the proof. □

Theorem 3.9. *Let ξ be a Poisson distributed Ra-Ra variable and $\bar{\lambda}$ holds that*

$$\begin{pmatrix} a_1 & a_2 & \cdots & a_m \\ p_1 & p_2 & \cdots & p_m \end{pmatrix} \tag{3.38}$$

where $0 < a_i < 1$ and $\sum_{i=1}^{m} p_i = 1$. Then we have

$$E[\xi] = \sum_{i=1}^{m} a_i p_i \tag{3.39}$$

$$V[\xi] = \sum_{i=1}^{m} a_i^2 p_i + \sum_{i=1}^{m} a_i p_i - \left(\sum_{i=1}^{m} a_i p_i \right)^2 \tag{3.40}$$

Proof. The proof is similar with Theorem 3.7, we have

$$E[\xi] = E[\bar{\lambda}] = \sum_{i=1}^{m} a_i p_i$$

and

$$E[\xi^2] = E[\bar{\lambda}^2] + E[\bar{\lambda}] = \sum_{i=1}^{m} a_i^2 p_i + \sum_{i=1}^{m} a_i p_i$$

Then

$$V[\xi] = E[\xi^2] - (E[\xi])^2 = \sum_{i=1}^{m} a_i^2 p_i + \sum_{i=1}^{m} a_i p_i - \left(\sum_{i=1}^{m} a_i p_i \right)^2$$

This completes the proof. □

For the continuous Ra-Ra variable, we have the following definition which is different from Definition 3.11 in form.

Definition 3.14. (Expected value of continuous Ra-Ra variables) Let ξ be a continuous Ra-Ra variable on $(\Omega, \mathscr{A}, Pr)$ with the density function $\bar{p}(x)$, then its expected value can be defined as follows,

$$E[\xi] = \int_0^{\infty} Pr\left\{ \int_{x \in \Omega^*} x \bar{p}(x) dx \geq r \right\} dr - \int_{-\infty}^{0} Pr\left\{ \int_{x \in \Omega^*} x \bar{p}(x) dx \leq r \right\} dr \tag{3.41}$$

where $\bar{p}(x)$ is a random function defined on $(\Omega^*, \mathscr{A}^*, Pr^*)$.

If ξ degenerates to be a random variable, the definition is identical with the expected value of random variables in Definition 2.9. Next, let's restrict to three special continuous Ra-Ra variables.

Theorem 3.10. *Assume that $\xi \sim \mathscr{U}(\bar{a}, \bar{b})$ is a uniformly distributed Ra-Ra variable, where $\bar{a} \sim \mathscr{U}(a_1, a_2)$ and $\bar{b} \sim \mathscr{U}(b_1, b_2)(b_1 > a_2)$ are both random variables defined on* $\mathbf{R}$. *Then we have*

$$E[\xi] = \frac{a_2 + b_2 + a_1 + b_1}{4}$$

$$V[\xi] = \frac{a_1^2 + a_1 a_2 + a_2^2}{9} + \frac{b_1^2 + b_1 b_2 + b_2^2}{9} + \frac{(a_1 + a_2)(b_1 + b_2)}{12} - \frac{(a_2 + b_2 + a_1 + b_1)^2}{16}$$

Proof. For any $\omega \in \Omega$, we have

$$E[\xi(\omega)] = \int_{-\infty}^{+\infty} x\bar{p}(\omega)(x)dx = \int_{\bar{a}}^{\bar{b}} \frac{x}{\bar{b}-\bar{a}}dx = \frac{\bar{b}+\bar{a}}{2}$$

Obviously, $E[\xi(\omega)]$ is a random variable. Since $\bar{a} \sim \mathscr{U}(a_1, a_2)$ and $\bar{b} \sim \mathscr{U}(b_1, b_2)$ $(b_1 > a_2)$, then $\frac{\bar{b}+\bar{a}}{2}$ is also a random variable with the distribution $\mathscr{U}(\frac{a_1+b_1}{2}, \frac{a_2+b_2}{2})$. Then we have

$$Pr\left\{\frac{\bar{b}+\bar{a}}{2} \geq t\right\} = \begin{cases} 1, & \text{if } t \leq \dfrac{a_1+b_1}{2} \\ \dfrac{a_2+b_2-2x}{a_2+b_2-(a_1+b_1)}, & \text{if } \dfrac{a_1+b_1}{2} < t \leq \dfrac{a_2+b_2}{2} \\ 0, & \text{if } t > \dfrac{a_2+b_2}{2} \end{cases}$$

and

$$Pr\left\{\frac{\bar{b}+\bar{a}}{2} \leq t\right\} = \begin{cases} 0, & \text{if } t \leq \dfrac{a_1+b_1}{2} \\ \dfrac{2x-(a_1+b_1)}{a_2+b_2-(a_1+b_1)}, & \text{if } \dfrac{a_1+b_1}{2} < t \leq \dfrac{a_2+b_2}{2} \\ 1, & \text{if } t > \dfrac{a_2+b_2}{2} \end{cases}$$

According to the definition of continuous Ra-Ra variables, the following formula can be obtained,

$$\begin{aligned} E[\xi] &= \int_0^{\infty} Pr\{\omega \in \Omega | E[\xi(\omega)] \geq t\}dt - \int_{-\infty}^{0} Pr\{\omega \in \Omega | E[\xi(\omega)] \leq t\}dt \\ &= \int_0^{\infty} Pr\left\{\omega \in \Omega | \frac{\bar{b}+\bar{a}}{2} \geq t\right\} dt - \int_{-\infty}^{0} Pr\left\{\omega \in \Omega | \frac{\bar{b}+\bar{a}}{2} \leq t\right\} dt \end{aligned}$$

Then it will be divided into the following three cases.

1. If $\frac{a_2+b_2}{2} \leq 0$, then

$$E[\xi] = -\int_{\frac{a_2+b_2}{2}}^{0} 1dt - \int_{\frac{a_1+b_1}{2}}^{\frac{a_2+b_2}{2}} \frac{2t-(a_1+b_1)}{a_2+b_2-(a_1+b_1)}dt = \frac{a_2+b_2+a_1+b_1}{4}$$

2. If $\frac{a_1+b_1}{2} \leq 0 < \frac{a_2+b_2}{2}$, then

$$\begin{aligned} E[\xi] &= \int_0^{\frac{a_2+b_2}{2}} \frac{(a_2+b_2)-2t}{a_2+b_2-(a_1+b_1)}dt - \int_{\frac{a_1+b_1}{2}}^{0} \frac{2t-(a_1+b_1)}{a_2+b_2-(a_1+b_1)}dt \\ &= \frac{a_2+b_2+a_1+b_1}{4} \end{aligned}$$

3. If $\frac{a_1+b_1}{2} > 0$, then

$$E[\xi] = \int_0^{\frac{a_1+b_1}{2}} 1dt + \int_{\frac{a_1+b_1}{2}}^{\frac{a_2+b_2}{2}} \frac{2t-(a_1+b_1)}{a_2+b_2-(a_1+b_1)}dt = \frac{a_2+b_2+a_1+b_1}{4}$$

Above all, the expected value of ξ is $(a_2+b_2+a_1+b_1)/4$. Similarly, we have

$$E[\xi^2(\omega)] = \int_{-\infty}^{+\infty} x^2 \bar{p}(\omega)(x)dx = \int_{\bar{a}}^{\bar{b}} \frac{x^2}{\bar{b}-\bar{a}}dx = \frac{\bar{a}^2+\bar{a}\bar{b}+\bar{b}^2}{3}$$

Then we have

$$E[\xi^2] = E[E[\xi^2(\omega)]] = E\left[\frac{\bar{a}^2+\bar{a}\bar{b}+\bar{b}^2}{3}\right] = \frac{1}{3}(E[\bar{a}^2]+E[\bar{a}\bar{b}]+E[\bar{b}^2])$$

Since $\bar{a} \sim \mathscr{U}(a_1, a_2)$ and $\bar{b} \sim \mathscr{U}(b_1, b_2)(b_1 > a_2)$ are independently random variables, it follows that

$$E[\bar{a}^2] = \int_{a_1}^{a_2} \frac{x^2}{a_2-a_1}dx = \frac{a_1^2+a_1a_2+a_2^2}{3}$$

$$E[\bar{b}^2] = \int_{b_1}^{b_2} \frac{x^2}{b_2-b_1}dx = \frac{b_1^2+b_1b_2+b_2^2}{3}$$

$$E[\bar{a}\bar{b}] = E[\bar{a}]E[\bar{b}] = \frac{(a_1+a_2)(b_1+b_2)}{4}$$

Then

$$V[\xi] = \frac{a_1^2+a_1a_2+a_2^2}{9} + \frac{b_1^2+b_1b_2+b_2^2}{9} + \frac{(a_1+a_2)(b_1+b_2)}{12} - \frac{(a_2+b_2+a_1+b_1)^2}{16}$$

This completes the proof. □

By the definition of uniformly distributed Ra-Ra variables, random variables can have many other distributions only satisfying $\bar{b}(\omega^*) > \bar{a}(\omega^*)$ for any $\omega^* \in \Omega^*$. Readers can similarly deduce their expected values and variances. Next, let's introduce the expected value and variance of normally distributed Ra-Ra variables.

Theorem 3.11. *Let ξ be a Ra-Ra variable, which is subject to the normal distribution $\mathscr{N}(\bar{\mu}, \sigma^2)$ with $\bar{\mu} \sim \mathscr{U}(a, b)$. Then*

$$E[\xi] = \frac{a+b}{2}$$

Proof. Since $\xi \sim N(\bar{\mu}, \sigma^2)$, it is obvious that $E[\xi(\omega)] = \bar{\mu}$, then we have

$$E[\xi] = \int_0^{\infty} Pr\{\mu \geq t\}dt - \int_{-\infty}^{0} Pr\{\mu \leq t\}dt \tag{3.42}$$

Since $\bar{\mu} \sim \mathscr{U}(a, b)$, then

$$Pr\{\bar{\mu} \geq t\} = \begin{cases} 1, & \text{if } t \leq a \\ \dfrac{b-t}{b-a}, & \text{if } a \leq x \leq b, \\ 0, & \text{if } t > b \end{cases} \qquad Pr\{\bar{\mu} \leq t\} = \begin{cases} 0, & \text{if } x \leq a, \\ \dfrac{x-a}{b-a}, & \text{if } a \leq x < b, \\ 1, & \text{if } x \geq b. \end{cases}$$

Then we have the following three cases.

1. If $a > 0$. Obviously, $\int_{-\infty}^{0} Pr\{\mu \leq t\}dt = 0$, then

$$E[\xi] = \int_0^a 1dt + \int_a^b \frac{b-t}{b-a}dt = \frac{a+b}{2}$$

2. If $a \leq 0 < b$, we have

$$E[\xi] = \int_0^b \frac{b-t}{b-a}dt - \int_a^0 \frac{t-a}{b-a}dt = \frac{a+b}{2}$$

3. If $b \leq 0$, then

$$E[\xi] = -\int_a^b \frac{t-a}{b-a}dt - \int_b^0 1dt = \frac{a+b}{2}$$

Above all, the expected value of ξ is $(a+b)/2$. □

Similarly, readers can compute the expected value if $\bar{\mu}$ follows other distributions or the variance $\bar{\sigma}$ is also a random variable on $(\Omega^*, \mathscr{A}^*, Pr^*)$. Then the expected value and variance of exponentially distributed Ra-Ra variables are introduced in the following part. For the exponential distribution, the parameter $\bar{\lambda}$ must be more than or equal to 0, then we can assume that $\Omega^* = [0, +\infty)$.

Theorem 3.12. *Let $\xi \sim exp(\bar{\lambda})$ be a Ra-Ra variable, where $\bar{\lambda} \sim \mathscr{U}(a, b)$ $(b > a > 0)$ defined on $(\Omega^*, \mathscr{A}^*, Pr^*)(\Omega^* = [0, +\infty))$. Then*

$$E[\xi] = \frac{ln\, b - ln\, a}{b-a} \tag{3.43}$$

$$V[\xi] = \frac{1}{ab} - \left(\frac{ln\, b - ln\, a}{b-a}\right)^2 \tag{3.44}$$

Proof. For any $\omega \in \Omega$, we have

$$E[\xi(\omega)] = \int_{-\infty}^{+\infty} x \cdot \bar{\lambda} e^{-\bar{\lambda}x} dx = \frac{1}{\bar{\lambda}} \tag{3.45}$$

Obviously, $E[\xi(\omega)] = \frac{1}{\bar{\lambda}}$ is a random variable on $(\Omega^*, \mathscr{A}^*, Pr^*)$. Then $\int_{-\infty}^{0} \frac{1}{\bar{\lambda}} \leq tdt = 0$ and

$$\int_0^{\infty} \frac{1}{\bar{\lambda}} \geq tdt = \int_0^{\frac{1}{b}} 1dt + \int_{\frac{1}{b}}^{\frac{1}{a}} \frac{\frac{1}{t} - a}{b-a} dt = \frac{lnb - lna}{b-a}$$

It follows from Definition 3.14 that

$$E[\xi] = \int_0^{+\infty} Pr\{E[\xi(\omega)] \geq t\}dt - \int_{-\infty}^{0} Pr\{E[\xi(\omega)] \leq t\}dt = \frac{lnb - lna}{b-a}$$

From $\xi \sim exp(\bar{\lambda})$, we also have

$$E[\xi^2(\omega)] = \int_{-\infty}^{+\infty} x^2 \cdot \bar{\lambda} e^{-\bar{\lambda}x} dx = \frac{2}{\bar{\lambda}^2}.$$

Obviously, $E[\xi^2(\omega)] = \frac{1}{\bar{\lambda}^2}$ is also a random variable on $(\Omega^*, \mathscr{A}^*, Pr^*)$. Then $\int_{-\infty}^{0} (\frac{1}{\bar{\lambda}^2} \leq t)dt = 0$ and for $t > 0$

$$Pr\left\{\frac{1}{\xi^2} \geq t\right\} = Pr\left\{-\frac{1}{\sqrt{t}} \leq \xi - \frac{1}{\sqrt{t}}\right\}$$

$$= \begin{cases} 0, & \text{if } t \geq \dfrac{1}{a^2} \\ \dfrac{\frac{1}{\sqrt{t}} - a}{b-a}, & \text{if } \dfrac{1}{b^2} \leq t \leq \dfrac{1}{a^2} \\ 1, & \text{if } t \leq \dfrac{1}{b^2} \end{cases}$$

It follows that

$$E[\xi^2] = \int_0^{+\infty} Pr\{E[\xi^2(\omega)] \geq t\}dt - \int_{-\infty}^{0} Pr\{E[\xi^2(\omega)] \leq t\}dt$$
$$= \int_0^{\frac{1}{b^2}} 1dt + \int_{\frac{1}{b^2}}^{\frac{1}{a^2}} \frac{\frac{1}{\sqrt{t}} - a}{b-a} dt = \frac{1}{ab}$$

Thus,

$$V[\xi] = E[\xi^2] - (E[\xi])^2 = \frac{1}{ab} - \left(\frac{lnb - lna}{b-a}\right)^2$$

This completes the proof. □

By the definition of exponentially distributed Ra-Ra variables, we know that the parameter $\bar{\lambda}$ can be a random variable with different distribution, but it must satisfy that $\bar{\lambda}(\omega^*) > 0$ for any $\omega^* \in \Omega^*$. Then readers can deduce the expected value and variance when $\bar{\lambda}$ follows other distributions.

Let's consider the typical single objective with Ra-Ra parameters,

$$\begin{cases} \max f(\boldsymbol{x}, \boldsymbol{\xi}) \\ \text{s.t.} \begin{cases} g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, j = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{3.46}$$

where $f(\boldsymbol{x}, \boldsymbol{\xi})$ and $g_j(\boldsymbol{x}, \boldsymbol{\xi})$, $j = 1, 2, \ldots, p$ are continuous functions in X and $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_n)$ is a Ra-Ra vector on the probability space $(\Omega, \mathscr{A}, Pr)$. Then it follows from the expected operator that

$$\begin{cases} \max E[f(\boldsymbol{x}, \boldsymbol{\xi})] \\ \text{s.t.} \begin{cases} E[g_j(\boldsymbol{x}, \boldsymbol{\xi})] \leq 0, j = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{3.47}$$

After being dealt with by expected value operator, the problem (3.46) has been converted into a certain programming and then decision makers can easily obtain the optimal solution.

Definition 3.15. $\boldsymbol{x}$ is said to be a *feasible solution* of problem (3.47) if and only if $E[g_j(\boldsymbol{x}, \boldsymbol{\xi})] \leq 0$ $(j = 1, 2, \ldots, p)$. For any feasible solution $\boldsymbol{x}$, if $E[f(\boldsymbol{x}^*, \boldsymbol{\xi})] \geq E[f(\boldsymbol{x}, \boldsymbol{\xi})]$, then $\boldsymbol{x}^*$ is the optimal solution of problem (3.47).

In many cases, there are usually multiple objectives decision makers must consider. Thus we have to employ the following expected value model,

$$\begin{cases} \max [E[f_1(\boldsymbol{x}, \boldsymbol{\xi})], E[f_2(\boldsymbol{x}, \boldsymbol{\xi})], \ldots, E[f_m(\boldsymbol{x}, \boldsymbol{\xi})]] \\ \text{s.t.} \begin{cases} E[g_j(\boldsymbol{x}, \boldsymbol{\xi})] \leq 0, j = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{3.48}$$

where $f_i(\boldsymbol{x}, \boldsymbol{\xi})$ are return functions for $i = 1, 2, \ldots, m$. $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_n)$ is a Ra-Ra vector.

Definition 3.16. $\boldsymbol{x}^*$ is said to be the Pareto solution of problem (3.48), if there doesn't exist feasible solutions $\boldsymbol{x}$ such that

$$E[f_i(\boldsymbol{x}, \boldsymbol{\xi})] \geq E[f_i(\boldsymbol{x}^*, \boldsymbol{\xi})], i = 1, 2, \ldots, m$$

and there is at least one $j (j = 1, 2, \ldots, m)$ such that $E[f_i(\boldsymbol{x}, \boldsymbol{\xi})] > E[f_i(\boldsymbol{x}^*, \boldsymbol{\xi})]$.

We can also formulate a random decision system as an expected value goal model (EVGM) according to the priority structure and target levels set by the decision-maker:

$$\begin{cases} \min \sum_{j=1}^{l} P_j \sum_{i=1}^{m} (u_{ij} d_i^+ + v_{ij} d_i^-) \\ \text{s.t.} \begin{cases} E[f_i(\boldsymbol{x}, \boldsymbol{\xi})] + d_i^- - d_i^+ = b_i, i = 1, 2, \ldots, m \\ E[g_j(\boldsymbol{x}, \boldsymbol{\xi})] \leq 0, j = 1, 2, \ldots, p \\ d_i^-, d_i^+ \geq 0, i = 1, 2, \ldots, m \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{3.49}$$

where P_j is the preemptive priority factor which expresses the relative importance of various goals, $P_j \gg P_{j+1}$, for all j, u_{ij} is the weighting factor corresponding to positive deviation for goal i with priority j assigned, v_{ij} is the weighting factor corresponding to negative deviation for goal i with priority j assigned, d_i^+ is the positive deviation from the target of goal i, defined as

$$d_i^+ = [E[f_i(\boldsymbol{x}, \boldsymbol{\xi})] - b_i] \vee 0,$$

d_i^- is the negative deviation from the target of goal i, defined as

$$d_i^- = [b_i - E[f_i(\boldsymbol{x}, \boldsymbol{\xi})]] \vee 0,$$

f_i is a function in goal constraints, g_j is a function in real constraints, b_i is the target value according to goal i, l is the number of priorities, m is the number of goal constraints, and p is the number of real constraints.

3.3.2 Linear Ra-Ra EVM and the Step Method

Generally, many uncertain problems cannot be directly converted into crisp ones unless they have favorable properties and their Ra-Ra parameters have crisp distribution. For those which cannot be transformed, Ra-Ra simulation is an useful tool to deal with them. Next, we will exhibit some examples which can be converted into crisp models. Let's consider the following linear multi-objective programming with Ra-Ra parameters,

$$\begin{cases} \max \left[\tilde{\tilde{c}}_1^T \boldsymbol{x}, \tilde{\tilde{c}}_2^T \boldsymbol{x}, \ldots, \tilde{\tilde{c}}_m^T \boldsymbol{x}\right] \\ \text{s.t.} \begin{cases} \tilde{\tilde{e}}_r^T \boldsymbol{x} \leq \tilde{\tilde{b}}_r, r = 1, 2, \ldots, p \\ \boldsymbol{x} \geq 0 \end{cases} \end{cases} \tag{3.50}$$

where $\boldsymbol{x} \in X \subset \mathbf{R}^n$, $\tilde{\tilde{c}}_i = (\tilde{\tilde{c}}_{i1}, \tilde{\tilde{c}}_{i2}, \ldots, \tilde{\tilde{c}}_{in})^T$, $\tilde{\tilde{e}}_r = (\tilde{\tilde{e}}_{r1}, \tilde{\tilde{e}}_{r2}, \ldots, \tilde{\tilde{e}}_{rn})^T$ are Ra-Ra vectors, and $\tilde{\tilde{b}}_r$ are Ra-Ra variables, $i = 1, 2, \ldots, m$, $r = 1, 2, \ldots, p$. This is

a typical linear Ra-Ra multi-objective problem. Because of the existence of Ra-Ra parameters $\tilde{\bar{c}}_i$, $\tilde{\bar{e}}_r$ and $\tilde{\bar{b}}_r$, we cannot easily obtain its optimal solutions. By Definition 3.13 and 3.14, we can obtain the following expected value model of problem (3.50),

$$\begin{cases} \max\left[E[\tilde{\bar{c}}_1^T\boldsymbol{x}], E[\tilde{\bar{c}}_2^T\boldsymbol{x}], \ldots, E[\tilde{\bar{c}}_m^T\boldsymbol{x}]\right] \\ \text{s.t.} \begin{cases} E[\tilde{\bar{e}}_r^T\boldsymbol{x}] \le E[\tilde{\bar{b}}_r], r = 1, 2, \ldots, p \\ \boldsymbol{x} \ge 0 \end{cases} \end{cases} \tag{3.51}$$

3.3.2.1 Crisp Equivalent Model

In the above model, if these Ra-Ra parameters have crisp distribution, we can get the equivalent model and easily solve them. Let's consider the following theorem in which the special distribution function is added to the Ra-Ra parameters.

Theorem 3.13. *Assume that Ra-Ra vector* $\tilde{\bar{c}}_i = (\tilde{\bar{c}}_{i1}, \tilde{\bar{c}}_{i2}, \ldots, \tilde{\bar{c}}_{in})^T$ *is normally distributed with mean vector* $\bar{\mu}_i^c = (\bar{\mu}_{i1}^c, \bar{\mu}_{i2}^c, \ldots, \bar{\mu}_{in}^c)^T$ *and positive definite covariance matrix* V_i^c*, written as* $\tilde{\bar{c}}_i \sim \mathcal{N}(\bar{\mu}_i^c, V_i^c)(i = 1, 2, \ldots, m)$ *and Ra-Ra vectors* $\tilde{\bar{e}}_r \sim \mathcal{N}(\bar{\mu}_r^e, V_r^e)$, $\tilde{\bar{b}}_r \sim \mathcal{N}(\bar{\mu}_r^b, (\sigma_r^b)^2)$ $(r = 1, 2, \ldots, p)$*, where* $\bar{\mu}_i^c$, $\bar{\mu}_r^e$ *and* $\bar{\mu}_r^b$ *are normally distributed random vectors respectively with mean vectors* $(\mu_i^c)_{n\times 1}$, $(\mu_r^e)_{n\times 1}$, $(\mu_r^b)_{n\times 1}$ *and covariance matrix* Ξ_i^c, Ξ_r^e, Ξ_r^b*. Assume that for any* $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$ *and* $r = 1, 2, \ldots, p$, $\tilde{\bar{c}}_{ij}$, $\tilde{\bar{e}}_{ij}$ *and* $\tilde{\bar{b}}_{ij}$ *are independent with each other. Then problem (3.51) is equivalent to*

$$\begin{cases} \max\left[H_1(\boldsymbol{x}), H_2(\boldsymbol{x}), \ldots, H_m(\boldsymbol{x})\right] \\ s.t. \begin{cases} G_r(\boldsymbol{x}) \le K_r, r = 1, 2, \ldots, p \\ \boldsymbol{x} \ge 0 \end{cases} \end{cases} \tag{3.52}$$

where

$$H_i(\boldsymbol{x}) = -\sqrt{\boldsymbol{x}^T\Xi_i^c\boldsymbol{x}}\left(F\left(\frac{\mu_i^{cT}\boldsymbol{x}}{\sqrt{\boldsymbol{x}^T\Xi_i^c\boldsymbol{x}}}\right) + F\left(-\frac{\mu_i^{cT}\boldsymbol{x}}{\sqrt{\boldsymbol{x}^T\Xi_i^c\boldsymbol{x}}}\right) - 2F(-\infty)\right),$$

$$G_r(\boldsymbol{x}) = -\sqrt{\boldsymbol{x}^T\Xi_r^e\boldsymbol{x}}\left(F\left(\frac{\mu_r^{eT}\boldsymbol{x}}{\sqrt{\boldsymbol{x}^T\Xi_r^e\boldsymbol{x}}}\right) + F\left(-\frac{\mu_r^{eT}x}{\sqrt{x^T\Xi_r^e x}}\right) - 2F(-\infty)\right),$$

$$K_r = -\sqrt{\Xi_r^b}\left(F\left(\frac{\mu_r^b}{\Xi_r^b}\right) + F\left(-\frac{\mu_r^b}{\Xi_r^b}\right) - 2F(-\infty)\right).$$

Proof. Since Ra-Ra variables $\tilde{\bar{c}}_{ij}$ are normally distributed and $\tilde{\bar{c}}_{ij} \sim \mathcal{N}(\bar{\mu}_{ij}^c, V_{ij}^c)$ $(i = 1, 2, \ldots, m, j = 1, 2, \ldots, n)$, then it follows that

$$E[\tilde{\bar{c}}_i^T\boldsymbol{x}] = E\left[\sum_{j=1}^{n} x_j\tilde{\bar{c}}_{ij}\right] = \sum_{j=1}^{n} x_j\bar{\mu}_{ij}^c = \bar{\mu}_i^{cT}\boldsymbol{x}$$

Obviously, $E[\tilde{\bar{c}}_i^T \boldsymbol{x}] = \bar{\mu}_i^{cT} \boldsymbol{x}$ is also a random variable. Since $\bar{\mu}_i^c \sim \mathcal{N}(\mu_{ij}^c, \Xi_i^c)$, it follows that $\bar{\mu}_i^{cT} \boldsymbol{x} \sim \mathcal{N}(\mu_i^{cT} \boldsymbol{x}, \boldsymbol{x}^T \Xi_i^c \boldsymbol{x})$. By Definition 3.11, we have that

$$
\begin{aligned}
E\left[\tilde{\bar{c}}_i^T \boldsymbol{x}\right] &= \int_0^{+\infty} Pr\left\{E\left[\tilde{\bar{c}}_i^T \boldsymbol{x}\right] \geq t\right\} dt - \int_{-\infty}^0 Pr\left\{E\left[\tilde{\bar{c}}_i^T \boldsymbol{x}\right] \leq t\right\} dt \\
&= \int_0^{+\infty} Pr\left\{\frac{E\left[\tilde{\bar{c}}_i^T \boldsymbol{x}\right] - \mu_i^{cT} \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T \Xi_i^c \boldsymbol{x}}} \geq \frac{t - \mu_i^{cT} \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T \Xi_i^c \boldsymbol{x}}}\right\} dt \\
&\quad - \int_{-\infty}^0 Pr\left\{\frac{E\left[\tilde{\bar{c}}_i^T \boldsymbol{x}\right] - \mu_i^{cT} \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T \Xi_i^c \boldsymbol{x}}} \leq \frac{t - \mu_i^{cT} \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T \Xi_i^c \boldsymbol{x}}}\right\} dt \\
&= \int_0^{+\infty} \left(1 - \Phi\left(\frac{t - \mu_i^{cT} \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T \Xi_i^c \boldsymbol{x}}}\right)\right) dt - \int_{-\infty}^0 \Phi\left(\frac{t - \mu_i^{cT} \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T \Xi_i^c \boldsymbol{x}}}\right) dt
\end{aligned}
$$

where $\Phi(x)$ is the standard normally distributed function. Since $\Phi(-x) = 1 - \Phi(x)$, and let $\frac{t - \mu_i^{cT} \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T \Xi_i^c \boldsymbol{x}}} = y$, then it follows that

$$
\begin{aligned}
E\left[\tilde{\bar{c}}_i^T \boldsymbol{x}\right] &= \int_0^{+\infty} \left(1 - \Phi\left(\frac{t - \mu_i^{cT} \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T \Xi_i^c \boldsymbol{x}}}\right)\right) dt - \int_{-\infty}^0 \Phi\left(\frac{t - \mu_i^{cT} \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T \Xi_i^c x \boldsymbol{x}}}\right) dt \\
&= -\sqrt{\boldsymbol{x}^T \Xi_i^c \boldsymbol{x}} \int_{-\infty}^{\frac{\mu_i^{cT} x}{\sqrt{x^T \Xi_i^c x}}} \Phi(y) dy - \sqrt{\boldsymbol{x}^T \Xi_i^c \boldsymbol{x}} \int_{-\infty}^{-\frac{\mu_i^{cT} x}{\sqrt{x^T \Xi_i^c x}}} \Phi(y) dy \\
&= -\sqrt{\boldsymbol{x}^T \Xi_i^c \boldsymbol{x}} \left(F\left(\frac{\mu_i^{cT} \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T \Xi_i^c \boldsymbol{x}}}\right) + F\left(-\frac{\mu_i^{cT} \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T \Xi_i^c \boldsymbol{x}}}\right) - 2F(-\infty)\right)
\end{aligned}
$$

where $F(x)$ is a convergent continuous function and $\frac{dF(x)}{dx} = \Phi(x)$. Similarly,

$$
\begin{aligned}
E\left[\tilde{\bar{e}}_r^T \boldsymbol{x}\right] &= -\sqrt{\boldsymbol{x}^T \Xi_r^e \boldsymbol{x}} \left(F\left(\frac{\mu_r^{eT} x}{\sqrt{\boldsymbol{x}^T \Xi_r^e \boldsymbol{x}}}\right) + F\left(-\frac{\mu_r^{eT} \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T \Xi_r^e \boldsymbol{x}}}\right) - 2F(-\infty)\right) \\
E\left[\tilde{\bar{b}}_r\right] &= -\sqrt{\Xi_r^b} \left(F\left(\frac{\mu_r^b}{\Xi_r^b}\right) + F\left(-\frac{\mu_r^b}{\Xi_r^b}\right) - 2F(-\infty)\right)
\end{aligned}
$$

Let

$$
H_i(\boldsymbol{x}) = -\sqrt{\boldsymbol{x}^T \Xi_i^c \boldsymbol{x}} \left(F\left(\frac{\mu_i^{cT} \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T \Xi_i^c \boldsymbol{x}}}\right) + F\left(\frac{\mu_i^{cT} \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T \Xi_i^c \boldsymbol{x}}}\right) - 2F(-\infty)\right),
$$

$$G_r(\boldsymbol{x}) = -\sqrt{\boldsymbol{x}^T \Xi_r^e \boldsymbol{x}} \left(F\left(\frac{\mu_r^{eT} \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T \Xi_r^e \boldsymbol{x}}} \right) + F\left(-\frac{\mu_r^{eT} \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T \Xi_r^e \boldsymbol{x}}} \right) - 2F(-\infty) \right),$$

$$K_r = -\sqrt{\Xi_r^b} \left(F\left(\frac{\mu_r^b}{\Xi_r^b} \right) + F\left(-\frac{\mu_r^b}{\Xi_r^b} \right) - 2F(-\infty) \right).$$

Then (3.51) is equivalent to the following formula,

$$\begin{cases} \max\left[H_1(\boldsymbol{x}), H_2(\boldsymbol{x}), \ldots, H_m(\boldsymbol{x})\right] \\ \text{s.t.} \begin{cases} G_r(\boldsymbol{x}) \le K_r, r = 1, 2, \ldots, p \\ \boldsymbol{x} \ge 0 \end{cases} \end{cases}$$

The proof is completed. □

Of course, in the real-life problems, Ra-Ra parameters in the linear multiobjective programming problem could follow many different distributed forms, we just only take the normal distribution as an example. Readers can obtain the similar result by the expected value operator.

3.3.2.2 The Step Method

In this section, we use the step method, which is also called STEM method and is the interactive programming method to deal with the multi-objective programming problem [20], to resolve the problem (3.52).

The STEM method is based on the norm ideal point method and its resolving process includes the analysis and decision stage. In the analysis stage, analyzer resolves the problem by the norm ideal point method and provides the decision makers with the solutions and the related objective values and the ideal objective values. In the decision stage, DM gives the tolerance level of the satisfied object to the dissatisfied object to make its objective value better after comparing the objective values obtained in the analysis stage with the ideal point, then provides the analyzer with the information to go on resolving. Do it repeatedly and DM will get the final satisfied solution.

Shimizu once extent the STEM method to deal with the general nonlinear multi-objective programming problem. Interested readers can refer to literatures [277, 287] and others [55, 237, 298] about its further development.

Consider the following multi-objective programming problem,

$$\begin{cases} \min \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})) \\ \text{s.t. } \boldsymbol{x} \in X \end{cases} \tag{3.53}$$

where $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ and $X = \{\boldsymbol{x} \in \mathbf{R}^n | \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \ge 0\}$. Let $\boldsymbol{x}^i$ be the optimal solution of the problem $\min_{\boldsymbol{x} \in X} f_i(\boldsymbol{x})$ and compute each objective function

Table 3.5 Payoff table

$\boldsymbol{f}$	$\boldsymbol{x}^1$	$\cdots$	$\boldsymbol{x}^i$	$\cdots$	$\boldsymbol{x}^m$	max
f_1	$f_{11} = f_1^*$	$\cdots$	f_{1i}	$\cdots$	f_{1m}	$f_1^{\max}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	
f_i	f_{i1}	$\cdots$	$f_{ii} = f_i^*$	$\cdots$	f_{im}	$f_i^{\max}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	
f_m	f_{m1}	$\cdots$	f_{mi}	$\cdots$	$f_{mm} = f_m^*$	$f_m^{\max}$

$f_i(\boldsymbol{x})$ at $\boldsymbol{x}^k$, then we get m^2 objective function value,

$$f_{ik} = f_i(\boldsymbol{x}^k),\ i,k = 1,2,\ldots,m.$$

Denote $f_i^* = f_{ii} = f_i(\boldsymbol{x}^i)$, $\boldsymbol{f}^* = (f_1^*, f_2^*, \ldots, f_m^*)^T$ and f_i^* is a ideal point of the problem (3.53). Compute the maximum value of the objective function $f_i(\boldsymbol{x})$ at every minimum point $\boldsymbol{x}^k$

$$f_i^{\max} = \max_{1\le k\le m} f_{ik}, i = 1,2,\ldots,m.$$

To make it more clearly, we list it in Table 3.5.

According to Table 3.5, we only look for the solution $\boldsymbol{x}$ such that the distance between $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{f}^*$ is minimum, that is, the solution such that each objective is close to the ideal point. Consider the following problem,

$$\min_{\boldsymbol{x}\in X} \max_{1\le i\le m} w_i\,|f_i(\boldsymbol{x}) - f_i^*| = \min_{\boldsymbol{x}\in X} \max_{1\le i\le m} w_i\,|\sum_{j=1}^{n} c_{ij}x_j - f_i^*| \tag{3.54}$$

where $\boldsymbol{w} = (w_1, w_2, \ldots, w_m)^T$ is the weight vector and w_i is the ith weight which can be decided as follows,

$$\alpha_i = \begin{cases} \frac{f_i^{\max} - f_i^*}{f_i^{\max}} \frac{1}{||\boldsymbol{c}_i||}, & f_i^{\max} > 0 \\ \frac{f_i^* - f_i^{\max}}{f_i^{\max}} \frac{1}{||\boldsymbol{c}_i||}, & f_i^{\max} \le 0 \end{cases}, \ i = 1,2,\ldots,m \tag{3.55}$$

$$w_i = \alpha_i / \sum_{i=1}^{m} \alpha_i,\ i = 1,2,\ldots,m, \tag{3.56}$$

where $||\boldsymbol{c}_i|| = \sqrt{\sum_{j=1}^{n} c_{ij}^2}$. Then the problem (3.53) is equivalent to

$$\begin{cases} \min \lambda \\ \text{s.t.} \begin{cases} w_i \left(\sum\limits_{j=1}^{n} c_{ij} x_j - f_i^* \right) \le \lambda, \ i = 1, 2, \ldots, m \\ \lambda \ge 0, \boldsymbol{x} \in X \end{cases} \end{cases} \tag{3.57}$$

Assume that the optimal solution of the problem (3.57) is $(\tilde{\boldsymbol{x}}, \tilde{\lambda})^T$. It is obvious that $(\tilde{\boldsymbol{x}}, \tilde{\lambda})^T$ is a weak efficient solution of the problem (3.5). In order to check if $\tilde{\boldsymbol{x}}$ is satisfied, DM needs to compare $f_i(\tilde{\boldsymbol{x}})$ with the ideal objective value f_i^*, $i = 1, 2, \ldots, m$. If DM has been satisfied with $f_s(\tilde{\boldsymbol{x}})$, but dissatisfied with $f_t(\tilde{\boldsymbol{x}})$, we add the following constraint in the next step in order to improve the objective value f_t,

$$f_t(\boldsymbol{x}) \le f_t(\tilde{\boldsymbol{x}}).$$

For the satisfied object f_s, we add one tolerance level δ_s,

$$f_s(\boldsymbol{x}) \le f_s(\tilde{\boldsymbol{x}}) + \delta_s.$$

Thus, in the problem (3.57), we replace X with the following constraint set,

$$X^1 = \{\boldsymbol{x} \in X \,|\, f_s(\boldsymbol{x}) \le f_s(\tilde{\boldsymbol{x}}) + \delta_s, f_t(\boldsymbol{x}) \le f_t(\tilde{\boldsymbol{x}})\},$$

and delete the objective f_s (do it by letting $w_s = 0$), then resolve the new problem to get better solutions.

In a word, the STEM method can be summarized as follows:

Step 1. Compute every single objective programming problem,

$$f_i(\boldsymbol{x}^i) = \min_{\boldsymbol{x} \in X} f_i(\boldsymbol{x}), \ i = 1, 2, \ldots, m.$$

If $\boldsymbol{x}^1 = \cdots = \boldsymbol{x}^m$, we obtain the optimal solution $\boldsymbol{x}^* = \boldsymbol{x}^1 = \cdots = \boldsymbol{x}^m$ and stop.

Step 2. Compute the objective value of $f_i(\boldsymbol{x})$ at every minimum point $\boldsymbol{x}^k$, then get m^2 objective values $f_{ik} = f_i(\boldsymbol{x}^k)(i, k = 1, 2, \ldots, m)$. List Table 3.5 and we have

$$f_i^* = f_{ii}, \ f_i^{\max} = \max_{1 \le k \le m} f_{ik}, \ i = 1, 2, \ldots, m.$$

Step 3. Give the initial constraint set and let $X^1 = X$.

Step 4. Compute the weight coefficients $w_1, w_2, \ldots, w_m$ by (3.55) and (3.56).

Step 5. Solve the auxiliary problem,

$$\begin{cases} \min \lambda \\ \text{s.t.} \begin{cases} w_i \left(\sum\limits_{j=1}^{n} c_{ij} x_j - f_i^* \right) \le \lambda, \ i = 1, 2, \ldots, m \\ \lambda \ge 0, \boldsymbol{x} \in X^k \end{cases} \end{cases} \tag{3.58}$$

Let the optimal of problem (3.58) be $(\boldsymbol{x}^k, \lambda^k)^T$.

Step 6. DM compare the reference value $f_i(\boldsymbol{x}^k)(i = 1, 2, \ldots, m)$ with the ideal objective value f_i^*. (1) If DM is satisfied with all objective values, output $\tilde{\boldsymbol{x}} = \boldsymbol{x}^k$. (2) If DM is dissatisfied with all objective values, there doesn't exists any satisfied solutions and stop the process. (3) If DM is satisfied with the object $f_{s_k}(1 \le s_k \le m, k < m)$, turn to **Step 7**.

Step 7. DM gives the tolerance level $\delta_{s_k} > 0$ to the object f_{s_k} and construct the new constraint set as follows,

$$X^{k+1} = \{\boldsymbol{x} \in X^k | f_{s_k}(\boldsymbol{x}) \le f_{s_k}(\boldsymbol{x}^k) + \delta_{s_k}, f_i(\boldsymbol{x}) \le f_i(\boldsymbol{x}^k), i \ne s_k\}.$$

Let $\delta_{s_k} = 0, k = k + 1$ and turn to **Step 4**.

3.3.3 Nonlinear Ra-Ra EVM and Ra-Ra Simulation-Based PSO

Let's introduce the process of the Ra-Ra simulation based PSO to deal with the expected value models. Consider the following multi-objective expected value model with Ra-Ra coefficients,

$$\begin{cases} \max \left[E[f_1(\boldsymbol{x}, \boldsymbol{\xi})], E[f_2(\boldsymbol{x}, \boldsymbol{\xi})], \ldots, E[f_m(\boldsymbol{x}, \boldsymbol{\xi})] \right] \\ \text{s.t.} \begin{cases} E[g_r(\boldsymbol{x}, \boldsymbol{\xi})] \le 0, \ r = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases}$$

where $f_i(\boldsymbol{x}, \boldsymbol{\xi})$, or $g_r(\boldsymbol{x}, \boldsymbol{\xi})$ or both of them are nonlinear functions with respect to $\boldsymbol{\xi}, i = 1, 2, \ldots, m, r = 1, 2, \ldots, p, \boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_n)$ is a Ra-Ra vector.

3.3.3.1 Ra-Ra Simulation for EVM

Assume that $\boldsymbol{\xi}$ is an n-dimensional Ra-Ra vector on the probability space $(\Omega, \mathscr{A}, Pr)$, and $f : \mathbf{R}^n \to \mathbf{R}$ is a measurable function. One problem is to calculate the expected value $E[f(\boldsymbol{x}, \boldsymbol{\xi})]$ for the given $\boldsymbol{x}$. Note that, for each $\omega \in \Omega$, we may calculate the expected value $E[f(\boldsymbol{x}, \boldsymbol{\xi}(\omega))]$ by random simulation. Since $E[f(\boldsymbol{x}, \boldsymbol{\xi})]$ is essentially the expected value of random variable $E[f(\boldsymbol{x}, \boldsymbol{\xi}(\omega))]$, we may produce a Ra-Ra simulation by the following process.

Firstly, we sample $\omega_1, \omega_2, \ldots, \omega_N$ from Ω according to Pr, where ω_k is an n-dimensional vector. For each $\omega_i (i = 1, 2, \ldots, N)$, $\boldsymbol{\xi}(\omega_i)$ are all random

variables. Then we can apply stochastic simulation to get their expected values, respectively. Randomly generate $\omega_i^1, \omega_i^2, \ldots, \omega_i^M$ from Ω according to the probability measure Pr for each $\boldsymbol{\xi}(\omega_i)(i = 1, 2, \ldots, N)$. Compute $f(\boldsymbol{x}, \boldsymbol{\xi}(\omega_i^1)), f(\boldsymbol{x}, \boldsymbol{\xi}(\omega_i^2)), \ldots, f(\boldsymbol{x}, \boldsymbol{\xi}(\omega_i^M))$. Then

$$E[f(\boldsymbol{x}, \boldsymbol{\xi}(\omega_i))] = \frac{\sum_{j=1}^{M} f(\boldsymbol{x}, \boldsymbol{\xi}(\omega_i^j))}{M}.$$

Next, we can get the expected value of $f(\boldsymbol{x}, \boldsymbol{\xi})$ as follows,

$$E[f(\boldsymbol{x}, \boldsymbol{\xi})] = \frac{\sum_{i=1}^{N} E[f(\boldsymbol{x}, \boldsymbol{\xi}(\omega_i))]}{N}.$$

Then the procedure simulating the expected value of the function $f(\boldsymbol{x}, \boldsymbol{\xi})$ can be summarized as follows:

Procedure `Ra-Ra simulation for EVM`

Input: The decision vector $\boldsymbol{x}$
Output: The expected value $E[f(\boldsymbol{x}, \boldsymbol{\xi})]$
Step 1. Set $e = l = 0$;
Step 2. Generate ω from Ω according to the probability measure Pr;
Step 3. For the ω, generate ω^i according to the probability measure Pr;
Step 4. Repeat the third step M times;
Step 5. $l \leftarrow l + f(\boldsymbol{x}, \boldsymbol{\xi}(\omega^i))$;
Step 6. Repeat the second to fifth step N times;
Step 7. $e \leftarrow e + l$;
Step 8. Return e/MN.

Example 3.15. We employ the Ra-Ra simulation to calculate the expected value of $\sqrt{\tilde{\tilde{\xi}}_1^2 + \tilde{\tilde{\xi}}_2^2 + \tilde{\tilde{\xi}}_3^2}$, where $\tilde{\tilde{\xi}}_1$, $\tilde{\tilde{\xi}}_2$ and $\tilde{\tilde{\xi}}_3$ are Ra-Ra variables defined as

$$\begin{aligned}
\tilde{\tilde{\xi}}_1 &= \mathscr{U}(\tilde{\rho}_1, \tilde{\rho}_1 + 2), \text{ with } \tilde{\rho}_1 \sim \mathscr{N}(0, 1), \\
\tilde{\tilde{\xi}}_2 &= \mathscr{N}(\tilde{\rho}_2, 1), \qquad \text{with } \tilde{\rho}_2 \sim \mathscr{U}(3, 5), \\
\tilde{\tilde{\xi}}_3 &= exp(\tilde{\rho}_3), \qquad \text{with } \tilde{\rho}_3 \sim \mathscr{U}(1, 2).
\end{aligned}$$

A run of Ra-Ra simulation with 1000 cycles shows that $E\left[\sqrt{\tilde{\tilde{\xi}}_1^2 + \tilde{\tilde{\xi}}_2^2 + \tilde{\tilde{\xi}}_3^2}\right] = 4.6928$.

3.3.3.2 Particle Swarm Optimization Algorithm

Population based stochastic local search techniques are a relatively new paradigm in the field of optimization. There are several nature inspired techniques belonging to this family that use metaphors as guides in solving problems. The most famous members are genetic algorithms that use the metaphor of genetic and evolutionary principles of fitness selection for reproduction to search solution spaces. In a similar fashion the collective behavior of insect colonies, bird flocks, fish schools and other animal societies are the motivation for Swarm Intelligence that is any attempt to design algorithms or distributed problem-solving devices inspired by the collective behavior of social insect colonies and other animal societies [29]. The Particle Swarm Optimization (abbr. PSO) method is a relatively new member of the Swarm Intelligence field for solving optimization problems.

PSO was proposed by Kennedy and Eberhard [158] and is one of the latest evolutionary optimization techniques for optimizing continuous nonlinear functions. PSO incorporates swarming behaviors observed in flocks of birds, schools of fish, or swarms of bees, and even human social behavior, from which the idea is emerged [60, 93, 159, 248]. Its biological inspiration is based on the metaphor of social interaction and communication in a flock of birds or school of fishes. In these groups, there is a leader who guides the movement of the whole swarm. The movement of every individual is based on the leader and on his own knowledge. Since PSO is population-based and evolutionary in nature, the individuals (i.e. particles) in a PSO algorithm tend to follow the leader of the group, i.e. the one with the best performance. In general, it can be said that the model that PSO is inspired, assumes that the behavior of every particle is a compromise between its individual memory and a collective memory. PSO is a population-based optimization tool, which could be implemented and applied easily to solve various function optimization problems, or the problems that can be transformed to function optimization problems. As an algorithm, the main strength of PSO is its fast convergence, which compares favorably with many global optimization algorithms like Genetic Algorithms (GA) [116], Simulated Annealing (SA) [241, 321] and other global optimization algorithms. For applying PSO successfully, one of the key issues is finding how to map the problem solution into the PSO particle, which directly affects its feasibility and performance.

Basic form of PSO. In the PSO algorithm, a solution of a specific problem is represented by an n-dimensional position of a particle. A swarm of fixed number of particles is generated and each particle is initialized with a random position in a multidimensional search space. Each particle files through the multidimensional search space with a velocity. In each step of the iteration the velocity of each particle is adjusted based on three components. The first component is the current velocity of the particle which represents the inertia term or momentum of the particle, i.e. the tendency to continue to move in the same direction. The second component is based on the position corresponds to the best solution is usually referred to as the personal best. The third component is based on the position corresponds to the best solution achieved so far by all the particles, i.e. the global best. Once the velocity of each particle is updated, the particle are then moved to the new positions.

The cycle repeats until the stopping criterion is met. The specific expressions used int he original particle swarm optimization algorithm will be discussed as follows. The PSO algorithm is consisted of a population of particle initialized with random position and velocity. This population of particle is usually called a swarm. In ont iteration step, each particle is first evaluated to find individual objective function value. For each particle, if a position is reached which has a better objective function than the previous best solution, the personal best position is updated. Also, if an objective function is found that is better than the previous best objective function of the swarm the global best position is updated. The velocity is then updated on the particle's personal best position and the global best position found so far by the swarm. Every particle is then moved from the current position to the new position based on its velocity. The precess repeats until the stopping criterion is met.

In PSO, a swarm of L particles served as searching agent for a specific problem solution. A particle's position (Θ_l), which consists of H dimensions, is representing (directly of indirectly) a solution of the problem. The ability of particle to search for solution is represented by its velocity vector (Ω), which drives the movement of particle. In each PSO iteration, every particle moves from one position to the next based on its velocity. By moving from one position to the next, a particle is reaching different prospective solution of the problem. The basic particle movement equation if presented below:

$$\theta_{lh}(t+1) = \theta_{lh}(t) + \omega_{lh}(t+1) \tag{3.59}$$

where, $\theta_{lh}(t+1)$ expresses position of the lth particle at the hth dimension in the $(t+1)$th iteration, $\theta_{lh}(t)$ is the position of the lth particle at the hth dimension in the tth iteration, $\omega_{lh}(t+1)$ is the velocity of the lth particle at the hth dimension in the $(t+1)$th iteration.

PSO also imitated swarm's cognitive and social behavior as local and global search abilities. In the basic version of PSO, the particle's personal best position (Ψ_l) and the global best position (Ψ_g) are always updated and maintained. The personal best position of a particle, which expresses the cognitive or self-learning behavior, is defined as the position that gives the best objective function among the positions that have been visited by that particle. Once a particle reaches a position that has a better objective function than the previous best objective function for this particle, i.e., $Z(\Theta_l) < Z(\Psi_l)$, the personal best position is updated, The global best position, which expresses the social behavior, is the position that gives the best objective function among the positions that have been visited by all particles in the swarm. Once a particle reaches a position that has a better objective function than the previous best objective function for whole swarm, i.e., $Z(\Psi_l) < Z(\Psi_g)$, the global best position is also updated.

The personal best and global best position are used as the basis to update velocity of particle. In each iteration step, the velocity Ω is updated based on three terms: inertia, cognitive learning and social learning terms.

The inertia term forces particle to move in the same direction as in previous iteration. This term is calculated as a product of current velocity with an inertia weight (w).

The cognitive term forces particle to go back to its personal best position. This term is calculated as a product of a random number (u), personal best acceleration constant (c_p), and the difference between personal best position Ψ_l and current position Θ_l.

The social term forces particle to move toward the global best position. This term is calculated as a product of random number (u), global best acceleration constant (c_g), and the difference between global best position Ψ_g and current position Θ_l. Specifically, the equation for velocity updated is expressed as follow:

$$\omega_{lh}(t+1) = w\omega_{lh}(t) + c_p u(\psi_{lh} - \theta_{lh}(t)) + c_g u(\psi_{gh} - \theta_{lh}(t)) \tag{3.60}$$

where, $\omega_{lh}(t+1)$: velocity of the lth particle at the hth dimension in the tth iteration, ψ_{lh}: personal best position of the lth particle at the hth dimension in the tth iteration, ψ_{gh}: global best position at the hth dimension in the tth iteration.

In the velocity-updating formula, random numbers is incorporated in order to randomize particle movement. Hence, two different particles may move to different position in the subsequent iteration even though they have similar position, personal best, and the global best.

The notation used in the algorithm is given as follows:

Notation	Meaning
t:	Iteration index, $t = 1 \ldots T$.
l:	Particle index, $l = 1 \ldots L$.
h:	Dimension index, $h = 1 \ldots H$.
u:	Uniform random number in the interval [0, 1].
$w(t)$:	Inertia weight in the tth iteration.
$\omega_{lh}(t+1)$:	Velocity of the lth particle at the hth dimension in the tth iteration.
$\theta_{lh}(t)$:	Position of the lth particle at the hth dimension in the tth iteration.
ψ_{lh}:	Personal best position of the lth particle at the hth dimension in the tth iteration.
ψ_{gh}:	Global best position at the hth dimension in the tth iteration.
c_p:	Personal best position acceleration constant.
c_g:	Global best position acceleration constant.
$\theta^{\max}$:	Maximum position value.
$\theta^{\min}$:	Minimum position value.
Θ_i:	Vector position of the lth particle, $[\theta_{l1}, \theta_{l2}, \ldots, \theta_{lH}]$.
Ω_l:	Vector velocity of the lth particle, $[\omega_{l1}, \omega_{l2}, \ldots, \omega_{lH}]$.
Ψ_l:	Vector personal best position of the lth particle, $[\psi_{l1}, \psi_{l2}, \ldots, \psi_{lH}]$.
Ψ_g:	Vector global best position, $[\psi_{l1}, \psi_{l2}, \ldots, \psi_{lH}]$.
R_l:	The lth set of solution.
$Z(\Theta_l)$:	Fitness value of Θ_l.

The basic procedure of PSO algorithm can be summarized as follows (Fig. 3.3):

Procedure `The general PSO algorithm`

Step 1. Initialize L particle as a swarm: generate the lth particle with random position Θ_l in the range $[\theta^{\min}, \theta^{\max}]$, velocity $\Omega_l = 0$ and personal best $\Psi_l = \Omega_l$ for $l = 1 \ldots L$. Set iteration $t = 1$;
Step 2. Decode particles into solutions: for $l = 1 \ldots L$, decode $\Theta_l(t)$ to a solution R_l. (This step is only needed if the particles are not directly representing the solutions.);
Step 3. Evaluate the particles: for $l = 1 \ldots L$, compute the performance measurement of R_l, and set this as the fitness value of Θ_l, represent by $Z(\Theta_l)$;
Step 4. Update pbest: (1) for $l = 1 \ldots L$, update $\Psi_l = \Theta_l$, if $Z(\Theta_l) < Z(\Psi_l)$; (2) update $\Psi_g = \Psi_l$, if $Z(\Psi_l) < Z(\Psi_g)$;
Step 5. Update the velocity and the position of each lth particle:

$$w(t) = w(T) + \frac{t - T}{1 - T}[w(1) - w(T)] \tag{3.61}$$

$$\omega_{lh}(t + 1) = w\omega_{lh}(t) + c_p u(\psi_{lh} - \theta_{lh}(t)) + c_g u(\psi_{gh} - \theta_{lh}(t)) \tag{3.62}$$

$$\theta_{lh}(t + 1) = \theta_{lh}(t) + \omega_{lh}(t + 1) \tag{3.63}$$

If $\theta_{lh}(t + 1) > \theta^{\max}$, then

$$\theta_{lh}(t + 1) = \theta^{\max} \tag{3.64}$$

$$\omega_{lh}(t + 1) = 0 \tag{3.65}$$

If $\theta_{lh}(t + 1) < \theta^{\max}$, then

$$\theta_{lh}(t + 1) = \theta^{\min} \tag{3.66}$$

$$\omega_{lh}(t + 1) = 0 \tag{3.67}$$

where the value $\theta^{\max}$ and $\theta^{\min}$ in (3.64) and (3.66) are the upper and lower bounds on the position of particles;
Step 6. If the stopping criteria is met, i.e., t = T, stop; otherwise, $t = t + 1$ and return to step 2.

Key parameters of PSO. Let's discuss possible qualifications and effects of each parameter on the performance of PSO. The parameters consist of the population size (L), two acceleration constants (c_p and c_g), and the inertia weight (w).

Population size (L): This parameter represents the number of particles in the system. It is one important parameter of PSO, because it affects the fitness value and computation time. Furthermore, increasing size of population always increases computation time, but might not improve the fitness value. Generally speaking, too small a population size can lead to poor convergence while too large a population size can yield good convergence at the expense of long running time.

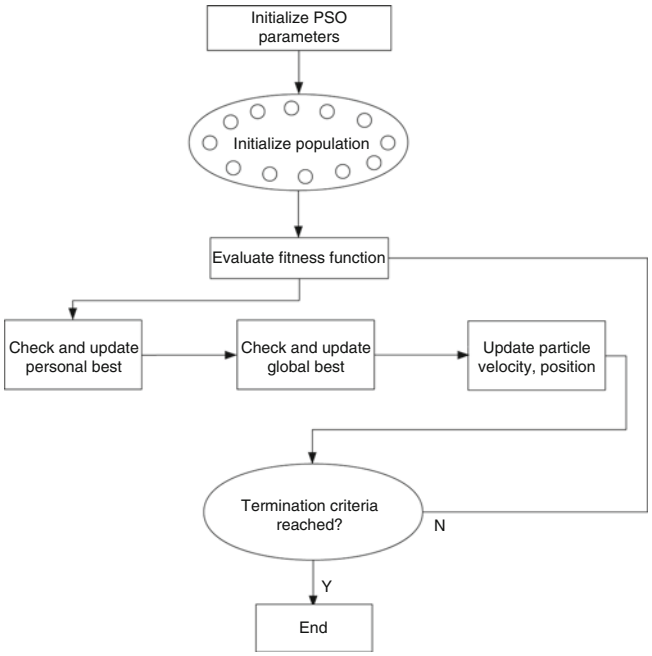


Fig. 3.3 Flow chart of PSO

Acceleration constants (c_p and c_g): The constants c_p and c_g are the acceleration constants of the personal best position and the global best position, respectively. Each acceleration constant controls the maximum distance that a particle is allowed to move from the current position to each best position. The new velocity can be viewed as a vector which combines the current velocity, and the vectors of the best positions. Each best positions's vector consists of the direction which is pointed from the particle's current position to the best position, and the magnitude of the movement can be between 0 to the acceleration constant of the best position times the distance between the best position and the current position.

Inertia weight (w): The new velocity is produced from the combination of vectors. One of these vectors is the current velocity. Inertia weight is a weight to control the magnitude of the current velocity on updating the new velocity. For $w = c$, it means that this vector has the same direction of the current velocity, and the parameters to control the search behavior if the swarm.

Velocity boundary ($V_{\max}$) and position boundary ($\theta_{\max}$): Some PSO algorithms are implemented with bound on velocity. For each dimension, the magnitude of a velocity cannot be greater than $V_{\max}$. This parameter is one of parameters to control the search behavior of the swarm. The smaller value of this parameter makes the particles in the population less aggressive in the search.

In the PSO particle movement mechanism, it is also common to limit the search space of particle location, i.e., the position value of particle dimension is bounded

in the interval $[\theta^{\min}, \theta^{\max}]$. The use of position boundary $\theta^{\max}$ is to force each particle to move within the feasible region to avoid solution divergence. Hence, the position value of certain particle dimension is being set at the minimum or maximum value whenever it moves beyond the boundary. In addition, the velocity of the corresponding dimension is reset to zero to avoid further movement beyond the boundary.

PSO for multi-objective decision-making model. Multi-objective optimization (MO) problems represent an important class of real-world problems. Typically such problems involve trade-offs. For example, a car manufacturer may wish to maximize its profit, but meanwhile also to minimize its production cost. These objectives are typically conflicting to each other. For example, a higher profit could increase the production cost. Generally, there is no single optimal solution. Often the manufacturer needs to consider many possible "trade-off" solutions before choosing the one that suits its need. The curve or surface (for more than two objectives) describing the optimal trade-off solutions between objectives is known as the Pareto front. A multi-objective optimization algorithm is required to find solutions as close as possible to the Pareto front, while maintaining a good solution diversity along the Pareto front.

To apply PSO to multi-objective optimization problems, several issues have to be taken into consideration:

1. How to choose pg (i.e., a leader) for each particle? The PSO needs to favor non-dominated particles over dominated ones, and drive the population towards different parts of the Pareto front, not just towards a single point. This requires that particles be allocated to different leaders.
2. How to identify non-dominated particles with respect to all particles current positions and personal best positions? And how to retain these solutions during the search process? One strategy is to combine all particles personal best positions and current positions, and then extract the non-dominated solutions from the combined population.
3. How to maintain particle diversity so that a set of well-distributed solutions can be found along the Pareto front? Some classic niching methods (e.g., crowding or sharing) can be adopted for this purpose.

The first PSO for solving multi-objective optimization was proposed by Moore and Chapman [224] in 1999. The main difference between single objective PSO and MOPSO is how to choose the global best. An *lbest* PSO was used, and p_g was chosen from a local neighborhood using a ring topology. All personal best positions were kept in an archive. At each particle update, the current position is compared with solutions in this archive to see if the current position represents a non-dominated solution. The archive is updated at each iteration to ensure it contains only non-dominated solutions.

Interestingly it was not until 2002 that the next publication on PSO for multi-objective optimization appeared. The diversity of solutions is maintained by keeping only one solution within each hypercube which is predefined by a user in the objective space. Parsopoulos and Vrahatis [247] adopted a more traditional weighted-sum approach. However, by using gradually changing weights, their approach was able

to find a diverse set of solutions along the Pareto front. Fieldsend and Singh [100] proposed a PSO using a dominated tree structure to store non-dominated solutions found. The selection of leaders was also based on this structure. To maintain a better diversity, a turbulence operator was adopted to function as a "mutation" operator in order to perturb the velocity value of a particle.

With the aim of increasing the efficiency of extracting non-dominated solutions from a swarm, Li [200] proposed NSPSO (Non-dominated Sorting PSO), which follows the principal idea of the well-known NSGA II algorithm [75]. In NSPSO, instead of comparing solely a particles personal best with its potential offspring, all particles personal best positions and offspring are first combined to form a temporary population. After this, domination comparisons for all individuals in this temporary population are carried out. This approach will ensure more non-dominated solutions can be discovered through the domination comparison operations than the above-mentioned multi-objective PSO algorithms.

Many more multi-objective PSO variants have been proposed in recent years. A survey conducted by Reyes–Sierra and Coello [259] in 2006 shows that there are currently 25 different PSO algorithms for handling multi-objective optimization problems. Interested readers should refer to for more information on these different approaches.

The principles that govern PSO algorithm can be stated as follows:

Step 1. Every particle k, a potential solution generated by the Ra-Ra simulation, in the swarm begins with a randomized position and randomized velocity. The position and velocity for particle k in the n-dimensional search space are represented by the vectors $X_k = (x_{k1}, x_{k2}, \ldots, x_{kn})$ and $V_k = (v_{k1}, v_{k2}, \ldots, v_{kn})$, respectively, where $x_{kd}(d = 1, \ldots, n)$ represents the location and $v_{kd}(d = 1, \ldots, n)$ represents the flying velocity of particle k in the dth dimension of the search space.

Step 2. Every particle k knows its position and the value of the objective function for that position. It also remembers at which position $P_k^t = (P_{k1}^t, P_{k2}^t, \ldots, P_{kn}^t)$it has achieved its highest performance.

Step 3. Every particle can generate a neighborhood from every position. Hence, it is also a member of some neighborhood of particles, and remembers which particle (given by the index g) has achieved the best overall position in that neighborhood. This neighborhood can either be a subset of the particles (local neighborhood), or all the particles (global neighborhood).

Step 4. In each iteration t the behavior of particle is a compromise among three possible alternatives: following its current pattern of exploration; going back towards its best previous position; going back towards the best historic value of all particles.

This compromise is executed by the following equations at the current iteration of the algorithm:

$$\begin{aligned} v_{kd}^{t+1} &= wv_{kd}^t + c_1 r_1 (p_{kd}^t - x_{kd}^t) + c_2 r_2 (p_{gd}^t - p_{gd}^t) \\ x_{kd}^{t+1} &= x_{kd}^t + v_{kd}^{t+1} \end{aligned}$$

where w, called the inertia weight, is a constant value chosen by the user to control the impact of the previous velocities on the current velocity. c_1 is the weight given to the attraction to the previous best location of the current particle and c_2 is the weight given to the attraction to the previous best location of the particle neighborhood. r_1 and r_2 are uniformly distributed random variables in [0,1].

The original PSO algorithm can only optimize problems in which the elements of the solution are continuous real numbers. Because it is not possible to continuously "fly" particles through a discrete-valued space, a modification of the PSO algorithm for solving problems with binary-valued solution elements is developed by the initiators of PSO [157]. Another approach for tackling discrete optimization problems by PSO also has been proposed by Laskari, Parsopoulos, and Vrahatis [187] which is based on the truncation of the real values to their nearest integer. In recent years a considerable amount of efforts has been put on solving sequencing problems by PSO algorithm. One of the basic approaches in solving sequencing problems by PSO lies on representing a sequence of n jobs with an array of n real numbers [315]. The decoding process then is done based on some rules. For example the position of the smallest value in the array determines the job placed first in the permutation. Then the position of the second smallest value in the array determines the job placed second in the permutation and so on. Another approach to tackle discrete optimization with PSO is done by generating equations similar to the original PSO equations for updating the particles position and velocity vectors [201,244]. However in these cases the proposed equations are not completely similar to the original ones.

3.3.4 Numerical Examples

Example 3.16. In order to illustrate the proposed model and method, let's consider the following multi-objective programming problem with Ra-Ra coefficients.

$$\begin{cases} \max\ f_1(\boldsymbol{x},\boldsymbol{\xi}) = 3\tilde{\tilde{\xi}}_1^2 x_1 - 2\tilde{\tilde{\xi}}_2 x_2 + 1.3\tilde{\tilde{\xi}}_3 x_3 \\ \max\ f_2(\boldsymbol{x},\boldsymbol{\xi}) = -2.5\tilde{\tilde{\xi}}_4 x_1 + 3\tilde{\tilde{\xi}}_5^2 x_2 + 5\tilde{\tilde{\xi}}_6^2 x_3 \\ \text{s.t.} \begin{cases} x_1 + x_2 + x_3 \le 10 \\ 3x_1 + 5x_2 + 3x_3 \ge 4 \\ x_1, x_2, x_3 \ge 0 \end{cases} \end{cases} \tag{3.68}$$

where $\tilde{\tilde{\xi}}_i\,(i = 1,\ldots,6)$ are all independently Ra-Ra variables as follows,

$$\begin{array}{lll} \tilde{\tilde{\xi}}_1 \sim B(20,k,\bar{p}_1), & \text{with } \bar{p}_1 \sim \mathscr{U}(0.4,0.6), & \tilde{\tilde{\xi}}_2 \sim B(k,\bar{\lambda}_2), \quad \text{with } \bar{\lambda}_2 \sim \mathscr{U}(4,6), \\ \tilde{\tilde{\xi}}_3 \sim \mathscr{U}(3,\bar{b}_3), & \text{with } \bar{b}_3 \sim \mathscr{U}(4,6), & \tilde{\tilde{\xi}}_4 \sim exp(\bar{\lambda}_4), \quad \text{with } \bar{\lambda}_4 \sim \mathscr{U}(2,4), \\ \tilde{\tilde{\xi}}_5 \sim \mathscr{U}(\bar{a}_5,7), & \text{with } \bar{a} \sim \mathscr{U}(1,3), & \tilde{\tilde{\xi}}_6 \sim \mathscr{N}(\bar{u}_6,1), \quad \text{with } \bar{u}_6 \sim \mathscr{N}(4,2). \end{array}$$

Table 3.6 Payoff table of Example 3.16

$\boldsymbol{H}$	$\boldsymbol{x}^1$	$\boldsymbol{x}^2$	min
H_1	3188.1	52	$H_1^{\min} = 52$
H_2	−8.576	1050	$H_2^{\min} = -8.756$

By the expected value operator of Ra-Ra variables, we have the following expected value model of problem (3.68),

$$\begin{cases} \max H_1(\boldsymbol{x}) = 3E[\tilde{\tilde{\xi}}_1^2]x_1 - 2E[\tilde{\tilde{\xi}}_2]x_2 + 1.3E[\tilde{\tilde{\xi}}_3]x_3 \\ \max H_2(\boldsymbol{x}) = -2.5E[\tilde{\tilde{\xi}}_4]x_1 + 3E[\tilde{\tilde{\xi}}_5^2]x_2 + 5E[\tilde{\tilde{\xi}}_6^2]x_3 \\ \text{s.t.} \begin{cases} x_1 + x_2 + x_3 \le 10 \\ 3x_1 + 5x_2 + 3x_3 \ge 4 \\ x_1, x_2, x_3 \ge 0 \end{cases} \end{cases} \tag{3.69}$$

Since,

$$\begin{aligned} &E[\tilde{\tilde{\xi}}_1^2] = 106.27, \quad E[\tilde{\tilde{\xi}}_2] = 5, \quad E[\tilde{\tilde{\xi}}_3] = 4, \\ &E[\tilde{\tilde{\xi}}_4] = 0.347, \quad E[\tilde{\tilde{\xi}}_5^2] = 22.44, \quad E[\tilde{\tilde{\xi}}_6^2] = 21. \end{aligned}$$

Then we have

$$\begin{cases} \max H_1(\boldsymbol{x}) = 318.81x_1 - 10x_2 + 5.2x_3 \\ \max H_2(\boldsymbol{x}) = -0.8675x_1 + 67.32x_2 + 105x_3 \\ \text{s.t.} \begin{cases} x_1 + x_2 + x_3 \le 10 \\ 3x_1 + 5x_2 + 3x_3 \ge 4 \\ x_1, x_2, x_3 \ge 0 \end{cases} \end{cases} \tag{3.70}$$

Then we use the step method to solve the above problem. Firstly, compute the optimal solution and value of each objective function as follows,

$$\begin{aligned} &H_1^* = 3188.1, \quad \boldsymbol{x}^1 = (10.00, 0, 0)^T, \\ &H_2^* = 1050, \quad \boldsymbol{x}^2 = (0, 0, 10.00)^T. \end{aligned} \tag{3.71}$$

By the step method, we get the payoff table as shown in Table 3.6.

Secondly, compute the weight coefficient by the follow method,

$$\begin{aligned} \alpha_1 &= \frac{H_1^* - H_1^{\min}}{H_1^{\min}} \frac{1}{\sqrt{318.81^2 + 10^2 + 5.2^2}} = 0.1891, \\ \alpha_2 &= \frac{H_2^{\min} - H_2^*}{H_2^{\min}} \frac{1}{\sqrt{0.8675^2 + 67.32^2 + 105^2}} = 0.9694. \end{aligned}$$

Then we get $w_1 = \alpha_1/(\alpha_1 + \alpha_2) = 0.1632$ and $w_2 = \alpha_2/(\alpha_1 + \alpha_2) = 0.8368$. The problem (3.70) can be rewritten as,

$$\begin{cases} \max \lambda \\ \text{s.t.} \begin{cases} 0.1632(3188.1 - (318.81x_1 - 10x_2 + 5.2x_3)) \geq \lambda \\ 0.8368(1050 - (-0.8675x_1 + 67.32x_2 + 105x_3)) \geq \lambda \\ x_1 + x_2 + x_3 \leq 10 \\ 3x_1 + 5x_2 + 3x_3 \geq 4 \\ x_1, x_2, x_3 \geq 0 \\ \lambda \geq 0 \end{cases} \end{cases} \tag{3.72}$$

By solving the above problem, we get the optimal solution $\boldsymbol{x}^* = (9.839, 0, 0)^T$.

Example 3.17. Let's consider the following problem,

$$\begin{cases} \max f_1(\boldsymbol{x}, \boldsymbol{\xi}) = \sqrt{(x_1 - \xi_1)^2 + (x_2 - \xi_2)^2} \\ \max f_2(\boldsymbol{x}, \boldsymbol{\xi}) = \sqrt{(x_1 + \xi_1)^2 + (x_2 + \xi_2)^2} \\ \text{s.t.} \begin{cases} x_1 + x_2 \leq 5 \\ x_1 \geq 0, x_2 \geq 0 \end{cases} \end{cases} \tag{3.73}$$

where $\xi_1 \sim \mathcal{N}(\bar{\mu}_1, 1)$ and $\xi_1 \sim \mathcal{N}(\bar{\mu}_2, 1)$ are independently Ra-Ra variables. $\bar{\mu}_1 = \mathcal{N}(3, 1)$ and $\bar{\mu}_1 = \mathcal{N}(2, 0.5)$ are normally distributed random variables. According to the expected value operator of random fuzzy variables, we get the following expected value model,

$$\begin{cases} \max H_1(\boldsymbol{x}, \boldsymbol{\xi}) = E\left[\sqrt{(x_1 - \xi_1)^2 + (x_2 - \xi_2)^2}\right] \\ \max H_2(\boldsymbol{x}, \boldsymbol{\xi}) = E\left[\sqrt{(x_1 + \xi_1)^2 + (x_2 + \xi_2)^2}\right] \\ \text{s.t.} \begin{cases} x_1 + x_2 \leq 5 \\ x_1 \geq 0, x_2 \geq 0 \end{cases} \end{cases} \tag{3.74}$$

Next we will use Ra-Ra simulation-based PSO to solve the above problem. The process of generating a new position for a selected individual in the swarm is depicted in the following equation:

$$v_{t+1}^i = wv_t^i + c_1 \cdot \text{rand}() \cdot (\boldsymbol{P}_t^i - \boldsymbol{x}_t^i) + c_2 \cdot \text{rand}() \cdot (\boldsymbol{G}_t - \boldsymbol{x}_t^i),$$

where v_t^i and $\boldsymbol{x}_t^i$ are the ith particle current velocity and position, $\boldsymbol{P}_t^i$ and $\boldsymbol{G}_t$ are the kth particle best position and the global best position visited so far, $w = 0.7298$ is the inertia weight, $c_1 = c_2 = 1.4962$ are learning factors and rand() is a random number in [0,1]. After the simulation with many cycles, we get the optimal solution under different weights as shown in Table 3.7. Figure 3.4 shows the search process when the weight is 0.5. The read line expresses the weight sum of two objective functions, and it shows that it gradually converges from Gen $=$ 40. Figure 3.5 shows the changes of two objective values when the generation increases.

Table 3.7 The optimal solution by Ra-Ra simulation-based PSO

w_1	w_2	x_1	x_2	$\bar{f}_1$	$\bar{f}_2$	$\bar{f}$	Gen
0.1	0.9	4.6731	0.3269	2.8234	8.1190	7.7906	500
0.2	0.8	4.4346	0.5654	2.3532	7.9339	7.1589	500
0.3	0.7	0.3303	4.6697	4.0580	7.6538	6.7975	500
0.4	0.6	3.5738	1.0343	3.9657	7.3728	5.8378	500
0.5	0.5	4.3234	0.6766	2.3304	8.0034	5.2275	500

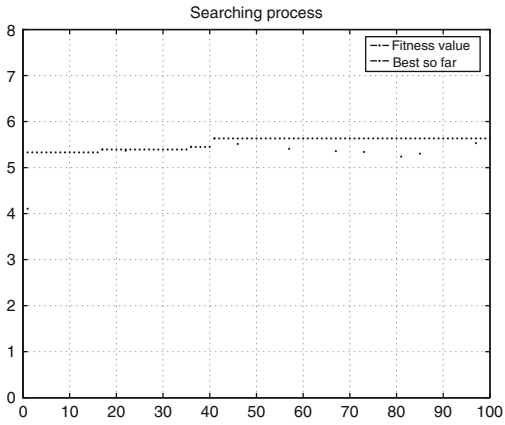


Fig. 3.4 Search process of Ra-Ra simulation-based PSO

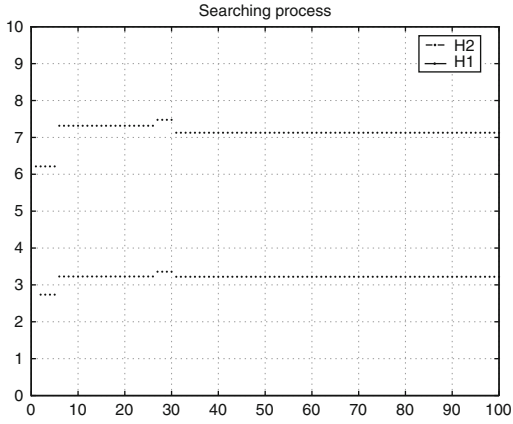


Fig. 3.5 Two objective values by Ra-Ra simulation-based PSO

3.4 Ra-Ra CCM

The chance measure of Ra-Ra variables is an extension of the chance of random variables introduced by Charnes and Cooper [45] and the purpose of introducing quantitative measures of the chance of a Ra-Ra event is manifold. First of all, it allows us to compare the chances of occurrence of two Ra-Ra events. Moreover, it is applicable to the further mathematical analysis. In the following part, we use CCM to substitute the chance constrained model.

3.4.1 General Model for Ra-Ra CCM

Assume that $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots \xi_n)$ is a Ra-Ra vector defined on the probability space $(\Omega, \mathscr{A}, Pr)$, and $g_r : \mathbf{R}^n \to \mathbf{R}$ are Borel measurable functions for $r = 1, 2, \ldots, p$. By Ra-Ra event we mean the system of inequalities

$$g_r(\boldsymbol{\xi}) \leq 0, r = 1, 2, \ldots, p \tag{3.75}$$

In the literature, the first attempt to develop the definition the chance of a Ra-Ra event is primitive chance, which is a function from [0,1] to [0,1]. Let's recall the basic definition and property of the chance of Ra-Ra events.

Definition 3.17. (Peng and Liu [251]) Let ξ be a Ra-Ra variable, and B a Borel set of $\mathbf{R}$. Then the chance of Ra-Ra event $\xi \in B$ is a function from $(0, 1]$ to $[0, 1]$, defined as

$$Ch\{\xi \in B\}(\alpha) = \sup_{Pr\{A\} \geq \alpha} \inf_{\omega \in A} Pr\{\xi(\omega) \in B\} \tag{3.76}$$

where α is a prescribed probability level. The value of primitive chance at α is called α-chance.

We all note that the symbol Pr appears twice in the right side of equality (3.76). In fact, they represent different meanings. In other words, the overloading allows us to use the same symbol Pr for different probability measures, because we can deduce the exact meaning in the context.

Based on Definition 3.17, a natural idea is to provide a confidence level α at which it is desired that the stochastic constrains hold. Let's still consider the following model,

$$\begin{cases} \max f(\boldsymbol{x}, \boldsymbol{\xi}) \\ \text{s.t.} \begin{cases} g_r(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, r = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{3.77}$$

where $f(\boldsymbol{x}, \boldsymbol{\xi})$ and $g_r(\boldsymbol{x}, \boldsymbol{\xi}), r = 1, 2 \ldots, p$ are continuous functions in X and $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_n)$ is a Ra-Ra vector on the probability space $(\Omega, \mathscr{A}, Pr)$. Based on the chance-constraint operator, the maximax Ra-Ra chance-constrained programming model (CCM) was proposed:

$$\begin{cases} \max \bar{f} \\ \text{s.t.} \begin{cases} Ch\{f(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}\}(\alpha) \geq \beta \\ Ch\{g_r(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\}(\eta_r) \geq \theta_r, r = 1, 2, \ldots, p \\ x \in X \end{cases} \end{cases} \tag{3.78}$$

where $\alpha, \beta, \eta_r, \theta_r$ are the predetermined confidence levels.

Definition 3.18. A solution $\mathbf{x} \in X$ is said to be the feasible solution of problem (3.78) if and only if $Ch\{f(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}\} \geq \beta$ and $Ch\{g_r(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \geq \alpha_r$ for all r. For any feasible $\boldsymbol{x}$, if there is a solution $\boldsymbol{x}^*$ such $\bar{f}^* > \bar{f}$, then $\boldsymbol{x}^*$ is called the optimal solution.

If the objective is to be minimized (for example, the objective is a cost function), the CCP model should be as follows,

$$\begin{cases} \min \bar{f} \\ \text{s.t.} \begin{cases} Ch\{f(\boldsymbol{x}, \boldsymbol{\xi}) \leq \bar{f}\}(\alpha) \geq \beta \\ Ch\{g_r(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\}(\eta_r) \geq \theta_r, r = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{3.79}$$

where $\alpha, \beta, \eta_r, \theta_r$ are the predetermined confidence levels. Similarly, we have the following definition,

Definition 3.19. A solution $\boldsymbol{x} \in X$ is said to be the feasible solution of problem (3.79) if and only if $Ch\{f(\boldsymbol{x}, \boldsymbol{\xi}) \leq \bar{f}\} \geq \beta$ and $Ch\{g_r(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \geq \alpha_r$ for all r. For any feasible $\mathbf{x}$, if there is a solution $\boldsymbol{x}^*$ such $\bar{f}^* < \bar{f}$, then $\boldsymbol{x}^*$ is called the optimal solution.

According to the above definition, we know that

$$Ch\{g_r(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\}(\eta_r) \geq \theta_r \Leftrightarrow Pr\{\omega | Pr\{g_r(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \geq \theta_r\} \geq \eta_r, r = 1, 2, \ldots, p.$$

For given confidence levels η_r, θ_r, using the primitive chance measure we have the chance constraints as follows,

$$Pr\{\omega | Pr\{g_r(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \geq \theta_r\} \geq \eta_r, r = 1, 2, \ldots, p.$$

Thus a point $\boldsymbol{x}(\geq 0)$ is called feasible for problem (3.79) if and only if the probability measures of the random events $\{\omega | Pr\{g_r(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \geq \theta_r\}$ are at least $\eta_r, r = 1, 2, \ldots, p$. Since

$$Ch\{f(\boldsymbol{x}, \boldsymbol{\xi}) \leq \bar{f}\}(\alpha) \geq \beta \Leftrightarrow Pr\{\omega | Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}) \leq \bar{f}\} \geq \beta\} \geq \alpha,$$

then problem (3.79) can be rewritten as

$$\begin{cases} \min \bar{f} \\ \text{s.t.} \begin{cases} Pr\{\omega | Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}) \leq \bar{f}\} \geq \beta\} \geq \alpha \\ Pr\{\omega | Pr\{g_r(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \geq \theta_r\} \geq \eta_r, r = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{3.80}$$

where $\alpha, \beta, \eta_r, \theta_r$ are the predetermined confidence levels, $r = 1, 2, \ldots, p$. Similarly, if decision makers want to maximize the objective value, then problem (3.78) can be rewritten as

$$\begin{cases} \max \bar{f} \\ \text{s.t.} \begin{cases} Pr\{\omega | Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}\} \geq \beta\} \geq \alpha \\ Pr\{\omega | Pr\{g_r(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \geq \theta_r\} \geq \eta_r, r = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{3.81}$$

where $\alpha, \beta, \eta_r, \theta_r$ are the predetermined confidence levels, $r = 1, 2, \ldots, p$.

Definition 3.20. A solution $\mathbf{x} \in X$ is said to be the feasible solution of problem (3.79) if and only if $Pr\{\omega | Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}) \leq \bar{f}\} \geq \beta\} \geq \alpha$ and $Pr\{\omega | Pr\{g_r(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \geq \theta_r\} \geq \eta_r$ for all r. For any feasible $\boldsymbol{x}$, if there is a solution $\boldsymbol{x}^*$ such that $\bar{f}^* < \bar{f}$, then $\boldsymbol{x}^*$ is called the optimal solution.

Remark 3.2. If the Ra-Ra vector $\boldsymbol{\xi}$ degenerates to a random vector $\bar{\boldsymbol{\xi}}$, then $Pr\{f(\boldsymbol{x}, \bar{\boldsymbol{\xi}}) \leq \bar{f}\} \geq \beta$ is a random event. For fixed $\omega \in \Omega$, $Pr\{f(\boldsymbol{x}, \bar{\boldsymbol{\xi}}) \leq \bar{f}\} \geq \beta$ implies $f(\boldsymbol{x}, \bar{\boldsymbol{\xi}}) \leq \bar{f}$. Then,

$$Pr\{\omega | Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}) \leq \bar{f}\} \geq \beta\} \geq \alpha$$

is equivalent to $Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}) \leq \bar{f}\} \geq \alpha$. Similarly, the constraint

$$Pr\{\omega | Pr\{g_r(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \geq \theta_r\} \geq \eta_r$$

is equivalent to $Pr\{g_r(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \geq \eta_r$, $r = 1, 2, \ldots, p$. Then, problem (3.80) is equivalent to

$$\begin{cases} \min \bar{f} \\ \text{s.t.} \begin{cases} Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}) \leq \bar{f}\} \geq \alpha \\ Pr\{g_r(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \geq \eta_r, r = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{3.82}$$

which is identical with the chance constrained programming model in the stochastic environment.

Consider the following multiobjective programming problem with Ra-Ra coefficients,

$$\begin{cases} \max[f_1(\boldsymbol{x}, \boldsymbol{\xi}), f_2(\boldsymbol{x}, \boldsymbol{\xi}), \ldots, f_m(\boldsymbol{x}, \boldsymbol{\xi})] \\ \text{s.t.} \begin{cases} g_r(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, r = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{3.83}$$

where $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T$ is an n-dimensional decision vector, $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_n)$ is a Ra-Ra vector, $f_i(\boldsymbol{x}, \boldsymbol{\xi})$ are objective functions, $i = 1, 2, \ldots, m$, $g_r(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0$ are Ra-Ra constraints, $r = 1, 2, \ldots, p$. For a fixed decision vector $\boldsymbol{x}$, it is meaningless to maximize the objectives $f_i(\boldsymbol{x}, \boldsymbol{\xi})$, $i = 1, 2, \ldots, m$. Before

we know the exact value of the Ra-Ra vector $\boldsymbol{\xi}$, just as we can not maximize a random function in stochastic programming. Also, we can not judge whether or not a decision $\boldsymbol{x}$ is feasible before we know the value of $\boldsymbol{\xi}$. Hence, both the objectives and constraints in problem (3.83) are ill-defined. For presenting a mathematically meaningful Ra-Ra programming, we build a new class of Ra-Ra programming to model Ra-Ra decision problems via chance measure which was proposed above. We present the chance-constrained multiobjective programming as follows,

$$\begin{cases} \max \{f_1, f_2, \ldots, f_m\} \\ \text{s.t.} \begin{cases} Ch\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) \geq f_i\}(\alpha_i) \geq \beta_i, i = 1, 2, \ldots, m \\ Ch\{g_r(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\}(\eta_r) \geq \theta_r, r = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{3.84}$$

where α_i, β_i, η_r and θ_r are predetermined confidence levels, $i = 1, 2, \ldots, m$, $r = 1, 2, \ldots, p$. By Definition 3.19, problem (3.84) can be rewritten as

$$\begin{cases} \max [f_1, f_2, \ldots, f_m] \\ \text{s.t.} \begin{cases} Pr\{\omega \in \Omega | Pr\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) \geq f_i\} \geq \beta_i\} \geq \alpha_i, i = 1, 2, \ldots, m \\ Pr\{\omega \in \Omega | Pr\{g_r(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \geq \theta_r\} \geq \eta_r, r = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{3.85}$$

where α_i, β_i, η_r and θ_r are predetermined confidence levels, $i = 1, 2, \ldots, m$, $r = 1, 2, \ldots, p$. If the objectives is to minimize the cost, then problem (3.85) should be formulated as follows,

$$\begin{cases} \min [f_1, f_2, \ldots, f_m] \\ \text{s.t.} \begin{cases} Pr\{\omega \in \Omega | Pr\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) \leq f_i\} \geq \beta_i\} \geq \alpha_i, i = 1, 2, \ldots, m \\ Pr\{\omega \in \Omega | Pr\{g_r(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \geq \theta_r\} \geq \eta_r, r = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{3.86}$$

Definition 3.21. Suppose a feasible solution $\boldsymbol{x}^*$ of problem (3.85) satisfies

$$Pr\{\omega | Pr\{f_i(\boldsymbol{x}^*, \boldsymbol{\xi}) \geq f_i(\boldsymbol{x}^*)\} \geq \beta_i\} \geq \alpha_i, i = 1, 2, \ldots, m,$$

where confidence levels $\alpha_i, \beta_i \in [0, 1]$. $\boldsymbol{x}^*$ is a Ra-Ra efficient solution to problem (3.85) if and only if there exists no other feasible solution $\boldsymbol{x}$ such that

$$Pr\{\omega | Pr\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) \geq f_i(\boldsymbol{x})\} \geq \beta_i\} \geq \alpha_i, i = 1, 2, \ldots, m,$$

$f_i(\boldsymbol{x}) \geq f_i(\boldsymbol{x}^*)$ for all i and $f_{i_0}(\boldsymbol{x}) > f_{i_0}(\boldsymbol{x}^*)$ for at least one $i_0 \in \{1, 2, \ldots, m\}$.

Sometimes, we may formulate a random decision system as a chance-constrained goal model (CCGM) proposed according to the priority structure and target levels set by the decision-maker,

$$\begin{cases} \min \sum_{j=1}^{l} P_j \sum_{i=1}^{m} (u_{ij} d_i^+ + v_{ij} d_i^-) \\ \text{s.t.} \begin{cases} Pr\{\omega | Pr\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) - b_i \le d_i^+\} \ge \beta_i^+\} \ge \alpha_i^+, i = 1, 2, \ldots, m \\ Pr\{\omega | Pr\{b_i - f_i(\boldsymbol{x}, \boldsymbol{\xi}) \le d_i^-\} \ge \beta_i^-\} \ge \alpha_i^-, i = 1, 2, \ldots, m \\ Pr\{\omega | Pr\{g_r(\boldsymbol{x}, \boldsymbol{\xi}) \le 0\} \ge \theta_r\} \ge \eta_r, r = 1, 2, \ldots, p \\ d_i^-, d_i^- \ge 0, i = 1, 2, \ldots, m \end{cases} \end{cases} \tag{3.87}$$

where P_j is the preemptive priority factor which express the relative importance of various goals, $P_j >> P_{j+1}$, for all j, u_{ij} is the weighting factor corresponding to positive deviation for goal i with priority j assigned, v_{ij} is the weighting factor corresponding to negative deviation for goal i with priority j assigned, d_i^+ is the α_i^+, β_i^+-optimistic positive deviation from the target of goal i, defined as

$$\min\{d \vee 0 | Pr\{\omega | Pr\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) - b_i \le d_i^+\} \ge \beta_i^+\} \ge \alpha_i^+\} \tag{3.88}$$

d_i^- is the α_i^-, β_i^--optimistic positive deviation from the target of goal i, defined as

$$\min\{d \vee 0 | Pr\{\omega | Pr\{b_i - f_i(\boldsymbol{x}, \boldsymbol{\xi}) \le d_i^-\} \ge \beta_i^-\} \ge \alpha_i^-\} \tag{3.89}$$

f_i is a function in goal constraints, g_r is a function in system constraints, b_i is the target value according to goal i, l is the number of priorities, m is the number of goal constraints, and p is the number of system constraints.

Remark 3.3. If the Ra-Ra vector ξ degenerates to the random variable, then the two random events $Pr\{\omega | Pr\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) - b_i \le d_i^+\} \ge \beta_i^+$ and $Pr\{\omega | Pr\{b_i - f_i(\boldsymbol{x}, \boldsymbol{\xi}) \le d_i^-\} \ge \beta_i^-$ should be always 1 for any $\omega \in \Omega$ provided that $\beta_i^+, \beta_i^- > 0$, then

$$Pr\{\omega | Pry\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) - b_i \le d_i^+\} \ge \beta_i^+\} \ge \alpha_i^+$$

is equivalent to $Pr\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) - b_i \le d_i^+\} \ge \alpha_i^+$, and

$$Pr\{\omega | Pr\{b_i - f_i(\boldsymbol{x}, \boldsymbol{\xi}) \le d_i^-\} \ge \beta_i^-\} \ge \alpha_i^-$$

is equivalent to $Pr\{b_i - f_i(\boldsymbol{x}, \boldsymbol{\xi}) \le d_i^-\} \ge \alpha_i^-$. Similarly, the constraint

$$Pr\{\omega | Pr\{g_r(\boldsymbol{x}, \boldsymbol{\xi}) \le 0\} \ge \theta_r\} \ge \eta_r$$

is equivalent to $Pr\{g_r(\boldsymbol{x}, \boldsymbol{\xi}) \le 0\} \ge \eta_r$, then problem (3.87) is rewritten as

$$\begin{cases} \min \sum_{j=1}^{l} P_j \sum_{i=1}^{m} (u_{ij} d_i^+ + v_{ij} d_i^-) \\ \text{s.t.} \begin{cases} Pr\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) - b_i \le d_i^+\} \ge \alpha_i^+, i = 1, 2, \ldots, m \\ Pr\{b_i - f_i(\boldsymbol{x}, \boldsymbol{\xi}) \le d_i^-\} \ge \alpha_i^-, i = 1, 2, \ldots, m \\ Pr\{g_r(\boldsymbol{x}, \boldsymbol{\xi}) \le 0\} \ge \eta_r, r = 1, 2, \ldots, p \\ d_i^-, d_i^- \ge 0, i = 1, 2, \ldots, m \end{cases} \end{cases} \tag{3.90}$$

This is identical with the goal programming in the random environment.

3.4.2 Linear Ra-Ra CCM and the Surrogate Worth Trade-Off Method

In this section, we concentrate on the multiobjective linear programming problem with Ra-Ra coefficients

$$\begin{cases} \max \left[\tilde{\tilde{c}}_1^T \boldsymbol{x}, \tilde{\tilde{c}}_2^T \boldsymbol{x}, \ldots, \tilde{\tilde{c}}_m^T \boldsymbol{x}\right] \\ \text{s.t.} \begin{cases} \tilde{\tilde{e}}_r^T \boldsymbol{x} \le \tilde{\tilde{b}}_r, r = 1, 2, \ldots, p \\ \boldsymbol{x} \ge 0 \end{cases} \end{cases} \tag{3.91}$$

where $\tilde{\tilde{c}}_i = (\tilde{\tilde{c}}_{i1}, \tilde{\tilde{c}}_{i2}, \ldots, \tilde{\tilde{c}}_{in})^T$, $\tilde{\tilde{e}}_r = (\tilde{\tilde{e}}_{r1}, \tilde{\tilde{e}}_{r2}, \ldots, \tilde{\tilde{e}}_{rn})^T$ and $\tilde{\tilde{b}}_r$ are Ra-Ra vectors, $i = 1, 2, \ldots, m, r = 1, 2, \ldots, p$. Then by the definition of chance measure of Ra-Ra variables, we have the following chance-constrained model of (3.91),

$$\begin{cases} \max[\bar{f}_1, \bar{f}_2, \ldots, \bar{f}_m] \\ \text{s.t.} \begin{cases} Ch\{\tilde{\tilde{c}}_i^T \boldsymbol{x} \le \bar{f}_i\}\alpha_i \ge \beta_i, i = 1, 2, \ldots, m \\ Ch\{\tilde{\tilde{e}}_r^T \boldsymbol{x} \le \tilde{\tilde{b}}_r\}(\eta_r) \ge \theta_r, r = 1, 2, \ldots, p \\ \boldsymbol{x} \ge 0, \end{cases} \end{cases} \tag{3.92}$$

For given confidence levels α_i, β_i, η_r and θ_r, we have

$$\begin{aligned} &Ch\{\tilde{\tilde{c}}_i^T \boldsymbol{x} \ge \bar{f}_i\}(\alpha_i) \ge \beta_i \Longleftrightarrow Pr\{\omega | Pr\{\tilde{\tilde{c}}_i^T \boldsymbol{x} \ge \bar{f}_i\} \ge \beta_i\} \ge \alpha_i, \\ &Ch\{\tilde{\tilde{e}}_r^T \boldsymbol{x} \le \tilde{\tilde{b}}_r\}(\eta_r) \ge \theta_r \Longleftrightarrow Pr\{\omega | Pr\{\tilde{\tilde{e}}_r^T \boldsymbol{x} \le \tilde{\tilde{b}}_r\} \ge \theta_r\} \ge \eta_r, \end{aligned}$$

where $i = 1, 2, \ldots, m$ and $r = 1, 2, \ldots, p$. Then the programming problem (3.92) can be rewritten as

$$\begin{cases} \max[f_1, f_2, \ldots, f_m] \\ \text{s.t.} \begin{cases} Pr\{\omega | Pr\{\tilde{\tilde{c}}_i^T \boldsymbol{x} \ge f_i\} \ge \beta_i\} \ge \alpha_i, i = 1, 2, \ldots, m \\ Pr\{\omega | Pr\{\tilde{\tilde{e}}_r^T \boldsymbol{x} \le \tilde{\tilde{b}}_r\} \ge \theta_r\} \ge \eta_r, r = 1, 2, \ldots, p \\ \boldsymbol{x} \ge 0 \end{cases} \end{cases} \tag{3.93}$$

where α_i, β_i, η_r and θ_r are predetermined confidence levels, $i = 1, 2, \ldots, m, r = 1, 2, \ldots, p$.

3.4.2.1 Crisp Equivalent Model

One way of solving a pr–pr constrained multiobjective programming model is to convert the constraints of (3.93) into their respective crisp equivalents. However, this process is usually quite difficult in many cases. Next, we will consider a special case and present the result in this section.

Theorem 3.14. *(Xu and Ding [338]) Assume that Ra-Ra vector $\tilde{\bar{c}}_i$ is characterized by $\tilde{\bar{c}}_i \sim \mathcal{N}(\bar{c}_i(\omega), V_i^c)$, where $\bar{c}_i(\omega) = (\bar{c}_{i1}(\omega), \bar{c}_{i2}(\omega), \ldots, c_{in}(\omega))^T$ is a random vector and V_i^c is a positive definite covariance matrix. Then, we have $Pr\{\omega | Pr\{\tilde{\bar{c}}_i^T \boldsymbol{x} \geq f_i\} \geq \beta_i\} \geq \alpha_i$ if and only if*

$$f_i \leq R + \mu_i + \sigma_i \Phi^{-1}(1 - \alpha_i) \tag{3.94}$$

where $R = \Phi^{-1}(1 - \beta_i)\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}$ and Φ is the standardized normal distribution and $\beta_i, \alpha_i \in [0, 1]$ are predetermined confidence levels.

Proof. From the above assumption we know that $\bar{c}_{ij}(\omega)$ is a random variable and $\bar{c}_i(\omega) = (\bar{c}_{i1}(\omega), \bar{c}_{i2}(\omega), \ldots, \bar{c}_{in}(\omega))^T$ is the random vector. We set

$$\bar{c}_{ij}(\omega) \sim \mathcal{N}(\mu_{ij}, \sigma_{ij}^2)$$

Since $x_i \geq 0$ for any i, it follows that $x_j \bar{c}_{ij}(\omega) \sim \mathcal{N}(x_j \mu_{ij}, x_j^2 \sigma_{ij}^2)$,

$$\bar{c}_i(\omega)^T \boldsymbol{x} = \sum_{j=1}^{n} \bar{c}_{ij}(\omega) x_j \sim \mathcal{N}\left(\sum_{j=1}^{n} \mu_{ij} x_j, \sum_{j=1}^{n} \sigma_{ij}^2 x_j^2 \right)$$

So $\bar{c}_i(\omega)^T \boldsymbol{x}$ is also a random variable. Now we set

$$\mu_i = \sum_{j=1}^{n} \mu_{ij} x_j, \sigma_i^2 = \sum_{j=1}^{n} \sigma_{ij}^2 x_j^2.$$

then $\bar{c}_i(\omega)^T \boldsymbol{x} \sim \mathcal{N}(\mu_i, \sigma_i^2)$. In addition, $\tilde{\bar{c}}_i$ is a Ra-Ra vector which is distributed with mean vector $\bar{c}_i(\omega)$ and positive definite covariance matrix V_i^c, written as $\tilde{\bar{c}}_i \sim \mathcal{N}(\bar{c}_i(\omega), V_i^c)$. It follows that $\tilde{\bar{c}}_i^T \boldsymbol{x} \sim \mathcal{N}(\bar{c}_i(\omega)^T \boldsymbol{x}, \boldsymbol{x}^T V_i^c \boldsymbol{x})$. Then, we have that

$$\begin{aligned}
&Pr\{\tilde{\bar{c}}_{ij}^T \boldsymbol{x} \geq f_i\} \geq \beta_i \\
&\Leftrightarrow Pr\left\{ \frac{\tilde{\bar{c}}_i^T \boldsymbol{x} - \bar{c}_i(\omega)^T \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}} \geq \frac{f_i - \bar{c}_i(\omega)^T \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}} \right\} \geq \beta_i \\
&\Leftrightarrow \bar{c}_i(\omega)^T \boldsymbol{x} + \Phi^{-1}(1 - \beta_i)\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}} \geq f_i
\end{aligned}$$

It follows that, for given confidence levels $\beta_i, \alpha_i \in [0,1]$,

$$\begin{aligned}
&Pr\{\omega | Pr\{\tilde{\bar{c}}_{ij}^T \boldsymbol{x} \geq f_i\} \geq \beta_i\} \geq \alpha_i \\
&\Leftrightarrow Pr\left\{\omega | \bar{c}_i(\omega)^T \boldsymbol{x} + \Phi^{-1}(1-\beta_i)\sqrt{x^T V_i^c \boldsymbol{x}} \geq f_i\right\} \geq \alpha_i \\
&\Leftrightarrow Pr\left\{\frac{\bar{c}_i(\omega)^T \boldsymbol{x} - \mu_i}{\sigma_i} \geq \frac{f_i - R - \mu_i}{\sigma_i}\right\} \geq \alpha_i \\
&\Leftrightarrow f_i \leq R + \mu_i + \sigma_i \Phi^{-1}(1-\alpha_i)
\end{aligned}$$

where $R = \Phi^{-1}(1-\beta_i)\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}$. This completes the proof. □

Similarly, the constraints

$$Pr\{\omega | Pr\{\tilde{\bar{e}}_r^T \boldsymbol{x} \leq \tilde{\bar{b}}_r\} \geq \theta_r\} \geq \eta_r, \quad r = 1, 2, \ldots, p$$

can be converted into crisp equivalent constraints.

Theorem 3.15. *(Xu and Ding [338]) Suppose that Ra-Ra vector $\tilde{\bar{e}}_r$ are characterized by $\tilde{\bar{e}}_r \sim \mathcal{N}(\bar{e}_r(\omega), V_r^e)$, where $\bar{e}_r(\omega)$ are the mean vector and V_r^e are the positive definite covariance; Ra-Ra variable $\tilde{\bar{b}}_r$ are characterized by $\tilde{\bar{b}}_r \sim \mathcal{N}(\bar{b}_r(\omega), (\sigma_r^b)^2)$, where $\bar{b}_r(\omega)$ are random variables and $(\sigma_r^b)^2$ are the variances, $r = 1, 2, \ldots, p$. Assume that $\bar{e}_{rj}(\omega)$ and $\bar{b}_r(\omega)$ $(r = 1, 2, \ldots, p; j = 1, 2, \ldots, n)$ are random variables, then $\bar{e}_r(\omega)^T \boldsymbol{x} - \bar{b}_r$ is a random variable. Let μ_r be the mean and σ_r^2 be the variance, then we have that $Pr\{\omega | Pr\{\tilde{\bar{e}}_r^T \boldsymbol{x} \leq \tilde{\bar{b}}_r(\omega)\} \geq \theta_r\} \geq \eta_r$ if and only if*

$$\Phi^{-1}(\theta_r)\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2} + \mu_r + \sigma_r \Phi^{-1}(\eta_r) \leq 0 \tag{3.95}$$

Proof. From the assumption $\tilde{\bar{e}}_r \sim \mathcal{N}(\bar{e}_r(\omega), V_r^e)$, $\tilde{\bar{b}}_r \sim \mathcal{N}(\bar{b}_r(\omega), (\sigma_r^b)^2)$, it follows that $\tilde{\bar{e}}_r^T \boldsymbol{x} \sim \mathcal{N}(\bar{e}_r(\omega)^T \boldsymbol{x}$. Then, $\tilde{\bar{e}}_r^T \boldsymbol{x} - \tilde{\bar{b}}_r \sim \mathcal{N}(\bar{e}_r(\omega)^T \boldsymbol{x} - \bar{b}_r(\omega), \boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2)$, we have that

$$\begin{aligned}
&Pr\{\tilde{\bar{e}}_r(\omega)^T \boldsymbol{x} \leq \tilde{\bar{b}}_r(\omega)\} \geq \theta_r \\
&\Leftrightarrow Pr\left\{\frac{\tilde{\bar{e}}_r^T \boldsymbol{x} - \tilde{\bar{b}}_r - (\bar{e}_r(\omega)^T \boldsymbol{x} - \bar{b}_r(\omega))}{\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2}} \leq \frac{\bar{b}_r(\omega) - \bar{e}_r(\omega)^T \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2}}\right\} \geq \theta r \\
&\Leftrightarrow \bar{e}_r(\omega)^T \boldsymbol{x} - \bar{b}_r(\omega) \leq -\Phi^{-1}(\theta_r)\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2}
\end{aligned}$$

Since $\bar{e}_r(\omega)^T \boldsymbol{x} - \bar{b}_r(\omega) \sim \mathcal{N}(\mu_r, \sigma_r^2)$, for given confidence levels $\theta_r, \eta_r \in [0,1]$, we have that,

$$\begin{aligned}
&Pr\{\omega | Pr\{\tilde{\bar{e}}_r^T \boldsymbol{x} \le \tilde{\bar{b}}_r\} \ge \theta_r\} \ge \eta_r \\
&\Leftrightarrow Pr\{\omega | \bar{e}_r(\omega)^T \boldsymbol{x} - \bar{b}_r(\omega) \le M\} \ge \eta_r \\
&\Leftrightarrow Pr\left\{\omega | \frac{e_r(\omega)^T \boldsymbol{x} - b_r(\omega) - \mu_r}{\sigma_r} \le \frac{M - \mu_r}{\sigma_r}\right\} \ge \eta_r \\
&\Leftrightarrow M \ge \mu_r + \sigma_r \Phi^{-1}(\eta_r) \\
&\Leftrightarrow \Phi^{-1}(\theta_r)\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2} + \mu_r + \sigma_r \Phi^{-1}(\eta_r) \le 0
\end{aligned}$$

where $M = -\Phi^{-1}(\theta_r)\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2}$. This completes the proof. □

Here we consider the special cases of Ra-Ra programming problem formulated in Theorems 3.14 and 3.15 and discuss its convexity. From Theorems 3.14 and 3.15, we have that problem (3.93) is equivalent to the following multiobjective programming problem

$$\begin{cases} \max\ [f_1, f_2, \ldots, f_m] \\ \text{s.t.} \begin{cases} f_i \le R + \mu_i + \sigma_i \Phi^{-1}(1-\alpha_i), i = 1, 2, \ldots, m \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{3.96}$$

or equivalently

$$\begin{cases} \max\ [H_1(\boldsymbol{x}), H_2(\boldsymbol{x}), \ldots, H_m(\boldsymbol{x})] \\ \text{s.t.} \boldsymbol{x} \in X \end{cases} \tag{3.97}$$

where $R = \Phi^{-1}(1-\beta_i)\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}$, $H_i(\boldsymbol{x}) = R + \mu_i + \sigma_i \Phi^{-1}(1-\alpha_i)$, $i = 1, 2, \ldots, m$, and $X := \{\boldsymbol{x} \in \mathbf{R}^n | \Phi^{-1}(\theta_r)\sqrt{\mathbf{x}^T V_r^e \mathbf{x}} + (\sigma_r^b)^2 + \mu_r + \sigma_r \Phi^{-1}(\eta_r) \le 0,\ r = 1, 2, \ldots, p; \boldsymbol{x} \ge 0\}$.

Suppose $H(\boldsymbol{x}) = \sum_{i=1}^m \lambda_i H_i(\boldsymbol{x})$, where the weight coefficient $\lambda_i \in [0,1]$ expresses the importance of $H_i(\boldsymbol{x})$ to the decision-maker satisfying $\sum_{i=1}^m \lambda_i = 1$. Then problem (3.97) can be converted into the following problem

$$\begin{cases} \max H(\boldsymbol{x}) \\ \text{s.t.} \boldsymbol{x} \in X \end{cases} \tag{3.98}$$

Theorem 3.16. *(Xu and Ding [338]) The efficient solution of problem (3.97) is the optimal solution of problem (3.98) and the optimal solution of problem (3.98) is the efficient solution of problem (3.97).*

Proof. To be proven, the efficient solution of problem (3.97) is the optimal solution of problem (3.98). Let's consider if the optimal solution of problem (3.98) is the efficient solution of problem (3.97).

Suppose $\boldsymbol{x}^*$ is an optimal solution of problem (3.98). If $\boldsymbol{x}^*$ is not an efficient solution of problem (3.97), then there exists $\boldsymbol{x}^0$ such that $H_i(\boldsymbol{x}^0) \geq H_i(\boldsymbol{x}^*)(i = 1, 2, \ldots, m)$, and there at least exists a k such that $H_k(\boldsymbol{x}^0) > H_k(\boldsymbol{x}^*)$. Then

$$\sum_{i=1}^{m} \lambda_i H_i(\boldsymbol{x}^0) > \sum_{i=1}^{m} \lambda_i H_i(\boldsymbol{x}^*) \tag{3.99}$$

That is

$$H(\boldsymbol{x}_0) > H(\boldsymbol{x}^*) \tag{3.100}$$

This conflicts that $\boldsymbol{x}^*$ is an optimal solution of problem (3.98). This completes the proof. □

From the above theorem, we know that problem (3.98) is equivalent to problem (3.97). To know about the existence of solution to (3.97), here we discuss the convexity of problem (3.98).

Theorem 3.17. *(Xu and Ding [338]) Let* $H(\boldsymbol{x}) = \sum_{i=1}^{m} \lambda_i H_i(\boldsymbol{x})$, $\lambda_i \in [0, 1]$, $H_i(\boldsymbol{x}) = \Phi^{-1}(1 - \delta_i)\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}} + \mu_i + \sigma_i \Phi^{-1}(1 - \gamma_i)$, *and* $X = \{\boldsymbol{x} \in \mathbf{R}^n | Pr\{\omega | Pr\{\tilde{\tilde{e}}_r^T \boldsymbol{x} \leq \tilde{\tilde{b}}_r\} \geq \theta_r\} \geq \eta_r, r = 1, 2, \ldots, p; \boldsymbol{x} \geq 0\}$. *If* $\gamma_i \geq 0.5$, $\delta_i \geq 0.5$, $\theta_i \geq 0.5$ *and* $\eta_i \geq 0.5$, *problem (3.98) is convex.*

Proof. According to [301], $\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}$ is a convex function. In addition, since $\gamma_i \geq 0.5$ and $\delta_i \geq 0.5$, it follows that $\Phi^{-1}(1 - \gamma_i) \leq 0$ and $\Phi^{-1}(1 - \delta_i) \leq 0$, we know $H_i(\boldsymbol{x})$ are concave functions, $i = 1, 2, \ldots, m$. So let $\boldsymbol{x}^1$ and $\boldsymbol{x}^2$ be any two points in feasible set, and $\alpha \in [0, 1]$, we have that

$$H_i[\alpha \boldsymbol{x}^1 + (1 - \alpha) x \boldsymbol{x}^2] \geq \alpha H_i(\boldsymbol{x}^1) + (1 - \alpha) H_i(\boldsymbol{x}^2)$$

for $\lambda_i \in [0, 1]$,

$$\lambda_i H_i[\alpha \boldsymbol{x}^1 + (1 - \alpha)\boldsymbol{x}^2] \geq \alpha \lambda_i H_i(\boldsymbol{x}^1) + (1 - \alpha)\lambda_i H_i(\boldsymbol{x}^2)$$

moreover,

$$\sum_{i=1}^{m} \lambda_i H_i[\alpha x \boldsymbol{x}^1 + (1 - \alpha)\boldsymbol{x}^2] \geq \alpha \sum_{i=1}^{m} \lambda_i H_i(\boldsymbol{x}^1) + (1 - \alpha) \sum_{i=1}^{m} \lambda_i H_i(\boldsymbol{x}^2)$$

that is

$$H[\alpha \boldsymbol{x}^1 + (1 - \alpha)\boldsymbol{x}^2] \geq \alpha H(\boldsymbol{x}^1) + (1 - \alpha) H(\boldsymbol{x}^2)$$

So the objective function $H(\boldsymbol{x})$ is concave. Next, we prove that X is convex. From Theorem 3.15, we know that

$$\begin{aligned} &Pr\{\omega | Pr\{\tilde{\tilde{e}}_r^T \boldsymbol{x} \leq \tilde{\tilde{b}}_r\} \geq \theta_r\} \geq \eta_r \\ &\Leftrightarrow \Phi^{-1}(\theta_r)\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2} + \mu_r + \sigma_r \Phi^{-1}(\eta_r) \leq 0 \end{aligned}$$

Let $g_r(\boldsymbol{x}) = \Phi^{-1}(\theta_r)\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2} + \mu_r + \sigma_r \Phi^{-1}(\eta_r)$. Then $X = \{\boldsymbol{x} \in \mathbf{R}^n | g_r(\boldsymbol{x}) \leq 0, r = 1, 2, \ldots, p; \boldsymbol{x} \geq 0\}$. Since $\theta_r \geq 0.5$ and $\eta_r \geq 0.5$, it follows that $\Phi^{-1}(\theta_r) \geq 0$ and $\Phi^{-1}(\eta_r) \geq 0$, and $g_r(\boldsymbol{x})$ are convex functions, $r = 1, 2, \ldots, p$. Let $\boldsymbol{x}^1$ and $\boldsymbol{x}^2$ be two feasible solutions, then

$$g_r(\boldsymbol{x}^1) \leq 0, g_r(\boldsymbol{x}^2) \leq 0 \tag{3.101}$$

according to g_r's convexity, we have

$$g_r[\alpha \boldsymbol{x}^1 + (1-\alpha)\boldsymbol{x}^2] \leq \alpha g_r(\boldsymbol{x}^1) + (1-\alpha) g_r(\boldsymbol{x}^2) \leq 0 \tag{3.102}$$

where $0 \leq \alpha \leq 1$, $r = 1, 2, \ldots, p$. This means that $\alpha \boldsymbol{x}^{12} + (1-\alpha)\boldsymbol{x}^2$ is also a feasible solution. So X is a convex set. Above all, we can conclude that problem $\max_{x \in X} H(\boldsymbol{x})$ is a convex programming. □

3.4.2.2 The Surrogate Worth Trade-Off Method

The surrogate worth trade-off method, which is called SWT method for short, was proposed by Haimes et al. [124] in 1974 to solve the multi-objective programming problem. It can be applied to continuous variables, objective functions and constraints which can be differentiated twice.

In its original version, SWT is, in principle, noninteractive and assumes continuous variables and twice differentiable objective functions and constraints. It consists of four steps: (1) generate a representative subset of efficient solutions, (2) obtain relevant trade-off information for each generated solution, (3) interact with DM t obtain information about preference expressed in terms of worth, and (4) retrieve the best-compromise solution from the information obtained.

Take the problem (3.97) as an example and list the detailed steps according to the book [44] as follows:

Step 1: *Generation of a Representative Subset of Efficient Solutions*. The ε-constraint method is recommended to obtain the representative subset of efficient solutions. Without loss of generality, we choose a reference objective H_1 and formulate the ε-constraint problem:

$$\begin{cases} \max H_1(\boldsymbol{x}) \\ \text{s.t.} \begin{cases} H_i(\boldsymbol{x}) \geq \varepsilon_i, i = 2, 3, \ldots, m \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{3.103}$$

Although there is no rule to specify which objective should be chosen as a reference, the most important objective is recommended. To guarantee that the ε-constraint problem has feasible solution, a reasonable ε_i should be selected, usually, in the range $[a_i, b_i]$, where $a_i = \min_{x \in X} H_i(\boldsymbol{x})$ and $b_i = \max_{x \in X} H_i(\boldsymbol{x})$.

Step 2: *Obtaining Trade-off Information*. In the process of solving the problem (3.103), the trade-off information can easily be obtained merely by observing the optimal Kuhn-Tucker multipliers corresponding to the ε-constraints. Let these multipliers be denoted by $\lambda_{1i}(\boldsymbol{x}(\varepsilon))$. If $\lambda_{1k}(\boldsymbol{x}(\varepsilon)) > 0 (k = 1, 2, \ldots, m)$, then the efficient surface in the objective function space around the neighborhood of $\boldsymbol{H}^{\varepsilon} = (H_1(\boldsymbol{x}(\varepsilon)), H_2(\boldsymbol{x}(\varepsilon)), \ldots, H_m(\boldsymbol{x}(\varepsilon)))^T$ can be represented by $\boldsymbol{H}_1 = (H_1, H_2, \ldots, H_m)$ and

$$\lambda_{1k}(\boldsymbol{x}(\varepsilon)) = -\left.\frac{\partial \boldsymbol{H}_1}{\partial H_k}\right|_{\boldsymbol{H}} = \boldsymbol{H}^{\varepsilon},\ k = 2, 3, \ldots, m \tag{3.104}$$

Thus each $\lambda_{1k}(\boldsymbol{x}(\varepsilon))$ represents the efficient partial trade-off rate between H_1 and H_k at $\boldsymbol{H}^{\varepsilon}$ when all other objective are held fixed at their respective values at $\boldsymbol{x}(\varepsilon)$. The adjective "efficient" is used to signify that after the trade-off is made the resulting point remains on the efficient surface. The detail can be referred to [44].

Step 3: *Interacting with the Decision Maker to Elicit Preference*. DM is supplied with trade-off information from Step 2 and the levels of all criteria. He then expresses his ordinal preference on whether or not (and by how much) he would like to make such a trade at that level. Haimes et al. [44] constructed the following surrogate worth function: DM is asked "How much would you like to improve H_1 by $\lambda_{1k}(\boldsymbol{x}(\varepsilon))$ units per one-unit degradation of H_k while all other objective remain fixed at $H_l(\boldsymbol{x}(\varepsilon))$, $l \neq 1, k$?" Indicate your preference on a scale of -10 to 10, where the values have the following meaning:

1. $+10$ means you have the greatest desire to improve H_1 by $\lambda_{1k}(\boldsymbol{x}(\varepsilon))$ units per one-unit degradation of H_k.
2. 0 means you are indifferent about the trade.
3. -10 means you have the greatest desire to degrade improve H_1 by $\lambda_{1k}(\boldsymbol{x}(\varepsilon))$ units per one-unit improvement in H_k.

Values between -10 and 0, and 0 and 10 show proportional desire to make the trade.

DM's response is recorded as $w_{1k}(\boldsymbol{x}(\varepsilon))$, called the *surrogate worth* of the trade-off between H_1 and H_k at the efficient solution $\boldsymbol{x}(\varepsilon)$. At a particular efficient solution, there will be $m - 1$ questions to obtain $w_{1k}(\boldsymbol{x}(\varepsilon)), k = 2, 3, \ldots, m$.

Step 4: *Retrieving the Best-Compromise Solution*. If there exists an efficient solution $\boldsymbol{x}(\varepsilon_0)$ such that

$$w_{1k}(\boldsymbol{x}(\varepsilon_0)) = 0,\ k = 2, 3, \ldots, m \tag{3.105}$$

the DM has obtained a best-compromise solution. Thus (3.105) is the best-compromise condition of $\boldsymbol{x}(\varepsilon_0)$. If there is such $\boldsymbol{x}(\varepsilon_0)$ in the representative set,

then stop and output $\boldsymbol{x}(\varepsilon_0)$. Otherwise we use multiple regression to construct the surrogate worth function as follows,

$$\boldsymbol{w}_{1k} = w_{1k}(H_1, H_2, \ldots, H_m),\ k = 2, 3, \ldots, m.$$

Then the system of equations

$$w_{1k}(H_1, H_2, \ldots, H_m) = 0,\ k = 2, 3, \ldots, m,$$

is solved to determine $(H_2^*, \ldots, H_m^*)$. Let $\varepsilon_{0k} = H_k^*(k = 2, \ldots, m)$, $\varepsilon_0 = (\varepsilon_{02}, \ldots, \varepsilon_{0m})^T$. The best-compromise solution $\boldsymbol{x}(\varepsilon_0)$ is then found by solving the problem (3.103).

3.4.3 Nonlinear Ra-Ra CCM and Ra-Ra Simulation-Based APSO

Let's introduce the process of the particle swarm optimization based on the Ra-Ra simulation to deal with the chance constraint models with Ra-Ra parameters. Consider the following multiobjective programming problem with Ra-Ra coefficients,

$$\begin{cases} \max[f_1, f_2, \ldots, f_m] \\ \text{s.t.} \begin{cases} Pr\{\omega | Pr\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) \geq f_i\} \geq \beta_i\} \geq \alpha_i, i = 1, 2, \ldots, m \\ Pr\{\omega | Pr\{g_r(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \geq \theta_r\} \geq \eta_r, r = 1, 2, \ldots, p \\ \boldsymbol{x} \geq 0 \end{cases} \end{cases}$$

where α_i, β_i, η_r and θ_r are predetermined confidence levels, $i = 1, \ldots, m$, $r = 1, \ldots, p$; $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T$ is an n-dimensional decision vector; $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_n)$ is a Ra-Ra vector; $f_i(\boldsymbol{x}, \boldsymbol{\xi})$ or $g_r(\boldsymbol{x}, \boldsymbol{\xi})$ or both them are nonlinear objective functions, $i = 1, 2, \ldots, m$, $r = 1, 2, \ldots, p$.

3.4.3.1 Ra-Ra Simulation for CCM

Let $\boldsymbol{\xi}$ be an n-dimensional Ra-Ra vector on the probability space $(\Omega, \mathscr{A}, Pr)$, and $f : \mathbf{R}^n \to \mathbf{R}$ be a measurable function. For any given confidence levels α and β, we find the maximal value $\bar{f}$ such that

$$Ch\{f(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}\}(\alpha) \geq \beta$$

holds. That is, we should compute the maximal value $\bar{f}$ such that

$$Pr\{\omega \in \Omega | Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}(\omega)) \geq \bar{f}\} \geq \beta\} \geq \alpha$$

holds. We sample $\omega_1, \omega_2, \ldots, \omega_N$ from Ω according to the probability measure Pr, where ω_k is an n-dimensional vector, and estimate $\bar{f}_k = \sup\{f_k | Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}(\omega_k))\} \geq \beta\}$ for $k = 1, 2, \ldots, N$ by the stochastic simulation. Let N' be the integer part of αN. Then $\bar{f}$ can be taken as the N'th largest element in the sequence $\{\bar{f}_1, \bar{f}_2, \ldots, \bar{f}_N\}$.

Then the procedure simulating the critical value $\bar{f}$ of the function $Pr\{\omega \in \Omega | Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}(\omega)) \geq \bar{f}\} \geq \beta\} \geq \alpha$ can be summarized as follows:

Procedure `Ra-Ra simulation for CCM`

Input: The decision vector $\boldsymbol{x}$
Output: The critical value $\bar{f}$
Step 1. Generate $\omega_1, \omega_2, \ldots, \omega_N$ from Ω according to the probability measure Pr;
Step 2. Find $\bar{f}_k = \sup\{f_k | Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}(\omega_k)) \geq f_k\} \geq \beta\}$ for $k = 1, 2, \ldots, N$ by random simulation;
Step 3. Set N' as the integer part of αN;
Step 4. Return the N'th largest element in $\{\bar{f}_1, \bar{f}_2, \ldots, \bar{f}_N\}$.

Example 3.18. Find the maximal value $\bar{f}$ such that $Ch\left\{\sqrt{\tilde{\tilde{\xi}}_1^2 + \tilde{\tilde{\xi}}_2^2 + \tilde{\tilde{\xi}}_3^2} \geq \bar{f}\right\}(0.9) \geq 0.9$, where $\tilde{\tilde{\xi}}_1$, $\tilde{\tilde{\xi}}_2$ and $\tilde{\tilde{\xi}}_3$ are Ra-Ra variables defined as

$$\begin{aligned}
\tilde{\tilde{\xi}}_1 &= \mathscr{U}(\tilde{\rho}_1, \tilde{\rho}_1 + 2), \text{ with } \tilde{\rho}_1 \sim \mathscr{N}(0, 1), \\
\tilde{\tilde{\xi}}_2 &= \mathscr{N}(\tilde{\rho}_2, 1), \qquad \text{with } \tilde{\rho}_2 \sim \mathscr{U}(3, 5), \\
\tilde{\tilde{\xi}}_3 &= exp(\tilde{\rho}_3), \qquad \text{with } \tilde{\rho}_3 \sim \mathscr{U}(1, 2).
\end{aligned}$$

A run of Ra-Ra simulation with 1000 cycles shows that $\bar{f} = 3.3623$.

3.4.3.2 Adaptive Particle Swarm Optimization

Since the particle swarm optimization algorithm (PSO) is proposed by Kennedy and Eberhard [158], it has been one of the newest techniques within the family of evolutionary optimization algorithms for optimizing continuous nonlinear functions. The algorithm is based on an analogy with the choreography of flight of a flock of birds. The performance of particle swarm optimization using an inertia weight in comparison with the performance using a constriction factor is also explained. Developments and resources in the particle swarm algorithm are reviewed in [191, 196]. Although PSO has shown some important advances by providing high speed of convergence in specific problems, it does exhibit some shortages. It found that PSO has a poor ability to search at a fine grain because it lacks velocity control mechanism [7]. Some improvements in the PSO algorithm are proposed in [134]. Many approaches are attempted to improve the performance of PSO by variable inertia weight. The inertia weight is critical for the performance of PSO, which balances global exploration and local exploitation abilities of the swarm. A big inertia weight

facilitates exploration, but it makes the particle long time to converge. Similar to other evolutionary algorithms, the particle swarm optimization method conducts its search using a population of particles, corresponding to individuals. Each particle in the swarm represents a candidate solution to the problem. It starts with a random initialization of a population of individuals in the search space and works on the social behavior of the particles in the swarm, like birds flocking, fish schooling and the swarm theory. Therefore, it finds the global optimum by simply adjusting the trajectory of each individual towards its own best location and towards the best particle of the swarm at each generation of evolution. However, the trajectory of each individual in the search space is adjusted by dynamically altering the velocity of each particle, according to the flying experience of its own and the other particles in the search space. This population based robust algorithm always ensures convergence to the global optimum solution as compared to a GA.

Many scholars [222, 245, 329, 353, 354] proposed the adaptive particle swarm optimization algorithm to dynamically adjust the velocity of each particle to accelerate the convergence. This section mainly introduces the APSO algorithm proposed by Panigrahi, Pandi and Das [245] and interested readers could refer to the related literatures. In the simple PSO method, the inertia weight is made constant for all the particles in a single generation, but the most important parameter that moves the current position towards the optimum position is the inertia weight ω. In order to increase the search ability, the algorithm should be redefined in the manner that the movement of the swarm should be controlled by the objective function. In the proposed adaptive PSO, the particle position is adjusted such that the highly fitted particle (best particle) moves slowly when compared to the lowly fitted particle. This can be achieved by selecting different ω values for each particle according to their rank, between $\omega_{\min}$ and $\omega_{\max}$ as in the following form:

$$\omega_i = \omega_{\min} + \frac{(\omega_{\max} - \omega_{\min}) \cdot \text{Rank}_i}{\text{Total population}} \tag{3.106}$$

Hence, it follows from (3.106) that the best particle takes the first rank, and the inertia weight for that particle is set to the minimum value while that for the lowest fitted particle takes the maximum inertia weight, which makes that particle move with a high velocity. The velocity of each particle is updated using (3.107), and if any updated velocity goes beyond Vmax, it is limited to Vmax using (3.108),

$$v_{ij}(t+1) = \omega_i v_{ij}(t) + c_1 r_1 (p_{ij}(t) - x_{ij}(t)) + c_2 r_2 (p_{gj}(t) - x_{gj}(t)) \tag{3.107}$$

$$v_{ij}(t+1) = \text{sign}(v_{ij}(t+1)) \cdot \min(v_{ij}(t+1), V_j\text{max}) \tag{3.108}$$

where $j = 1, 2, \ldots, d$ and $i = 1, 2, \ldots, n$. The new particle position is obtained by using (3.109), and if any particle position goes beyond the range specified, it is

adjusted to its boundary using (3.110),

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \tag{3.109}$$

$$\begin{aligned} x_{ij}(t+1) &= \min(x_{ij}(t), \text{range}_{j}\max, \\ x_{ij}(t+1) &= \max(x_{ij}(t), \text{range}_{j}\min) \end{aligned} \tag{3.110}$$

where $j = 1, 2, \ldots, d$ and $i = 1, 2, \ldots, n$. The concept of re-initialization is introduced to the proposed APSO algorithm after a specific number of generations if there is no improvement in the convergence of the algorithm. The population of the proposed APSO at the end of the above mentioned specific generation is re-initialized with new randomly generated individuals. The number of these new individuals is selected from the k least fit individuals of the original population, where 'k' is the percentage of the total population to be changed. This effect of population re-initialization is, in a sense, similar to the mutation operator in a GA [174]. This effect is favorable when the algorithm prematurely converges to a local optimum and further improvement is not noticeable. This re-initialization of population is performed after checking the changes in the 'Fbest' value in each and every specific number of generations. The procedure of APSO algorithm can be summarized as follows:

Procedure `The procedure of APSO`

Step 1. Get the input parameters like range [min max] for each of the variables, c_1, c_2, iteration counter $= 0$, $V\max$, $\omega\max$ and $\omega_{\min}$;
Step 2. Initialize n number of population of particles of dimension d with random positions and velocities;
Step 3. Increment iteration counter by one;
Step 4. Evaluate the fitness function of all particles in the population, find particles best position Pbest of each particle and update its objective value. Similarly, find the global best position (Gbest) among all the particles and update its objective value;
Step 5. If stopping criterion is met go to step (11). Otherwise continue;
Step 6. Evaluate the inertia factor according to (3.106), so that each particles movement is directly controlled by its fitness value;
Step 7. Update the velocity using (3.107) and correct it using (3.108);
Step 8. Update the position of each particle according to (3.109), and if the new position goes out of range, set it to the boundary value using (3.110);
Step 9. The elites are inserted in the first position of the new population in order to maintain the best particle found so far;
Step 10. For every 5 generations, this $F_{\text{Best,new}}$ value (at the end of these 5 generations) is compared with the $F_{\text{Best,old}}$ value (at the beginning of these 5 generations, if there is no noticeable change, then re-initialize $k\%$ of the population. Go to step (3);
Step 11. Output the Gbest particle and its objective value.

3.4.4 Numerical Examples

Example 3.19. Let's consider the following problem,

$$\begin{cases} \max f_1(\boldsymbol{x}, \boldsymbol{\xi}) = \tilde{\tilde{\xi}}_1 x_1 + \tilde{\tilde{\xi}}_2 x_2 + \tilde{\tilde{\xi}}_3 x_3 + \tilde{\tilde{\xi}}_4 x_4 + \tilde{\tilde{\xi}}_5 x_5 \\ \max f_2(\boldsymbol{x}, \boldsymbol{\xi}) = \tilde{\tilde{\xi}}_6 x_1 + \tilde{\tilde{\xi}}_7 x_2 + \tilde{\tilde{\xi}}_8 x_3 + \tilde{\tilde{\xi}}_9 x_4 + \tilde{\tilde{\xi}}_{10} x_5 \\ \text{s.t.} \begin{cases} x_1 + x_2 + x_3 + x_4 + x_5 \le 350 \\ x_1 + x_2 + x_3 + x_4 + x_5 \ge 300 \\ 4x_1 + 2x_2 + 1.5x_3 + x_4 + 2x_5 \le 1085 \\ x_1 + 4x_2 + 2x_3 + 5x_4 + 3x_5 \le 660 \\ x_1 \ge 20, x_2 \ge 20, x_3 \ge 20, x_4 \ge 20, x_5 \ge 20 \end{cases} \end{cases} \tag{3.111}$$

where $\tilde{\tilde{\xi}}_i (i = 1, \ldots, 10)$ are Ra-Ra variables as follows,

$$\begin{aligned} &\tilde{\tilde{\xi}}_1 \sim \mathcal{N}(\bar{u}_1, 1), \textit{ with } \bar{u}_1 \sim \mathcal{N}(113, 2), \ \tilde{\tilde{\xi}}_2 \sim \mathcal{N}(\bar{u}_2, 4), \quad \textit{with } \bar{u}_2 \sim \mathcal{N}(241, 2), \\ &\tilde{\tilde{\xi}}_3 \sim \mathcal{N}(\bar{u}_3, 1), \textit{ with } \bar{u}_3 \sim \mathcal{N}(87, 3), \ \tilde{\tilde{\xi}}_4 \sim \mathcal{N}(\bar{u}_4, 2), \quad \textit{with } \bar{u}_4 \sim \mathcal{N}(56, 1), \\ &\tilde{\tilde{\xi}}_5 \sim \mathcal{N}(\bar{u}_5, 1), \textit{ with } \bar{u}_5 \sim \mathcal{N}(92, 1), \ \tilde{\tilde{\xi}}_6 \sim \mathcal{N}(\bar{u}_6, 1), \quad \textit{with } \bar{u}_6 \sim \mathcal{N}(628, 2), \\ &\tilde{\tilde{\xi}}_7 \sim \mathcal{N}(\bar{u}_7, 2), \textit{ with } \bar{u}_7 \sim \mathcal{N}(143, 1), \ \tilde{\tilde{\xi}}_8 \sim \mathcal{N}(\bar{u}_8, 2), \quad \textit{with } \bar{u}_8 \sim \mathcal{N}(476, 1), \\ &\tilde{\tilde{\xi}}_9 \sim \mathcal{N}(\bar{u}_9, 2), \textit{ with } \bar{u}_9 \sim \mathcal{N}(324, 2), \ \tilde{\tilde{\xi}}_{10} \sim \mathcal{N}(\bar{u}_{10}, 2), \textit{ with } \bar{u}_{10} \sim \mathcal{N}(539, 1). \end{aligned}$$

and $\bar{u}_i (i = 1, 2, \ldots, 10)$ are independently random variables. Set $\alpha_i = \beta_i = 0.9$ $(i = 1, 2)$, we can get the following chance-constrained model,

$$\begin{cases} \max\{\bar{f}_1, \bar{f}_2\} \\ \text{s.t.} \begin{cases} Pr\{\omega | Pr\{\tilde{\tilde{\xi}}_1 x_1 + \tilde{\tilde{\xi}}_2 x_2 + \tilde{\tilde{\xi}}_3 x_3 + \tilde{\tilde{\xi}}_4 x_4 + \tilde{\tilde{\xi}}_5 x_5 \ge \bar{f}_1\} \ge \beta_1\} \ge \alpha_1 \\ Pr\{\omega | Pr\{\tilde{\tilde{\xi}}_6 x_1 + \tilde{\tilde{\xi}}_7 x_2 + \tilde{\tilde{\xi}}_8 x_3 + \tilde{\tilde{\xi}}_9 x_4 + \tilde{\tilde{\xi}}_{10} x_5 \ge \bar{f}_2\} \ge \beta_2\} \ge \alpha_2 \\ x_1 + x_2 + x_3 + x_4 + x_5 \le 350 \\ x_1 + x_2 + x_3 + x_4 + x_5 \ge 300 \\ 4x_1 + 2x_2 + 1.5x_3 + x_4 + 2x_5 \le 1085 \\ x_1 + 4x_2 + 2x_3 + 5x_4 + 3x_5 \le 660 \\ x_1 \ge 20, x_2 \ge 20, x_3 \ge 20, x_4 \ge 20, x_5 \ge 20 \end{cases} \end{cases} \tag{3.112}$$

It follows from Theorem 3.14 and 3.15 that, problem (3.112) is equivalent to

$$\begin{cases} \max H_1(\boldsymbol{x}) = 113x_1 + 241x_2 + 87x_3 + 56x_4 + 92x_5 \\ \qquad\qquad - 1.28\sqrt{x_1^2 + 4x_2^2 + x_3^2 + 2x_4^2 + x_5^2} \\ \qquad\qquad - 1.28\sqrt{2x_1^2 + 2x_2^2 + 3x_3^2 + x_4^2 + x_5^2} \\ \max H_2(\boldsymbol{x}) = 628x_1 + 143x_2 + 476x_3 + 324x_4 + 539x_5 \\ \qquad\qquad - 1.28\sqrt{x_1^2 + 2x_2^2 + 2x_3^2 + 2x_4^2 + 2x_5^2} \\ \qquad\qquad - 1.28\sqrt{2x_1^2 + x_2^2 + x_3^2 + 2x_4^2 + x_5^2} \\ \text{s.t.} \begin{cases} x_1 + x_2 + x_3 + x_4 + x_5 \le 350 \\ x_1 + x_2 + x_3 + x_4 + x_5 \ge 300 \\ 4x_1 + 2x_2 + 1.5x_3 + x_4 + 2x_5 \le 1085 \\ x_1 + 4x_2 + 2x_3 + 5x_4 + 3x_5 \le 660 \\ x_1 \ge 20, x_2 \ge 20, x_3 \ge 20, x_4 \ge 20, x_5 \ge 20 \end{cases} \end{cases} \tag{3.113}$$

Next, we use the SWT method to solve the above problem. Firstly, let $H_1(\boldsymbol{x})$ be the reference objective and compute $a_2 = \min_{x \in X} H_2(\boldsymbol{x}) = 141354.3$ and $b_2 = \max_{x \in X} H_2(\boldsymbol{x}) = 192528.7$. Take $\varepsilon = 160{,}000$ in $[a_2, b_2]$ and construct the following ε-constraint problem,

$$\begin{cases} \max H_1(\boldsymbol{x}) = 113x_1 + 241x_2 + 87x_3 + 56x_4 + 92x_5 \\ \qquad\qquad - 1.28\sqrt{x_1^2 + 4x_2^2 + x_3^2 + 2x_4^2 + x_5^2} \\ \qquad\qquad - 1.28\sqrt{2x_1^2 + 2x_2^2 + 3x_3^2 + x_4^2 + x_5^2} \\ \text{s.t.} \begin{cases} 628x_1 + 143x_2 + 476x_3 + 324x_4 + 539x_5 \\ -1.28\sqrt{x_1^2 + 2x_2^2 + 2x_3^2 + 2x_4^2 + 2x_5^2} \\ -1.28\sqrt{2x_1^2 + x_2^2 + x_3^2 + 2x_4^2 + x_5^2} \le 160000 \\ x_1 + x_2 + x_3 + x_4 + x_5 \le 350 \\ x_1 + x_2 + x_3 + x_4 + x_5 \ge 300 \\ 4x_1 + 2x_2 + 1.5x_3 + x_4 + 2x_5 \le 1085 \\ x_1 + 4x_2 + 2x_3 + 5x_4 + 3x_5 \le 660 \\ x_1 \ge 20, x_2 \ge 20, x_3 \ge 20, x_4 \ge 20, x_5 \ge 20 \end{cases} \end{cases} \tag{3.114}$$

Secondly, we get the optimal solution $\boldsymbol{x}(\varepsilon) = (198.28, 65.43, 20.00, 20.00, 20.00)^T$ and Kuhn-Tucker multiplier $\lambda_{12}(\boldsymbol{x}(\varepsilon)) = 0.176$ of the constraint $H_2(\boldsymbol{x}) \le 160{,}000$.

Thirdly, to interact with the decision maker, we prepare Table 3.8, listing a few representative noninferior solutions. We can now go to the decision maker to elicit W_{12} for each noninferior point in Table 3.8.

Table 3.8 Data generated for Example 3.19

H_2	H_1	x_1	x_2	x_3	x_4	x_5	λ_{12}	w_{12}
141354.3	40583.37	166.67	73.33	20.00	20.00	20.00	0.1808	−10
146471.7	41023.49	175.34	71.16	20.00	20.00	20.00	0.1808	−8
151589.2	41462.95	184.02	68.99	20.00	20.00	20.00	0.1808	−6
156706.6	41901.77	192.70	66.83	20.00	20.00	20.00	0.1808	−4
161824.1	42340.01	201.38	64.66	20.00	20.00	20.00	0.1808	−2
166941.5	42777.71	210.06	62.49	20.00	20.00	20.00	0.1808	0
166941.5	43206.53	218.57	60.36	20.00	20.00	20.00	0.1808	+2
177176.4	43206.53	218.57	60.36	20.00	20.00	20.00	0.1807	+4
182293.8	43206.53	218.57	60.36	20.00	20.00	20.00	0.1807	+6
182293.8	43206.53	218.57	60.36	20.00	20.00	20.00	0.1807	+8
187411.3	43206.53	218.57	60.36	20.00	20.00	20.00	0.1807	+10

Example 3.20. Let's consider the following problem,

$$\begin{cases} \max[\bar{f}_1, \bar{f}_2] \\ \text{s.t.} \begin{cases} Pr\left\{\omega | Pr\left\{\sqrt{(x_1-\xi_1)^2+(x_2-\xi_2)^2} \geq \bar{f}_1\right\} \geq 0.8\right\} \geq 0.8 \\ Pr\left\{\omega | Pr\left\{\sqrt{(x_1+\xi_1)^2+(x_2+\xi_2)^2} \geq \bar{f}_2\right\} \geq 0.8\right\} \geq 0.8 \\ x_1 + x_2 \leq 5 \\ x_1 \geq 0, x_2 \geq 0 \end{cases} \end{cases} \tag{3.115}$$

where $\xi_1 \sim \mathcal{N}(\bar{\mu}_1, 1)$ and $\xi_1 \sim \mathcal{N}(\bar{\mu}_2, 1)$ are independently Ra-Ra variables. $\bar{\mu}_1 = \mathcal{N}(3, 1)$ and $\bar{\mu}_1 = \mathcal{N}(2, 0.5)$ are normally distributed random variables. Next we will use Ra-Ra simulation-based APSO to solve the above problem. The process of generating a new position for a selected individual in the swarm is depicted in the following equation:

$$v_{t+1}^i = wv_t^i + c_1 \cdot \text{rand}() \cdot (\boldsymbol{P}_t^i - \boldsymbol{x}_t^i) + c_2 \cdot \text{rand}() \cdot (\boldsymbol{G}_t - \boldsymbol{x}_t^i),$$

where v_t^i and $\boldsymbol{x}_t^i$ are the ith particle current velocity and position, $\boldsymbol{P}_t^i$ and $\boldsymbol{G}_t$ are the kth particle best position and the global best position visited so far, $w = 0.7298$ is the inertia weight, $c_1 = c_2 = 1.4962$ are learning factors and rand() is a random number in [0,1]. After the simulation with many cycles, we get the optimal solution under different weights as shown in Table 3.9. Figure 3.6 shows the search process when the weight is 0.5. The read line expresses the weight sum of two objective functions, and it shows that it gradually converges from Gen = 40. Figure 3.7 shows the changes of two objective values when the generation increases.

Table 3.9 The optimal solution by Ra-Ra simulation-based APSO

w_1	w_2	x_1	x_2	$\bar{f}_1$	$\bar{f}_2$	$\bar{f}$	Gen
0.1	0.9	3.5162	1.4838	0.2582	3.1564	4.8016	500
0.2	0.8	2.1930	2.8070	0.3677	4.7229	3.9686	500
0.3	0.7	0.3120	4.6880	1.2843	4.8389	4.3708	500
0.4	0.6	4.5370	0.4630	0.2726	5.5330	3.4103	500
0.5	0.5	4.0007	0.9993	0.2020	4.7898	2.8924	500

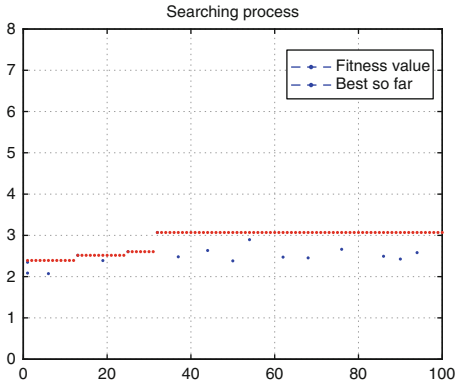


Fig. 3.6 Search process of Ra-Ra simulation-based APSO

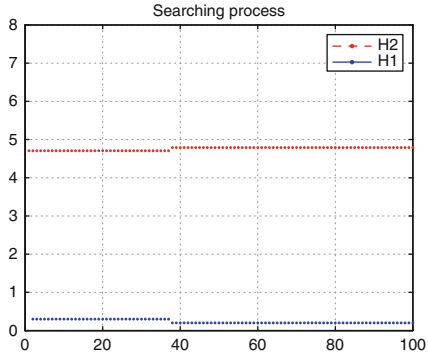


Fig. 3.7 Two objective values by Ra-Ra simulation-based APSO

3.5 Ra-Ra DCM

Uncertain environment, event, and the chance function are key elements in DCM. Let us redefine them in Ra-Ra decision systems, and introduce the principle of uncertainty. By uncertain environment (in this case the Ra-Ra environment) we mean the Ra-Ra constraints represented by

$$g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, j = 1, 2, \ldots, p \tag{3.116}$$

where $\boldsymbol{x}$ is a decision vector, and $\boldsymbol{\xi}$ is a Ra-Ra vector. By the event we mean the system of inequalities

$$h_k(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, k = 1, 2, \ldots, q \tag{3.117}$$

The chance function of an event ε characterized by (3.117) is defined as the chance measure of the event ε, i.e.,

$$f(\boldsymbol{x}) = Ch\{h_k(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, k = 1, 2, \ldots, q\} \tag{3.118}$$

subject to the uncertain environment (3.116).

For each decision $\boldsymbol{x}$ and realization $\boldsymbol{\xi}$, an event ε is said to be consistent in the uncertain environment if the following two conditions hold: (a) $h_k(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, k = 1, 2, \ldots, q$; and (b) $g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, j \in J$, where J is the index set of all dependent constraints.

Assume that there are m events ε_i characterized by $h_{ik}(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, k = 1, 2, \ldots, q_i$ for $i = 1, 2, \ldots, m$ in the uncertain environment $g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, j = 1, 2, \ldots, p$. The principle of uncertainty implies that the chance function of the ith event ε_i in the uncertain environment is

$$f_i(\boldsymbol{x}) = Ch \left\{ \begin{array}{l} h_{ik}(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, k = 1, 2, \ldots, q \\ g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, j \in J_i \end{array} \right\} \tag{3.119}$$

where J_i are defined by

$$J_i = \{j \in \{1, 2, \ldots, p\} | g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0 \textit{ is a dependent constraint of } \varepsilon_i\}$$

for $i = 1, 2, \ldots, m$.

3.5.1 General Model for Ra-Ra DCM

When α-chance measure is used, we may formulate a Ra-Ra DCM as follows:

$$\left\{ \begin{array}{l} \max Ch\{f(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\}(\alpha) \\ \text{s.t. } g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, j = 1, 2, \ldots, p \end{array} \right. \tag{3.120}$$

where $\boldsymbol{x}$ is an n-dimensional decision vector, $\boldsymbol{\xi}$ is a Ra-Ra vector, the event ε is characterized by $f(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0$, α is a given probability level, and the uncertain environment is described by the Ra-Ra constraints $g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, j = 1, 2, \ldots, p$.

Remark 3.4. If the Ra-Ra vector $\boldsymbol{\xi}$ degenerates to a random vector, then for any given $\alpha > 0$,

$$Ch\{f(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\}(\alpha) \equiv Pr\{h(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\}.$$

Thus the model (3.120) becomes

$$\begin{cases} \max Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \\ \text{s.t. } g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, j = 1, 2, \ldots, p \end{cases} \tag{3.121}$$

which is a standard random DCM.

If there are multiple events in the uncertain environment, then we have the following Ra-Ra dependent-chance multiobjective decision making model,

$$\begin{cases} \max \begin{bmatrix} Ch\{f_1(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\}(\alpha_1) \\ Ch\{f_2(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\}(\alpha_2) \\ \cdots \\ Ch\{f_m(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\}(\alpha_m) \end{bmatrix} \\ \text{s.t. } g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, j = 1, 2, \ldots, p \end{cases} \tag{3.122}$$

where the events ε_i are characterized by $f_i(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0$ and α_i are given probability levels, $i = 1, 2, \ldots, m$, respectively.

The Ra-Ra dependent-chance goal model (DCGM) to formulate bi-random decision systems according to the priority structure and target levels set by the decision-maker is given as follows,

$$\begin{cases} \min \sum_{j=1}^{l} P_j \sum_{i=1}^{m} (u_{ij} d_i^+ + v_{ij} d_i^-) \\ \text{s.t.} \begin{cases} Ch\left\{ f_i(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0 \right\}(\alpha_i) + d_i^- - d_i^+ = b_i \\ g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, j = 1, 2, \ldots, p \\ d_i^-, d_i^+ \geq 0, i = 1, 2, \ldots, m \end{cases} \end{cases} \tag{3.123}$$

where P_j is the preemptive priority factor which express the relative importance of various goals, $P - j >> P_{j+1}$, for all j, u_{ij} is the weighting factor corresponding to positive deviation for goal i with priority j assigned, u_{ij} is the weighting factor corresponding to negative deviation for goal i with priority j assigned, d_i^+ is the positive deviation from the target of goal i, d_i^- is the negative deviation from the target of goal i, α_i is the given probability level, g_j is a function in system constraints, b_i is the target value according to goal i, l is the number of priorities, m is the number of goal constraints, and p is the number of system constraints.

3.5.2 *Linear Ra-Ra DCM and the Satisfying Trade-Off Method*

In this section, we restrict our attention on the linear multiobjective programming with Ra-Ra parameters. Let's consider the following model,

$$\begin{cases} \max\left[\tilde{\tilde{c}}_1^T \boldsymbol{x}, \tilde{\tilde{c}}_2^T \boldsymbol{x}, \ldots, \tilde{\tilde{c}}_m^T \boldsymbol{x}\right] \\ \text{s.t.} \begin{cases} \tilde{\tilde{e}}_r^T \boldsymbol{x} \le \tilde{\tilde{b}}_r, \ r = 1, 2, \ldots, p \\ \boldsymbol{x} \ge 0 \end{cases} \end{cases} \tag{3.124}$$

where $\tilde{\tilde{c}}_i = (\tilde{\tilde{c}}_{i1}, \tilde{\tilde{c}}_{i2}, \ldots, \tilde{\tilde{c}}_{in})^T$, $\tilde{\tilde{e}}_r = (\tilde{\tilde{e}}_{r1}, \tilde{\tilde{e}}_{r2}, \ldots, \tilde{\tilde{e}}_{rn})^T$ are Ra-Ra vectors and $\tilde{\tilde{b}}_r$ are Ra-Ra variables, $i = 1, 2, \ldots, m$, $r = 1, 2, \ldots, p$. Then by the definition of chance measure of Ra-Ra variables, we have the following chance-constrained model of (3.124),

$$\begin{cases} \max\left[Ch\{\tilde{\tilde{c}}_i^T \boldsymbol{x} \ge \bar{f}_i\}(\alpha_i), i = 1, 2, \ldots, m\right] \\ \text{s.t.} \begin{cases} Ch\{\tilde{\tilde{e}}_r^T \boldsymbol{x} \le \tilde{\tilde{b}}_r\}(\eta_r) \ge \theta_r, \ r = 1, 2, \ldots, p \\ \boldsymbol{x} \ge 0 \end{cases} \end{cases} \tag{3.125}$$

where $\alpha_i, \eta_r, \theta_r$ are given confidence levels, f_i are predetermined objective value, $i = 1, 2, \ldots, m, r = 1, 2, \ldots, p$. Then the problem (3.125) can be rewritten as

$$\begin{cases} \max\left[\sup\{\beta_i | Pr\{\omega | Pr\{\tilde{\tilde{c}}_i^T \boldsymbol{x} \ge \bar{f}_i\} \ge \beta_i\} \ge \alpha_i\}, i = 1, 2, \ldots, m\right] \\ \text{s.t.} \begin{cases} Pr\{\omega | Pr\{\tilde{\tilde{e}}_r^T \boldsymbol{x} \le \tilde{\tilde{b}}_r\} \ge \theta_r\} \ge \eta_r, \ r = 1, 2, \ldots, p \\ \boldsymbol{x} \ge 0 \end{cases} \end{cases} \tag{3.126}$$

3.5.2.1 Crisp Equivalent Model

One way of solving the dependent-chance multiobjective programming model is to convert the objectives and constraints of problem (3.126) into their respective crisp equivalents. As we know, this process is usually a hard work and only successful for some special cases. Next, we will consider a special case and present the result in this section.

Theorem 3.18. *Assume that the Ra-Ra vector* $(\tilde{\tilde{c}}_i = (\tilde{\tilde{c}}_{i1}, \tilde{\tilde{c}}_{i2}, \ldots, \tilde{\tilde{c}}_{in})^T$ *is normally distributed with mean vector* $\bar{d}_i^c(\omega) = (\bar{d}_{i1}^c(\omega), \bar{d}_{i2}^c(\omega), \ldots, \bar{d}_{in}^c(\omega))^T$ *and positive definite covariance matrix* V_i^c*, written as* $(\tilde{\tilde{c}}_{ij} \sim \mathcal{N}(\bar{d}_i^c(\omega), V_i^c)$*, where* $\tilde{d}_{ij}^c(\omega)$ *is a random variable characterized by* $\tilde{d}_{ij}^c(\omega) \sim \mathcal{N}(d_{ij}^c, \sigma_{ij}^2)$*. Then we have that*

$$\begin{aligned} &\sup\{\beta_i | Pr\{Pr\{\tilde{\tilde{c}}_i^T \boldsymbol{x} \ge \bar{f}_i\} \ge \beta_i\} \ge \alpha_i\} \\ &= \Phi\left(\frac{\Phi^{-1}(1-\alpha_i)\sqrt{\sum_{j=1}^{n} x_{ij}^2 \sigma_{ij}^2} + d_i^{cT} \boldsymbol{x} - \bar{f}_i}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right). \end{aligned}$$

Proof. From the assumption and Theorem 3.14, we have that

$$\begin{aligned} &Pr\{\omega | Pr\{\tilde{\tilde{c}}_i^T \boldsymbol{x} \geq \bar{f}_i\} \geq \beta_i\} \geq \alpha_i \\ &\quad \Leftrightarrow \bar{f}_i \leq \Phi^{-1}(1-\beta_i)\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}} + \Phi^{-1}(1-\alpha_i)\sqrt{\sum_{j=1}^{n} x_{ij}^2 \sigma_{ij}^2} + d_i^{cT}\boldsymbol{x} \end{aligned}$$

Then it follows that

$$\beta_i \leq 1 - \Phi\left(\frac{\bar{f}_i - (\Phi^{-1}(1-\alpha_i)\sqrt{\sum_{j=1}^{n} x_{ij}^2 \sigma_{ij}^2} + d_i^{cT}\boldsymbol{x})}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right).$$

Since $\Phi(-x) = 1 - \Phi(x)$, then we have

$$\beta_i \leq \Phi\left(\frac{\Phi^{-1}(1-\alpha_i)\sqrt{\sum_{j=1}^{n} x_{ij}^2 \sigma_{ij}^2} + d_i^{cT}\boldsymbol{x} - \bar{f}_i}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right)$$

Thus

$$\begin{aligned} &\sup\{\beta_i | Pr\{Pr\{\tilde{\tilde{c}}_i^T \boldsymbol{x} \geq \bar{f}_i\} \geq \beta_i\} \geq \alpha_i\} \\ &\quad = \Phi\left(\frac{\Phi^{-1}(1-\alpha_i)\sqrt{\sum_{j=1}^{n} x_{ij}^2 \sigma_{ij}^2} + d_i^{cT}\boldsymbol{x} - \bar{f}_i}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right) \end{aligned}$$

This completes the proof. □

Similarly, we have that $Pr\{\omega | Pr\{\tilde{\tilde{e}}_r^T \boldsymbol{x} \leq \tilde{\tilde{b}}_r\} \geq \theta_r\} \geq \eta_r$ is equivalent to

$$\Phi^{-1}(\theta_r)\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2} - (d_r^b - d_r^e \boldsymbol{x}) - \Phi^{-1}(\eta_r)\sqrt{(\delta_r^b)^2 + \sum_{j=1}^{n} x_{ij}^2 (\delta_r^e)^2} \leq 0.$$

Then problem (3.126) can be rewritten as

$$\begin{cases} \max \left[\Phi \left(\dfrac{\Phi^{-1}(1-\alpha_i)\sqrt{\sum\limits_{j=1}^{n} x_{ij}^2 \sigma_{ij}^2} + d_i^{cT}\boldsymbol{x} - \bar{f_i}}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}} \right), i = 1, 2, \ldots, m \right] \\ \text{s.t. } \boldsymbol{x} \in X \end{cases} \tag{3.127}$$

where $X = \{\boldsymbol{x} | \Phi^{-1}(\theta_r)\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2} - \Phi^{-1}(\eta_r)\sqrt{(\delta_r^b)^2 + \sum\limits_{j=1}^{n} x_{ij}^2 (\delta_r^e)^2} - (d_r^b - d_r^e \boldsymbol{x}) \leq 0, \boldsymbol{x} \geq 0\}$.

3.5.2.2 The Satisfying Trade-Off Method

The satisfying trade-off method for multi-objective programming problems was proposed by Nakayama [233, 277]. It is an interactive method combining the satisfying level method with the ideal point method. This method can be applied to not only the linear multi-objective but also the nonlinear multi-objective programming.

Take the problem (3.127) as an example. Let $H_i(\boldsymbol{x}) = \Phi\left(\frac{R^{-1}(\alpha_i)\gamma_{ij}^c + d_i^{cT}x - \bar{f_i}}{\sqrt{x^T V_i^c x}}\right)$, and $X = \{\boldsymbol{x} | \Phi^{-1}(\theta_r)\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2} - (d_r^b - d_r^e \boldsymbol{x}) - R^{-1}(\eta_r)(\gamma_r^b + \delta_r^e \boldsymbol{x}) \leq 0, \boldsymbol{x} \geq 0\}$, then problem (3.127) is equivalent to

$$\begin{cases} \max[H_1(\boldsymbol{x}), H_2(\boldsymbol{x}), \ldots, H_m(\boldsymbol{x})] \\ \text{s.t. } \boldsymbol{x} \in X \end{cases} \tag{3.128}$$

In the begin, let's briefly introduce the simple satisfying level method which in mainly referred in [234]. In some real decision making problems, DM usually provides a reference objective values $\bar{\boldsymbol{H}} = (\bar{H}_1, \bar{H}_2, \ldots, \bar{H}_m)^T$. If the solution satisfies the reference value, take it. The simple satisfying level method can be summarized as follows:

Step 1. DM gives the reference objective values $\bar{\boldsymbol{H}}$.

Step 2. Solve the following programming problem,

$$\begin{cases} \max \sum\limits_{i=1}^{m} H_i(\boldsymbol{x}) \\ \text{s.t.} \begin{cases} H_i(\boldsymbol{x}) \geq \bar{H}_i, \ i = 1, 2, \ldots, m \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{3.129}$$

Step 3. If the problem (3.129) doesn't have the feasible solution, turn to Step 4. If the problem (3.129) has the optimal solution $\bar{\boldsymbol{x}}$, output $\bar{\boldsymbol{x}}$.

Step 4. DM re-gives the reference objective values $\bar{\boldsymbol{H}}$ and turn to Step 2.

The satisfying trade-off method can be summarized as follows:

Step 1. Take the ideal point $\boldsymbol{H}^* = (H_1^*, H_2^*, \ldots, H_m^*)^T$ such that $H_i^* > \max_{\boldsymbol{x}\in X} f_i(\boldsymbol{x})$ $(i = 1, 2, \ldots, m)$.

Step 2. DM gives the objective level $\bar{\boldsymbol{H}}^k = (\bar{H}_1^k, \bar{H}_2^k, \ldots, \bar{H}_m^k)^T$ and $\bar{H}_i^k < \bar{H}_i^*(i = 1, 2, \ldots, m)$. Let $k = 1$.

Step 3. Compute the weight and solve the following problem to get the efficient solution.

$$w_i^k = \frac{1}{H_i^* - \bar{H}_i^k},\ i = 1, 2, \ldots, m \tag{3.130}$$

$$\min_{\boldsymbol{x}\in X} \max_{1\le i\le m} w_i^k |H_i^* - H_i(\boldsymbol{x})| \tag{3.131}$$

or the equivalent problem,

$$\begin{cases} \min \lambda \\ \text{s.t.} \begin{cases} w_i^k (H_i^* - H_i(\boldsymbol{x})) \le \lambda,\ i = 1, 2, \ldots, m \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{3.132}$$

Suppose that the optimal solution is $\boldsymbol{x}^k$.

Step 4. According to the objective value $\boldsymbol{H}(\boldsymbol{x}^k) = (H_1(\boldsymbol{x}^k), H_2(\boldsymbol{x}^k), \ldots, H_m(\boldsymbol{x}^k))^T$, DM divide them into three classes: (1) which needs to improve, denote the related subscript set I_I^k, (2) which is permitted to release, denote the related subscript set I_R^k, (3) which is accepted, denote the related subscript set I_A^k. If $I_I^k = \Phi$, stop the iteration and output $\boldsymbol{x}^k$. Otherwise, DM gives the new reference objective values $\tilde{H}_i^k, i \in I_I^k \cup I_R^k$ and let $\tilde{H}_i^k = H_i(\boldsymbol{x}^k), i \in I_A^k$.

Step 5. Let $u_i (i = 1, 2, \ldots, m)$ be the optimal Kuhn-Tucker operator of the first constraints. If there exists a minimal nonnegative number ε such that

$$\sum_{i=1}^{m} u_i w_i^k (\tilde{H}_i^k - H_i(\boldsymbol{x}^k)) \ge -\varepsilon,$$

then we deem that $\tilde{H}_i^k$ passes the check for feasibility. Let $\tilde{H}_{i+1} = \tilde{H}_i^k$ $(i = 1, 2, \ldots, m)$, turn to Step 3. Otherwise, $\tilde{H}_i^k$ isn't feasible. The detail can be referred in [277]. DM should re-give $\tilde{H}_i^k, i \in I_I^k \cup I_R^k$ and recheck it.

3.5.3 Nonlinear Ra-Ra DCM and Ra-Ra Simulation-Based Tribe-PSO

Consider the Ra-Ra DCM as follows,

$$\begin{cases} \max [Ch\{f_k(\boldsymbol{x}, \boldsymbol{\xi}) \ge f_k\}(\alpha_k), k = 1, 2, \ldots, q] \\ \text{s.t. } Ch\{g_r(\boldsymbol{x}, \boldsymbol{\xi}) \le 0\}(\theta_r) \ge \eta_r,\ r = 1, 2, \ldots, p \end{cases} \tag{3.133}$$

where $\boldsymbol{x}$ is an n-dimensional decision vector, $\boldsymbol{\xi}$ is a Ra-Ra vector, the event ε is characterized by $f_k(\boldsymbol{x}, \boldsymbol{\xi}) \geq f_k$, f_k is the predetermined value, $f_k(\boldsymbol{x}, \boldsymbol{\xi})$ is nonlinear functions with respect to $\boldsymbol{\xi}$, α_k, θ_r and η_r are the given probability levels, and the uncertain environment is described by the Ra-Ra constraints $g_r(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0$, $k = 1, 2, \ldots, q$, $r = 1, 2, \ldots, p$.

3.5.3.1 Ra-Ra Simulation for DCM

Since the existence of nonlinear functions $f_k(\boldsymbol{x}, \boldsymbol{\xi})$, it is difficult to convert the objective functions into the crisp ones. Then the Ra-Ra simulation technique is used to obtain the approximative chance measure. Suppose that $\boldsymbol{\xi}$ is an n-dimensional Ra-Ra vector defined on the probability space $(\Omega, \mathscr{A}, Pr)$, and $f : \mathbf{R}^n \to \mathbf{R}$ is a measurable function. For any real number $\alpha \in (0, 1]$, we design a Ra-Ra simulation to compute the α-chance $Ch\{f(\boldsymbol{x}, \boldsymbol{\xi}) \geq f\}(\alpha)$. That is, we should find the supremum $\bar{\beta}$ such that

$$Pr\{\omega \in \Omega | Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}(\omega)) \geq f\} \geq \bar{\beta}\} \geq \alpha.$$

First, we sample $\omega_1, \omega_2, \ldots, \omega_N$ from Ω according to the probability measure Pr, and estimate $\beta_k = Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}(\omega_k)) \geq f\}$ for $k = 1, 2, \ldots, N$ by random simulation. Let N' be the integer part of αN. Then the value $\bar{\beta}$ can be taken as the N'th largest element in the sequence $\{\beta_1, \beta_2, \ldots, \beta_N\}$.

Then the procedure simulating the α-chance $Ch\{f(\boldsymbol{x}, \boldsymbol{\xi}) \geq f\}(\alpha)$ can be summarized as follows:

Procedure `Ra-Ra simulation for DCM`

Input: The decision vector $\boldsymbol{x}$
Output: α-chance $Ch\{f(\boldsymbol{x}, \boldsymbol{\xi}) \geq f\}(\alpha)$
Step 1. Generate $\omega_1, \omega_2, \omega_N$ from Ω according to the probability measure Pr;
Step 2. Compute the probability $\beta_k = Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}(\omega_k)) \geq f\}$ for $k = 1, 2, \ldots, N$ by random simulation;
Step 3. Set N' as the integer part of αN;
Step 4. Return the N'th largest element in $\{\beta_1, \beta_2, \ldots, \beta_N\}$.

Example 3.21. Consider the following Ra-Ra variables,

$$\begin{aligned}
\tilde{\tilde{\xi}}_1 &= \mathscr{U}(\tilde{\rho}_1, \tilde{\rho}_1 + 2), \text{ with } \tilde{\rho}_1 \sim \mathscr{N}(0, 1), \\
\tilde{\tilde{\xi}}_2 &= \mathscr{N}(\tilde{\rho}_2, 1), \qquad \text{with } \tilde{\rho}_2 \sim \mathscr{U}(3, 5), \\
\tilde{\tilde{\xi}}_3 &= exp(\tilde{\rho}_3), \qquad \text{with } \tilde{\rho}_3 \sim \mathscr{U}(1, 2).
\end{aligned}$$

A run of Ra-Ra simulation with 1000 cycles shows that

$$Ch\left\{\sqrt{\tilde{\tilde{\xi}}_1^2 + \tilde{\tilde{\xi}}_2^2 + \tilde{\tilde{\xi}}_3^2} \geq 6\right\}(0.8) = 0.5150.$$

3.5.3.2 Tribe Particle Swarm Optimization

Since the particle swarm optimization (PSO) algorithm was proposed by Kennedy and Eberhard [158], many improvements have bee made by some scholars. Tribe-PSO is one of them, which is proposed by Chen, Li and Cao [51]. In classical PSO, a parameter that the user must specify is the sociometry (topology, type and quantity of social relationships) of the swarm. In order to create a more robust PSO variant, Clerc [58, 59] proposed a parameter-free PSO method called Tribes, in which details of the topology, including the size of the population, evolve over time in response to performance feedback. TRIBES has attracted attention from researchers in different application areas such as the optimization of milling operations [239], flow shop scheduling [240], particle swarm optimization [51, 82] and the Loney's Solenoid [82]. Tribe-PSO is inspired from the concept of hierarchical fair competition (HFC) developed by Goodman [136]. The main principles in HFC are that the competition is allowed only among individuals with comparable fitness and it should be organized into hierarchical levels. Such principles help the algorithms to prevent the optimization procedure from too early loss of diversity and help the population to escape from the prematurity. Goodman et al. [136] have successfully introduced these principles into genetic algorithm and other EAs and developed the AHFC model. Keeping diversity in the population is very important for complex optimization problems, such as flexible docking. Thus, Chen, Li and Cao [51] proposed Tribe-PSO (Tribe Particle Swarm Optimization), a hybrid PSO model based on HFC principles. In this model, particles are divided into two layers and the whole procedure of convergence is divided into three phases. Particles on different layers or in different phases are strictly controlled in order to preserve population diversities.

A tribe is a sub-swarm formed by particles which have the property that all particles inform all others belonging to the tribe (a symmetrical clique in graph theoretical language). The concept is therefore related to the "cultural vicinity" (information neighborhood) and not on "spatial vicinity" (parameter-space neighborhood). It should be noted that, due to the above definition, the set of informers of a particle (its so-called i-group) contains the whole of its tribe but is not limited to it. This is shown in Fig. 3.8 where the i-group of particle B1 contains all particles of its tribe (black) and particle W1 belonging to the white tribe.

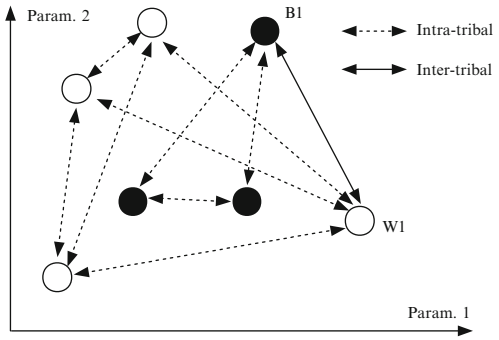


Fig. 3.8 Tribal relationships

In general PSO algorithm, Chen, Li and Cao [51] found that the part of $(pBest - p)$ and especially the part of $(pBest_{gBest} - p)$ still dominates the whole velocity function although these two vectors have random factors before each item. It results that the current position p is located in the neighborhood of $pBest$ while it is much far away from the position of $pBest_{gBest}$ and furthermore the norm of the vector towards $pBest_{gBest}$ is much larger than that of the vector towards $pBest$. In this way, particles with worse fitness but promising diversities are strongly attracted by the best particle of the swarm and get entrapped in the neighborhood of $gBest$ and lose their diversities. They also found that this kind of competition is inadequate and does harm to the healthy procedure of optimization. In order to prevent the prematurity, we introduce the principle of HFC and propose the Tribe-PSO model.

In the following part, we will introduce the detail of Tribe-PSO model proposed by Chen, Li and Cao [51]. Tribe-PSO has two important concepts: layer and phase. In a word, particles in the swarm are divided into two layers and the procedure of optimization is divided into three phases. Assume that there are totally $l \times m$ particles in the swarm. In the initiation step of Tribe-PSO, the swarm is divided into l sub-populations, called as tribes. Each tribe has the same structure of the basic PSO model: it has m particles and the best particle from them is called $tBest$. Tribes form the basic layer while the best particles from the l tribes form the upper layer in the two-layered structure of Tribe-PSO, which is shown in Fig. 3.9. The convergence procedure of Tribe-PSO consists of three phases: *isolated phase*, *communing phase* and *united phase*. Each phase occupies a portion of iterations. In the first phase, the tribes are isolated and work as l independent PSO models. No information is exchanged between each pair of tribes. The isolated phase ensures the tribes having enough time to develop before the possible premature convergence. In the second phase, Tribe-PSO works in the standard two-layered model: tribe members p form the basic layer (the tribes) and the best particles from each tribe $tBest$ form the

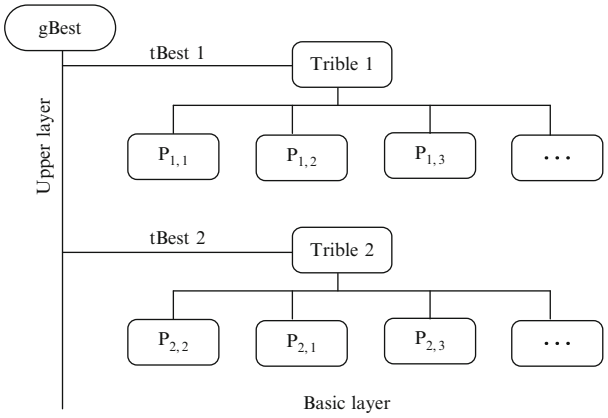


Fig. 3.9 Two-layer structure of Tribe-PSO

upper layer. Information about searching experience is exchanged among certain basic tribe and among the upper layer as well. The particle $tBest$, i.e. the best particle from tribe in the basic layer, serves as the information-exchanging agency between the tribe members p and the best particle of the whole swarm $gBest$. The communing phase leads the procedure to convergence in a more moderate way than basic PSO. Therefore the diversity of most particles is well preserved. In the last phase, all the tribes are united into one group. The model becomes a basic PSO model. The concept of $tBest$ does not exist any longer. The united phase helps the swarm to converge as quickly as possible. In the Tribe-PSO model, particles in different layers or in different phases have different velocity functions. The velocity functions for a specific layer and phase are defined as follows.

Phase I (Isolated phase). There is no upper layer or $gBest$ in this phase. All the $tBest$ and p from different tribes have the same velocity function as (3.134).

$$\begin{aligned} v = v \times w + 2 \times rand() \times (pBest - p) + 2 \times rand() \\ \times (pBest_{tBest} - p) \end{aligned} \tag{3.134}$$

Phase II (Communing phase). For tribe members, the velocity function is the same as they have in the first phase (3.134). For $tBest$ and $gBest$ particles, their velocity function is described as (3.135). In this equation, $gBest$ is regarded as the leader in the upper layer.

$$\begin{aligned} v = v \times w + 2 \times rand() \times (pBest - p) + 2 \times rand() \\ \times (pBest_{gBest} - p) \end{aligned} \tag{3.135}$$

Phase III (United phase). Theres no *tBest* in this phase. All the tribes are united into one swarm. Thus, the velocity function for all the particles becomes the original one in the basic PSO (3.135). Compared with basic PSO, Tribe-PSO has two more parameters. One is the ratio of tribe number l to tribe size m. The other is the partition of three phases. These two parameters have considerable effects on the global search ability and performance of Tribe-PSO model. The discussion on the influence of the two parameters can be found in [51].

3.5.4 Numerical Examples

Example 3.22. Let's consider the following problem,

$$\begin{cases} \max f_1(\boldsymbol{x}, \boldsymbol{\xi}) = Ch\{\tilde{\xi}_1 x_1 + \tilde{\xi}_2 x_2 + \tilde{\xi}_3 x_3 \geq f_1\}(\alpha) \\ \max f_2(\boldsymbol{x}, \boldsymbol{\xi}) = Ch\{c_1\tilde{\xi}_4 x_1 + c_2\tilde{\xi}_5 x_2 + c_3\tilde{\xi}_6 x_3 \geq f_2\}(\beta) \\ \text{s.t.} \begin{cases} x_1 + x_2 + x_3 \leq 15 \\ x_1 + x_2 + x_3 \geq 10 \\ x_1 + 4x_2 + 2x_3 \leq 30 \\ 2 \leq x_1, x_2, x_3 \leq 6 \end{cases} \end{cases} \tag{3.136}$$

where $c = (c_1, c_2, c_3) = (1.2, 0.8, 1.5)$ and $\tilde{\tilde{\xi}}_i (i = 1, \ldots, 10)$ are Ra-Ra variables as follows,

$$\begin{array}{llll}
\tilde{\tilde{\xi}}_1 \sim \mathcal{N}(\bar{u}_1, 1), & with\ \bar{u}_1 \sim \mathcal{N}(11, 2), & \tilde{\tilde{\xi}}_2 \sim \mathcal{N}(\bar{u}_2, 4), & with\ \bar{u}_2 \sim \mathcal{N}(12, 2), \\
\tilde{\tilde{\xi}}_3 \sim \mathcal{N}(\bar{u}_3, 1), & with\ \bar{u}_3 \sim \mathcal{N}(8, 1), & \tilde{\tilde{\xi}}_4 \sim \mathcal{N}(\bar{u}_4, 2), & with\ \bar{u}_4 \sim \mathcal{N}(6, 1), \\
\tilde{\tilde{\xi}}_5 \sim \mathcal{N}(\bar{u}_5, 1), & with\ \bar{u}_5 \sim \mathcal{N}(9, 1), & \tilde{\tilde{\xi}}_6 \sim \mathcal{N}(\bar{u}_6, 1), & with\ \bar{u}_6 \sim \mathcal{N}(8, 2),
\end{array}$$

and $\bar{u}_i (i = 1, 2, \ldots, 6)$ are independently random variables. Set $\alpha = \beta = 0.9$, $f_1 = 65$ and $f_2 = 73$, then we get the following equivalent model by Theorem 3.18 and (3.127),

$$\begin{cases}
\max H_1(\boldsymbol{x}) = \Phi\left(\dfrac{11x_1 + 12x_2 + 8x_3 - 1.28\sqrt{2x_1^2 + 2x_2^2 + x_3^2} - 65}{\sqrt{x_1^2 + 4x_2^2 + x_3^2}}\right) \\
\max H_2(\boldsymbol{x}) = \Phi\left(\dfrac{6x_1 + 9x_2 + 8x_3 - 1.28\sqrt{x_1^2 + x_2^2 + 2x_3^2} - 73}{\sqrt{2x_1^2 + x_2^2 + x_3^2}}\right) \\
\text{s.t.} \begin{cases}
x_1 + x_2 + x_3 \le 15 \\
x_1 + x_2 + x_3 \ge 10 \\
x_1 + 4x_2 + 2x_3 \le 30 \\
2 \le x_1, x_2, x_3 \le 6
\end{cases}
\end{cases} \tag{3.137}$$

Since $\Phi(x)$ is a monotone function, the problem (3.137) can be rewritten as

$$\begin{cases}
\max H_1(\boldsymbol{x}) = \dfrac{11x_1 + 12x_2 + 8x_3 - 1.28\sqrt{2x_1^2 + 2x_2^2 + x_3^2} - 65}{\sqrt{x_1^2 + 4x_2^2 + x_3^2}} \\
\max H_2(\boldsymbol{x}) = \dfrac{6x_1 + 9x_2 + 8x_3 - 1.28\sqrt{x_1^2 + x_2^2 + 2x_3^2} - 73}{\sqrt{2x_1^2 + x_2^2 + x_3^2}} \\
\text{s.t.} \begin{cases}
x_1 + x_2 + x_3 \le 15 \\
x_1 + x_2 + x_3 \ge 10 \\
x_1 + 4x_2 + 2x_3 \le 30 \\
2 \le x_1, x_2, x_3 \le 6
\end{cases}
\end{cases} \tag{3.138}$$

Next, we use the satisfying trade-off method to solve the above problem.

Firstly, calculate the problem $\max_{x \in X} H_i(\boldsymbol{x})$ and we get $H_1^1 = 6.80$ and $H_2^1 = 2.23$. Then we set the ideal point $\boldsymbol{H}^* = (H_1^*, H_2^*)^T = (7.00, 2.50)^T$.

Secondly, let DM give the objective level $\bar{\boldsymbol{H}}^k = (\bar{H}_1^k, \bar{H}_2^k) = (6.90, 2.40)^T$ $(k = 1)$.

Thirdly, compute the weight coefficients by the following equation,

$$w_i^k = \frac{1}{H_i^* - \bar{H}_i^k},\ i = 1, 2,$$

and we get $w_1^1 = 10$ and $w_2^1 = 10$. Solve the following problem,

$$\begin{cases} \min \lambda \\ \text{s.t.} \begin{cases} 10(7.00 - \dfrac{11x_1 + 12x_2 + 8x_3 - 1.28\sqrt{2x_1^2 + 2x_2^2 + x_3^2} - 65}{\sqrt{x_1^2 + 4x_2^2 + x_3^2}}) \le \lambda \\ 10(2.50 - \dfrac{6x_1 + 9x_2 + 8x_3 - 1.28\sqrt{x_1^2 + x_2^2 + 2x_3^2} - 73}{\sqrt{2x_1^2 + x_2^2 + x_3^2}}) \le \lambda \\ x_1 + x_2 + x_3 \le 15 \\ x_1 + x_2 + x_3 \ge 10 \\ x_1 + 4x_2 + 2x_3 \le 30 \\ 2 \le x_1, x_2, x_3 \le 6 \end{cases} \end{cases} \quad (3.139)$$

and we get $\boldsymbol{x}^1 = (6.00, 3.00, 6.00)^T$.

Fourthly, compute each objective value and we get $H_1(\boldsymbol{x}^1) = 6.80$ and $H_2(\boldsymbol{x}^1) = 2.23$. If DM figures that the two objective values needn't be changed, output $\boldsymbol{x}^* = \boldsymbol{x}^1$.

Example 3.23. Let's consider the following problem,

$$\begin{cases} \max H_1(\boldsymbol{x}, \boldsymbol{\xi}) = Ch\left\{\sqrt{(x_1 - \xi_1)^2 + (x_2 - \xi_2)^2} \ge 2.34\right\}(0.8) \\ \max H_2(\boldsymbol{x}, \boldsymbol{\xi}) = Ch\left\{\sqrt{(x_1 + \xi_1)^2 + (x_2 + \xi_2)^2} \ge 6.20\right\}(0.8) \\ \text{s.t.} \begin{cases} x_1 + x_2 \le 5 \\ x_1 \ge 0, x_2 \ge 0 \end{cases} \end{cases} \quad (3.140)$$

where $\xi_1 \sim \mathcal{N}(\bar{\mu}_1, 1)$ and $\xi_1 \sim \mathcal{N}(\bar{\mu}_2, 1)$ are independently Ra-Ra variables. $\bar{\mu}_1 = \mathcal{N}(3, 1)$ and $\bar{\mu}_1 = \mathcal{N}(2, 0.5)$ are normally distributed random variables. Next we will use Ra-Ra simulation-based Tribe-PSO to solve the above problem. The process of generating a new position for a selected individual in the swarm is depicted in the following equation:

$$v_{t+1}^i = wv_t^i + c_1 \cdot \text{rand}() \cdot (\boldsymbol{P}_t^i - \boldsymbol{x}_t^i) + c_2 \cdot \text{rand}() \cdot (\boldsymbol{G}_t - \boldsymbol{x}_t^i),$$

where v_t^i and $\boldsymbol{x}_t^i$ are the ith particle current velocity and position, $\boldsymbol{P}_t^i$ and $\boldsymbol{G}_t$ are the kth particle best position and the global best position visited so far, $w = 0.7298$ is the inertia weight, $c_1 = c_2 = 1.4962$ are learning factors and rand() is a random number in [0,1]. After the simulation with many cycles, we get the optimal solution

Table 3.10 The optimal solution obtained by Ra-Ra simulation-based Tribe-PSO

w_1	w_2	x_1	x_2	$\bar{f}_1$	$\bar{f}_2$	$\bar{f}$	Gen
0.1	0.9	4.8498	0.1502	0.5506	0.4500	0.5236	500
0.2	0.8	0.9169	4.0831	0.7681	0.4500	0.6967	500
0.3	0.7	1.1836	3.8164	0.5575	0.3500	0.6612	500
0.4	0.6	4.6811	0.3189	0.5199	0.2000	0.6231	500
0.5	0.5	0.8084	4.1916	0.8189	0.5000	0.8815	500

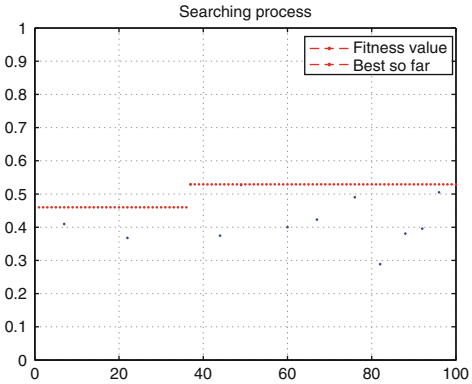


Fig. 3.10 Search process of Ra-Ra simulation-based Tribe-PSO

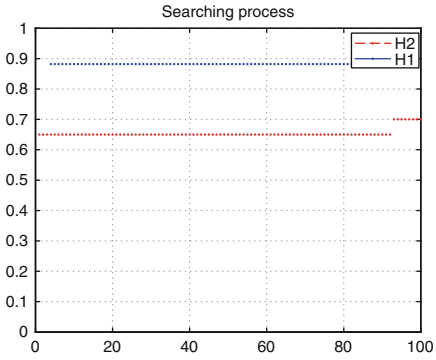


Fig. 3.11 Two objective values by Ra-Ra simulation-based Tribe-PSO

under different weights as shown in Table 3.10. Figure 3.10 shows the search process when the weight is 0.5. The read line expresses the weight sum of two objective functions, and it shows that it gradually converges from Gen $=$ 40. Figure 3.11 shows the changes of two objective values when the generation increases.

3.6 Application to GEELY Haoqing Automotive Holdings Group Limited Company

In this section, we discuss a flow shop scheduling problem under a Ra-Ra environment and its application to the FSSP of GEELY Haoqing Automotive Holdings Group Limited Company. We first introduce the basic model of the flow shop scheduling problem.

3.6.1 Background Introduction

GEELY Haoqing Automotive Holdings Group Limited Company is a famous automotive manufacturing company in China. Built in 1986, its main products are economy cars. The company also produces automotive fittings, such as car seats, stamping pieces, rubber pieces, redirectors, electrical products and so on. GEELY has contributed significantly a lot to the development of the automotive industry in China. In this study, we consider the scheduling system of the engine machining workshop in Zhe Jiang Linhai. The engine is a combination of two major components and five systems. There are several components some of which are produced by GEELY, and for various reasons, there are many workpieces (pistons, screws and so on) that are manufactured by other companies. In this section we only consider the following six components: the connecting rod, cylinder, cylinder head, left tank of the gearbox, right tank of the gearbox and the fuel pump sub-oil block. The work procedures of these workpieces are approximately the same, and are produced on the same machines. There are three milling machines, three grinders and two lathes in the Linhai city GEELY Haoqing engine workshop.

The working procedures and machines used for each component are as follows:

Connecting rod:

1. Rough mill the small and the large side faces, on milling machine no. 1
2. Precision mill the small and the large side faces, on milling machine no. 2
3. Cut the connecting rod and it's cover, on milling machine no. 3
4. Rough lathe the hole on the small side, on lathe no. 1
5. Precision lathe the hole on the small side, on lathe no. 2
6. Half rough lathe the hole on the large side, on lathe no. 3
7. Grind the small location side, on grinder no. 1
8. Grind the hole of the large side, on grinder no. 2.

Cylinder:

1. Rough mill the upper side face, on milling machine no. 1
2. Rough mill the lower side face, on milling machine no. 2
3. Precision mill the upper side face, on milling machine no. 3
4. Rough lathe the hole on the main axes, on lathe no. 1
5. Half precision lathe the hole on the main axes, on lathe no. 2

6. Precision lathe the hole on the main axes, on lathe no. 3
7. Grind the lower side, on grinder no. 1
8. Grind the front and back sides, on grinder no. 2.

Cover of cylinder:

1. Rough mill the upper side face, on milling machine no. 1
2. Rough mill the lower side face, on milling machine no. 2
3. Precision mill the upper side face, on milling machine no. 3
4. Rough lathe the hole on the valve seat, on lathe no. 1
5. Precision lathe the hole on the valve seat, on lathe no. 2
6. Rough lathe the hole on the camshaft, on lathe no. 3
7. Grind the upper side, on grinder no. 1
8. Grind the input exhaust side, on grinder no. 2

Left tank of the gear box:

1. Rough mill the combining surface, on milling machine no. 1
2. Rough and precision mill the right convex surface, on milling machine no. 2
3. Precision mill the combining surface, on milling machine no. 3
4. Rough lathe the hole on the input axes, on lathe no. 1
5. Rough lathe the hole on the output axes, on lathe no. 2
6. Precision lathe the hole on the input and output axes, on lathe no. 3
7. Rough grind the support plane, on grinder no. 1
8. Precision grind the support plane, on grinder no. 2

Right tank of the gear box:

1. Rough mill the combining surface, on milling machine no. 1
2. Rough and precision mill the left convex surface, on milling machine no. 2
3. Precision mill the combining surface, on milling machine no. 3
4. Rough lathe the hole on the input axes, on lathe no. 1
5. Rough lathe the hole on the output axes, on lathe no. 2
6. Precision lathe the hole on the input and output axes, on lathe no. 3
7. Rough grind the support plane, on grinder no. 1
8. Precision grind the support plane, on grinder no. 2

Fuel pump sub-oil block:

1. Rough mill the small side surface, on milling machine no. 1
2. Rough mill the combining surface, on milling machine no. 2
3. Precision mill the combining surface, on milling machine no. 3
4. Rough lathe the side surface and ladder excircle, on lathe no. 1
5. Precision lathe the side surface and ladder excircle, on lathe no. 2
6. Rough lathe the φ 27 hole, on lathe no. 3
7. Grind the combining surface , on grinder no. 1
8. Precision grind the φ 27 hole, on grinder no. 2

For convenience, we denote each workpiece as follows:

connecting rod: J_1, cylinder: J_2, cylinder head: J_3,
left tank: J_4, right tank: J_5, fuel pump sub-oil block: J_6;

We also denote each machine as follows:

milling machine no.1, no.2, no.3: M_1, M_2, M_3,
grinder no.1, no.2, no.3: M_4, M_5, M_6,
lathe no.1, no.2: M_7, M_8.

It's natural to follow the philosophy of multi-objective expected value modelling, which optimizes some expected objective functions, subject to some expected constraints, to model a Ra-Ra flow shop scheduling problem as a multi-objective expected value model. Then we can reasonably provide the multi-objective expected value model for a Ra-Ra flow shop scheduling problem.

3.6.2 Problem Description and Notation

The flow shop scheduling problem is generally described as follows: a solution to a permutation flow shop problem consists of sequencing N jobs $(J_1, J_2, \ldots, J_N)$ on M machines $(M_1, M_2, \ldots, M_M)$. A job J_i (for $i = 1, \ldots, N$) has at most M operations $(O_{i,1}, O_{i,2}, \ldots, O_{i,M})$. The operation $O_{k,i}$ must be processed on machine M_k without interruption in a $P_{k,j}$ unit of time. These times are fixed and nonnegative, some of which are set to zero if not processed on a machine. Two operations of the same job cannot be executed simultaneously. Every machine can process at most one job at a time and every job has to be processed at most once on each machine following the same sequence $1, 2, \ldots, M$. The flow shop scheduling problem is shown in Fig. 3.12.

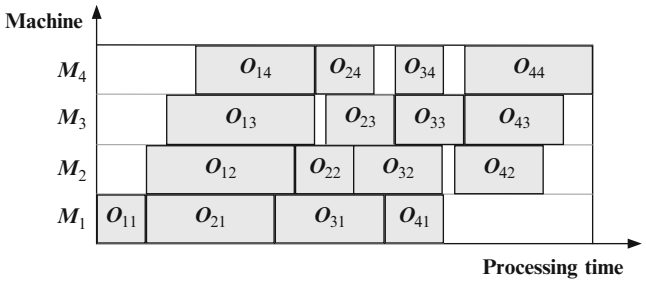


Fig. 3.12 Flow shop scheduling problem with 4 machines and 4 jobs

Based on the above considerations of the flow shop scheduling problem, a mathematical model is developed based on the following assumptions:

1. The job sequence is the same on each machine
2. A machine can only process one job at a time
3. Only one job can be processed on one machine at one time
4. The processing times are Ra-Ra variables, and accordingly the completion time and the earliness time are also Ra-Ra variables

The following notations are used to describe the flow shop scheduling problem.

Parameters:

N: the number of jobs to be processed.

M: the number of machines.

$\bar{P}_{k,j}$: the processing time of job j on machine k.

$\bar{C}_{k,j}$: the completion time of the job at the jth position in the job sequence on machine k.

$\bar{E}_{k,j}$: the earliness time of the job at the jth position in the job sequence on machine k.

$\bar{T}_{k,j}$: the tardiness time of the job at the jth position in the job sequence on machine k.

U_j: denotes whether the job at the jth position is tardy ($U_j = 1$) or not ($U_j = 0$).

$\bar{I}_k$: denotes the idleness time on machine k.

w_j: the importance factor related to job j.

D_j: the due date of job j.

Decision variables:

$$Z_{i,j} = \begin{cases} 1, \text{ if job } i \text{ is in the sequence } j \\ 0, \text{ otherwise} \end{cases}$$

3.6.3 Modelling and Analysis

In the proposed model, the objective is to seek a schedule to both minimize the weighted mean completion time and the weighted mean earliness of the manufacturing system.

Weighted mean completion time:

$$min\ F_1 = \frac{1}{\sum_{j=1}^{N} w_j} \sum_{j=1}^{N} \sum_{k=1}^{M} w_j \bar{C}_{k,j} \tag{3.141}$$

Weighted mean earliness time:

$$min\ F_2 = \frac{1}{\sum_{j=1}^{N} w_j} \sum_{j=1}^{N} \sum_{k=1}^{M} w_j \bar{E}_{k,j} \tag{3.142}$$

In the real-world of flow shop scheduling in the decision making process of solving problems, environmental coefficients and operation parameters are typically uncertain because some information is incomplete or unobtainable in a reasonable time period. Accordingly, (3.141) and (3.142) are Ra-Ra and these conflicting goals are required to be simultaneously optimized.

The objectives are subjected to the following constraints.

$$\sum_{j=1}^{N} Z_{i,j} = 1,\ i = 1, 2, \ldots, N \tag{3.143}$$

Constraints (3.143) ensure that each job is assigned to one position in the sequence.

$$\sum_{i=1}^{N} Z_{i,j} = 1,\ j = 1, 2, \ldots, N \tag{3.144}$$

Constraints (3.144) specify that each position is assigned to one and only one job.

$$\bar{C}_{k,j} + \sum_{j=1}^{N} \bar{P}_{k,i} Z_{i,j+1} = \bar{C}_{k,j+1},\ k = 1, 2, \ldots, M; j = 1, 2, \ldots, N-1 \tag{3.145}$$

Constraints (3.145) ensure that the completion time of the job in position $j + 1$ on machine k is equal to the sum of the completion time of the job in the position j on machine k and the processing time of the job in position $j + 1$ on machine k. No machine idle time is allowed after running until the last job is processed.

$$\bar{C}_{k,j} + \sum_{j=1}^{N} \bar{P}_{k+1,i} Z_{i,j} \leq \bar{C}_{k,j+1},\ k = 1, 2, \ldots, M-1; j = 1, 2, \ldots, N-1 \tag{3.146}$$

Constraints (3.146) show that the completion time of the job in position j on machine $k + 1$ is at least equal to the sum of the completion time of the job in position j of machine k and the processing time of the job in position j on machine $k + 1$; a job can wait between two machines.

$$\sum_{i=1}^{N} \bar{P}_{1,i} Z_{i,1} = \bar{C}_{11} \tag{3.147}$$

Constraints (3.147) specify that the completion time of the first job on the first machine is equal to its processing time on the machine.

$$\bar{E}_{k,j} = max\{0, \sum_{i=1}^{N}[(D_j - \bar{C}_{k,j})Z_{i,j}]\}\ k = 1, 2, \ldots, M; j = 1, 2, \ldots, N-1 \tag{3.148}$$

Constraint (3.148) specifies the earliness of each job on each machine.

$$Z_{i,j} \in [0, 1]\ \forall i, j \tag{3.149}$$

$$\bar{E}_{k,j}, \bar{C}_{k,j}, D_j \geq 0\ \forall k, j \tag{3.150}$$

Constraint (3.149) and (3.150) are logical constraints.

Above all, the Ra-Ra decision-making model for the presented flow shop scheduling problem is as follows:

$$\begin{cases} \min F_1 = \frac{1}{\sum_{j=1}^{N} w_j} \sum_{j=1}^{N} \sum_{k=1}^{M} w_j \bar{C}_{k,j} \\ \min F_2 = \frac{1}{\sum_{j=1}^{N} w_j} \sum_{j=1}^{N} \sum_{k=1}^{M} w_j \bar{E}_{k,j} \\ \text{s.t.} \begin{cases} \sum_{j=1}^{N} Z_{i,j} = 1,\ i = 1, 2, \ldots, N \\ \sum_{i=1}^{N} Z_{i,j} = 1,\ j = 1, 2, \ldots, N \\ C_{k,j} + \sum_{j=1}^{N} \bar{P}_{k,i} Z_{i,j+1} = \bar{C}_{k,j+1},\ k = 1, 2, \ldots, M; j = 1, 2, \ldots, N-1 \\ C_{k,j} + \sum_{j=1}^{N} \bar{P}_{k+1,i} Z_{i,j} \leq \bar{C}_{k,j+1},\ k = 1, 2, \ldots, M-1; j = 1, 2, \ldots, N-1 \\ \sum_{i=1}^{N} \bar{P}_{1,i} Z_{i,1} = \bar{C}_{11} \\ \bar{E}_{k,j} = max\{0, \sum_{i=1}^{N}[(D_j - \bar{C}_{k,j})Z_{i,j}]\},\ k = 1, 2, \ldots, M; j = 1, 2, \ldots, N-1 \\ Z_{i,j} \in [0, 1],\ i = 1, 2, \ldots, N; j = 1, 2, \ldots, N \\ \bar{E}_{k,j}, \bar{C}_{k,j}, D_j \geq 0,\ k = 1, 2, \ldots, M; j = 1, 2, \ldots, N \end{cases} \end{cases} \tag{3.151}$$

Then we transform the above model to a linear model by substituting the following constraints (3.152)–(3.154) for constraint (3.148).

$$\bar{E}^{+}_{k,j} = \sum_{i=1}^{N}[(D_j - \bar{C}_{k,j})Z_{i,j}] \tag{3.152}$$

$$\bar{E}^{-}_{k,j} = \sum_{i=1}^{N}[(\bar{C}_{k,j} - D_j)Z_{i,j}] \tag{3.153}$$

$$\bar{E}^{+}_{k,j}, \bar{E}^{-}_{k,j} \geq 0 \tag{3.154}$$

So we obtain a linear multi-objective expected value model (3.155).

$$\begin{cases} \min F_1 = \frac{1}{\sum_{j=1}^{N} w_j} \sum_{j=1}^{N} \sum_{k=1}^{M} w_j \bar{C}_{k,j} \\ \min F_2 = \frac{1}{\sum_{j=1}^{N} w_j} \sum_{j=1}^{N} \sum_{k=1}^{M} w_j (\bar{E}_{k,j}^{+} + \bar{E}_{k,j}^{-}) \\ \text{s.t.} \begin{cases} \sum_{j=1}^{N} Z_{i,j} = 1, \ i = 1, 2, \ldots, N \\ \sum_{i=1}^{N} Z_{i,j} = 1, \ j = 1, 2, \ldots, N \\ C_{k,j} + \sum_{j=1}^{N} \bar{P}_{k,i} Z_{i,j+1} = \bar{C}_{k,j+1}, \ k = 1, 2, \ldots, M; j = 1, 2, \ldots, N-1 \\ C_{k,j} + \sum_{j=1}^{N} \bar{P}_{k+1,i} Z_{i,j} \le \bar{C}_{k,j+1}, \ k = 1, 2, \ldots, M-1; j = 1, 2, \ldots, N-1 \\ \sum_{i=1}^{N} \bar{P}_{1,i} Z_{i,1} = \bar{C}_{11} \\ \bar{E}_{k,j}^{+} = \sum_{i=1}^{N} [(D_j - \bar{C}_{k,j}) Z_{i,j}], \ k = 1, 2, \ldots, M; j = 1, 2, \ldots, N-1 \\ \bar{E}_{k,j}^{-} = \sum_{i=1}^{N} [(\bar{C}_{k,j} - D_j) Z_{i,j}], \ k = 1, 2, \ldots, M; j = 1, 2, \ldots, N-1 \\ Z_{i,j} \in [0, 1], \ i = 1, 2, \ldots, N; j = 1, 2, \ldots, N \\ \bar{E}_{k,j}^{+}, \bar{E}_{k,j}^{-}, \bar{C}_{k,j}, D_j \ge 0, \ k = 1, 2, \ldots, M; j = 1, 2, \ldots, N. \end{cases} \end{cases} \tag{3.155}$$

After statistical analysis, we have summarized the processing time in Table 3.11; the due time and importance weight are shown in Table 3.12.

The particle swarm optimization algorithm has the following important aspects:

1. Coding: we use an order-based coding method and use sequence to code a particles, if there are eight jobs, the processing sequence is J_1, J_3, J_4, J_5, J_6, J_8, J_2, J_7, so the particle is $\{1, 3, 4, 5, 6, 8, 2, 7\}$.
2. Initialization: first we randomly generate some particles, then we choose the best particle from them, then we repeat this progress until we obtain the predetermined population.
3. Acceleration for this particle, several different genes are chosen partially mapped crossover (PMX) and ordered crossover (OX) are adopted to generate the offsprings as shown in Fig. 3.13, then we regard the offsprings as new parents and use OX to obtain the new offsprings, as shown in Figs. 3.14 and 3.15.

We implemented the APSO using Visual C++ language and ran it on Pentium 4 processor, 2.40 GHz clock pulse with 1024 MB memory, and tested the performance of this method with actual data from the workshop of GEELY Haoqing company in Linhai, China. A run of the Ra-Ra simulation-based PSO (2000 cycles in Ra-Ra simulation, 300 generations in PSO) shows that the result of the optimal schedule is

$$J_6 \to J_3 \to J_5 \to J_2 \to J_1 \to J_4.$$

Table 3.11 Processing time of every workpiece on each machine

Workpieces	Processing time (minute)			
	M_1	M_2	M_3	M_4
J_1	$N(\mu, 0.1)$, with $\mu \sim U(3.9, 4.1)$	$N(\mu, 0.1)$, with $\mu \sim U(3.95, 4.05)$	$N(\mu, 0.1)$, with $\mu \sim U(2.9, 3.1)$	$N(\mu, 0.1)$, with $\mu \sim U(2.9, 3.1)$
J_2	$N(\mu, 0.1)$, with $\mu \sim U(2.45, 2.55)$	$N(\mu, 0.1)$, with $\mu \sim U(2.45, 2.55)$	$N(\mu, 0.1)$, with $\mu \sim U(2.9, 3.1)$	$N(\mu, 0.1)$, with $\mu \sim U(3.9, 4.1)$
J_3	$N(\mu, 0.1)$, with $\mu \sim U(2.45, 2.55)$	$N(\mu, 0.1)$, with $\mu \sim U(2.45, 2.55)$	$N(\mu, 0.1)$, with $\mu \sim U(2.9, 3.1)$	$N(\mu, 0.1)$, with $\mu \sim U(3.9, 4.1)$
J_4	$N(\mu, 0.1)$, with $\mu \sim U(2.45, 2.55)$	$N(\mu, 0.1)$, with $\mu \sim U(1.45, 1.55)$	$N(\mu, 0.1)$, with $\mu \sim U(2.9, 3.1)$	$N(\mu, 0.1)$, with $\mu \sim U(1.95, 2.05)$
J_5	$N(\mu, 0.1)$, with $\mu \sim U(2.45, 2.55)$	$N(\mu, 0.1)$, with $\mu \sim U(1.45, 1.55)$	$N(\mu, 0.1)$, with $\mu \sim U(2.9, 3.1)$	$N(\mu, 0.1)$, with $\mu \sim U(1.95, 2.05)$
J_6	$N(\mu, 0.1)$, with $\mu \sim U(1.45, 1.55)$	$N(\mu, 0.1)$, with $\mu \sim U(1.95, 2.05)$	$N(\mu, 0.1)$, with $\mu \sim U(1.95, 2.05)$	$N(\mu, 0.1)$, with $\mu \sim U(1.95, 2.05)$
Workpieces	Processing time (minute)			
	M_5	M_6	M_7	M_8
J_1	$N(\mu, 0.1)$, with $\mu \sim U(3.9, 4.1)$	$N(\mu, 0.1)$, with $\mu \sim U(3.9, 4.1)$	$N(\mu, 0.1)$, with $\mu \sim U(2.95, 3.05)$	$N(\mu, 0.1)$, with $\mu \sim U(3.9, 4.1)$
J_2	$N(\mu, 0.1)$, with $\mu \sim U(3.9, 4.1)$	$N(\mu, 0.1)$, with $\mu \sim U(5.9, 6.1)$	$N(\mu, 0.1)$, with $\mu \sim U(3.9, 4.1)$	$N(\mu, 0.1)$, with $\mu \sim U(3.9, 4.1)$
J_3	$N(\mu, 0.1)$, with $\mu \sim U(3.9, 4.1)$	$N(\mu, 0.1)$, with $\mu \sim U(2.95, 3.05)$	$N(\mu, 0.1)$, with $\mu \sim U(2.9, 3.1)$	$N(\mu, 0.1)$, with $\mu \sim U(4.9, 5.1)$
J_4	$N(\mu, 0.1)$, with $\mu \sim U(1.95, 2.05)$	$N(\mu, 0.1)$, with $\mu \sim U(2.45, 2.55)$	$N(\mu, 0.1)$, with $\mu \sim U(1.95, 2.05)$	$N(\mu, 0.1)$, with $\mu \sim U(1.95, 2.05)$
J_5	$N(\mu, 0.1)$, with $\mu \sim U(1.95, 2.05)$	$N(\mu, 0.1)$, with $\mu \sim U(2.45, 2.55)$	$N(\mu, 0.1)$, with $\mu \sim U(1.95, 2.05)$	$N(\mu, 0.1)$, with $\mu \sim U(1.95, 2.05)$
J_6	$N(\mu, 0.1)$, with $\mu \sim U(2.45, 2.55)$	$N(\mu, 0.1)$, with $\mu \sim U(1.95, 2.05)$	$N(\mu, 0.1)$, with $\mu \sim U(1.45, 1.55)$	$N(\mu, 0.1)$, with $\mu \sim U(2.9, 3.1)$

Table 3.12 Due time and the importance factor of the six jobs

Jobs	J_1	J_2	J_3	J_4	J_5	J_6
Due time	40	40	30	40	30	20
Importance	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$

We then obtain the optimal sequence of the makespan $C_{\max} = 43.5$ min, the weighted completion time $C = 33$ min and the weighted earliness time $E = 1.25$ min. And we drew a Gantt Chart as in Fig. 3.16 according to the results.

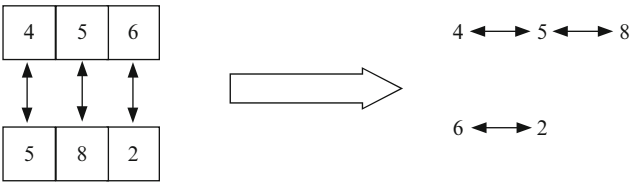
In the GEELY Linhai workshop, the scheduling of the six selected workpieces with the same working procedures are: fuel pump sub-oil block, cylinder head, right tank, cylinder, connecting rod, and right tank; it will take 33 min to complete all of these workpieces which results in the decision maker being satisfied.

To check the efficiency and the effectiveness of the APSO, we also present, under deterministic conditions, the APSO and Tribe-PSO by using the same program language with the same PSO paraments. In this paper, 10 problems that were

Parent 1 1, 2, 3, 4, 5, 6, 7, 8

Parent 2 1, 3, 4, 5, 8, 3, 7, 6

Mapping relations:



Offsrpings generated by PMX

Offsprings 1 1, 6, 3, 5, 8, 2, 7, 4

Offsprings 2 1, 3, 8, 4, 5, 6, 7, 2

Fig. 3.13 Illustration of partially mapped crossover

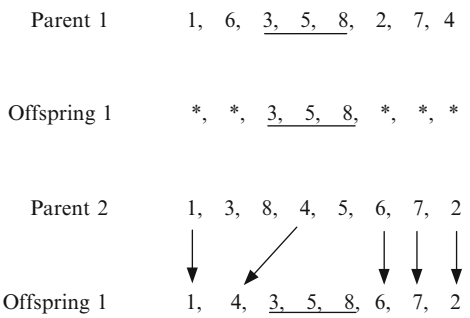


Fig. 3.14 Illustration of generating offspring 1 by OX

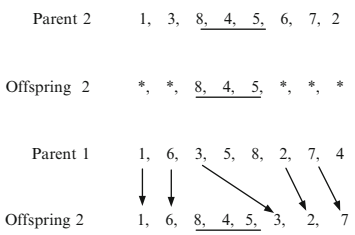


Fig. 3.15 Illustration of generating offspring 2 by OX

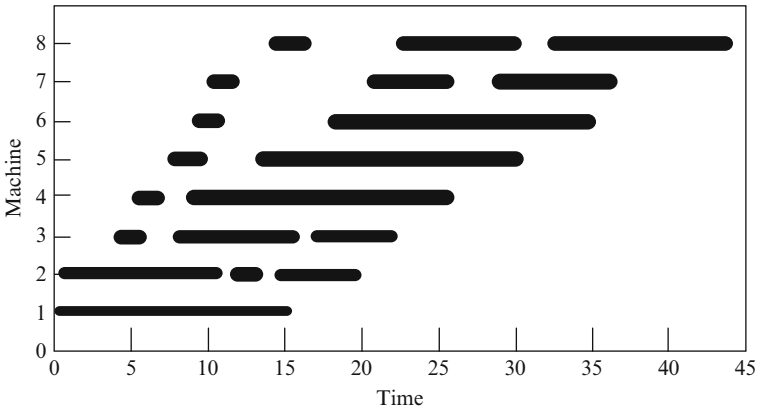


Fig. 3.16 Gantt chart

Table 3.13 Statistical results of 2 testing algorithm

No.	Problem	N, M	C_{max}	APSO			Tribe-PSO	
				BRE	ARE	WRE	BRE	ARE
1	Car1	11,5	7038	0	0	0	0	0.28
2	Car2	13,4	7170	0	0	0	0	4.03
3	Car4	14,4	8003	0	0	0	0	2.31
4	Car5	10,6	7724	0	0	0	0	1.45
5	Car7	7,7	6590	0	0	0	0	1.57
6	Rec1	20,5	1245	0	0.05	0.18	2.81	6.96
7	Rec13	20,15	1942	0.26	1.08	1.66	3.68	5.94
8	Rec21	30,10	2019	1.44	1.64	3.17	3.42	6.08
9	Rec27	30,15	2373	0.96	2.09	3.58	4.92	6.85
10	Rec41	75,20	4961	2.30	3.43	4.69	7.44	8.92

BRE: the relative error of the best result obtained to C_{max}
ARE: the relative error of the average result obtained to C_{max}
WRE: the relative error of the worst result obtained to C_{max}

contributed to the OR-Library by Mattfeld and Vaessens are selected. The first five problems were called car1, car2, car4, car5, car7, respectively by Carlier [36]. The other five problems were called rec01, rec13, rec21, rec27, rec41, respectively by Reeves [257], who used them to compare the performances of APSO and Tribe-PSO. All these problems can be downloaded from http://mscmga.ms.ic.ac.uk. Thus far these problems have been used as benchmarks with different methods by many researchers.

Based on the implementation discussed in the above section, ten testing problems are carried out for the Tribe-PSO and APSO, the statistical results are summarized in Table 3.13.

From Table 3.13, it can obviously be concluded that the APSO provides better optimization performance for flow shop scheduling problems. The results obtained by the APSO are close to being the best results available. Compared with the

Tribe-PSO, not only can the APSO achieve much better results, but its performance is also very stable as its BRE is always very close to the corresponding WRE. Especially when solving larger scale problems, the average performance of the APSO can be better than the Tribe-PSO. Thus, it can be concluded that the APSO is superior to the Tribe-PSO with respect to optimization quality and stability. It can also be concluded that the APSO can achieve the same performance as a traditional PSO with a much reduced computational effort.

The effectiveness of the APSO can be attributed to the order-based coding, the initialization process, the hybrid acceleration operator which combines PMX and OX, the INV mutation operators, and the self-learning evolution mechanism of PSO.

Chapter 4
Random Fuzzy Multiple Objective Decision Making

Since the fuzzy set was initialized by Zadeh [356], it has been applied to many fields. Later, many scholars proposed the concept of two-fold uncertain variables, combined fuzzy variables, and random variables. Two different definitions from different perspectives have been proposed. The first comes from Kwakernaak [180], who coined the term "fuzzy random variable" and who regarded that a fuzzy random variable is a random variable whose value is not real, but fuzzy number. Since Kruse and Meyer [176] worked on an expanded version of a similar model, they are often mentioned with as Kwakernaak. The second comes from Puri and Ralescu [256] who regarded fuzzy random variables as random fuzzy sets. Once again, because of Klement et al. [162], and other collaborations, these three authors often are jointly credited with the second definition. In this chapter, we mainly take Kwakernaak's point and regard it as a random fuzzy variable (Abbr. Ra-Fu variable) which has been renamed by some scholars [32, 209, 236, 341] in order to avoid the confusion. It has been widely extended to many fields. Xu and Liu [341] discussed a class of supply chain networks optimal problems with random fuzzy shipping costs and customer demand, and proposed a random fuzzy multi-objective mixed-integer nonlinear programming model to gain an optimal strategy. Xu and He [339] provided an auxiliary programming model of the random fuzzy programming, converted it to a deterministic mixed 0–1 integer programming model, of which the solutions are proved to exist and showed its efficiency by application to a supply chain problem. Zhou and Xu [363] discussed a class of integrated logistics network models under random fuzzy environments and applied them to a Chinese beer company.

This chapter mainly considers some multi-objective programming problems under a random fuzzy environment. We first propose two kinds of random fuzzy variables, i.e., discrete random fuzzy variables and continuous random fuzzy variables, and introduce some special examples. After introducing those basic concepts and properties, three parts are presented respectively from different viewpoints:

1. Random fuzzy expected value model (abbr. Ra-Fu EVM). Usually, decision makers find it is difficult to make a decision when they encounter uncertain parameter. A clear criteria must be brought forward to assist in the decision. The expected value operator of random fuzzy variables is introduced and the crisp equivalent model is deduced when the distribution is clear.

J. Xu and L. Yao, *Random-Like Multiple Objective Decision Making*, Lecture Notes in Economics and Mathematical Systems 647, DOI 10.1007/978-3-642-18000-2_4,

2. Random fuzzy chance-constraint model (abbr. Ra-Fu CCM). Sometimes, decision makers don't strictly require the objective value to be the maximal benefit but only want to obtain the maximum benefit under a predetermined confidence level. Then the chance constrained model is proposed and the crisp equivalent model is deduced when the distribution is clear.
3. Random fuzzy dependent-chance model (abbr. Ra-Fu DCM). When decision makers predetermine an objective value and require the maximal probability that objective values exceed the predetermined one.

Finally, an application to the supply chain network problem is presented to show the effectiveness of the above three models. Readers can refer to the following more details.

4.1 Supply Chain Network Design with Ra-Fu Phenomena

In recent years, the supply chain network (abbr. SCN) design problem has been gaining importance due to increasing competitiveness introduced by market globalization. A supply chain, beginning with the production of raw material by a supplier and ending with the consumption of a product by the customer, is a set of suppliers, facilities, products, customers and of controlling inventory, purchasing, and distribution. Traditionally, marketing, distribution, planning, manufacturing, and purchasing organizations along the supply chain operate independently. These organizations have their own objectives and these objectives are often conflicting. But, there is a need for a mechanism through which these different functions can be integrated [5]. Supply chain management (SCM) (see Fig. 4.1), which appeared in the early 1990s and involved planning and managing production/manufacturing,

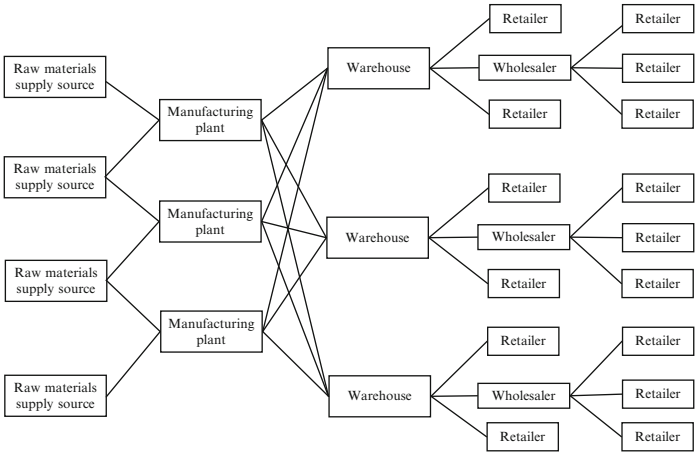


Fig. 4.1 Supply chain network

transportation and distribution, offers a strategy through which such integration can be achieved and a way to improve industrial environments [283].

SCN design problems cover a wide range of formulations ranging from simple single product types to complex multi-product types, and from linear deterministic models to complex non-linear uncertain models. The network design problem is one of the most comprehensive strategic decision problems that needs to be optimized for the long-term efficient operation of the whole supply chain. It determines the number, location, capacity and type of plants, warehouses, and distribution centers (DCs) to be used. It also establishes distribution channels, and the amount of materials and items to be consumed, produced, and shiped from suppliers to customers. In the literature, different studies tackle the design problems of supply networks and these studies have been surveyed in Vidal and Goetschalckx [326], Beamon [19], Erenguc et al. [94], Shen et al. [286], Ko et al. [166] and Romeijn et al. [264]. However, the majority of this research assumes that the operational characteristics and design parameters of the supply chain are deterministic. Unfortunately, real-world situations are often not deterministic, and some factors such as demand, allocation, cost of shipments, even location of customers and facilities are usually changing, hence we need to consider the supply chain network design problem under an uncertain environment. A number of research papers on stochastic parameters involving distribution of raw materials and products and the facility location of the supply chain have been done at the strategic and tactical levels [81,355]. Beginning with the seminal work of Geoffrion and Graves [112] on multi-commodity distribution system design, a large number of optimization-based approaches have been proposed for the design of supply chain networks [9, 53]. While reviewing the literatures, we find that there are three methods of solving supply chain design network uncertain problems: Benders decomposition algorithm, branch-and-fix heuristic, and the hybrid approach. MirHassani et al. [221], Tsiakis et al. [322], Choudhary et al. [56] and Santoso et al. [276] proposed the Benders decomposition algorithm (or modification of the Benders decomposition algorithm) which is commonly used for deterministic network design problems to generate robust designs, and they created network configurations that are good (nearly optimal) for a variety of scenarios of the design parameters. Alonso–Ayuso and Escudero et al. [4] proposed an approach of branch-and-fix heuristic for solving two-stage stochastic supply chain Computational results on networks involving 6 plants, 12 products, 24 markets, and 23 scenarios were presented. Chan, et al. [42], Chan and Chung [43], and Chen and Lee [49] developed a hybrid approach based on genetic algorithm and Analytic Hierarchy Process (AHP) (or two-phase fuzzy decision-making method) for production and distribution problems in multi-factory, multi-product, multi-stage, and multi-period scheduling SCN with uncertain market demands and product prices. As to the fuzziness, since Zadeh's pioneering work [356], the fuzzy sets theory has been applied to different management problems. Many successful applications of the theory in the area of fuzzy optimization can be found in the literature. Zimmermann [364, 365], Chen and Lee [49], Amid et al. [6], Kulak et al. [178] and Wang [331] proposed different methods for solving fuzzy multi-objective linear programming problems.

Unfortunately, the SCN design problem is subject to many sources of uncertainty besides random uncertainty and fuzzy uncertainty [137]. In a practical decision-making process, we often face a hybrid of uncertain environments. To deal with this twofold uncertainty, fuzzy random variable was proposed by Kwakernaak [180,181] to depict the phenomena in which fuzziness and randomness appear simultaneously [96, 256, 325]. Several research works have been published in recent years [210, 318]. However, we consider the amount of demand on the products as a normally distributed variable $\mathcal{N}(\mu, \sigma^2)$ from the view point of probability theory, and the values of μ as a triangular fuzzy variable (a, b, c) because of the lack of data to analyze. Therefore, probability SCN with fuzzy parameters appears. In this case, the Ra-Fu variable which was presented by Buckley [32] can be used to deal with this kind of combined uncertainty of randomness and fuzziness. How to model and solve the problem of SCN design in random fuzzy environment is a new area of research interest. To the best of the authors knowledge, so far, there is little research in this area. In this chapter, we focus on these conflicts and discuss the random fuzzy multiobjective decision making and its application to the supply chain network.

4.2 Random Fuzzy Variable

Before introducing the concept of Ra-Fu variables, let's recall some definitions and properties of the fuzzy set and fuzzy variable.

4.2.1 Fuzzy Variable

Fuzzy set theory is developed for solving problems in which descriptions of activities and observations are imprecise, vague, and uncertain. The term "fuzzy" refers to the situation in which there are no well defined boundaries of the set of activities or observations to which the descriptions apply. For example, one can easily assign a person seven feet tall to the "class of tall men". But it would be difficult to justify the inclusion or exclusion of a 173 cm tall person to that class, because the term "tall" does not constitute a well defined boundary. This notion of fuzziness exists almost everywhere in our daily life, such as the "class of red flowers," the "class of good shooters," the "class of comfortable speeds for traveling, the "numbers close to 10," etc. These classes of objects cannot be well represented by classical set theory. In classical set theory, an object is either in a set or not in a set. An object cannot partially belong to a set.

To cope with this difficulty, Zadeh [356] proposed the fuzzy set theory in 1965. A fuzzy set is a class of objects with a continuum of membership grades. A membership function, which assigns to each object a grade of membership, is associated with each fuzzy set. Usually, the membership grade are in [0, 1]. When the grade of membership for an object in a set is one, this object is absolutely in that set; when the

grade of membership is zero, the object is absolutely not in that set. Borderline cases are assigned numbers between zero and one. Precise membership grades do not convey any absolute significance. They are context-dependent can be subjectively assessed.

Let U be the universe which is a classical set of object, and the generic elements are denoted by x. The membership in a crisp subset of A is often viewed as characteristic function π_A from U to $\{0, 1\}$ such that:

$$\pi_A(x) = \begin{cases} 1 \text{ iff } x \in A \\ 0 \text{ otherwise} \end{cases} \tag{4.1}$$

where $\{0, 1\}$ is called a valuation set.

If the valuation set is allowed to be the real interval $[0, 1]$, A is called a fuzzy set. $\mu_A(x)$ is the degree of membership of x in fuzzy set A. The closer the value of $\mu_A(x)$ is to 1, the more x belongs to A. Therefore, A is characterized by the set of ordered pairs:

$$A = \{(x, \mu_A(x))|x \in U\} \tag{4.2}$$

Sometimes, we might only need objects of a fuzzy set instead of its characteristic function, that is, to transfer a fuzzy set into a crisp set. In order to do so, we need two concepts, support and α-cut.

It is often necessary to consider those elements in a fuzzy set which have non-zero membership grades. These element are the support of that fuzzy set.

Definition 4.1. (Zadeh [357]) Given a fuzzy set A, its support $S(A)$ is an ordinary crisp subset on U defined as

$$S(A) = \{x|\mu_A(x) > 0 \text{ and } x \in U\} \tag{4.3}$$

Definition 4.2. (Zadeh [357]) Given a fuzzy set A, its α-cut A_α defined as

$$A_\alpha = \{x|\mu_A(x) \geq \alpha \text{ and } x \in U\} \tag{4.4}$$

where, α is the confidence level.

It is obviously that the α-cut of a fuzzy set A is an ordinary crisp subset whose elements belongs to fuzzy set A – at least to the degree of α. That is, for fuzzy set A its α-cut is defining as (4.4). The α-cut is a more general case of the support of a fuzzy set, when $\alpha = 0$, $A_\alpha = \text{supp}(A)$.

The term fuzzy number is used to handle imprecise numerical quantities, such as "close to 10," "about 60," "several," etc. A general definition of a fuzzy number is given by Dubois and Prade [84, 85]: any fuzzy subset $M = \{(x, \mu(x))\}$ where x takes its number on the real line R and $\mu_M(x) \in [0, 1]$.

Definition 4.3. (Dubois and Prade [84]) Let A be a fuzzy set, its membership function is $\mu_A : R \to [0, 1]$, if

i. A is upper semi-continuous, i.e., α-cut A_α is close set, for $0 < \alpha \leq 1$.
ii. A is normal, i.e., $A_1 \neq \emptyset$.

iii. A is convex, i.e., A_α is a convex subset of R, for $0 < \alpha \leq 1$.
iv. The closed convex hull of A $A_0 = cl[co\{x \in R, \mu_A(x) > 0\}]$ is cored.

then A is a fuzzy number.

By Definition 4.3, the α-cut A_α of the fuzzy number A is actually a close interval of the real number field, that is,

$$A_\alpha = \{x \in R | \mu_A(x) \geq \alpha\} = [A_\alpha^{\mathrm{L}}, A_\alpha^{\mathrm{R}}], \quad \alpha \in [0, 1],$$

where A_α^{L} and A_α^{R} are the left and the right extreme points of the close interval.

Example 4.1. Given fuzzy number A with membership function

$$\mu_{\widetilde{A}}(x) = \begin{cases} L\left(\dfrac{a-x}{l}\right), & \text{if } a - l \leq x < a, l > 0 \\ 1 & \text{if } x = a \\ R\left(\dfrac{x-a}{r}\right), & \text{if } a < x \leq a + r, r > 0 \end{cases}$$

and the basis functions $L(x)$, $R(x)$ are continuous un-increasing functions, and $L, R : [0, 1] \to [0, 1]$, $L(0) = R(0) = 1$, $L(1) = R(1) = 0$, then $\widetilde{A}$ is LR fuzzy number, denoted by $\widetilde{A} = (a, l, r)_{LR}$, where a is the central value of $\widetilde{A}$, $l, r > 0$ is the left and the right spread. α-cut A_α of the LR fuzzy number $\widetilde{A}$ is

$$A_\alpha = [A_\alpha^L, A_\alpha^R] = [a - L^{-1}(\alpha)l, a + R^{-1}(\alpha)r], \quad \alpha \in [0, 1].$$

After that, the concept of fuzzy variable was proposed. Let's introduce the basic knowledge about fuzzy variable, which including the measure, the definition and the properties of fuzzy random variables. We give some basic knowledge about fuzzy variable. Since its introduction in 1965 by Zadeh [356], fuzzy set theory has been well developed and applied in a wide variety of real problems. The term fuzzy variable was first introduced by Kaufmann [154], then it appeared in Zadeh [358, 359] and Nahmias [231]. Possibility theory was proposed by Zadeh [358], and developed by many researchers such as Dubois and Prade [84].

In order to provide an axiomatic theory to describe fuzziness, Nahmias [231] suggested a theoretical framework. Let us give the definition of possibility space (also called pattern space by Nahmias).

Definition 4.4. (Dubois and Prade [84]) Let Θ be a nonempty set, and $P(\Theta)$ be the power set of Θ. For each $A \subseteq P(\Theta)$, there is a nonnegative number $Pos\{A\}$, called its possibility, such that

i. $Pos\{\emptyset\} = 0$;
ii. $Pos\{\Theta\} = 1$;
iii. $Pos\{\bigcup_k A_k\} = \sup_k Pos\{A_k\}$ for any arbitrary collection $\{A_k\}$ in $P(\Theta)$.

The triplet $(\Theta, P(\Theta), Pos)$ is called a possibility space, and the function Pos is referred to as a possibility measure.

It is easy to obtain the following properties of *Pos* from the axioms above.

Property 4.1. The properties of *Pos* measure:

(i) $0 \leq Pos\{A\} \leq 1, \forall A \in P(\Theta)$;
(ii) $Pos\{A\} \leq Pos\{B\}$, if $A \subseteq B$

Several researchers have defined fuzzy variable in different ways, such as Kaufman [154], Zadah [358, 359] and Nahmias [231]. In this book we use the following definition of fuzzy variable.

Definition 4.5. (Nahmias [231]) A fuzzy variable is defined as a function from the possibility space $(\Theta, P(\Theta), Pos)$ to the real line $\mathbf{R}$.

Definition 4.6. (Dubois and Prade [84]) Let ξ be a fuzzy variable on the possibility space $(\Theta, P(\Theta), Pos)$. Then its membership function $\mu : \mathbf{R} \mapsto [0, 1]$ is derived from the possibility measure *Pos* by

$$\mu(x) = Pos\{\theta \in \Theta | \xi(\theta) = x\} \tag{4.5}$$

In order to measure the chances of occurrence of fuzzy events, the possibility and necessity of a fuzzy event is given as follows.

Definition 4.7. [87] Let $\tilde{a}_1, \tilde{a}_2, \ldots, \tilde{a}_n$ be fuzzy variables, and $f : \mathbf{R}^n \to \mathbf{R}$ be continuous functions. Then the possibility of the fuzzy event characterized by $f(\tilde{a}_1, \tilde{a}_2, \ldots, \tilde{a}_n) \leq 0$ is

$$\begin{aligned} &Pos\{f(\tilde{a}_1, \tilde{a}_2, \ldots, \tilde{a}_n) \leq 0\} \\ &= \sup_{x_1, x_2, \ldots, x_n \in R} \left\{ \min_{1 \leq i \leq n} \mu_{\tilde{a}_i}(x_i) | f(x_1, x_2, \ldots, x_n) \leq 0 \right\} \end{aligned} \tag{4.6}$$

The necessity measure of a set A is defined as the impossibility of the opposite set A^c.

Definition 4.8. (Dubois [87]) Let $(\Theta, P(\Theta), Pos)$ be a possibility space, and A be a set in $P(\Theta)$. Then the necessity measure of A is

$$Nec\{A\} = 1 - Pos\{A^c\}.$$

Thus the necessity measure is the dual of possibility measure, that is, $Pos\{A\} + Nec\{A^c\} = 1$ for any $A \in P(\Theta)$.

In order to measure the mean of a fuzzy variable, several researchers defined an expected value for fuzzy variables with different ways, such as Dubois and Prade [86], Campos and Verdegay [34], González [117] and Yager [349, 350]. Readers could refer to them to know the detail, and we don't introduce it here. Next, we will introduce the definition of random fuzzy variable.

4.2.2 Ra-Fu Variable

Consider the following question. A client's retirement savings plan has approximately 100 participants. Several of their portfolios have too much risk. What is the probability that a participant chosen at random is maintaining a portfolio that has too much risk? If the descriptive variables, namely "approximately", "several" and "too much", were crisp numbers, the answer to the question would be a numerical probability. However, since these terms are fuzzy, rather than crisp values, the solution, like the data upon which it is based, is a fuzzy number. Situations of this sort, which involve a function from a possibility space to the set of random variables, give rise to the notion of a Ra-Fu variable. In this section, we mainly refer to these literatures [32, 162, 176, 180, 181, 256].

Definition 4.9. (Buckley [32]) A Ra-Fu variable is a random variable with a fuzzy parameter.

A random variable is a mapping from the probability space Ω to the real space. In fact, the random fuzzy variable is defined as a mapping $\xi : \mathscr{F}(B) \to \Omega$ such that for any $\mu \in \mathscr{F}(B)$, $\xi(\mu)$ is a random variable, where $\mathscr{F}(B)$ is combined by some fuzzy numbers. For example, if $\xi \sim \mathscr{N}(\mu, \sigma^2)$ is a normally distributed random variable, where μ is a fuzzy variable, then ξ is a random fuzzy variable as shown in Fig. 4.2. Hence, it is available to take those like its independence, distribution, expected value, variance, and so on into account.

Example 4.2. Let $\eta_1, \eta_2, \ldots, \eta_m$ be random variables and $u_1, u_2, \ldots, u_m$ be real numbers in [0,1]. Then

$$\xi = \begin{cases} \eta_1, & \text{with possibility } u_1 \\ \eta_2, & \text{with possibility } u_2 \\ \cdots & \\ \eta_m, & \text{with possibility } u_m \end{cases}$$

is clearly a Ra-Fu variable. Is it a function from a possibility space $(\Theta, \mathscr{P}(\Theta), Pos)$ to a collection of random variables $\mathscr{R}$? Yes. For example, we define $\Theta =$

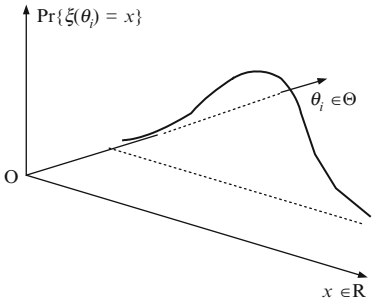


Fig. 4.2 Representation of a Ra-Fu variable

$\{1, 2, \ldots, m\}$, $Pos\{i\} = u_i, i = 1, 2, \ldots, m$, $\mathscr{R} = \{\eta_1, \eta_2, \ldots, \eta_m\}$, and the function is $\xi(i) = \eta_i, i = 1, 2, \ldots, m$.

Example 4.3. If η is a random variable, and $\tilde{a}$ is a fuzzy variable defined on the possibility space $(\Theta, \mathscr{P}(\Theta), Pos)$, then $\xi = \eta + \tilde{a}$ is a fuzzy random variable. In fact, ξ is also a Ra-Fu variable, defined by

$$\xi(\theta) = \eta + \tilde{a}(\theta), \forall \theta \in \Theta$$

Example 4.4. In many statistics problems, the probability distribution is completely known except for the values of one or more parameters. For example, it might be known that the lifetime ξ of a modern engine is an exponentially distributed variable with an unknown mean θ,

$$\phi(x) = \begin{cases} \dfrac{1}{\theta} e^{-x/\theta}, & \text{if } 0 \leq x < \infty \\ 0, & \text{otherwise} \end{cases}$$

Usually, there is some relevant information in practice. It is thus possible to specify an interval in which the value of θ is likely to lie, or to give an approximate estimate of the value of θ. It is typically not possible to determine the value of θ exactly. If the value of θ is provided as a fuzzy variable defined on the possibility space $(\Theta, \mathscr{P}(\Theta), Pos)$, then ξ is a Ra-Fu variable defined as

$$\xi(\theta) \sim exp(\theta), \theta \in \Theta.$$

Example 4.5. Let $\xi \sim \mathscr{N}(\rho, 1)$, where ρ is a fuzzy variable with membership function $\mu_\rho(x) = [1 - |x - 2|] \vee 0$. Then ξ is a Ra-Fu variable taking "normally distributed variable $\mathscr{N}(\rho, 1)$" values.

Remark 4.1. Roughly speaking, if Θ consists of a single element, then the Ra-Fu variable degenerates to a random variable. If $\mathscr{R}$ is a collection of real numbers (rather than random variables), then the Ra-Fu variable degenerates to a fuzzy variable.

4.2.2.1 Discrete Ra-Fu Variable

From the definition of Ra-Fu variable, we know that Ra-Fu variables can be divided into two kinds, one is the discrete Ra-Fu variable, and the other is continuous Ra-Fu variable. There exists a class of special Ra-Fu variables, which are functions from the possibility space $(\Theta, \mathscr{P}(\Theta), Pos)$ to the collection of discrete random variables or a discrete random variable. Next, let's discuss the detail.

Definition 4.10. (Discrete Ra-Fu variable) Let ξ be a Ra-Fu variable on the possibility space $(\Theta, \mathscr{P}(\Theta), Pos)$. If $\xi(\theta)$ is a discrete random variable for any $\theta \in \Theta$, then ξ is said to be a discrete Ra-Fu variable.

Example 4.6. Let ξ be a 0–1 distributed random variable with the probability p of success. Now we assume that p is not known exactly and is to be estimated from a fuzzy space Θ. We substitute a fuzzy number $\tilde{p}$ for p, then ξ is obviously a Ra-Fu variable.

Example 4.7. Let ξ be a Ra-Fu variable on $(\Theta, \mathscr{P}(\Theta), Pos)$, where $\Theta = \{\theta_1, \theta_2\}$, $Pos\{\theta = \theta_1\} = Pos\{\theta = \theta_2\} = 0.5$, and ξ is defined as follows

$$\xi(\theta) = \begin{cases} \eta_1, \text{ with the possibility } 0.5 \\ \eta_2, \text{ with the possibility } 0.5 \end{cases}$$

where η_1 is a binomially distributed random variable and η_2 is a poisson distributed random variable. Obviously, ξ is a discrete Ra-Fu variable on $(\Theta, \mathscr{P}(\Theta), Pos)$.

Then, let's discuss some special Ra-Fu variables in the following part.

The crisp binomial probability function, usually written $b(n, p)$ where n is the number of independent experiments and p is the probability of a "success" in each experiment, has one parameter p. In these experiments let us assume that p is not known precisely and it needs to be estimated, or obtained from expert opinion. So the p value is uncertain and we substitute a fuzzy number $\tilde{p}$ for p to get the Ra-Fu binomial distribution.

Definition 4.11. (Ra-Fu binomial distribution) Let ξ be a discrete Ra-Fu variable on $(\Theta, \mathscr{P}(\Theta), Pos)$, then $\xi(\theta)$ is a random variable for $\theta \in \Theta$. Assume that $\xi(\theta)$ has the binomial distribution with the following probability,

$$Pr\{\xi(\theta) = k\} = \binom{n}{k} \tilde{p}^k \tilde{q}^{n-k}, \ k = 0, 1, \ldots, n \tag{4.7}$$

where $\tilde{p}$ is a fuzzy variable from $(\Theta, \mathscr{P}(\Theta), Pos)$ to (0,1) and $\tilde{q} = 1 - \tilde{p}$. Then ξ is said to be a binomially distributed Ra-Fu variable, denoted by $b(n, \tilde{p})$.

Since $\tilde{p}$ is a fuzzy variable, then $Pr\{\xi(\theta) = k\}$ is a function about the fuzzy parameter $\tilde{p}$. By the fuzzy arithmetic, we know $\binom{n}{k} \tilde{p}^k \tilde{q}^{n-k}$ is a fuzzy variable defined on the product space. Obviously, $\tilde{p}$ and $\tilde{q}$ are two mappings from the possibility space $(\Theta, \mathscr{P}(\Theta), Pos)$ to $(0, 1)$. Then for any $\theta^* \in \theta^*$, $\tilde{p}(\theta^*)$ and $\tilde{q}(\theta^*)$ become two certain probability values. It follows that

$$\sum_{i=1}^{n} \binom{n}{k} \tilde{p}^k(\theta^*) \tilde{q}^{n-k}(\theta^*) = (\tilde{p}(\theta^*) + \tilde{q}(\theta^*))^n = 1$$

then Definition 4.11 is well defined. In (4.7), if $\tilde{p}$ is a fixed number, then $Pr\{\xi(\theta) = k\}(0 < Pr\{\xi(\theta) = k\} < 1)$ is also a fixed number and ξ degenerates to a random variable subject to binomial distribution from $(\Omega, \mathscr{A}, Pr)$ to $\mathbf{R}$.

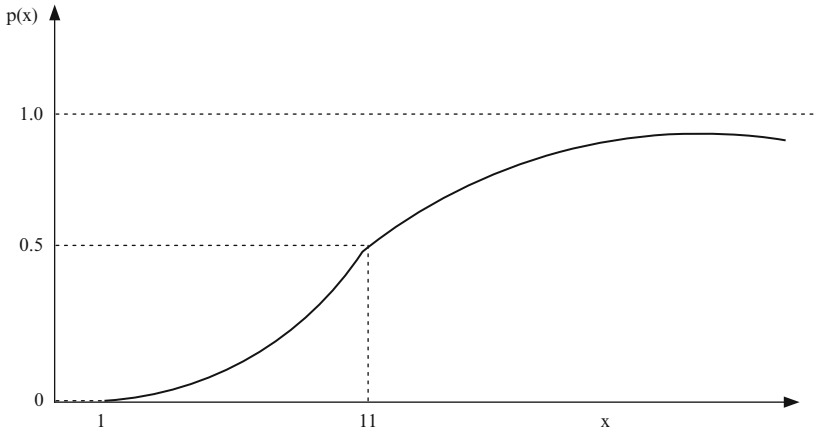


Fig. 4.3 The membership function of fuzzy variable $\tilde{p}$

Table 4.1 The numerical result of $\xi \sim b(20, \tilde{p})$

k	$b(20, \tilde{p})$		k	$b(20, \tilde{p})$	
	$\theta_1 \leq 1$	$\theta_2 = 11$		$\theta_1 \leq 1$	$\theta_2 = 11$
0	–	–	11	–	0.1602
1	–	–	12	–	0.1201
2	–	0.0002	13	–	0.0739
3	–	0.0011	14	–	0.0370
4	–	0.0046	15	–	0.0148
5	–	0.0148	16	–	0.0046
6	–	0.0370	17	–	0.0011
7	–	0.0739	18	–	0.0002
8	–	0.1201	19	–	–
9	–	0.1602	20	–	–
10	–	0.1762			

Example 4.8. Let $\xi \sim b(n, \tilde{p})$ be a Ra-Fu variable, where $\tilde{p}$ is a fuzzy variable on $(\Theta, \mathscr{P}(\Theta), Pos)$ with the following membership function (Fig. 4.3),

$$\mu_{\tilde{p}}(x) = \begin{cases} 0, & x \leq 1 \\ \left(1 + \dfrac{100}{(x-1)^2}\right)^{-1}, & x > 1, \end{cases} \tag{4.8}$$

where $\Theta = (-\infty, \infty)$. Then we get the distribution of ξ, see Table 4.1.

In the part of random poisson, we have known that when n is a large number and p is a small number, the random binomial distribution becomes toward a Poisson distribution. So is the random fuzzy variable.

Definition 4.12. (Ra-Fu Poisson distribution) Let ξ be a random variable having the Poisson probability function. If $Pr\{\xi = k\}$ stands for the probability that $\xi = k$,

then

$$Pr\{\xi = k\} = \frac{\lambda^k e^{-\lambda}}{k!},$$

for $k = 0, 1, 2, 3, \ldots$, and parameter $\lambda > 0$. Now substitute fuzzy number $\tilde{\lambda}$ for λ to produce the random fuzzy Poisson probability function, denoted by $\xi \sim P(\tilde{\lambda})$.

From the definition, we know that $Pr\{\xi(\theta) = k\}$ is a fuzzy function about $\tilde{\lambda}$, where $\tilde{\lambda}$ is a fuzzy variable from the possibility space $(\Theta, \mathscr{P}(\Theta), Pos)$ to $(0, +\infty)$. For any $\theta^* \in \Theta$, $\tilde{\lambda}(\theta^*)$ is a real number in $(0, +\infty)$, then

$$\sum_{k=0}^{\infty} \frac{\tilde{\lambda}(\theta^*)^k}{k!} e^{-\tilde{\lambda}(\theta^*)} = e^{\tilde{\lambda}(\theta^*)} \cdot e^{-\tilde{\lambda}(\theta^*)} = 1.$$

So Definition 4.12 is well defined. If $(\tilde{\lambda})$ degenerates to a fixed number, then ξ degenerates to a random variable with Poisson distribution.

Example 4.9. $\xi \sim P(\tilde{\lambda})$ be a Poisson distributed Ra-Fu variable, where $\tilde{\lambda}$ is an L-R fuzzy variable with the following membership function,

$$\mu_{\tilde{\lambda}} = \begin{cases} L\left(\dfrac{\lambda - x}{\alpha}\right), & x \leq \lambda, \alpha > 0 \\ R\left(\dfrac{x - \lambda}{\beta}\right), & x > \lambda, \beta > 0 \end{cases}$$

where λ is the "mean" of $\tilde{\lambda}$, α, β are the left and right spread of ξ, respectively. Especially, if $L(x)$ and $R(x)$ are linear functions, then $\tilde{\lambda}$ is a triangular fuzzy variable.

4.2.2.2 Continuous Ra-Fu Variable

Similarly to the discrete Ra-Fu variable, there exists another Ra-Fu variables, that is, the continuous Ra-Fu variable. It can be defined as follows.

Definition 4.13. (Continuous Ra-Fu variable) Let ξ be a Ra-Fu variable on the possibility space $(\Theta, \mathscr{P}(\Theta), Pos)$. If $\xi(\theta)$ is a continuous random variable for any $\theta \in \Theta$, then ξ is said to be a continuous Ra-Fu variable.

Example 4.10. Let ξ be a Ra-Fu variable on $(\Theta, \mathscr{P}(\Theta), Pos)$ with the following density function,

$$\phi(x) = \begin{cases} \dfrac{1}{\tilde{\lambda}} e^{-x/\tilde{\lambda}}, & \text{if } 0 \leq x < \infty \\ 0, & \text{otherwise} \end{cases}$$

where $\tilde{\lambda}$ is a triangular fuzzy variable on $(\Theta, \mathscr{P}(\Theta), Pos)$. Then ξ is a continuous Ra-Fu variable.

Definition 4.14. (Ra-Fu uniform distribution) Let ξ be a Ra-Fu variable on $(\Theta, \mathscr{P}(\Theta), Pos)$ with the following density function,

$$\bar{p}(x) = \begin{cases} \dfrac{1}{\tilde{b} - \tilde{a}}, & \text{if } \tilde{a} \le x \le \tilde{b} \\ 0, & \text{otherwise} \end{cases}$$

where $\tilde{a}$ and $\tilde{b}$ are fuzzy variables on $(\Theta, \mathscr{P}(\Theta), Pos)$ and $\tilde{a} < \tilde{b}$. Then ξ is said to be an uniformly distributed Ra-Fu variable, denoted by $\xi \sim \mathscr{U}(\tilde{a}, \tilde{b})$.

Since we must guarantee that $\tilde{b} - \tilde{a} \neq 0$, $\tilde{a}$ and $\tilde{b}$ needs to be two fuzzy variables, for which one must be bigger than the other. Then it is obvious that $Pos\{\tilde{a} < \tilde{b}\} = 1$ holds. It follows that

$$Pos\{\tilde{a} < \tilde{b}\} = 1 \Leftrightarrow \sup_{x,y \in \mathbf{R}} \{\mu_{\tilde{a}}(x) \wedge \mu_{\tilde{b}}(y) | x \le y\} = 1 \tag{4.9}$$

where $\mu_{\tilde{a}}(x)$ and $\mu_{\tilde{b}}(y)$ are the membership functions of $\tilde{a}$ and $\tilde{b}$, respectively. Then let's consider the following example.

Example 4.11. Let $\xi \sim \mathscr{U}(\tilde{a}, \tilde{b})$ be a Ra-Fu variable, $\tilde{a}$ and $\tilde{b}$ are fuzzy variables with the following membership,

$$\mu_{\tilde{a}}(x) = \frac{10 - x}{10},$$

$$\mu_{\tilde{b}}(x) = \frac{1}{1 + (x - 7)^2}.$$

Really,

$$\mu_{\tilde{a}}(x) \wedge \mu_{\tilde{b}}(x) = \begin{cases} \dfrac{1}{1 + (x - 7)^2}, & 0 \le x < 5.815 \\ \dfrac{10 - x}{10}, & 5.815 \le x \le 10 \\ 0, & x > 10 \end{cases}$$

Then we have $\sup_{x,y \in \mathbf{R}} \{\mu_{\tilde{a}}(x) \wedge \mu_{\tilde{b}}(y) | x \le y\} = 0.42$, see Fig. 4.4. Then ξ cannot be considered as a Ra-Fu variable, for there exists the possibility such that $\tilde{b} - \tilde{a} = 0$.

Definition 4.15. (Ra-Fu normal distribution) Let $\xi \sim \mathscr{N}(\rho, \sigma^2)$ be a random fuzzy variable on $(\Theta, \mathscr{P}(\Theta), Pos)$. We substitute the fuzzy variable $\tilde{\rho}$ for the mean value ρ or the fuzzy variable $\tilde{\sigma}$ for the variance σ, or both. Then ξ is said to be a normally distributed Ra-Fu variable, denoted by $\xi \sim \mathscr{N}(\tilde{\rho}, \sigma^2)$ or $\xi \sim \mathscr{N}(\rho, \tilde{\sigma}^2)$.

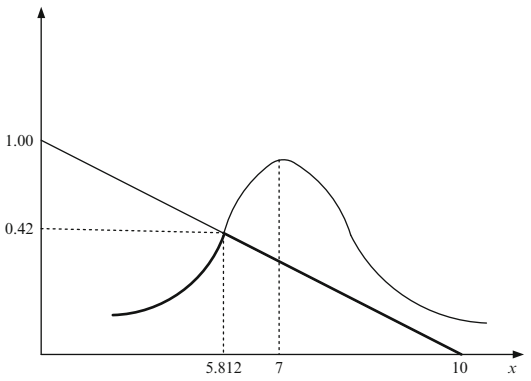


Fig. 4.4 The membership function of $\tilde{a} < \tilde{b}$

In fact, there are many cases about Ra-Fu normal distribution in our real-life world. For example, by the statistical data, the sold amount of some seasonal products are subject to the normal distribution, however, the average sold amount of each year, i.e. "mean", varies from year to year, and it can be described as a fuzzy number by the historical data. Thus, this is an example of normally distributed Ra-Fu variables. It is useful to evaluate the sold amount in the next year. Next, let's discuss a numerical example.

Example 4.12. Let $\xi \sim \mathcal{N}(\tilde{\rho}, 1)$ be a normally distributed Ra-Fu variable, where $\tilde{\rho}$ is a fuzzy variable with the following membership function,

$$\mu_{\tilde{\rho}}(x) = \begin{cases} x - 2, & 2 \le x < 3 \\ -\dfrac{x}{2} + \dfrac{5}{2}, & 3 \le x \le 5 \end{cases}$$

By the definition of α-cut set, we know that $\tilde{\rho}(\theta) \in [\alpha + 2, -2\alpha + 5]$, then for any $\theta \in \Theta$, $\xi(\theta) \sim \mathcal{N}(\tilde{\rho}(\theta), 1)$ is a normally distributed random variable, where $\tilde{\rho}(\theta)$ varies between $\alpha + 2$ and $-2\alpha + 5$.

Definition 4.16. (Ra-Fu exponential distribution) Let ξ be an exponentially distributed random variable with the parameter λ. If we substitute a fuzzy variable $\tilde{\lambda}$ for λ, ξ becomes a exponentially distributed Ra-Fu variable, denoted by $\xi \sim exp(\tilde{\lambda})$.

Really, there are still a lot of real-life cases about Ra-Fu exponential distribution.

Example 4.13. In many statistics problems, the probability distribution is completely known except for the values of one or more parameters. For example, it might be known that the lifetime ξ of a modern engine is an exponentially distributed variable with an unknown mean $\tilde{\lambda}$,

$$\phi(x) = \begin{cases} \dfrac{1}{\tilde{\lambda}} e^{-x/\tilde{\lambda}}, & \text{if } 0 \le x < \infty \\ 0, & \text{otherwise} \end{cases}$$

Usually, there is some relevant information in practice. It is thus possible to specify an interval in which the value of $\tilde{\lambda}$ is likely to lie, or to give an approximate estimate of the value of $\tilde{\lambda}$. It is typically not possible to determine the value of $\tilde{\lambda}$ exactly. If the value of $\tilde{\lambda}$ is provided as a fuzzy variable defined on the possibility space $(\Theta, \mathscr{P}(\Theta), Pos)$, then ξ is a Ra-Fu variable defined as

$$\xi \sim exp(\tilde{\lambda}),$$

where $\tilde{\lambda}$ is a fuzzy variable on $(\Theta, \mathscr{P}(\Theta), Pos)$.

There are still many Ra-Fu variables following other distributions except the above two discrete Ra-Fu variables and three continuous Ra-Fu variables. Here we don't introduce them one by one, interesting readers can define them and deduce them by imitating the above definitions. In the following part we will introduce their expected values and variances.

First of all, let us recall that the expected value of a fuzzy variable ξ on $(\Theta, \mathscr{P}(\Theta), Pos)$ can be defined as follows,

$$E[\xi] = \int_0^{+\infty} Cr\{\theta | \xi(\theta) \geq t\} dt - \int_{-\infty}^0 Cr\{\theta | \xi(\theta) \leq t\} dt \tag{4.10}$$

As we can see in the following part, the expected value of a random fuzzy variable that we defined is somewhat similar to this expression in form. The expected value operator of a Ra-Fu variable can be defined as follows.

Definition 4.17. (Liu and Liu [209]) Let ξ be a Ra-Fu variable defined on the possibility space $(\Theta, \mathscr{P}(\Theta), Pos)$. Then the expected value of ξ is defined as

$$E[\xi] = \int_0^{\infty} Cr\{\theta \in \Theta | E[\xi(\theta)] \geq r\} dr - \int_{-\infty}^0 Cr\{\theta \in \Theta | E[\xi(\theta)] \leq r\} dr$$

provided that at least one of the above two integrals is finite.

Remark 4.2. If the Ra-Fu variable ξ degenerates to a random variable, then the expected value operator becomes

$$E[\xi] = \int_0^{\infty} Pr\{\xi \geq r\} dr - \int_{\infty}^0 Pr\{\xi \leq r\} dr$$

which is just the conventional mathematical expectation of the random variable ξ.

Remark 4.3. If the Ra-Fu variable ξ degenerates to a fuzzy variable, then the expected value operator becomes

$$E[\xi] = \int_0^{\infty} Cr\{\xi \geq r\} dr - \int_{\infty}^0 Cr\{\xi \leq r\} dr$$

which is just the expected value of the fuzzy variable ξ.

Remark 4.4. Let ξ be a Ra-Fu variable on the possibility space $(\Theta, \mathscr{P}(\Theta), Pos)$. If the expected value $E[\xi(\theta)]$ of random variable $\xi(\theta)$ is finite for each $\theta \in \Theta$, then $E[\xi(\theta)]$ is a fuzzy variable on $(\Theta, \mathscr{P}(\Theta), Pos)$.

Example 4.14. Suppose that ξ is a Ra-Fu variable defined as

$$\xi \sim \mathscr{U}(\tilde{\rho}, \tilde{\rho} + 2), \text{ with } \tilde{\rho} = (0, 1, 2).$$

Without loss of generality, we assume that $\tilde{\rho}$ is defined on the possibility space $(\Theta, \mathscr{P}(\Theta), Pos)$. Then for each $\theta \in \Theta$, $\xi(\theta)$ is a random variable and $E[\xi(\theta)] = \tilde{\rho}(\theta) + 1$. Thus the expected value of ξ is $E[\xi] = E[\tilde{\rho}] + 1 = 2$.

By the definition of expected value operator of Ra-Fu variables, we can compute the expected value of some Ra-Fu variables with special distribution. Next, let's introduce their expected value and variance.

Definition 4.18. (Expected value of discrete Ra-Fu variable) Let ξ be a discrete Ra-Fu variable on $(\Theta, \mathscr{P}(\Theta), Pos)$, its expected value can be defined as follows,

$$E[\xi] = \sum_{i=1}^{\infty} p_i E[\tilde{u}_i(\theta)] \tag{4.11}$$

where $\tilde{u}_i(\theta)(i = 1, 2, \ldots)$ are fuzzy variables on the possibility space $(\Theta_i, \mathscr{P}(\Theta_i)_i, Pos_i)$, p_i is the probability of $\xi = \tilde{u}_i(\theta)$. Sometimes, the probability p_i may be a fuzzy variable, then (4.11) can be rewritten as

$$E[\xi] = \sum_{i=1}^{\infty} \tilde{p}_i u_i \tag{4.12}$$

where $\xi = u_i$ with the probability $\tilde{p}_i$.

In the above definition, ξ may be a finite or an infinite random fuzzy variable, such as the binomially distributed Ra-Fu variable and the Poisson distributed Ra-Fu variable. At the same time, for the fuzzy variable $\xi_i(\theta)$ may be in the same possibility space or in different possibility spaces.

By the above definition, we know that if $\xi_i(\theta)$ degenerates to be a certain number or a certain function, then ξ degenerates to be a random variable on $(\Theta, \mathscr{P}(\Theta), Pos)$ and Definition 4.11 is equivalent to Definition 2.8. Next, let's restrict our attention to two kinds of special discrete Ra-Fu variables and discuss their expected values.

Theorem 4.1. *Let ξ be a binomially distributed Ra-Fu variable and $\tilde{p}$ be a fuzzy variable with the following membership,*

$$\mu_{\tilde{p}}(t) = \begin{cases} L\left(\dfrac{p-t}{\alpha}\right), t \leq p, 0 < \alpha < 1 \\ R\left(\dfrac{t-p}{\beta}\right), t \geq p, 0 < \beta < 1 \end{cases}$$

where $p(0 < p < 1)$ *is the "mean" value of* $\tilde{p}$, α, β *are positive numbers expressing the left and right spreads of* $\tilde{p}$ *such that* $p - \alpha > 0$ *and* $p + \beta < 1$, *and reference functions* $L, R : [0, 1] \to [0, 1]$ *with* $L(1) = R(1) = 0$ *and* $L(0) = R(0) = 1$ *are non-increasing, continuous functions. Then we have*

$$E[\xi] = nE[\tilde{p}(\theta)] = nK \tag{4.13}$$

where $K = p + \Upsilon(0) - \Upsilon(\frac{\alpha}{2}) + \chi(\frac{\beta}{2}) - \chi(0)$.

Proof. For each trial, ξ_i is 0–1 distributed Ra-Fu variable, respectively, $i = 1, 2, \ldots, n$. Since $\xi_i(\theta)$ is a random variable, then

$$E[\xi_i(\theta)] = 0 \cdot \tilde{q}(\theta) + 1 \cdot \tilde{p}(\theta) = \tilde{p}(\theta),$$

where $\tilde{q} = 1 - \tilde{p}$ is the probability of "failure". It follows that the expected value of the random variable $\xi(\theta)$ is as follows,

$$E[\xi(\theta)] = E\left[\sum_{i=1}^{n} \xi_i(\theta)\right] = n\tilde{p}(\theta).$$

Then

$$E[\xi] = E[E[\xi(\theta)]] = nE[\tilde{p}(\theta)].$$

Since $\tilde{p}$ is an L-R fuzzy variable with the following membership function,

$$\mu_{\tilde{p}}(t) = \begin{cases} L\left(\dfrac{p-t}{\alpha}\right), & t \le p, 0 < \alpha < 1 \\ R\left(\dfrac{t-p}{\beta}\right), & t \ge p, 0 < \beta < 1 \end{cases}$$

then we have

$$Cr\{\theta|\xi(\theta) \ge t\} = \begin{cases} 1, & t \le p - \alpha \\ 1 - \dfrac{1}{2}L\left(\dfrac{p-t}{\alpha}\right), & p - \alpha \le t < p \\ \dfrac{1}{2}R\left(\dfrac{t-p}{\beta}\right), & p \le t \le p + \beta \\ 0, & t > p + \beta \end{cases}$$

$$Cr\{\theta|\xi(\theta) \le t\} = \begin{cases} 0, & t \le p-\alpha \\ \frac{1}{2}L\left(\frac{p-t}{\alpha}\right), & p-\alpha \le t < p \\ 1-\frac{1}{2}R\left(\frac{t-p}{\beta}\right), & p \le t \le p+\beta \\ 1, & t > p+\beta \end{cases}$$

It follows from (4.10) that

$$\int_{-\infty}^{0} Cr\{\theta|\xi(\theta) \le t\}dt = 0$$

and

$$\begin{aligned} E[\tilde{p}(\theta)] &= \int_0^{+\infty} Cr\{\theta|\xi(\theta) \ge t\}dt - \int_{-\infty}^{0} Cr\{\theta|\xi(\theta) \le t\}dt \\ &= \int_0^{p-\alpha} 1dt + \int_{p-\alpha}^{p} \left[1-\frac{1}{2}L\left(\frac{p-t}{\alpha}\right)\right]dt + \int_p^{p+\beta} \frac{1}{2}R\left(\frac{t-p}{\beta}\right)dt \\ &= p + \frac{\alpha}{2}(\Upsilon(0)-\Upsilon(1)) + \frac{\beta}{2}(\chi(1)-\chi(0)) \end{aligned}$$

where $\Upsilon(x)$ is a continuous function on $[0,1]$ and $\frac{\partial \Upsilon(x)}{\partial x} = L(x)$, $\chi(x)$ is a continuous function on $[0,1]$ and $\frac{\partial \chi(x)}{\partial x} = R(x)$. Then

$$E[\xi] = nE[\tilde{p}(\theta)] = n\left[p + \frac{\alpha}{2}(\Upsilon(0)-\Upsilon(1)) + \frac{\beta}{2}(\chi(1)-\chi(0))\right].$$

This completes the proof. □

Especially, when $\tilde{p}$ is a triangular fuzzy number (a,b,c), $0 < a < b < c < 1$, then $E[\xi] = \frac{n}{4}(a+2b+c)$. If $\tilde{p}$ is a trapezoidal fuzzy variable (a,b,c,d), then $E[\xi] = \frac{n}{4}(a+b+c+d)$. For the poisson distributed Ra-Fu variables, we also obtain the following results.

Theorem 4.2. *Let $\xi \sim P(k,\tilde{\lambda})$ be a Poisson distributed Ra-Fu variable and $\tilde{\lambda}$ is a fuzzy variable with the following membership function,*

$$\mu_{\tilde{\lambda}}(t) = \begin{cases} L\left(\frac{\lambda-t}{\alpha}\right), t \le \lambda, \alpha > 0 \\ R\left(\frac{t-\lambda}{\beta}\right), t \ge \lambda, \beta > 0 \end{cases}$$

where $\lambda(\lambda > 0)$ is the "mean" value of $\tilde{\lambda}$, α, β are positive numbers expressing the left and right spreads of $\tilde{\lambda}$ such that $\lambda - \alpha > 0$, and reference functions $L, R : [0,1] \to [0,1]$ with $L(1) = R(1) = 0$ and $L(0) = R(0) = 1$ are non-increasing, continuous functions. Then we have,

$$E[\xi] = \lambda + \frac{\alpha}{2}(\Upsilon(0) - \Upsilon(1)) + \frac{\beta}{2}(\chi(1) - \chi(0)).$$

Proof. By the definition of the expected value of discrete Ra-Fu variables, we have

$$\begin{aligned} E[\xi] &= \sum_{k=0}^{\infty} kE[p_k] = E\left[\sum_{k=0}^{\infty} k \cdot \frac{\tilde{\lambda}^k}{k!} e^{-\tilde{\lambda}}\right] \\ &= E\left[\tilde{\lambda} e^{-\tilde{\lambda}} \sum_{k=1}^{\infty} \frac{\tilde{\lambda}^{k-1}}{(k-1)!}\right] = E[\tilde{\lambda}] \end{aligned}$$

By the proving process of Theorem 4.1, we have that

$$E[\xi] = E[\tilde{\lambda}] = \lambda + \frac{\alpha}{2}(\Upsilon(0) - \Upsilon(1)) + \frac{\beta}{2}(\chi(1) - \chi(0)).$$

This completes the proof. □

For the continuous Ra-Fu variable, we have the following definition.

Definition 4.19. (Expected value of continuous Ra-Fu variable) Let ξ be a continuous Ra-Fu variable on $(\Theta, \mathscr{P}(\Theta), Pos)$ with the density function $\tilde{p}(x)$, then its expected value can be defined as follows,

$$E[\xi] = \int_0^{\infty} Cr\left\{\int_{x\in\Theta} x\tilde{p}(x)dx \geq r\right\} dr - \int_{-\infty}^{0} Cr\left\{\int_{x\in\Theta} x\tilde{p}(x)dx \leq r\right\} dr \tag{4.14}$$

where $\tilde{p}(x)$ is a density function with fuzzy parameters defined on $(\Theta, \mathscr{P}(\Theta), Pos)$.

If ξ degenerates to be a random variable, the definition is identical with the expected value of random variables in Definition 2.8. Next, let's restrict to three special continuous Ra-Fu variables.

Theorem 4.3. *Assume that $\xi \sim \mathscr{U}(\tilde{a}, \tilde{b})$ is a uniformly distributed Ra-Fu variable, where $\tilde{a}$ and $\tilde{b}$ are both fuzzy variables defined on $(\Theta, \mathscr{P}(\Theta), Pos)$ with the following membership functions, respectively,*

$$\mu_{\tilde{a}}(t) = \begin{cases} L\left(\dfrac{a-t}{\alpha}\right), t \leq a, \alpha > 0 \\ R\left(\dfrac{t-a}{\beta}\right), t \geq a, \beta > 0 \end{cases}$$

and

$$\mu_{\tilde{b}}(t) = \begin{cases} L\left(\dfrac{b-t}{\delta}\right), t \leq b, \delta > 0 \\ R\left(\dfrac{t-b}{\gamma}\right), t \geq b, \gamma > 0 \end{cases}$$

where a, b are the "mean" values of $\tilde{a}$ and $\tilde{b}$, respectively, α and β are the left and right spreads of $\tilde{a}$, δ and γ are the left and right spreads of $\tilde{b}$ such that $b-\delta > a+\beta$. Reference functions $L, R : [0,1] \to [0,1]$ with $L(1) = R(1) = 0$ and $L(0) = R(0) = 1$ are non-increasing, continuous functions. Then we have

$$E[\xi] = \frac{1}{2}(a+b) + \frac{1}{4}(\alpha+\delta)(\Upsilon(0)-\Upsilon(1)) + \frac{1}{4}(\beta+\gamma)(\chi(1)-\chi(0)) \tag{4.15}$$

Proof. For any $\theta \in \Theta$, we have

$$E[\xi(\theta)] = \int_{-\infty}^{+\infty} x\tilde{p}(\theta)(x)dx = \int_{\tilde{a}}^{\tilde{b}} \frac{x}{\tilde{b}-\tilde{a}}dx = \frac{\tilde{b}+\tilde{a}}{2}$$

Obviously, $E[\xi(\theta)]$ is a fuzzy variable since $\tilde{a}$ and $\tilde{b}$ are both fuzzy variables. In fact, assume that $w \in [0,1]$, let $L(\frac{a-x}{\alpha}) = w = L(\frac{b-y}{\delta})$, then

$$x = a - \alpha L^{-1}(w), \quad y = b - \delta L^{-1}(w).$$

It follows that

$$z = \frac{x+y}{2} = \frac{a+b}{2} - \frac{1}{2}(\alpha+\delta)L^{-1}(w).$$

Thus, we have $L\left(\frac{\frac{1}{2}(a+b)-z}{\frac{1}{2}(\alpha+\delta)}\right) = w$. Similarly, we can prove that $R\left(\frac{z-\frac{1}{2}(a+b)}{\frac{1}{2}(\beta+\gamma)}\right) = w$. Then we know that $\frac{\tilde{b}+\tilde{a}}{2}$ is also a fuzzy variable with the following membership,

$$\mu_{\frac{\tilde{b}+\tilde{a}}{2}}(t) = \begin{cases} L\left(\dfrac{\frac{1}{2}(a+b)-t}{\frac{1}{2}(\alpha+\delta)}\right), & t \le \dfrac{1}{2}(a+b) \\ R\left(\dfrac{t-\frac{1}{2}(a+b)}{\frac{1}{2}(\beta+\gamma)}\right), & t \ge \dfrac{1}{2}(a+b) \end{cases}$$

By the proving process of Theorem 4.1, we have

$$E[\xi] = E[E[\xi(\theta)]] = E\left[\frac{\tilde{b}+\tilde{a}}{2}\right]$$

$$= \frac{1}{2}(a+b) + \frac{1}{4}(\alpha+\delta)(\Upsilon(0)-\Upsilon(1)) + \frac{1}{4}(\beta+\gamma)(\chi(1)-\chi(0)).$$

This completes the proof. □

By the definition of uniformly distributed Ra-Fu variables, fuzzy variables can have many other membership functions only satisfying $Pos\{\tilde{b}(\theta) > \tilde{a}(\theta)\} = 1$

for any $\theta \in \Theta$. Readers can similarly deduce their expected values and variances. Next, let's introduce the expected value and variance of normally distributed Ra-Fu variables.

Theorem 4.4. *Let $\xi \sim \mathcal{N}(\tilde{\mu}, \sigma^2)$ be a normally distributed Ra-Fu variable, where $\tilde{\mu}$ is a fuzzy variable on $(\Theta, \mathcal{P}(\Theta), Pos)$ with the following membership function,*

$$\mu_{\tilde{\mu}}(t) = \begin{cases} L\left(\dfrac{\mu - t}{\alpha}\right), t \leq \mu, \alpha > 0 \\ R\left(\dfrac{t - \mu}{\beta}\right), t \geq \mu, \beta > 0 \end{cases}$$

where μ is the "mean" value of $\tilde{\mu}$, α and β are the left and right spreads of $\tilde{\mu}$, respectively. Reference functions $L, R : [0, 1] \to [0, 1]$ with $L(1) = R(1) = 0$ and $L(0) = R(0) = 1$ are non-increasing, continuous functions. Then we have

$$E[\xi] = \mu + \frac{\alpha}{2}(\Upsilon(0) - \Upsilon(1)) + \frac{\beta}{2}(\chi(1) - \chi(0)) \tag{4.16}$$

Proof. By Definition 4.19, we know

$$E[\xi] = \int_0^{\infty} Pr\{\theta \in \Theta | E[\xi(\theta)] \geq t\}dt - \int_{-\infty}^{0} Pr\{\theta \in \Theta | E[\xi(\theta)] \leq t\}dt$$

Since $\xi \sim N(\tilde{\mu}, \sigma^2)$, and obviously we know that $E[\xi(\theta)] = \tilde{\mu}$, then by the proving process of Theorem 4.1, we have

$$E[\xi] = E[\xi(\theta)] = E[\tilde{\mu}] = \mu + \frac{\alpha}{2}(\Upsilon(0) - \Upsilon(1)) + \frac{\beta}{2}(\chi(1) - \chi(0)).$$

This completes the proof. □

Similarly, readers can compute the expected value if $\tilde{\mu}$ follows other distributions or the variance $\tilde{\sigma}$ is also a random variable on $(\Theta, \mathcal{P}(\Theta), Pos)$. Then the expected value and variance of exponentially distributed random fuzzy variables are introduced in the following part. For the exponential distribution, the parameter $\tilde{\lambda}$ must be more than or equal to 0, then we can assume that $\Theta = (0, +\infty)$.

Theorem 4.5. *Let $\xi \sim exp(\tilde{\lambda})$ be an exponentially distributed Ra-Fu variable, where $\tilde{\lambda}$ is a fuzzy variable on $(\Theta, \mathcal{P}(\Theta), Pos)$ with the following membership function,*

$$\mu_{\tilde{\lambda}}(t) = \begin{cases} L\left(\dfrac{\lambda - t}{\alpha}\right), t \leq \mu, \alpha > 0 \\ R\left(\dfrac{t - \lambda}{\beta}\right), t \geq \mu, \beta > 0 \end{cases}$$

where μ is the "mean" value of $\tilde{\lambda}$, α and β are the left and right spreads of $\tilde{\lambda}$ such that $\lambda - \alpha > 0$, respectively. Reference functions $L, R : [0,1] \to [0,1]$ with $L(1) = R(1) = 0$ and $L(0) = R(0) = 1$ are non-increasing, continuous functions. Then we have

$$E[\xi] = -\frac{1}{2\mu} - K - T$$

where $K = \int_0^1 \frac{R'(x)}{\mu + x\beta} dx$ and $T = \int_0^1 \frac{L'(x)}{\mu - x\alpha} dx$.

Proof. For any $\theta \in \Theta$, we have

$$E[\xi(\theta)] = \int_{-\infty}^{+\infty} x \cdot \tilde{\lambda} e^{-\tilde{\lambda}x} dx = \frac{1}{\tilde{\lambda}} \tag{4.17}$$

Obviously, $E[\xi(\theta)] = \frac{1}{\tilde{\lambda}}$ is a fuzzy variable on $(\Theta, \mathscr{P}(\Theta), Pos)$. In fact, assume that $w \in [0,1]$, let $L(\frac{\mu - t}{\alpha}) = w$, then

$$t = \mu - \alpha L^{-1}(w).$$

It follows from that

$$z = \frac{1}{t} = \frac{1}{\mu - \alpha L^{-1}(w)}.$$

Thus, we have $L\left(\frac{\mu - \frac{1}{z}}{\alpha}\right) = w$. Similarly, we can prove that $R\left(\frac{\frac{1}{z} - \mu}{\beta}\right) = w$. Then we know that $\frac{1}{\tilde{\lambda}}$ is also a fuzzy variable with the following membership,

$$\mu_{\frac{1}{\tilde{\lambda}}}(t) = \begin{cases} L\left(\dfrac{\mu - \frac{1}{t}}{\alpha}\right), & \dfrac{1}{t} \le \mu \\ R\left(\dfrac{\frac{1}{t} - \mu}{\beta}\right), & \dfrac{1}{t} \ge \mu \end{cases} \tag{4.18}$$

It follows from the definition of the credibility of fuzzy variables that

$$Cr\left\{\frac{1}{\tilde{\lambda}} \ge t\right\} = \begin{cases} 1, & t \le \dfrac{1}{\mu + \beta} \\ 1 - \dfrac{1}{2} R\left(\dfrac{\frac{1}{t} - \mu}{\beta}\right), & \dfrac{1}{\mu + \beta} < t \le \dfrac{1}{\mu} \\ \dfrac{1}{2} L\left(\dfrac{\mu - \frac{1}{t}}{\alpha}\right), & \dfrac{1}{\mu} < t \le \dfrac{1}{\mu - \alpha} \\ 0, & t > \dfrac{1}{\mu - \alpha} \end{cases}$$

Since $\Theta = (0, +\infty)$ and $\mu - \alpha > 0$, then $\int_{-\infty}^{0} Cr\left\{\frac{1}{\bar{\lambda}} \leq t\right\} dt = 0$ and

$$\begin{aligned}\int_0^{\infty} Cr\left\{\frac{1}{\bar{\lambda}} \geq t\right\} dt &= \int_0^{\frac{1}{\mu+\beta}} 1dt + \int_{\frac{1}{\mu+\beta}}^{\frac{1}{\mu}} \left(1 - \frac{1}{2}R\left(\frac{\frac{1}{t} - \mu}{\beta}\right)\right) dt \\ &\quad + \int_{\frac{1}{\mu}}^{\frac{1}{\mu-\alpha}} \frac{1}{2}L\left(\frac{\mu - \frac{1}{t}}{\alpha}\right) dt \\ &= -\frac{1}{2\mu} - K - T\end{aligned}$$

where $K = \int_0^1 \frac{R'(x)}{\mu + x\beta} dx$ and $T = \int_0^1 \frac{L'(x)}{\mu - x\alpha} dx$. Then

$$E[\xi] = \int_0^{+\infty} Pr\{E[\xi(\theta)] \geq t\} dt - \int_{-\infty}^{0} Pr\{E[\xi(\theta)] \leq t\} dt = -\frac{1}{2\mu} - K - T$$

This completes the proof. □

By the definition of exponentially distributed Ra-Fu variables, we know that the parameter $\bar{\lambda}$ can be a random variable with different distribution, but it must satisfy that $\bar{\lambda}(\theta^*) > 0$ for any $\theta^* \in \Theta^*$. Then readers can deduce the expected value and variance when $\bar{\lambda}$ follows other distribution.

4.3 Ra-Fu EVM

For the complicated real-life uncertain problems, we should convert them into crisp models which can be easily solved. Usually, we can apply the expected value operator to compute the objective function, then get the deterministic expected value model. Before introducing the expected value model, we must propose the definition of the expected value operator of Ra-Fu variables. And the expected value of uncertain variables serves as a powerful tool for a wide variety of applications.

4.3.1 General Model for Ra-Fu EVM

Let's consider the typical single objective with Ra-Fu parameters,

$$\begin{cases} \max f(\boldsymbol{x}, \boldsymbol{\xi}) \\ \text{s.t.} \begin{cases} g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, j = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{4.19}$$

where $f(\boldsymbol{x}, \boldsymbol{\xi})$ and $g_j(\boldsymbol{x}, \boldsymbol{\xi}), j = 1, 2 \ldots, p$ are continuous functions in X and $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_n)$ is a Ra-Fu vector on possibility space $(\Theta, \mathscr{P}(\Theta), Pos)$. Then

it follows from the expected operator that,

$$\begin{cases} \max E[f(\boldsymbol{x}, \boldsymbol{\xi})] \\ \text{s.t.} \begin{cases} E[g_j(\boldsymbol{x}, \boldsymbol{\xi})] \leq 0, j = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{4.20}$$

After being dealt with by expected value operator, the problem (4.19) has been converted into a certain programming and then decision makers can easily obtain the optimal solution.

Definition 4.20. $\boldsymbol{x}$ is said to be called a *feasible solution* of problem (4.20) if and only if $E[g_j(\boldsymbol{x}, \boldsymbol{\xi})] \leq 0 (j = 1, 2, \ldots, p)$. For any feasible solution $\boldsymbol{x}$, if $E[f(\boldsymbol{x}^*, \boldsymbol{\xi})] \geq E[f(\boldsymbol{x}, \boldsymbol{\xi})]$, then $\boldsymbol{x}^*$ is the optimal solution of problem (4.20).

In many cases, there are usually multiple objectives decision makers must consider. Thus we have the following expected value multiobjective model (EVM),

$$\begin{cases} \max [E[f_1(\boldsymbol{x}, \boldsymbol{\xi})], E[f_2(\boldsymbol{x}, \boldsymbol{\xi})], \ldots, E[f_m(\boldsymbol{x}, \boldsymbol{\xi})]] \\ \text{s.t.} \begin{cases} E[g_j(\boldsymbol{x}, \boldsymbol{\xi})] \leq 0, j = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{4.21}$$

where $f_i(\boldsymbol{x}, \boldsymbol{\xi})$ are return functions for $i = 1, 2, \ldots, m$. $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_n)$ is a Ra-Fu vector on possibility space $(\Theta, \mathscr{P}(\Theta), Pos)$.

Definition 4.21. $\boldsymbol{x}^*$ is said to be the Pareto solution of problem (4.21), if there doesn't exist feasible solutions $\boldsymbol{x}$ such that

$$E[f_i(\boldsymbol{x}, \boldsymbol{\xi})] \geq E[f_i(\boldsymbol{x}^*, \boldsymbol{\xi})], i = 1, 2, \ldots, m$$

and there is at least one $j (j = 1, 2, \ldots, m)$ such that $E[f_i(\boldsymbol{x}, \boldsymbol{\xi})] > E[f_i(\boldsymbol{x}^*, \boldsymbol{\xi})]$.

We can also formulate a Ra-Fu decision system as an expected value goal model (EVGM) according to the priority structure and target levels set by the decision-maker,

$$\begin{cases} \min \sum_{j=1}^{l} P_j \sum_{i=1}^{m} (u_{ij} d_i^+ + v_{ij} d_i^-) \\ \text{s.t.} \begin{cases} E[f_i(\boldsymbol{x}, \boldsymbol{\xi})] + d_i^- - d_i^+ = b_i, & i = 1, 2, \ldots, m \\ E[g_j(\boldsymbol{x}, \boldsymbol{\xi})] \leq 0, & j = 1, 2, \ldots, p \\ d_i^-, d_i^+ \geq 0, & i = 1, 2, \ldots, m \\ \boldsymbol{x} \in X \end{cases} \end{cases}$$

where P_j is the preemptive priority factor which expresses the relative importance of various goals, $P_j >> P_{j+1}$, for all j, u_{ij} is the weighting factor corresponding to positive deviation for goal i with priority j assigned, v_{ij} is the weighting factor

corresponding to negative deviation for goal i with priority j assigned, d_i^+ is the positive deviation from the target of goal i, defined as

$$d_i^+ = [E[f_i(\boldsymbol{x}, \boldsymbol{\xi})] - b_i] \vee 0,$$

d_i^- is the negative deviation from the target of goal i, defined as

$$d_i^- = [b_i - E[f_i(\boldsymbol{x}, \boldsymbol{\xi})]] \vee 0,$$

f_i is a function in goal constraints, g_j is a function in real constraints, b_i is the target value according to goal i, l is the number of priorities, m is the number of goal constraints, and p is the number of real constraints.

4.3.2 Linear Ra-Fu EVM and the Maximin Point Method

Generally, many uncertain problems cannot be directly converted into crisp ones unless they have favorable properties and their random fuzzy parameters have crisp distribution. For those which cannot be transformed, Ra-Fu simulation is an useful tool to deal with them. Next, we will exhibit some examples which can be converted into crisp models. Let's consider the following linear multi-objective programming with Ra-Fu parameters,

$$\begin{cases} \max \left[\tilde{\tilde{c}}_1^T \boldsymbol{x}, \tilde{\tilde{c}}_2^T \boldsymbol{x}, \ldots, \tilde{\tilde{c}}_m^T \boldsymbol{x}\right] \\ \text{s.t.} \begin{cases} \tilde{\tilde{e}}_r^T \boldsymbol{x} \leq \tilde{\tilde{b}}_r, r = 1, 2, \ldots, p \\ \boldsymbol{x} \geq 0 \end{cases} \end{cases} \tag{4.22}$$

where $\boldsymbol{x} \in X \subset \mathbf{R}^n$, $\tilde{\tilde{c}}_i = (\tilde{\tilde{c}}_{i1}, \tilde{\tilde{c}}_{i2}, \ldots, \tilde{\tilde{c}}_{in})^T$, $\tilde{\tilde{e}}_r = (\tilde{\tilde{e}}_{r1}, \tilde{\tilde{e}}_{r2}, \ldots, \tilde{\tilde{e}}_{rn})^T$ are Ra-Fu vectors, and $\tilde{\tilde{b}}_r$ are random fuzzy variables, $i = 1, 2, \ldots, m$, $r = 1, 2, \ldots, p$.

4.3.2.1 Crisp Equivalent Model

Because of the uncertainty of Ra-Fu parameters $\tilde{\tilde{c}}_i$, $\tilde{\tilde{e}}_r$ and $\tilde{\tilde{b}}_r$ in the problem (4.22), we cannot easily obtain its optimal solutions. By Definition 4.18 and 4.19, we can obtain the following expected value model,

$$\begin{cases} \max \left[E[\tilde{\tilde{c}}_1^T \boldsymbol{x}], E[\tilde{\tilde{c}}_2^T \boldsymbol{x}], \ldots, E[\tilde{\tilde{c}}_m^T \boldsymbol{x}]\right] \\ \text{s.t.} \begin{cases} E[\tilde{\tilde{e}}_r^T \boldsymbol{x}] \leq E[\tilde{\tilde{b}}_r], r = 1, 2, \ldots, p \\ \boldsymbol{x} \geq 0 \end{cases} \end{cases} \tag{4.23}$$

Theorem 4.6. *Assume that random vector* $\tilde{\tilde{c}}_i = (\tilde{\tilde{c}}_{i1}, \tilde{\tilde{c}}_{i2}, \ldots, \tilde{\tilde{c}}_{in})^T$ *is normally distributed with mean vector* $\tilde{\mu}_i^c = (\tilde{\mu}_{i1}^c(\theta), \tilde{\mu}_{i2}^c(\theta), \ldots, \tilde{\mu}_{in}^c(\theta))^T$ *and positive*

definite covariance matrix V_i^c on the probability space $(\Theta, \mathscr{A}, Pr)$, written as $\tilde{\tilde{c}}_i \sim \mathscr{N}(\tilde{\mu}_i^c(\theta), V_i^c)(i = 1, 2, \ldots, m)$ and random fuzzy vectors $\tilde{\tilde{e}}_r \sim \mathscr{N}(\tilde{\mu}_r^e(\theta), V_r^e)$, $\tilde{\tilde{b}}_r \sim \mathscr{N}(\tilde{\mu}_r^b(\theta), (\sigma_r^b)^2)(r = 1, 2, \ldots, p)$, where $\tilde{\mu}_i^c(\theta)$, $\tilde{\mu}_r^e(\theta)$ are fuzzy vectors and $\tilde{\mu}_r^b(\theta)$ are fuzzy variables defined on $(\Theta, \mathscr{P}(\Theta), Pos)$ respectively characterized by the following membership functions,

$$\mu_{\tilde{\mu}_{ij}^c(\theta)}(t) = \begin{cases} L\left(\dfrac{\mu_{ij}^c - t}{\delta_{ij}^c}\right) & t \le \mu_{ij}^c, \delta_{ij}^c > 0 \\ & \theta \in \Theta \\ R\left(\dfrac{t - \mu_{ij}^c}{\gamma_{ij}^c}\right) & t \ge \mu_{ij}^c, \gamma_{ij}^c > 0 \end{cases} \tag{4.24}$$

and

$$\mu_{\tilde{\mu}_{rj}^e(\theta)}(t) = \begin{cases} L\left(\dfrac{\mu_{rj}^e - t}{\delta_{rj}^e}\right) & t \le \mu_{rj}^e, \delta_{rj}^e > 0 \\ & \theta \in \Theta \\ R\left(\dfrac{t - \mu_{rj}^e}{\gamma_{rj}^e}\right) & t \ge \mu_{rj}^e, \gamma_{rj}^e > 0 \end{cases} \tag{4.25}$$

and

$$\mu_{\tilde{\mu}_r^b(\theta)}(t) = \begin{cases} L\left(\dfrac{\mu_r^b - t}{\delta_r^b}\right) & t \le \mu_r^b, \delta_r^b > 0 \\ & \theta \in \Theta \\ R\left(\dfrac{t - \mu_r^b}{\gamma_r^b}\right) & t \ge \mu_r^b, \gamma_r^b > 0 \end{cases} \tag{4.26}$$

where $\delta_{ij}^c, \gamma_{ij}^c$ are positive numbers expressing the left and right spreads of $\tilde{d}_{ij}^c(\theta)$, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$, $\delta_{rj}^e, \gamma_{rj}^e$ are positive numbers expressing the left and right spreads of $\tilde{\mu}_{rj}^e(\theta)$, $r = 1, 2, \ldots, p$, $j = 1, 2, \ldots, n$, δ_r^b, γ_r^b are positive numbers expressing the left and right spreads of $\tilde{\mu}_r^b(\theta)$, $r = 1, 2, \ldots, p$ and reference functions $L, R : [0, 1] \to [0, 1]$ with $L(1) = R(1) = 0$ and $L(0) = R(0) = 1$ are non-increasing, continuous functions. Assume that for any $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$ and $r = 1, 2, \ldots, p$, $\tilde{\tilde{c}}_{ij}(\theta)$, $\tilde{\tilde{e}}_{ij}(\theta)$ and $\tilde{\tilde{b}}_{ij}(\theta)$ are independently random variables. Then problem (4.23) is equivalent to

$$\begin{cases} \max[H_1(\boldsymbol{x}), H_2(\boldsymbol{x}), \ldots, H_m(\boldsymbol{x})] \\ s.t. \begin{cases} K_r(\boldsymbol{x}) \le B, \ r = 1, 2, \ldots, p \\ \boldsymbol{x} \ge 0 \end{cases} \end{cases} \tag{4.27}$$

where

$$H_i(\boldsymbol{x}) = \frac{1}{2}\delta_i^c\boldsymbol{x}\left[F(\mu_i^c\boldsymbol{x}) - F(\mu_i^c\boldsymbol{x} - \delta_i^c\boldsymbol{x})\right] + \frac{1}{2}\gamma_i^c\boldsymbol{x}\left[G(\mu_i^c\boldsymbol{x} + \gamma_i^c\boldsymbol{x}) - G(\mu_i^c\boldsymbol{x})\right],$$
$$K_r(\boldsymbol{x}) = \frac{1}{2}\delta_r^e\boldsymbol{x}\left[F(\mu_r^e\boldsymbol{x}) - F(\mu_r^e\boldsymbol{x} - \delta_r^e\boldsymbol{x})\right] + \frac{1}{2}\gamma_r^e\boldsymbol{x}\left[G(\mu_r^e\boldsymbol{x} + \gamma_r^e\boldsymbol{x}) - G(\mu_r^e\boldsymbol{x})\right],$$
$$B = \frac{1}{2}\delta_r^b\left[F(\mu_r^b) - F(\mu_r^b - \delta_r^b)\right] + \frac{1}{2}\gamma_r^b\left[G(\mu_r^b + \gamma_r^b) - G(\mu_r^b)\right].$$

Proof. Since Ra-Fu variables $\tilde{\tilde{c}}_{ij}$ are normally distributed on the probability space $(\Theta, \mathscr{A}, Pr)$ and $\tilde{\tilde{c}}_{ij} \sim \mathscr{N}(\tilde{\mu}_{ij}^c(\theta), V_{ij}^c)(i = 1, 2, \ldots, m, j = 1, 2, \ldots, n)$, then it follows from the linearity of Ra-Fu variables that

$$E[\tilde{\tilde{c}}_i^T\boldsymbol{x}] = E\left[\sum_{j=1}^{n} x_j\tilde{\tilde{c}}_{ij}\right] = \sum_{j=1}^{n} x_j E[\tilde{\mu}_{ij}^c(\theta)] = E[\tilde{\mu}_i^{cT}(\theta)\boldsymbol{x}].$$

Since $\tilde{\mu}_i^c(\theta) = (\tilde{\mu}_{i1}^c(\theta), \tilde{\mu}_{i2}^c(\theta), \ldots, \tilde{\mu}_{in}^c(\theta))^T$ is a fuzzy vector. It follows that, for $x_{ij} > 0$, $\tilde{\mu}_i^{cT}(\theta)\boldsymbol{x}$ is a fuzzy variable characterized by the following membership function,

$$\mu_{\tilde{\mu}_i^c(\theta)\boldsymbol{x}}(t) = \begin{cases} L\left(\dfrac{\mu_i^c\boldsymbol{x} - t}{\delta_i^c\boldsymbol{x}}\right) & t \le \mu_i^c\boldsymbol{x},\ \delta_i^c\boldsymbol{x} > 0 \\ & \theta \in \Theta, \\ R\left(\dfrac{t - \mu_i^c\boldsymbol{x}}{\gamma_i^c\boldsymbol{x}}\right) & t \ge \mu_i^c\boldsymbol{x},\ \gamma_i^c\boldsymbol{x} > 0 \end{cases}$$

then we have,

$$Cr\{\tilde{\mu}_i^c(\theta)\boldsymbol{x} \ge t\} = \begin{cases} 1, & \text{if } t \le \mu_i^c\boldsymbol{x} - \delta_i^c\boldsymbol{x} \\ 1 - \dfrac{1}{2}L\left(\dfrac{\mu_i^c\boldsymbol{x} - t}{\delta_i^c\boldsymbol{x}}\right), & \text{if } \mu_i^c\boldsymbol{x} - \delta_i^c\boldsymbol{x} \le t \le \mu_i^c\boldsymbol{x} \\ \dfrac{1}{2}R\left(\dfrac{t - \mu_i^c\boldsymbol{x}}{\gamma_i^c\boldsymbol{x}}\right), & \text{if } \mu_i^c\boldsymbol{x} \le t \le \mu_i^c\boldsymbol{x} + \gamma_i^c\boldsymbol{x} \\ 0, & \text{if } t > \mu_i^c x + \gamma_i^c\boldsymbol{x} \end{cases}$$

It follows from the expected value of fuzzy variables that,

$$\begin{aligned} E[\tilde{\mu}_i^c(\theta)\boldsymbol{x}] &= \int_0^{+\infty} Cr\{\tilde{\mu}_i^c(\theta)\boldsymbol{x} \ge t\}dt - \int_{-\infty}^{0} Cr\{\tilde{\mu}_i^c(\theta)\boldsymbol{x} \le t\}dt \\ &= \frac{1}{2}\delta_i^c\boldsymbol{x}\left[F(\mu_i^c\boldsymbol{x}) - F(\mu_i^c x - \delta_i^c\boldsymbol{x})\right] \\ &\quad + \frac{1}{2}\gamma_i^c\boldsymbol{x}\left[G(\mu_i^c\boldsymbol{x} + \gamma_i^c\boldsymbol{x}) - G(\mu_i^c\boldsymbol{x})\right] \end{aligned}$$

where $F(x)$ and $G(x)$ are respectively continuous functions on $[\mu_i^c \boldsymbol{x} - \delta_i^c \boldsymbol{x}, \mu_i^c \boldsymbol{x}]$ and $[\mu_i^c \boldsymbol{x}, \mu_i^c \boldsymbol{x} + \gamma_i^c \boldsymbol{x}]$ such that

$$\begin{aligned} \frac{\partial F(x)}{\partial x} &= L(x), \text{ for } x \in [\mu_i^c \boldsymbol{x} - \delta_i^c \boldsymbol{x}, \mu_i^c \boldsymbol{x}], \\ \frac{\partial G(x)}{\partial x} &= R(\boldsymbol{x}), \text{ for } x \in [\mu_i^c \boldsymbol{x}, \mu_i^c \boldsymbol{x} + \gamma_i^c \boldsymbol{x}]. \end{aligned}$$

Similarly,

$$E[\tilde{\tilde{e}}_r^T \boldsymbol{x}] = \frac{1}{2}\delta_r^e \boldsymbol{x}\left[F(\mu_r^e \boldsymbol{x}) - F(\mu_r^e \boldsymbol{x} - \delta_r^e \boldsymbol{x})\right] + \frac{1}{2}\gamma_r^e \boldsymbol{x}\left[G(\mu_r^e \boldsymbol{x} + \gamma_{rj}^e \boldsymbol{x}) - G(\mu_{rj}^e \boldsymbol{x})\right],$$

$$E[\tilde{\tilde{b}}_r] = \frac{1}{2}\delta_r^b\left[F(\mu_r^b) - F(\mu_r^b - \delta_r^b)\right] + \frac{1}{2}\gamma_r^b\left[G(\mu_r^b + \gamma_r^b) - G(\mu_r^b)\right].$$

Denote

$$\begin{aligned} H_i(\boldsymbol{x}) &= \frac{1}{2}\delta_i^c \boldsymbol{x}\left[F(\mu_i^c \boldsymbol{x}) - F(\mu_i^c \boldsymbol{x} - \delta_i^c \boldsymbol{x})\right] + \frac{1}{2}\gamma_i^c \boldsymbol{x}\left[G(\mu_i^c \boldsymbol{x} + \gamma_i^c \boldsymbol{x}) - G(\mu_i^c \boldsymbol{x})\right], \\ K_r(\boldsymbol{x}) &= \frac{1}{2}\delta_r^e \boldsymbol{x}\left[F(\mu_r^e \boldsymbol{x}) - F(\mu_r^e \boldsymbol{x} - \delta_r^e \boldsymbol{x})\right] + \frac{1}{2}\gamma_r^e \boldsymbol{x}\left[G(\mu_r^e \boldsymbol{x} + \gamma_r^e \boldsymbol{x}) - G(\mu_r^e \boldsymbol{x})\right], \\ B &= \frac{1}{2}\delta_r^b\left[F(\mu_r^b) - F(\mu_r^b - \delta_r^b)\right] + \frac{1}{2}\gamma_r^b\left[G(\mu_r^b + \gamma_r^b) - G(\mu_r^b)\right]. \end{aligned}$$

Then (4.23) is equivalent to the following formula,

$$\begin{cases} \max[H_1(\boldsymbol{x}), H_2(\boldsymbol{x}), \ldots, H_m(\boldsymbol{x})] \\ \text{s.t.} \begin{cases} K_r(\boldsymbol{x}) \le B, \ r = 1, 2, \ldots, p \\ \boldsymbol{x} \ge 0 \end{cases} \end{cases}$$

The proof is completed. □

Of course, in the real-life problems, Ra-Fu parameters in the linear multiobjective programming problem could follows many different distributed forms, we just only take the normal distribution as an example. Readers can obtain the similar result by the expected value operator.

4.3.2.2 The Maximin Point Method

In this section, we use the maximin point method proposed in [340] to deal with the crisp multiobjective problem (4.27). To maximize the objectives, the maximin point method firstly constructing an evaluation function by seeking the minimal objective value after respectively computing all objective functions, that is, $u(\mathbf{H}(\boldsymbol{x})) = \min_{1 \le i \le m} H_i(x)$, where $\mathbf{H}(\boldsymbol{x}) = (H_1(\boldsymbol{x}), H_2(\boldsymbol{x}), \ldots, H_m(\boldsymbol{x}))^T$. Then

the objective function of problem (4.27) is came down to solve the maximization problem as follows,

$$\max_{\boldsymbol{x}\in X'} u(\mathbf{H}(\boldsymbol{x})) = \max_{\boldsymbol{x}\in X'} \min_{1\le i\le m} H_i(\boldsymbol{x}) \tag{4.28}$$

Sometimes, decision makers need considering the relative importance of various goals, then the weight can be combined into the evaluation function as follows,

$$\max_{\boldsymbol{x}\in X'} u(\mathbf{H}(\boldsymbol{x})) = \max_{\boldsymbol{x}\in X'} \min_{1\le i\le m} \{\omega_i H_i(\boldsymbol{x})\} \tag{4.29}$$

where the weight $\sum_{i=1}^{m} \omega_i = 1(\omega_i > 0)$ and is predetermined by decision makers.

Theorem 4.7. *The optimal solution $\boldsymbol{x}^*$ of problem (4.29) is the weak efficient solution of problem (4.27).*

Proof. Assume that $\boldsymbol{x}^* \in X'$ is the optimal solution of the problem (4.29). If there exists an $\boldsymbol{x}$ such that $H_i(\boldsymbol{x}) \ge H_i(\boldsymbol{x}^*)(i = 1, 2, \ldots, m)$, we have

$$\min_{1\le i\le m} \{\omega_i H_i(\boldsymbol{x}^*)\} \le \omega_i H_i(\boldsymbol{x}^*) \le \omega_i H_i(\boldsymbol{x}),\ 0 < \omega_i < 1$$

Denote $\delta = \min_{1\le i\le m}\{\omega_i H_i(\boldsymbol{x})\}$, then $\delta \ge \min_{1\le i\le m}\{\omega_i H_i(\boldsymbol{x}^*)\}$. This means that $\boldsymbol{x}^*$ isn't the optimal solution of the problem (4.29). This conflict with the condition. Thus, there doesn't exist $\boldsymbol{x} \in X'$ such that $H_i(\boldsymbol{x}) \ge H_i(\boldsymbol{x}^*)$, namely, $\boldsymbol{x}^*$ is a weak efficient solution of the problem (4.27). □

By introducing an auxiliary variable, the maximin problem (4.29) can be converted into a single objective problem. Let

$$\lambda = \min_{1\le i\le m} \{\omega_i H_i(\boldsymbol{x})\},$$

then the problem (4.29) is converted into

$$\begin{cases} \max \lambda \\ \text{s.t.} \begin{cases} \omega_i H_i(\boldsymbol{x}) \ge \lambda, i = 1, 2, \ldots, m \\ \boldsymbol{x} \in X' \end{cases} \end{cases} \tag{4.30}$$

Theorem 4.8. *The problem (4.29) is equivalent to the problem (4.30).*

Proof. Assume that $\boldsymbol{x}^* \in X'$ is the optimal solution of the problem (4.29) and let $\lambda^* = \min_{1\le i\le m}\{\omega_i H_i(\boldsymbol{x}^*)\}$, then it is apparent that $H_i(\boldsymbol{x}^*) \ge \lambda^*$. This means that $(\boldsymbol{x}^*, \lambda^*)$ is a feasible solution of the problem (4.30). Assume that $(\boldsymbol{x}, \lambda)$ is any feasible solution of the problem (4.30). Since $\boldsymbol{x}^*$ is the optimal solution of the problem (4.29), we have

$$\lambda^* = \min_{1\le i\le m}\{\omega_i H_i(\boldsymbol{x}^*)\} \ge \min_{1\le i\le m}\{\omega_i H_i(\boldsymbol{x})\} \ge \lambda,$$

namely, $(\boldsymbol{x}^*, \lambda^*)$ is the optimal solution of the problem (4.30).

On the contrary, assume that $(\boldsymbol{x}^*, \lambda^*)$ is an optimal solution of the problem (4.30). Then $\omega_i H_i(\boldsymbol{x}^*) \ge \lambda^*$ holds for any i, this means $\min_{1\le i\le m}\{\omega_i H_i(\boldsymbol{x}^*)\} \ge \lambda^*$. It follows that for any feasible $\boldsymbol{x} \in X'$,

$$\min_{1\le i\le m}\{\omega_i H_i(\boldsymbol{x})\} = \lambda \le \lambda^* \le \min_{1\le i\le m}\{\omega_i H_i(\boldsymbol{x}^*)\}$$

holds, namely, $\boldsymbol{x}^*$ is the optimal solution of the problem (4.29). □

In a word, the maximin point method can be summarized as follows:

Step 1. Compute the weight for each objective function by solving the two problems, $\max_{\boldsymbol{x}\in X'} H_i(\boldsymbol{x})$ and $\omega_i = H_i(\boldsymbol{x}^*)/\sum_{i=1}^m H_i(\boldsymbol{x}^*)$.

Step 2. Construct the auxiliary problem as follows,

$$\begin{cases} \max \lambda \\ \text{s.t.} \begin{cases} \omega_i H_i(\boldsymbol{x}) \ge \lambda, i = 1, 2, \ldots, m \\ \boldsymbol{x} \in X' \end{cases} \end{cases}$$

Step 3. Solve the above problem to obtain the optimal solution.

4.3.3 Nonlinear Ra-Fu EVM and Ra-Fu Simulation-Based aw-GA

For many decision-making problems, they are usually complicated with nonlinear objectives or constraints or even both of them so that we can not apply the technique proposed above to convert them into crisp ones. For example, let's consider the following problem,

$$\begin{cases} \max [E[f_1(\boldsymbol{x}, \boldsymbol{\xi})], E[f_2(\boldsymbol{x}, \boldsymbol{\xi})], \ldots, E[f_m(\boldsymbol{x}, \boldsymbol{\xi})]] \\ \text{s.t.} \begin{cases} E[g_j(\boldsymbol{x}, \boldsymbol{\xi})] \le 0, j = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{4.31}$$

where $f_i(\boldsymbol{x}, \boldsymbol{\xi})$ are return functions with respect to $\boldsymbol{\xi}$ for $i = 1, 2, \ldots, m$. $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_n)$ is a Ra-Fu vector on possibility space $(\Theta, \mathscr{P}(\Theta), Pos)$. If m is a great large number, and f_i are nonlinear functions, the traditional methods are difficult to deal with it. Then we will introduce another method in the following part which is aimed at the large scale decision making problems with nonlinear objectives or constraints.

4.3.3.1 Ra-Fu Simulation for EVM

The Ra-Fu simulation is used to deal with those which cannot be converted into crisp ones. Next, let's introduce the process of the Ra-Fu simulation dealing with the expected value models.

Assume that $\boldsymbol{\xi}$ is an n-dimensional Ra-Fu vector defined on the possibility space $(\Theta, \mathscr{P}(\Theta), Pos)$, and $f : \mathbf{R}^n \to \mathbf{R}$ is a measurable function. One problem is to calculate the expected value $E[f(\boldsymbol{x}, \boldsymbol{\xi})]$ for given $\boldsymbol{x}$. Then $f(\boldsymbol{x}, \boldsymbol{\xi})$ is a Ra-Fu variable whose expected value $E[f(\boldsymbol{x}, \boldsymbol{\xi})]$ is

$$\int_0^{+\infty} Cr\{\theta \in \Theta | E[f(\boldsymbol{x}, \boldsymbol{\xi}(\theta))] \geq r\} dr - \int_{-\infty}^{0} Cr\{\theta \in \Theta | E[f(\boldsymbol{x}, \boldsymbol{\xi}(\theta))] \leq r\} dr.$$

A Ra-Fu simulation will be introduced to compute the expected value $E[f(\boldsymbol{x}, \boldsymbol{\xi})]$. We randomly sample θ_k from Θ such that $Pos\{\theta_k\} \geq \varepsilon$, and denote $v_k = Pos\{\theta_k\}$ for $k = 1, 2, \ldots, N$, where ε is a sufficiently small number. Then for any number $r \geq 0$, the credibility $Cr\{\theta \in \Theta | E[f(\boldsymbol{x}, \boldsymbol{\xi}(\theta))] \geq r\}$ can be estimated by

$$\frac{1}{2}\left(\max_{1 \leq k \leq N} \{v_k | E[f(\boldsymbol{x}, \boldsymbol{\xi}(\theta_k))] \geq r\} + \min_{1 \leq k \leq N} \{1 - v_k | E[f(\boldsymbol{x}, \boldsymbol{\xi}(\theta_k))] < r\}\right)$$

and for any number $r < 0$, the credibility $Cr\{\theta \in \Theta | E[f(\boldsymbol{x}, \boldsymbol{\xi}(\theta))] \leq r\}$ can be estimated by

$$\frac{1}{2}\left(\max_{1 \leq k \leq N} \{v_k | E[f(\boldsymbol{x}, \boldsymbol{\xi}(\theta_k))] \leq r\} + \min_{1 \leq k \leq N} \{1 - v_k | E[f(\boldsymbol{x}, \boldsymbol{\xi}(\theta_k))] > r\}\right)$$

provided that N is sufficiently large, where $E[f(\boldsymbol{x}, \boldsymbol{\xi}(\theta_k))], k = 1, 2, \ldots, N$ may be estimated by the stochastic simulation.

Then the procedure simulating the expected value of the function $f(\boldsymbol{x}, \boldsymbol{\xi})$ can be summarized as follows:

Procedure `Ra-Fu simulation for EVM`

Input: The decision vector $\boldsymbol{x}$
Output: The expected value $E[f(\boldsymbol{x}, \boldsymbol{\xi})]$
Step 1. Set $e = 0$;
Step 2. Randomly sample θ_k from Θ such that $Pos\{\theta_k\} \geq \varepsilon$ for $k = 1, 2, \ldots, N$, where ε is a sufficiently small number;
Step 3. Let $a = \min_{1 \leq k \leq N} E[f(\boldsymbol{x}, \boldsymbol{\xi}(\theta_k))]$ and $b = \max_{1 \leq k \leq N} E[f(\boldsymbol{x}, \boldsymbol{\xi}(\theta_k))]$;
Step 4. Randomly generate r from $[a, b]$;
Step 5. If $r \geq 0$, then $e \leftarrow e + Cr\{\theta \in \Theta | E[f(\boldsymbol{x}, \boldsymbol{\xi}(\theta_k))] \geq r\}$;
Step 6. If $r < 0$, then $e \leftarrow e - Cr\{\theta \in \Theta | E[f(\boldsymbol{x}, \boldsymbol{\xi}(\theta_k))] \leq r\}$;
Step 7. Repeat the fourth to sixth steps for N times;
Step 8. $E[f(\boldsymbol{x}, \boldsymbol{\xi})] = a \vee 0 + b \wedge 0 + e \cdot (b - a)/N$.

Example 4.15. We employ the Ra-Fu simulation to calculate the expected value of $\sqrt{\tilde{\tilde{\xi}}_1^2 + \tilde{\tilde{\xi}}_2^2 + \tilde{\tilde{\xi}}_3^2}$, where $\tilde{\tilde{\xi}}_1$, $\tilde{\tilde{\xi}}_2$ and $\tilde{\tilde{\xi}}_3$ are random fuzzy variables defined as

$$\begin{aligned}
\tilde{\tilde{\xi}}_1 &\sim \mathscr{N}(\tilde{\rho}_1, 1), \text{ with } \tilde{\rho}_1 = (1, 2, 3), \\
\tilde{\tilde{\xi}}_2 &\sim \mathscr{U}(\tilde{\rho}_2, 2), \text{ with } \tilde{\rho}_2 = (2, 5, 6), \\
\tilde{\tilde{\xi}}_3 &\sim \mathscr{U}(\tilde{\rho}_3, 1), \text{ with } \tilde{\rho}_3 = (2, 3, 4).
\end{aligned}$$

where $\tilde{\rho}_i$ are all triangular fuzzy numbers. We perform the Ra-Fu simulation with 1000 cycles and obtain that $E\left[\sqrt{\tilde{\tilde{\xi}}_1^2 + \tilde{\tilde{\xi}}_2^2 + \tilde{\tilde{\xi}}_3^2}\right] = 6.0246$.

4.3.3.2 The Adaptive Weight Genetic Algorithm

Since evolutionary computation was proposed, ingrowing researchers has been interested in simulating evolution to solve complex optimization problems. Among them, the genetic algorithm introduced by Holland [135] is paid more and more attention to. As a kind of meta-heuristics, it could search the optimal solution without regard to the specific inner connections of the problem. Especially, the application of GA to multiobjective optimization problems has caused a theoretical and practical challenge to the mathematical community. In the past two decades, there are many approaches on GA developed by the scholars in all kinds of field. Globerg [116] firstly suggested the Pareto ranking based fitness assignment method to find the next set of nondominated individuals. Then the multiobjective genetic algorithm in which the rank of individual corresponds to the number of current parent population was proposed by Fonseca and Fleming [103]. There are still two weighted sum genetic algorithms to solve multiobjective optimization problems. One is the random-weight genetic algorithm proposed by Ishibuchi et al.[144], the other is adaptive-weight genetic algorithm proposed by Gen and Chen [109]. Xu, Liu and Wang [341] applied spanning tree based on genetic algorithm to solve a class multiobjective programming problems with Ra-Fu coefficients.

Genetic algorithms (GAs) are a stochastic search method for optimization problems based on the mechanics of natural selection and natural genetics-survival of the fittest. GAs have demonstrated considerable success in providing good solutions to many complex optimization problems and received more and more attentions during the past three decades. When the objective functions to be optimized in the optimization problems are multi-modal or the search spaces are particularly irregular, algorithms need to be highly robust in order to avoid getting stuck at a local optimal solution. The advantage of GAs is just able to obtain the global optimal solution fairly. In addition, GAs do not require the specific mathematical analysis of optimization problems, which makes GAs easily coded by users who are not necessarily good at mathematical and algorithms. GAs have been applied to a wide variety of problems, such as optimal control problem, transportation problem, traveling salesman problem, scheduling, facility layout problem and network optimization and so on.

One of the important technical terms in GAs is chromosome, which is usually a string of symbols or numbers. A chromosome is a coding of a solution of an optimization problem, not necessarily the solution itself. GAs start with an initial set of random-generated chromosomes called population size. All chromosomes are evaluated by the so-called evaluation function, which is some measure of fitness. A new population will be formed by a selection process using some sampling mechanism based on the fitness values. The cycle from one population to the next one is called a generation. In each new generation, all chromosomes will be updated by the crossover and mutation operations. The revised chromosomes are also called offspring. The selection process enters a new generation. After performing the genetic system a given number of cycles, we decode the best chromosome into a solution which is regarded as the optimal solution of the optimization problem.

1. Coding. How to encode a solution of the problem into a chromosome is a key issue when using GAs. The issue has been investigated from many aspects, such as mapping characters from genotype space to phenotype space when individuals are decoded into solutions, and metamorphosis properties when individuals are manipulated by genetic operators.

During the last 15 years, various encoding methods have been created for particular problems to provide effective implementation of GAs. According to what kind of symbol is used as the alleles of a gene, the encoding methods can be classified as follows:

(a) Binary encoding
(b) Real-number encoding
(c) Integer or literal permutation encoding
(d) General data structure encoding.

Genetic algorithm work on two types of spaces alternatively: coding space and solution space, or in other words, genotype space and phenotype space. Genetic operators work on genotype space, and evaluation and selection work on the phenotype space. Natural selection is the link between chromosomes and the performance of decoded solutions. The mapping from genotype space to phenotype space has a considerable influence on the performance of genetic algorithms. One outstanding problem associated with mapping is that some individuals correspond to infeasible solutions to a given problem. This problem may become very severe for constrained optimization problems and combinatorial optimization problems.

2. Genetic operators. Search is one of the more universal problem-solving methods for problems in which one cannot determine a priori the sequence of steps leading to a solution. Typically, there are two types of search behaviors: random search and local search. Random search explores the entire solution and is capable of achieving escape from a local optimum. Local search exploits the best solution and is capable of climbing upward toward a local search optimum. The two types of search abilities from the mutual complementary components of a search. An ideal search should possess both types simultaneously. It is nearly impossible to design such a search method with conventional techniques. Genetic algorithms are a class of general-purpose search methods combining elements of directed and stochastic searches which can make a good balance between exploration and exploitation of

the search space. In genetic algorithms, accumulated information is exploited by the selection mechanism, while mew regions of the search space are explored by means of genetic operators.

In conventional genetic algorithms, the crossover operator is used as the principle operator and the performance of a genetic system is heavily dependent on it. The mutation operator which produces spontaneous random changes in various chromosomes, is used as a background operator. In essence, genetic operators perform a random search and cannot guarantee to yield improved offspring. It has been discovered that the speed of convergence problems. There are many empirical studies on a comparison between crossover and mutation. It is confirmed that mutation can sometimes play a more important role than crossover.

How we conceptualize the genetic search will affect how we design genetic operators. From the point of view of search abilities, it is expected that a search provided by a method can possess the abilities of random search and directed search simultaneously. Cheng and Gen [108, 109] suggest the following approach for designing genetic operators. For the two genetic operators, crossover and mutation, one is used to perform a random search to try to explore the area beyond a local optimum, and the other is used to perform a local search to try to find an improved solution, The genetic search then possesses two types of the search abilities. With this approach, the mutation operator will play the same important role as that of the crossover operator in a genetic search.

3. Selection. The principle behind genetic algorithms is essentially Darwinian natural selection. Selection provides the driving force in a genetic algorithm. With woo much force, genetic search will terminate prematurely; with too little force, evolutionary progress will be slower than necessary. Typically, a lower selection pressure is indicated at the start of a genetic search in favor of a wide exploration of the search space, while a higher selection pressure is recommended at the end to narrow the search space. The selection directs the genetic search toward promising regions in the search space. During the past two decades, many selection methods have been proposed, examined, and compared. There are the following types.

(a) Roulette wheel selection: Proposed by Holland [135], is the best known selection type. The basis idea is to determine selection probability or survival probability for each chromosome proportional to the fitness value. Then a model roulette wheel can be made displaying these probabilities. The selection process is based on spinning the wheel the number of times equal to population size, each time selecting a dingle chromosome for the new population, The wheel features the selection method as a stochastic sampling method that uses a single wheel spin. The wheel is constructed in the same way as is a standard roulette wheel, with a number of equally spaced markers equal to the population size. The basic strategy underlying this approach is to keep the expected number of copies of each chromosome in the next generation.

(b) $(\mu + \lambda)$-selection: In contrast with proportional selection, $(\mu + \lambda)$-selection and (μ, λ)-selection as proposed by B$\ddot{a}$ck are deterministic procedures that select the best chromosomes from parents and offspring. Note that both methods prohibit selection of duplicate chromosome from the population, so many researcher

prefer to use this method to deal with combinational optimization problems. Truncation selection and block selection are also deterministic procedures that rank all individuals according to their fitness and select the best as parents.

(c) Tournament selection: This type of selection contains random and deterministic features simultaneously. A special example is the tournament selection of Goldberg [116]. This method randomly chooses a set of chromosomes and picks out the best chromosome for reproduction. The number of chromosomes in the set is called the tournament size. A common tournament size is 2; this is called a binary tournament. Stochastic tournament selection was suggested by Wetzel [335]. In this method, selection probabilities are calculated normally and successive pairs of chromosome are drawn using roulette wheel selection. After drawing a pair, the chromosome with higher fitness is inserted in the new population. The process continues until the population is full. Reminder stochastic sampling, proposed by Brindle, is a modified version of his deterministic sampling. In this method, each chromosome is allocated samples according to the fractional parts of the number expected.

(d) Steady-state reproduction: Generational replacement, replacing an entire set of parents by their offspring, can be viewed as another version of the deterministic approach. The steady-state reproduction of Whitely and Syswerdra [310] belongs to this class, in which the n worst parents are replaced by offspring (n is the number of offspring).

(e) Ranking and scaling: The ranking and scaling mechanisms are proposed to mitigate these problems. The scaling method maps raw objective function values to positive real values, and the survival probability for each chromosome is determined according to these values. Fitness scaling has a twofold intension: to maintain a reasonable differential between relative fitness ratings of chromosomes, and to prevent too-rapid takeover by some superchromosomes to meet the requirement to limit competition early but to stimulate it later. Since De Jong's work [74], use of scaling objective functions has become widely accepted, and several scaling mechanisms have been proposed. According to the type of function used to transform the raw fitness into scaled fitness, scaling methods can be classified as linear scaling, sigma truncation, power law scaling, logarithmic scaling, an so on. If the transformation relation between scaled fitness and raw fitness is constant, it is called a static scaling method; if the transformation is variable with respect to some factors, it is called a dynamic scaling method. The windowing technique introduces a moving baseline into fitness proposition selection to maintain more constant selection pressure. The normalizing technique is also one type of dynamic scaling proposed by Cheng and Gen [109]. For most scaling methods, scaling parameters are problem dependent. Fitness ranking has an effect similar to that of fitness scaling but avoids the need for extra scaling parameters. Baker introduced the notion of ranking selection with genetic algorithm to overcome the scaling problems of the direct fitness-based approach. The ranking method ignores the actual object function values; instead, it uses a ranking of chromosome to determine survival probability. The idea is straightforward: Sort the population according to the ranking

but not its raw fitness, Two methods are in common use: linear ranking and exponential ranking.

(f) Sharing: The sharing techniques, introduced by Goldberg and Richardson [116] for multi-model function optimization, are used to maintain the diversity of population. A sharing function is a way of determining the degradation of an individual's fitness due to a neighbor at some distance. With the degradation, the reproduction probability of individuals in a crowd peak is restrained while other individuals are encouraged to give offspring.

Next, we will introduce the adaptive weight-based GA which used in this section. This section attempts to apply Ra-Fu simulation to compute the expected value and convert the uncertain multi-objective problem into deterministic one and make use of adaptive weight genetic algorithm to solve this multi-objective problem. For the following model,

$$\begin{cases} \max\left[E[f_1(\boldsymbol{x},\boldsymbol{\xi})], E[f_2(\boldsymbol{x},\boldsymbol{\xi})], \ldots, E[f_m(\boldsymbol{x},\boldsymbol{\xi})]\right] \\ \text{s.t. } E[g_r(\boldsymbol{x},\boldsymbol{\xi})] \leq 0,\ r = 1,2,\ldots,p. \end{cases} \tag{4.32}$$

No matter that the random variable is discrete or continuous, we can simulate its expected value by stochastic simulation and apply the genetic algorithm to solve the multi-objective programming problem.

Adaptive weight approach

The adaptive weight approach proposed by Gen and Cheng [109] makes use of the useful information from current generation to readjust the weights of every objectives, then obtains a search pressure towards to positive ideal points. Let P denote the set of the current population, we can define the maximal and minimal values as follows for each objective in problem (4.32), respectively.

$$\begin{aligned} z_k^{max} &= \max\{E[f_k(\boldsymbol{x},\boldsymbol{\xi})] | \boldsymbol{x} \in P\}, k = 1,2,\ldots,m. \\ z_k^{min} &= \min\{E[f_k(\boldsymbol{x},\boldsymbol{\xi})] | \boldsymbol{x} \in P\}, k = 1,2,\ldots,m. \end{aligned}$$

Then the adaptive weight for objective k is calculated by the following equation:

$$w_k = \frac{1}{z_k^{max} - z_k^{min}}, k = 1,2,\ldots,m.$$

For a given individual x, the weighted-sum objective function is given by the following equation:

$$z(\boldsymbol{x}) = \sum_{k=1}^{m} \frac{E[f_k(\boldsymbol{x},\boldsymbol{\xi})] - z_k^{min}}{z_k^{max} - z_k^{min}}.$$

Comprise approach

The purpose of the comprise approach is to construct fitness function and select the better chromosomes in current generation. For each feasible solution x, the regret function is defined by the following weighted L_p-norm,

$$r(\boldsymbol{x}, p) = \left(\sum_{k=1}^{m} w_k^p |E[f_k(\boldsymbol{x}, \boldsymbol{\xi})] - z_k^*|^p \right)^p$$

where w_k is the adaptive weight generated by the above section, z_k^* is the ideal value of the decision maker for each objective k. Sometimes, it is difficult for DM to obtain the ideal point for many complex problems. Gen and Cheng [109] proposed the concept of a proxy ideal point $z^* = (z_1^{max}, z_2^{max}, \ldots, z_m^{max})$ to replace the ideal point. In problem (4.32), DM wants to obtain the maximal objective values, then it is necessary to convert the regret value to the fitness value in order to ensure that the excellent chromosome has larger fitness value. The fitness function of each chromosome x can be computed by

$$eval(\boldsymbol{x}) = \frac{r_{max} - r(\boldsymbol{x}, p) + \gamma}{r_{max} - r_{min} + \gamma},$$

where γ is the real number in (0,1), r_{max} and r_{min} denote the maximal and minimal regret value in the current generation, respectively.

Genetic operators. When GA proceeds, some important factors should be considered, such as, the search direction to optimal solution and the search speed and so on. In general, the exploitation of the accumulated information resulting from GA search is done by the selection mechanism, while the exploitation to new regions of the search space is accounted by genetic operators. Readers could to refer to Fig. 4.5 to get the intuitive understanding.

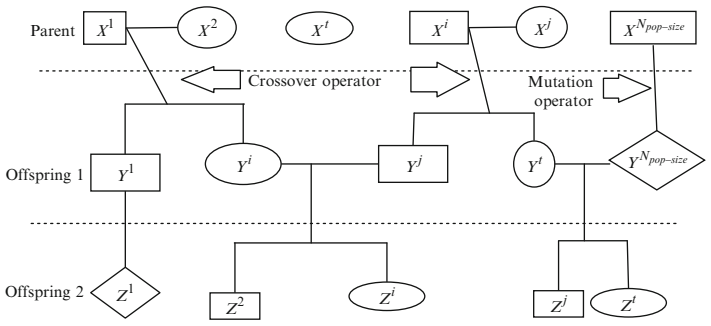


Fig. 4.5 Genetic operators

1. Selection process. Selection provides the driving force in a GA. With too much force, genetic search will be slower than necessary. The selection directs the genetic search toward promising regions in the search space. Roulette wheel selection, proposed by Holland [135] is the best known selection type. The basic idea is to determine selection probability or survival probability for each chromosome proportional to the fitness value. We can apply the roulette wheel method to develop the selection process. Each time a single chromosome for a new population is selected in the following way: Compute the total probability q,

$$q = \sum_{j=1}^{N_{pop-size}} eval(\boldsymbol{x}^j).$$

Then compute the probability of the ith chromosome q_i, $q_i = \frac{eval(\boldsymbol{x}^i)}{q}$. Generate a random number r in [0, 1] and select the ith chromosome $\boldsymbol{x}_i$ such that $q_{i-1} < r \leq q_i, 1 \leq i \leq N_{pop-size}$. Repeat the above process $N_{pop-size}$ times and we obtain $N_{pop-size}$ copies of chromosomes. The selection probability can be computed by the following function,

$$p_i = \frac{eval(\boldsymbol{x}^i) - eval(\boldsymbol{x})_{min}}{\sum_{j=1}^{pop-size} eval(\boldsymbol{x}^i) - eval(\boldsymbol{x})_{min}}$$

where $eval(\boldsymbol{x})_{min}$ is the minimum fitness value of current population.

2. Crossover operation. Crossover is the main genetic operator. It operates on two chromosomes at a time and generates offspring by combing both chromosomes' features. The crossover probability (denoted by P_c) is defined as the probability of the number of offspring produced in each generation to the population size. This probability controls the expected number $P_c \cdot N_{pop-size}$ of chromosomes to undergo the crossover operation. The detailed step is as follows. Generate a random number c from the open interval (0, 1) and the chromosome $\boldsymbol{x}^i$ is selected as a parent provided that $c < P_c$, where parameter P_c is the probability of crossover operation. Repeat this process $N_{pop-size}$ times and $P_c \cdot N_{pop-size}$ chromosomes are expected to be selected to undergo the crossover operation. The crossover operator on x^1 and x^2 will produce two children y^1 and y^2 as follows:

$$\boldsymbol{y}^1 = c\boldsymbol{x}^1 + (1-c)\boldsymbol{x}^2, \qquad \boldsymbol{y}^2 = c\boldsymbol{x}^2 + (1-c)\boldsymbol{x}^1.$$

If both children are feasible, then we replace the parents with them, or else we keep the feasible one if it exists. Repeat the above operation until two feasible children are obtained or a given number of cycles is finished.

3. Mutation operation. Mutation is a background operator which produces spontaneous random changes in various chromosomes. In GA, mutation serves the crucial role of either replacing the genes lost from the population during the

selection process so that they can be tried in a new context, or providing the genes that were not present in the initial population. The mutation probability (denoted with P_m) is defined as the percentage of the total number genes in the population. The mutation probability controls the probability at which new genes are introduced into the population for trial. The detailed process is as follows. Similar to the crossover process, the chromosome $\boldsymbol{x}^i$ is selected as a parent to undergo the mutation operation provided that random number $m < P_m$, where parameter P_m as the probability of mutation operation. $P_m \cdot N_{pop-size}$ are expected to be selected after repeating the process $N_{pop-size}$ times. Suppose that $\boldsymbol{x}$ is chosen as a parent. Choose a mutation direction $\mathbf{d} \in \mathbf{R^n}$ randomly. Replace $\boldsymbol{x}$ with $\boldsymbol{x} + M \cdot \mathbf{d}$ if $\boldsymbol{x} + M \cdot \mathbf{d}$ is feasible, otherwise we set M as a random between 0 and M until it is feasible or a given number of cycle is finished. Here, M is a sufficiently large positive number.

Procedure for GA

We illustrate the Ra-Fu simulation-based genetic algorithm procedure as follows (see Fig. 4.6):

Procedure `The procedure for GA`

Input: The parameters $N_{pop-size}$, P_c and P_m
Output: The optimal chromosomes
Step 1. Initialize $N_{pop-size}$ chromosomes whose feasibility may be checked by Ra-Fu simulation;
Step 2. Update the chromosomes by crossover and mutation operations and Ra-Fu simulation is used to check the feasibility of offspring. Compute the fitness of each chromosome based on weight-sum objective;
Step 3. Select the chromosomes by spinning the roulette wheel;
Step 4. Make the crossover operation;
Step 5. Make the mutation operation for the chromosomes generated by crossover operation;
Step 6. Repeat the second to fourth steps for a given number of cycles;
Step 7. Report the best chromosome as the optimal solution.

4.3.4 Numerical Examples

Example 4.16. In order to illustrate the proposed model and method, let's consider the following multi-objective programming problem with Ra-Fu coefficients.

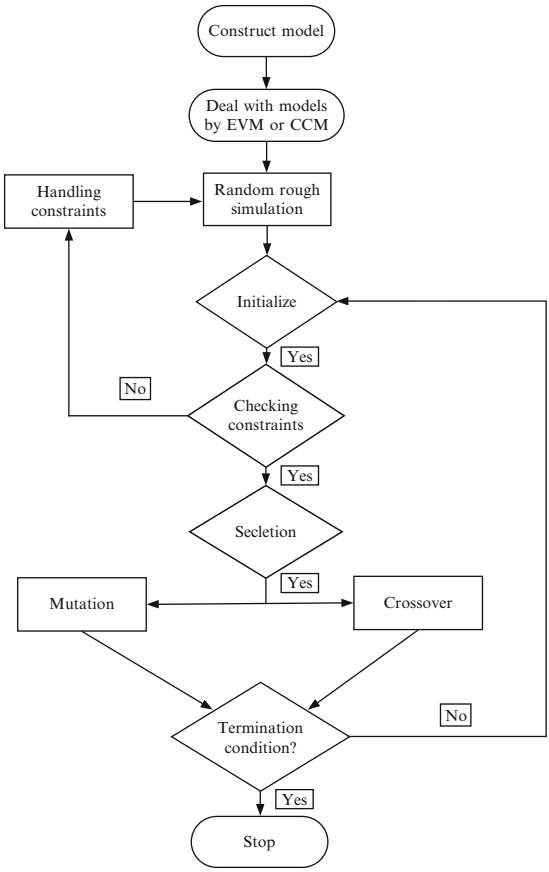


Fig. 4.6 Flow chart of GA

$$\begin{cases} \max f_1(\boldsymbol{x}, \boldsymbol{\xi}) = \tilde{\tilde{\xi}}_1 x_1 + \tilde{\tilde{\xi}}_2 x_2 + \tilde{\tilde{\xi}}_3 x_3 + \tilde{\tilde{\xi}}_4 x_4 + \tilde{\tilde{\xi}}_5 x_5 \\ \max f_2(\boldsymbol{x}, \boldsymbol{\xi}) = c_1 \tilde{\tilde{\xi}}_6 x_1 + c_2 \tilde{\tilde{\xi}}_7 x_2 + c_3 \tilde{\tilde{\xi}}_8 x_3 + c_4 \tilde{\tilde{\xi}}_9 x_4 + c_5 \tilde{\tilde{\xi}}_{10} x_5 \\ \text{s.t.} \begin{cases} x_1 + x_2 + x_3 + x_4 + x_5 \le 350 \\ x_1 + x_2 + x_3 + x_4 + x_5 \ge 300 \\ 4x_1 + 2x_2 + 1.5x_3 + x_4 + 2x_5 \le 1085 \\ x_1 + 4x_2 + 2x_3 + 5x_4 + 3x_5 \le 660 \\ x_1 \ge 20, x_2 \ge 20, x_3 \ge 20, x_4 \ge 20, x_5 \ge 20 \end{cases} \end{cases} \tag{4.33}$$

where $c = (c_1, c_2, c_3, c_4, c_5) = (1.2, 0.5, 1.3, 0.8, 0.9)$,

$\tilde{\tilde{\xi}}_1 \sim \mathcal{N}(\tilde{u}_1, 1)$, with $\tilde{u}_1 \sim (113, 4, 4)_{LR}$, $\tilde{\tilde{\xi}}_2 \sim \mathcal{N}(\tilde{u}_2, 4)$, with $\tilde{u}_2 \sim (241, 7, 7)_{LR}$,
$\tilde{\tilde{\xi}}_3 \sim \mathcal{N}(\tilde{u}_3, 1)$, with $\tilde{u}_3 \sim (87, 2, 2)_{LR}$, $\tilde{\tilde{\xi}}_4 \sim \mathcal{N}(\tilde{u}_4, 2)$, with $\tilde{u}_4 \sim (56, 2, 2)_{LR}$,
$\tilde{\tilde{\xi}}_5 \sim \mathcal{N}(\tilde{u}_5, 1)$, with $\tilde{u}_5 \sim (92, 3, 3)_{LR}$, $\tilde{\tilde{\xi}}_6 \sim \mathcal{N}(\tilde{u}_6, 1)$, with $\tilde{u}_6 \sim (628, 8, 8)_{LR}$,

$\tilde{\tilde{\xi}}_7 \sim \mathcal{N}(\tilde{u}_7, 2)$, with $\tilde{u}_7 \sim (143, 4, 4)_{LR}$, $\tilde{\tilde{\xi}}_8 \sim \mathcal{N}(\tilde{u}_8, 2)$, with $\tilde{u}_8 \sim (476, 5, 5)_{LR}$,
$\tilde{\tilde{\xi}}_9 \sim \mathcal{N}(\tilde{u}_9, 2)$, with $\tilde{u}_9 \sim (324, 4, 4)_{LR}$, $\tilde{\tilde{\xi}}_{10} \sim \mathcal{N}(\tilde{u}_{10}, 2)$, with $\tilde{u}_{10} \sim (539, 7, 7)_{LR}$.

and $\tilde{u}_i (i = 1, 2, \ldots, 10)$ are all assumed as triangular fuzzy variables which are a class of L-R fuzzy variables. Because of the uncertainty of Ra-Fu variables, DM usually wants to get the maximum expected value, then we have the following expected value model,

$$\begin{cases} \max H_1(\boldsymbol{x}) = E[\tilde{\tilde{\xi}}_1]x_1 + E[\tilde{\tilde{\xi}}_2]x_2 + E[\tilde{\tilde{\xi}}_3]x_3 + E[\tilde{\tilde{\xi}}_4]x_4 + E[\tilde{\tilde{\xi}}_5]x_5 \\ \max H_2(\boldsymbol{x}) = c_1 E[\tilde{\tilde{\xi}}_6]x_1 + c_2 E[\tilde{\tilde{\xi}}_7]x_2 + c_3 E[\tilde{\tilde{\xi}}_8]x_3 + c_4 E[\tilde{\tilde{\xi}}_9]x_4 + c_5 E[\tilde{\tilde{\xi}}_{10}]x_5 \\ \text{s.t.} \begin{cases} x_1 + x_2 + x_3 + x_4 + x_5 \le 350 \\ x_1 + x_2 + x_3 + x_4 + x_5 \ge 300 \\ 4x_1 + 2x_2 + 1.5x_3 + x_4 + 2x_5 \le 1085 \\ x_1 + 4x_2 + 2x_3 + 5x_4 + 3x_5 \le 660 \\ x_1 \ge 20, x_2 \ge 20, x_3 \ge 20, x_4 \ge 20, x_5 \ge 20 \end{cases} \end{cases} \tag{4.34}$$

It follows from Theorem 4.6 that, (4.34) is equivalent to

$$\begin{cases} \max H_1(\boldsymbol{x}) = 113x_1 + 241x_2 + 87x_3 + 56x_4 + 92x_5 \\ \max H_2(\boldsymbol{x}) = 753.6x_1 + 71.5x_2 + 618.8x_3 + 259.2x_4 + 485.1x_5 \\ \text{s.t.} \begin{cases} x_1 + x_2 + x_3 + x_4 + x_5 \le 350 \\ x_1 + x_2 + x_3 + x_4 + x_5 \ge 300 \\ 4x_1 + 2x_2 + 1.5x_3 + x_4 + 2x_5 \le 1085 \\ x_1 + 4x_2 + 2x_3 + 5x_4 + 3x_5 \le 660 \\ x_1 \ge 20, x_2 \ge 20, x_3 \ge 20, x_4 \ge 20, x_5 \ge 20 \end{cases} \end{cases} \tag{4.35}$$

Next, we use the maximin method to compute the optimal solution.

Step 1. We can compute the weights of two objective functions by solving $\max_{x \in X} H_i(x)$ and $\omega_i = H_i(\boldsymbol{x}^*)/(H_1(\boldsymbol{x}^*) + H_2(\boldsymbol{x}^*))$ as follows,

$$\begin{aligned} &H_1(\boldsymbol{x}^*) = 43944.64, \ H_2(\boldsymbol{x}^*) = 225424, \\ &\omega_1 = 0.163, \qquad\qquad \omega_2 = 0.837. \end{aligned}$$

Step 2. Construct the auxiliary problem as follows,

$$\begin{cases} \max \lambda \\ \text{s.t.} \begin{cases} 0.163 * (113x_1 + 241x_2 + 87x_3 + 56x_4 + 92x_5) \ge \lambda \\ 0.837 * (753.6x_1 + 71.5x_2 + 618.8x_3 + 259.2x_4 + 485.1x_5) \ge \lambda \\ x_1 + x_2 + x_3 + x_4 + x_5 \le 350 \\ x_1 + x_2 + x_3 + x_4 + x_5 \ge 300 \\ 4x_1 + 2x_2 + 1.5x_3 + x_4 + 2x_5 \le 1085 \\ x_1 + 4x_2 + 2x_3 + 5x_4 + 3x_5 \le 660 \\ x_1 \ge 20, x_2 \ge 20, x_3 \ge 20, x_4 \ge 20, x_5 \ge 20 \end{cases} \end{cases}$$

Step 3. Resolve the above problem, and we get the weak efficient solution of the problem (4.35): $\boldsymbol{x}^* = (218.57, 60.36, 20.00, 20.00, 20.00)^T$.

Example 4.17. Let's consider the another multi-objective programming problem with Ra-Fu coefficients as follows, and we will apply the Ra-Fu simulation-based GA to resolve it.

$$\begin{cases} \max\ f_1(\boldsymbol{x}, \boldsymbol{\xi}) = 3\tilde{\tilde{\xi}}_1^2 x_1 - 2\tilde{\tilde{\xi}}_1\tilde{\tilde{\xi}}_2 x_2 + 1.3\tilde{\tilde{\xi}}_2^2 x_3 \\ \max\ f_2(\boldsymbol{x}, \boldsymbol{\xi}) = 2.5\tilde{\tilde{\xi}}_3^2 x_1 + 3\tilde{\tilde{\xi}}_3\tilde{\tilde{\xi}}_4 x_2 + 5\tilde{\tilde{\xi}}_4^2 x_3 \\ \text{s.t.} \begin{cases} x_1 + x_2 + x_3 \le 10 \\ 3x_1 + 5x_2 + 3x_3 \ge 4 \\ x_1, x_2, x_3 \ge 0 \end{cases} \end{cases} \tag{4.36}$$

where $\xi_i\,(i = 1, \ldots, 6)$ are all independently Ra-Fu variables as follows,

$$\tilde{\tilde{\xi}}_1 \sim \mathcal{N}(\tilde{\mu}_1, 2), \quad \text{with } \tilde{\mu}_1 = (5, 6, 7),\ \tilde{\tilde{\xi}}_2 \sim \mathcal{N}(\tilde{\mu}_2, 1), \text{ with } \tilde{\mu}_2 = (6.5, 8, 10),$$
$$\tilde{\tilde{\xi}}_3 \sim \mathcal{N}(\tilde{\mu}_3, 1.5), \text{ with } \tilde{\mu}_3 = (4, 5, 6),\ \tilde{\tilde{\xi}}_4 \sim \mathcal{N}(\tilde{\mu}_4, 2), \text{ with } \tilde{\mu}_4 = (5, 7, 8).$$

where $\tilde{\mu}_i$ are all triangular fuzzy numbers, $i = 1, \ldots, 4$. By the expected value operator of Ra-Fu variables, we have the following expected model of problem (4.36),

$$\begin{cases} \max H_1(\boldsymbol{x}) = 3E[\tilde{\tilde{\xi}}_1^2] x_1 - 2E[\tilde{\tilde{\xi}}_1\tilde{\tilde{\xi}}_2] x_2 + 1.3E[\tilde{\tilde{\xi}}_2^2] x_3 \\ \max H_2(\boldsymbol{x}) = 2.5E[\tilde{\tilde{\xi}}_3^2] x_1 + 3E[\tilde{\tilde{\xi}}_3\tilde{\tilde{\xi}}_4] x_2 + 5E[\tilde{\tilde{\xi}}_4^2] x_3 \\ \text{s.t.} \begin{cases} x_1 + x_2 + x_3 \le 10 \\ 3x_1 + 5x_2 + 3x_3 \ge 4 \\ x_1, x_2, x_3 \ge 0 \end{cases} \end{cases} \tag{4.37}$$

Let the probability P_c of crossover process be 0.6 and the probability P_m of the mutation process be 0.3, perform the Ra-Fu simulation-based GA with 5000 cycles and we obtain the optimal solutions under different weights as shown in Table 4.2, and Figs. 4.7 and 4.8.

4.4 Ra-Fu CCM

In practice, the goal of the decision-maker is to minimize the total cost or maximize the total profit on the condition of possibility β at probability α, where α and β are the predetermined confidence levels. Then Charnes and Cooper [45] proposed the chance constraint model to deal with this kind of problems. Similar to the definition, some scholars [205] initialized the concept of the chance measure of Ra-Fu variables which is also considered in the multi-objective problems with Ra-Fu coefficients.

Table 4.2 Optimal solutions under different weights by GA

w_1	w_2	H	x_1	x_2	x_3	Gen
0.1	0.9	2288.20	0.002	0	0.998	2000
0.2	0.8	2126.40	0.012	0	0.987	2000
0.3	0.7	1964.60	0.001	0	0.999	2000
0.4	0.6	1802.80	0.002	0	0.998	2000
0.5	0.5	1641.00	0.014	0	0.986	2000

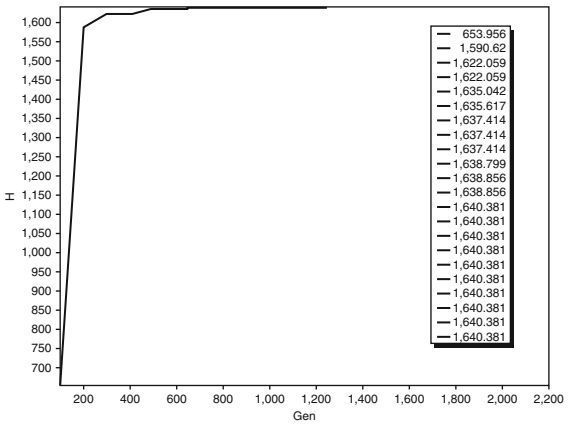


Fig. 4.7 Search process of Ra-Fu simulation-based GA

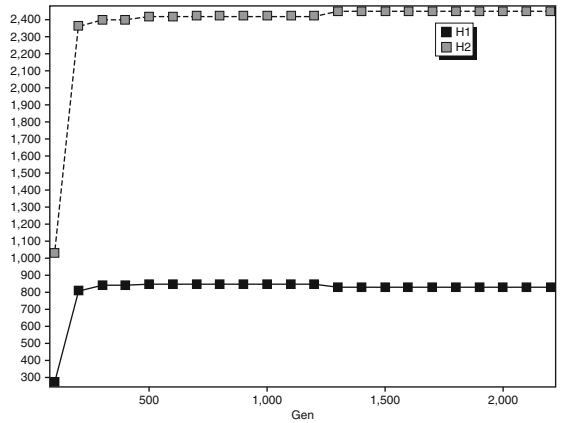


Fig. 4.8 Two objective values by Ra-Fu simulation-based GA

Next, let's briefly introduce the definition and property of the chance measure of Ra-Fu variables.

4.4.1 General Model for Ra-Fu CCM

The primitive chance measure of Ra-Fu event is defined as a function rather than a number.

Definition 4.22. (Liu [205]) Let $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_n)$ be a Ra-Fu vector on the possibility space $(\Theta, \mathscr{P}(\Theta), Pos)$, and $f : \mathbf{R}^n \to \mathbf{R}$ be continuous functions. Then the primitive chance of Ra-Fu event characterized by $f(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0$ is a function from [0,1] to [0,1], defined as

$$Ch\{f(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\}(\alpha) = \sup\{\beta \,|Pos\{\theta \in \Theta \,|Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}(\theta)) \leq 0\} \geq \beta\} \geq \alpha\}.$$

From Definition 4.22, we know that

$$Ch\{f(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\}(\alpha) \geq \beta \Leftrightarrow Pos\{Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \geq \beta\} \geq \alpha \tag{4.38}$$

Remark 4.5. The primitive chance represents "the Ra-Fu event holds with probability $Ch\{f(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\}(\alpha)$ at possibility α".

Remark 4.6. It is obvious that $Ch\{f(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\}(\alpha)$ is a decreasing function of α.

Remark 4.7. If the Ra-Fu vector $\boldsymbol{\xi}$ becomes a random vector, then the chance $Ch\{f(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\}(\alpha)$ (with $\alpha > 0$) is exactly the probability of the event. That is,

$$Ch\{f(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\}(\alpha) \equiv Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\}$$

Remark 4.8. If the Ra-Fu vector $\boldsymbol{\xi}$ becomes a fuzzy vector, then the chance $Ch\{f(\boldsymbol{\xi}) \leq 0\}(\alpha)$ (with $\alpha > 0$) takes the values either 0 or 1. That is,

$$Ch\{f(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\}(\alpha) = \begin{cases} 1, \text{ if } Pos\{f(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \geq \alpha \\ 0, \text{ otherwise} \end{cases}$$

Assume that $\boldsymbol{x}$ is a decision vector, $\boldsymbol{\xi}$ is a Ra-Fu vector, $f(\boldsymbol{x}, \boldsymbol{\xi})$ is a return function, and $g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, j = 1, 2, \ldots, p$ do not define a deterministic feasible set, it is naturally desired that the Ra-Fu constraints hold with probability β at possibility α, where α and β are specified confidence levels. Then we have the chance constraints as follows,

$$Ch\{g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\}(\alpha_j) \geq \beta_j, j = 1, 2, \ldots, p \tag{4.39}$$

Remark 4.9. If the Ra-Fu vector $\boldsymbol{\xi}$ degenerates to a random vector, and $\alpha_j > 0$, then the chance constraint (4.39) degenerates to

$$Pr\{g_j(\boldsymbol{x},\boldsymbol{\xi}) \le 0\} \ge \beta_j, j = 1,2,\ldots,p$$

which are standard stochastic chance constraints.

Remark 4.10. If the Ra-Fu vector ξ degenerates to a fuzzy vector, and $\beta_j > 0$, then the chance constraint (4.39) degenerates to

$$Pos\{g_j(\boldsymbol{x},\boldsymbol{\xi}) \le 0\} \ge \alpha_j, j = 1,2,\ldots,p$$

which are standard fuzzy chance constraints.

Consider the following multiobjective programming problem with Ra-Fu coefficients,

$$\begin{cases} \max\left[f_1(\boldsymbol{x},\boldsymbol{\xi}), f_2(\boldsymbol{x},\boldsymbol{\xi}),\ldots,f_m(\boldsymbol{x},\boldsymbol{\xi})\right] \\ \text{s.t.} \begin{cases} g_r(\boldsymbol{x},\boldsymbol{\xi}) \le 0,, r = 1,2,\ldots,p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{4.40}$$

where $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T$ is an n-dimensional decision vector; $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_n)$ is a Ra-Fu vector; $f_i(\boldsymbol{x},\boldsymbol{\xi})$ are objective functions, $i = 1,2,\ldots,m$; $g_r(\boldsymbol{x},\boldsymbol{\xi}) \le 0$ are Ra-Fu constraints, $r = 1,2,\ldots,p$. For a fixed decision vector $\boldsymbol{x}$, it is meaningless to maximize the objectives $f_i(\boldsymbol{x},\boldsymbol{\xi}), i = 1,2,\ldots,m$, before we know the exact value of the Ra-Fu vector $\boldsymbol{\xi}$, just as we can not maximize a random function in stochastic programming. Also, we can not judge weather or not a decision $\boldsymbol{x}$ is feasible before we know the value of $\boldsymbol{\xi}$. Hence, both the objectives and constraints in problem (4.40) are ill-defined. For presenting a mathematically meaningful Ra-Fu programming, we build a new class of Ra-Fu programming to model Ra-Fu decision problems via chance measure which was proposed above. We present the chance-constrained multi-objective programming as follows,

$$\begin{cases} \max\left[\bar{f}_1, \bar{f}_2, \ldots, \bar{f}_m\right] \\ \text{s.t.} \begin{cases} Ch\{f_i(\boldsymbol{x},\boldsymbol{\xi}) \ge \bar{f}_i\}(\alpha_i) \ge \beta_i, i = 1,2,\ldots,m \\ Ch\{g_r(\boldsymbol{x},\boldsymbol{\xi}) \le 0\}(\eta_r) \ge \theta_r, \ \ r = 1,2,\ldots,p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{4.41}$$

where α_i, β_i, η_r and θ_r are predetermined confidence levels, $i = 1,2,\ldots,m, r = 1,2,\ldots,p$. By (4.38), problem (4.41) can be rewritten as

$$\begin{cases} \max\left[\bar{f}_1, \bar{f}_2, \ldots, \bar{f}_m\right] \\ \text{s.t.} \begin{cases} Pos\{\theta | Pr\{f_i(\boldsymbol{x},\boldsymbol{\xi}) \ge \bar{f}_i\} \ge \beta_i\} \ge \alpha_i, i = 1,2,\ldots,m \\ Pos\{\theta | Pr\{g_r(\boldsymbol{x},\boldsymbol{\xi}) \le 0\} \ge \theta_r\} \ge \eta_r, \ \ r = 1,2,\ldots,p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{4.42}$$

where α_i, β_i, η_r and θ_r are predetermined confidence levels, $i = 1, 2, \ldots, m, r = 1, 2, \ldots, p$. If the objectives is to minimize the cost, then problem (4.41) should be formulated as follows,

$$\begin{cases} \min [\bar{f}_1, \bar{f}_2, \ldots, \bar{f}_m] \\ \text{s.t.} \begin{cases} Pos\{\theta | Pr\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) \le \bar{f}_i\} \ge \beta_i\} \ge \alpha_i, \ i = 1, 2, \ldots, m \\ Pos\{\theta | Pr\{g_r(\boldsymbol{x}, \boldsymbol{\xi}) \le 0\} \ge \theta_r\} \ge \eta_r, \ \ r = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{4.43}$$

Definition 4.23. Suppose a feasible solution $\boldsymbol{x}^*$ of problem (4.42) satisfies

$$Pos\{\theta | Pr\{f_i(\boldsymbol{x}^*, \boldsymbol{\xi}) \ge \bar{f}_i(\boldsymbol{x}^*)\} \ge \beta_i\} \ge \alpha_i, \ i = 1, 2, \ldots, m,$$

where confidence levels $\alpha_i, \beta_i \in [0, 1]$. $\boldsymbol{x}^*$ is said to be an efficient solution to problem (4.42) if and only if there exists no other feasible solution $\boldsymbol{x}$ such that

$$Pos\{\theta | Pr\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) \ge \bar{f}_i(\boldsymbol{x})\} \ge \beta_i\} \ge \alpha_i, \ i = 1, 2, \ldots, m,$$

$f_i(\boldsymbol{x}) \ge f_i(\boldsymbol{x}^*)$ for all i and $f_{i_0}(\boldsymbol{x}) > f_{i_0}(\boldsymbol{x}^*)$ for at least one $i_0 \in \{1, 2, \ldots, m\}$.

Sometimes, we may formulate a Ra-Fu decision system as a chance-constrained goal model (CCGM) according to the priority structure and target levels set by the decision-maker:

$$\begin{cases} \min \sum_{j=1}^{l} P_j \sum_{i=1}^{m} (u_{ij} d_i^+ + v_{ij} d_i^-) \\ \text{s.t.} \begin{cases} Pos\{\theta | Pr\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) - b_i \le d_i^+\} \ge \beta_i^+\} \ge \alpha_i^+, & i = 1, 2, \ldots, m \\ Pos\{\theta | Pr\{b_i - f_i(\boldsymbol{x}, \boldsymbol{\xi}) \le d_i^-\} \ge \beta_i^-\} \ge \alpha_i^-, & i = 1, 2, \ldots, m \\ Pos\{\theta | Pr\{g_r(\boldsymbol{x}, \boldsymbol{\xi}) \le 0\} \ge \theta_r\} \ge \eta_r, & r = 1, 2, \ldots, p \\ d_i^-, d_i^- \ge 0, & i = 1, 2, \ldots, m \end{cases} \end{cases} \tag{4.44}$$

where P_j is the preemptive priority factor which express the relative importance of various goals, $P_j >> P_{j+1}$, for all j, u_{ij} is the weighting factor corresponding to positive deviation for goal i with priority j assigned, v_{ij} is the weighting factor corresponding to negative deviation for goal i with priority j assigned, d_i^+ is the α_i^+, β_i^+-optimistic positive deviation from the target of goal i, defined as

$$\min\{d \vee 0 | Pos\{\theta | Pr\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) - b_i \le d_i^+\} \ge \beta_i^+\} \ge \alpha_i^+\} \tag{4.45}$$

d_i^- is the α_i^-, β_i^--optimistic positive deviation from the target of goal i, defined as

$$\min\{d \vee 0 | Pos\{\theta | Pr\{b_i - f_i(\boldsymbol{x}, \boldsymbol{\xi}) \le d_i^-\} \ge \beta_i^-\} \ge \alpha_i^-\} \tag{4.46}$$

f_i is a function in goal constraints, g_r is a function in system constraints, b_i is the target value according to goal i, l is the number of priorities, m is the number of goal constraints, and p is the number of system constraints.

Remark 4.11. If the Ra-Fu vector $\boldsymbol{\xi}$ degenerates to the random variable, then the two events $Pos\{\theta|Pr\{f_i(\boldsymbol{x},\boldsymbol{\xi}) - b_i \leq d_i^+\} \geq \beta_i^+\}$ and $Pos\{\theta|Pr\{b_i - f_i(\boldsymbol{x},\boldsymbol{\xi}) \leq d_i^-\} \geq \beta_i^-\}$ should be always stochastic at possibility 1 for any $\theta \in \Theta$ provided that $\beta_i^+, \beta_i^- > 0$, then

$$Pos\{\theta|Pr\{f_i(\boldsymbol{x},\boldsymbol{\xi}) - b_i \leq d_i^+\} \geq \beta_i^+\} \geq \alpha_i^+$$

is equivalent to $Pr\{f_i(\boldsymbol{x},\boldsymbol{\xi}) - b_i \leq d_i^+\} \geq \beta_i^+$, and

$$Pos\{\theta|Pr\{b_i - f_i(\boldsymbol{x},\boldsymbol{\xi}) \leq d_i^-\} \geq \beta_i^-\} \geq \alpha_i^-$$

is equivalent to $Pr\{b_i - f_i(\boldsymbol{x},\boldsymbol{\xi}) \leq d_i^-\} \geq \beta_i^-$. Similarly, the constraint

$$Pos\{\theta|Pr\{g_r(\boldsymbol{x},\boldsymbol{\xi}) \leq 0\} \geq \theta_r\} \geq \eta_r$$

is equivalent to $Pr\{g_r(\boldsymbol{x},\boldsymbol{\xi}) \leq 0\} \geq \theta_r$, then problem (4.44) is rewritten as

$$\begin{cases} \min \sum_{j=1}^{l} P_j \sum_{i=1}^{m} (u_{ij} d_i^+ + v_{ij} d_i^-) \\ \text{s.t.} \begin{cases} Pr\{f_i(\boldsymbol{x},\boldsymbol{\xi}) - b_i \leq d_i^+\} \geq \beta_i^+, & i = 1,2,\ldots,m \\ Pr\{b_i - f_i(\boldsymbol{x},\boldsymbol{\xi}) \leq d_i^-\} \geq \beta_i^-, & i = 1,2,\ldots,m \\ Pr\{g_r(\boldsymbol{x},\boldsymbol{\xi}) \leq 0\} \geq \theta_r, & r = 1,2,\ldots,p \\ d_i^-, d_i^- \geq 0, & i = 1,2,\ldots,m \end{cases} \end{cases} \tag{4.47}$$

This is identical with the goal programming in the stochastic environment.

Remark 4.12. If the Ra-Fu vector ξ degenerates to the fuzzy variable, then $Pr\{f_i(x,\xi) - b_i \leq d_i^+\}$ and $Pr\{b_i - f_i(x,\xi) \leq d_i^-\}$ should be always 1 provided that $\beta_i^+, \beta_i^- > 0$, then

$$Pos\{\theta|Pr\{f_i(\boldsymbol{x},\boldsymbol{\xi}) - b_i \leq d_i^+\} \geq \beta_i^+\} \geq \alpha_i^+$$

is equivalent to $Pos\{\theta|f_i(\boldsymbol{x},\boldsymbol{\xi}) - b_i \leq d_i^+\} \geq \alpha_i^+$, and

$$Pos\{\theta|Pr\{b_i - f_i(\boldsymbol{x},\boldsymbol{\xi}) \leq d_i^-\} \geq \beta_i^-\} \geq \alpha_i^-$$

is equivalent to $Pos\{\theta|b_i - f_i(\boldsymbol{x},\boldsymbol{\xi}) \leq d_i^-\} \geq \alpha_i^-$. Similarly, the constraint

$$Pos\{\theta|Pr\{g_r(\boldsymbol{x},\boldsymbol{\xi}) \leq 0\} \geq \theta_r\} \geq \eta_r$$

is equivalent to $Pos\{\theta|g_r(\boldsymbol{x},\boldsymbol{\xi}) \le 0\} \ge \eta_r$, then problem (4.44) is rewritten as

$$\begin{cases} \min \sum_{j=1}^{l} P_j \sum_{i=1}^{m} (u_{ij} d_i^+ + v_{ij} d_i^-) \\ \text{s.t.} \begin{cases} Pos\{\theta|f_i(\boldsymbol{x},\boldsymbol{\xi}) - b_i \le d_i^+\} \ge \alpha_i^+, \ i = 1,2,\ldots,m \\ Pos\{\theta|b_i - f_i(\boldsymbol{x},\boldsymbol{\xi}) \le d_i^-\} \ge \alpha_i^-, \ i = 1,2,\ldots,m \\ Pos\{\theta|g_r(\boldsymbol{x},\boldsymbol{\xi}) \le 0\} \ge \eta_r, \qquad r = 1,2,\ldots,p \\ d_i^-, d_i^- \ge 0, \qquad i = 1,2,\ldots,m \end{cases} \end{cases} \tag{4.48}$$

This is identical with the goal programming in the fuzzy environment.

4.4.2 Linear Ra-Fu CCM and ε-Constraint Method

In this section, let's restrict our attention on the linear multiobjective programming with Ra-Fu parameters. Firstly, we use some mathematical technique to transform the Ra-Fu CCM with special parameters into a crisp one. Secondly, the ε-constraint method is used to solve the crisp multiobjective programming problem.

4.4.2.1 Crisp Equivalent Model

Let's consider the following model,

$$\begin{cases} \max\left[\tilde{\tilde{c}}_1^T \boldsymbol{x}, \tilde{\tilde{c}}_2^T \boldsymbol{x}, \ldots, \tilde{\tilde{c}}_m^T \boldsymbol{x}\right] \\ \text{s.t.} \begin{cases} \tilde{\tilde{e}}_r^T \boldsymbol{x} \le \tilde{\tilde{b}}_r, r = 1,2,\ldots,p \\ \boldsymbol{x} \ge 0 \end{cases} \end{cases} \tag{4.49}$$

where $\boldsymbol{x} \in X \subset \mathbf{R}^n$, $\tilde{\tilde{c}}_i = (\tilde{\tilde{c}}_{i1}, \tilde{\tilde{c}}_{i2}, \ldots, \tilde{\tilde{c}}_{in})^T$, $\tilde{\tilde{e}}_r = (\tilde{\tilde{e}}_{r1}, \tilde{\tilde{e}}_{r2}, \ldots, \tilde{\tilde{e}}_{rn})^T$ and $\tilde{\tilde{b}}_r$ are Ra-Fu vectors, $i = 1,2,\ldots,m$, $r = 1,2,\ldots,p$.

Then by the definition of chance measure of Ra-Fu variables and the formula (4.38), we have the following chance-constrained model of (4.49),

$$\begin{cases} \max[\bar{f}_1, \bar{f}_2, \ldots, \bar{f}_m] \\ \text{s.t.} \begin{cases} Pos\{\theta \in \Theta | Pr\{\tilde{\tilde{c}}_i^T \boldsymbol{x} \ge \bar{f}_i\} \ge \beta_i\} \ge \alpha_i, \ i = 1,2,\ldots,m \\ Pos\{\theta \in \Theta | Pr\{\tilde{\tilde{e}}_r^T \boldsymbol{x} \le \tilde{\tilde{b}}_r\} \ge \theta_r\} \ge \eta_r, \ r = 1,2,\ldots,p \\ \boldsymbol{x} \ge 0 \end{cases} \end{cases} \tag{4.50}$$

To deal with the uncertain programming problem (4.50), we usually divide them into two kinds, one is the problems with clearly distributed Ra-Fu parameters and the other is the problems with unclearly distributed random fuzzy parameters. For the former, we can transform them into the crisp ones by the chance-constrained

operator, but for the latter, it is difficult to convert them into crisp ones, we must employ the technique of simulation to simulate them and obtain the certain value.

Theorem 4.9. *Assume that the Ra-Fu vector* $\tilde{\bar{c}}_i = (\tilde{\bar{c}}_{i1}, \tilde{\bar{c}}_{i2}, \ldots, \tilde{\bar{c}}_{in})^T$ *is normally distributed with mean vector* $\tilde{d}_i^c(\theta) = (\tilde{d}_{i1}^c(\theta), \tilde{d}_{i2}^c(\theta), \ldots, \tilde{d}_{in}^c(\theta))^T$ *and positive definite covariance matrix* V_i^c*, written as* $\tilde{\bar{c}}_i \sim \mathcal{N}(d_i^c(\theta), V_i^c)$*, where* $\tilde{d}_{ij}^c(\theta)$ *is a fuzzy variable characterized by the following membership function,*

$$\mu_{\tilde{d}_{ij}^c(\theta)}(t) = \begin{cases} L\left(\dfrac{d_{ij}^c - t}{\delta_{ij}^c}\right), & t \le d_{ij}^c, \delta_{ij}^c > 0 \\ & \theta \in \Theta \\ R\left(\dfrac{t - d_{ij}^c}{\gamma_{ij}^c}\right), & t \ge d_{ij}^c, \gamma_{ij}^c > 0 \end{cases} \tag{4.51}$$

where δ_{ij}^c*,* γ_{ij}^c *are positive numbers expressing the left and right spreads of* $\tilde{d}_{ij}(\theta)$*,* $i = 1, 2, \ldots, m$*,* $j = 1, 2, \ldots, n$*, and reference functions* $L, R : [0, 1] \to [0, 1]$ *with* $L(1) = R(1) = 0$ *and* $L(0) = R(0) = 1$ *are non-increasing, continuous functions. Then, we have* $Pos\{\theta \in \Theta | Pr\{\tilde{\bar{c}}_i^T \boldsymbol{x} \le \bar{f}_i\} \ge \beta_i\} \ge \alpha_i$ *if and only if*

$$\bar{f}_i \le \Phi^{-1}(1 - \beta_i)\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}} + R^{-1}(\alpha_i)\gamma_{ij}^c + d_i^{cT}\boldsymbol{x}$$

Proof. From the assumption we know that $\tilde{\bar{c}}_i$ is a normally distributed Ra-Fu vector with mean vector $\tilde{d}_i^c(\theta)$ and positive definite covariance matrix V_i^c, written as $\tilde{\bar{c}}_{ij} \sim \mathcal{N}(\tilde{d}_i^c(\theta), V_i^c)$, it follows that $\tilde{\bar{c}}_i^T \boldsymbol{x} \sim \mathcal{N}(\tilde{d}_i^c(\theta)^T \boldsymbol{x}, \boldsymbol{x}^T V_i^c \boldsymbol{x})$, then we have

$$\begin{aligned} & Pr\{\tilde{\bar{c}}_i^T \boldsymbol{x} \ge \bar{f}_i\} \ge \beta_i \\ & \Leftrightarrow Pr\left\{\frac{\tilde{\bar{c}}_i^T \boldsymbol{x} - \tilde{d}_i^c(\theta)^T \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}} \ge \frac{\bar{f}_i - \tilde{d}_i^c(\theta)^T \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right\} \ge \beta_i \\ & \Leftrightarrow 1 - \Phi\left(\frac{\bar{f}_i - \tilde{d}_i^c(\theta)^T \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right) \ge \beta_i \\ & \Leftrightarrow \tilde{d}_i^c(\theta)^T \boldsymbol{x} \ge \bar{f}_i - \Phi^{-1}(1 - \beta_i)\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}} \end{aligned}$$

Since $\tilde{d}_{ij}^c(\theta) = (d_{ij}^c, \delta_{ij}^c, \gamma_{ij}^c)$ is an L-R fuzzy variable, it follows from the fuzzy arithmetic that, for $x_{ij} \ge 0$, $\tilde{d}_i^c(\theta)^T \boldsymbol{x}$ is still an L-R fuzzy variable characterized by the following membership function,

$$\mu_{\tilde{d}_{ij}^{c}(\theta)\boldsymbol{x}}(t)=\begin{cases} L\left(\dfrac{d_i^{cT}\boldsymbol{x}-t}{\delta_i^{cT}\boldsymbol{x}}\right) & t\le d_i^{cT}\boldsymbol{x},\delta_i^{cT}\boldsymbol{x}>0 \\ & \theta\in\Theta \\ R\left(\dfrac{t-d_i^{cT}\boldsymbol{x}}{\gamma_i^{cT}\boldsymbol{x}}\right) & t\ge d_i^{cT}\boldsymbol{x},\gamma_i^{cT}\boldsymbol{x}>0 \end{cases}$$

where $\tilde{d}_i^{cT}=(d_{i1},d_{i2},\ldots,d_{in})^T$, $\delta_i^{cT}=(\delta_{i1},\delta_{i2},\ldots,\delta_{in})^T$ and $\gamma_i^{cT}=(\gamma_{i1},\gamma_{i2},\ldots,\gamma_{in})^T$. Denote $K(\boldsymbol{x})=\bar{f_i}-\Phi^{-1}(1-\beta_i)\sqrt{\boldsymbol{x}^TV_i^c\boldsymbol{x}}$, then it follows that

$$\begin{aligned} &Pos\left\{\theta|\tilde{d}_i^{cT}\boldsymbol{x}\ge K(\boldsymbol{x})\right\}\ge\alpha_i \\ &\Leftrightarrow \alpha_i\le\begin{cases} 1, & \text{if } K(\boldsymbol{x})\le d_i^{cT}\boldsymbol{x} \\ R\left(\dfrac{K(\boldsymbol{x})-d_i^{cT}\boldsymbol{x}}{\gamma_i^{cT}\boldsymbol{x}}\right), & \text{if } d_i^{cT}\boldsymbol{x}<K(\boldsymbol{x})\le d_i^{cT}\boldsymbol{x}+\gamma_{ij}^c\boldsymbol{x} \\ 0, & \text{if } K(\boldsymbol{x})>d_i^{cT}\boldsymbol{x}+\gamma_{ij}^c\boldsymbol{x} \end{cases} \end{aligned}$$

For $\alpha_i\in(0,1)$, we have

$$K(\boldsymbol{x})\le R^{-1}(\alpha_i)\gamma_{ij}^c x+d_i^{cT}\boldsymbol{x}$$

That is

$$\bar{f_i}\le\Phi^{-1}(1-\beta_i)\sqrt{\boldsymbol{x}^TV_i^c\boldsymbol{x}}+R^{-1}(\alpha_i)\gamma_{ij}^c\boldsymbol{x}+d_i^{cT}\boldsymbol{x}$$

This completes the proof. □

Similar, the constraints $Pos\{\theta|Pr\{\tilde{\tilde{e}}_r^T\boldsymbol{x}\le\tilde{\tilde{b}}_r\}\ge\theta_r\}\ge\eta_r$, $r=1,2,\ldots,p$ can be converted into crisp equivalent constraints.

Theorem 4.10. *Suppose that $\tilde{\tilde{e}}_r=(\tilde{\tilde{e}}_{r1},\tilde{\tilde{e}}_{r2},\ldots,\tilde{\tilde{e}}_{rn})^T$ is a normally distributed Ra-Fu vector with the fuzzy mean vector $\tilde{d}_r^e(\theta)=(\tilde{d}_{r1}^e(\theta),\tilde{d}_{r2}^e(\theta),\ldots,\tilde{d}_{nr}^e(\theta))^T$ and covariance matrix V_r^e, written as $\tilde{\tilde{e}}_r\sim\mathcal{N}(\tilde{d}_r^e(\theta),V_r^e)$, $\tilde{\tilde{b}}_r$ is a Ra-Fu variable with the fuzzy mean variable $\tilde{d}_r^b(\theta)$ and the variance $(\sigma_r^b)^2$, written as $\tilde{\tilde{b}}_r\sim\mathcal{N}(\tilde{d}_r^b(\theta),(\sigma_r^b)^2)$, where $\tilde{d}_{rj}^e(\theta)$ and $\tilde{d}_r^b(\theta)$ are fuzzy variables characterized by the following membership functions, respectively,*

$$\mu_{\tilde{d}_{rj}^e(\theta)}(t)=\begin{cases} L\left(\dfrac{d_{rj}^e-t}{\delta_{rj}^e}\right), & t\le d_{rj}^e,\delta_{rj}^e>0 \\ & \theta\in\Theta \\ R\left(\dfrac{t-d_{rj}^e}{\gamma_{rj}^e}\right), & t\ge d_{rj}^e,\gamma_{rj}^e>0 \end{cases} \tag{4.52}$$

and

$$\mu_{\tilde{d}_r^b(\theta)}(t) = \begin{cases} L\left(\dfrac{d_r^b - t}{\delta_r^b}\right), & t \le d_r^b, \delta_r^b > 0 \\ R\left(\dfrac{t - d_r^b}{\gamma_r^b}\right), & t \ge d_r^b, \gamma_r^b > 0 \end{cases} \quad \theta \in \Theta \tag{4.53}$$

where $\delta_{rj}^e, \gamma_{rj}^e$ *are positive numbers expressing the left and right spreads of* $\tilde{d}_{rj}^e(\theta)$, δ_r^b, γ_r^b *are the left and right spreads of* $\tilde{d}_r^b(\theta)$, $r = 1, 2, \ldots, p$, $j = 1, 2, \ldots, n$, *and reference functions* $L, R : [0, 1] \to [0, 1]$ *with* $L(1) = R(1) = 0$ *and* $L(0) = R(0) = 1$ *are non-increasing, continuous functions. Assume that for any* $\theta \in \Theta$, $\tilde{\tilde{e}}_{rj}(\theta)$, $\tilde{\tilde{b}}_r(\theta)$ *are independent random variables,* $r = 1, 2, \cdots p$, $j = 1, 2, \ldots, n$. *Then, we have* $Pos\{\theta \in \Theta | Pr\{\tilde{\tilde{e}}_r^T \boldsymbol{x} \le \tilde{\tilde{b}}_r\} \ge \theta_r\} \ge \eta_r$ *if and only if*

$$\Phi^{-1}(\theta_r)\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2} - (d_r^b - d_r^e \boldsymbol{x}) - R^{-1}(\eta_r)(\gamma_r^b + \delta_r^e \boldsymbol{x}) \le 0$$

Proof. From the assumption, we know that for any $\theta \in \Theta$, $(\tilde{\tilde{e}}_{rj}(\theta))_{n\times 1} \sim \mathcal{N}(\tilde{d}_r^e(\theta), V_r^e)$ and $\tilde{\tilde{d}}_r(\theta) \sim \mathcal{N}(\tilde{d}_r^b(\theta), (\sigma_r^b)^2)$ are independent random variables, it follows that

$$\tilde{\tilde{e}}_r(\theta)^T \boldsymbol{x} - \tilde{\tilde{b}}_r(\theta) \sim \mathcal{N}(\tilde{d}_r^e(\theta)\boldsymbol{x} - \tilde{d}_r^b(\theta), \boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2)$$

is also a normally distributed random variable for any $\theta \in \Theta$. Then we have

$$\begin{aligned} & Pr\{\tilde{\tilde{e}}_r^T \boldsymbol{x} \le \tilde{\tilde{b}}_r\} \ge \theta_r \\ & \Leftrightarrow Pr\left\{\frac{(\tilde{\tilde{e}}_r^T \boldsymbol{x} - \tilde{\tilde{b}}_r) - (\tilde{d}_r^e(\theta)\boldsymbol{x} - \tilde{d}_r^b(\theta))}{\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2}} \le \frac{-(\tilde{d}_r^e(\theta)\boldsymbol{x} - \tilde{d}_r^b(\theta))}{\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2}}\right\} \ge \theta_r \\ & \Leftrightarrow \Phi\left(\frac{-(\tilde{d}_r^e(\theta)\boldsymbol{x} - \tilde{d}_r^b(\theta))}{\sqrt{x^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2}}\right) \ge \theta_r \\ & \Leftrightarrow \tilde{d}_r^b(\theta) - \tilde{d}_r^e(\theta)x \ge \Phi^{-1}(\theta_r)\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2} \end{aligned}$$

Since $\tilde{d}_{rj}^e(\theta) = (d_{rj}^e, \delta_{rj}^e, \gamma_{rj}^e)$ and $\tilde{d}_r^b(\theta) = (d_r^b, \delta_r^b, \gamma_r^b)$ are respectively L-R fuzzy variables, then it follows from the fuzzy arithmetic that for $x_{ij} > 0$, $\tilde{d}_r^b(\theta) - \tilde{d}_r^e(\theta)\boldsymbol{x} = (d_r^b - d_r^e \boldsymbol{x}, \delta_r^b + \gamma_r^e \boldsymbol{x}, \gamma_r^b + \delta_r^e \boldsymbol{x})$ is also an L-R fuzzy variable characterized by the following membership function,

$$\mu_{\tilde{d}_r^b(\theta)-\tilde{d}_r^e(\theta)\boldsymbol{x}}(t)=\begin{cases} L\left(\dfrac{d_r^b-d_r^e\boldsymbol{x}-t}{\delta_r^b+\gamma_r^e\boldsymbol{x}}\right) & t\le d_r^b-d_r^e\boldsymbol{x},\delta_r^b+\gamma_r^e\boldsymbol{x}>0 \\ R\left(\dfrac{t-(d_r^b-d_r^e\boldsymbol{x})}{\gamma_r^b+\delta_r^e\boldsymbol{x}}\right) & t\ge d_r^b-d_r^e\boldsymbol{x},\gamma_r^b+\delta_r^e\boldsymbol{x}>0 \end{cases}$$

Denote $G(\boldsymbol{x})=\Phi^{-1}(\theta_r)\sqrt{\boldsymbol{x}^T V_r^e\boldsymbol{x}+(\sigma_r^b)^2}$, then it follows that

$$Pos\{\theta|\tilde{d}_r^b(\theta)-\tilde{d}_r^e(\theta)\boldsymbol{x}\ge G(\boldsymbol{x})\}\ge\eta_r$$

$$\Leftrightarrow \eta_r\le\begin{cases} 1, & \text{if } G(\boldsymbol{x})\le d_r^b-d_r^e\boldsymbol{x} \\ R\left(\dfrac{G(\boldsymbol{x})-(d_r^b-d_r^e\boldsymbol{x})}{\gamma_r^b+\delta_r^e\boldsymbol{x}}\right), & \text{if } d_r^b-d_r^e\boldsymbol{x}<G(\boldsymbol{x})\le d_r^b-d_r^e\boldsymbol{x}+\gamma_r^b+\delta_r^e\boldsymbol{x} \\ 0, & \text{if } G(\boldsymbol{x})>d_r^b-d_r^e\boldsymbol{x}+\gamma_r^b+\delta_r^e\boldsymbol{x} \end{cases}$$

For $\eta_r\in(0,1)$, the above formula is equivalent to

$$G(\boldsymbol{x})-(d_r^b-d_r^e\boldsymbol{x})-R^{-1}(\eta_r)(\gamma_r^b+\delta_r^e\boldsymbol{x})\le 0$$

That is,

$$\Phi^{-1}(\theta_r)\sqrt{\boldsymbol{x}^T V_r^e\boldsymbol{x}+(\sigma_r^b)^2}-(d_r^b-d_r^e\boldsymbol{x})-R^{-1}(\eta_r)(\gamma_r^b+\delta_r^e\boldsymbol{x})\le 0$$

This completes the proof. □

Denote $X=\{\boldsymbol{x}\in\mathbf{R}^n|\Phi^{-1}(\theta_r)\sqrt{\boldsymbol{x}^T V_r^e\boldsymbol{x}+(\sigma_r^b)^2}-(d_r^b-d_r^e\boldsymbol{x})-R^{-1}(\eta_r)(\gamma_r^b+\delta_r^e\boldsymbol{x})\le 0; x_j\ge 0, j=1,2,\ldots,n\}$, then from Theorems 4.9 and 4.10, we have that (4.50) is equivalent to the following multiobjective programming problem,

$$\begin{cases} \max[\bar{f}_1,\bar{f}_2,\ldots,\bar{f}_m] \\ \text{s.t.} \begin{cases} \bar{f}_i\le\Phi^{-1}(1-\beta_i)\sqrt{\boldsymbol{x}^T V_i^c\boldsymbol{x}}+R^{-1}(\alpha_i)\gamma_{ij}^c\boldsymbol{x}+d_i^{cT}\boldsymbol{x} \\ \boldsymbol{x}\in X \end{cases} \end{cases} \tag{4.54}$$

or equivalently

$$\begin{cases} \max[H_1(\boldsymbol{x}),H_2(\boldsymbol{x}),\ldots,H_m(\boldsymbol{x})] \\ \text{s.t. } \boldsymbol{x}\in X \end{cases} \tag{4.55}$$

where $H_i(\boldsymbol{x})=\Phi^{-1}(1-\beta_i)\sqrt{\boldsymbol{x}^T V_i^c\boldsymbol{x}}+R^{-1}(\alpha_i)\gamma_{ij}^c\boldsymbol{x}+d_i^{cT}\boldsymbol{x}, i=1,2,\ldots,m.$

4.4.2.2 ε-Constraint Method

ε-constraint method was proposed by Haimes [124, 125] in 1971. The idea of this method is that we choose a main referenced objective f_{i0}, put the other objective functions into the constraints. Let's consider the following multi-objective model,

$$\begin{cases} \min\ [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})] \\ \text{s.t. } \boldsymbol{x} \in X \end{cases} \tag{4.56}$$

So we use the ε-constraint method, we can get the single objective model,

$$\begin{cases} \min\ f_{i_0}(\boldsymbol{x}) \\ \text{s.t.} \begin{cases} f_i(\boldsymbol{x}) \le \varepsilon_i,\ i = 1, 2, \ldots, m, i \ne i_0 \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{4.57}$$

where the parameter ε_i is predetermined by the decision maker, it denote the threshold value that the decision maker will accept, we denote the feasible domain of model (4.57) as X_1.

Theorem 4.11. *If $\bar{\boldsymbol{x}}$ is the optimal solution of model (4.57), then $\bar{\boldsymbol{x}}$ is a weak efficient solution of model (4.56).*

Proof. Let $\bar{\boldsymbol{x}}$ be the optimal solution of model (4.57), but it is not a weak efficient solution of model (4.56), then there exists $\boldsymbol{x}' \in X$, such that for $\forall\ i \in \{1, 2, \ldots, m\}$, $f_i(\boldsymbol{x}') < f_i(\bar{\boldsymbol{x}})$ holds. Since $\bar{\boldsymbol{x}} \in X_1$, $f_i(\bar{\boldsymbol{x}}) \le \varepsilon_i$ $(i = 1, 2, \ldots, m, i \ne i_0)$, we have

$$f_i(\boldsymbol{x}') < f_i(\bar{\boldsymbol{x}}) \le \varepsilon_i,\ i = 1, 2, \ldots, m, i \ne i_0. \tag{4.58}$$

We can obtain from (4.58) that $\boldsymbol{x}' \in X_1$, and $f_{i_0}(\boldsymbol{x}') < f_{i_0}(\bar{\boldsymbol{x}})$. This conflicts with that $\bar{\boldsymbol{x}}$ is the optimal solution. □

Theorem 4.12. *Let $\bar{\boldsymbol{x}}$ be an efficient solution of model (4.56), then there exists a parameter $\varepsilon_i (i = 1, 2, \ldots, m, i \ne i_0)$, such that $\bar{\boldsymbol{x}}$ is the optimal solution of model (4.57).*

Proof. Take $\varepsilon_i = f_i(\bar{\boldsymbol{x}})$, $(i = 1, 2, \ldots, m, i \ne i_0)$, by the definition of the efficient solution, $\bar{\boldsymbol{x}}$ is an optimal solution of model (4.57). □

So the advantages of the ε-constraint method are as follows:

1. Every efficient solution of model (4.56) can be got by properly choosing parameter $\varepsilon_i (i = 1, 2, \ldots, m, i \ne i_0)$.
2. The i_0th objective is mainly guaranteed, and the other objectives are considered meanwhile.

It is worth for us noticing that the parameter ε_i is important, we should carefully choose it. If the value of every ε_i is too small, then it is possible that the model (4.57) will have no solutions; otherwise, the value of ε_i is too large, then besides the main

objective, the other objective will lose more with higher possibility. Commonly, we can offer the decision maker $f_i^0 = \min_{x \in X} f_i(\boldsymbol{x})$ $(i = 1, 2, \ldots, m)$ and the objective value $(f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))^T$ of a certain feasible solution $\boldsymbol{x}$. And then the decision maker can decide ε_i.

4.4.3 Nonlinear Ra-Fu CCM and Ra-Fu Simulation-Based st-GA

Consider the following model,

$$\begin{cases} \max[\bar{f}_1, \bar{f}_2, \ldots, \bar{f}_m] \\ \text{s.t.} \begin{cases} Pos\{\theta | Pr\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}_i\} \geq \beta_i\} \geq \alpha_i, \ i = 1, 2, \ldots, m \\ Pos\{\theta | Pr\{g_r(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \geq \theta_r\} \geq \eta_r, \ \ r = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases}$$

where α_i, β_i, η_r and θ_r are predetermined confidence levels, $i = 1, 2, \ldots, m, r = 1, 2, \ldots, p$. If f_i, or g_r or even both of them are nonlinear functions with respect to $\boldsymbol{\xi}$, we cannot directly convert it into crisp model, then another method is introduced to solve it.

4.4.3.1 Ra-Fu Simulation for CCM

Assume that $\boldsymbol{\xi}$ is an n-dimensional Ra-Fu vector defined on the possibility space $(\Theta, \mathscr{P}(\Theta), Pos)$, and $f : \mathbf{R}^n \to \mathbf{R}$ is a measurable function. For any given confidence level α and β, we need to design the Ra-Fu simulation to find the maximal value $\bar{f}$ such that

$$Ch\{f(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}\}(\alpha) \geq \beta$$

holds. That is, we must find the maximal value $\bar{f}$ such that

$$Pos\{\theta | Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}(\theta)) \geq \bar{f}\} \geq \beta\} \geq \alpha.$$

We randomly generate θ_k from Θ such that $Pos\{\theta_k\} \geq \varepsilon$, and write $v_k = Pos\{\theta_k\}, k = 1, 2, \ldots, N$, respectively, where ε is a sufficiently small number. For any number θ_k, we search for the maximal value $\bar{f}(\theta_k)$ such that $Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}(\theta_k)) \geq \bar{f}(\theta_k)\} \geq \beta$ by stochastic simulation. For any number r, we have

$$H(r) = \frac{1}{2}\left(\max_{1 \leq k \leq N}\{v_k | \bar{f}(\theta_k) \geq r\} + \min_{1 \leq k \leq N}\{1 - v_k | \bar{f}(\theta_k) < r\}\right)$$

If follows from monotonicity that we may employ bisection search to find the maximal value r such that $H(r) \geq \alpha$. This value is an estimation of $\bar{f}$. Then the

procedure simulating the critical value of $Pos\{\theta|Pr\{f(\boldsymbol{x},\boldsymbol{\xi}(\theta)) \geq \bar{f}\} \geq \beta\} \geq \alpha$ can be summarized as follows:

Procedure `Ra-Fu simulation for CCM`

Input: The decision vector $\boldsymbol{x}$
Output: The critical value $\bar{f}$ of $Pos\{\theta|Pr\{f(\boldsymbol{x},\boldsymbol{\xi}(\theta)) \geq \bar{f}\} \geq \beta\} \geq \alpha$
Step 1. Generate θ_k from Θ such that $Pos\{\theta_k\} \geq \varepsilon$ for $k = 1, 2, \ldots, N$, where ε is a sufficiently small number;
Step 2. Find the maximal value r such that $H(r) \geq \alpha$ holds;
Step 3. Return r.

Example 4.18. In order to find the maximal value $\bar{f}$ such that

$$Ch\left\{\sqrt{\tilde{\tilde{\xi}}_1^2 + \tilde{\tilde{\xi}}_2^2 + \tilde{\tilde{\xi}}_3^2} \geq \bar{f}\right\}(0.8) \geq 0.8,$$

where $\tilde{\tilde{\xi}}_1$, $\tilde{\tilde{\xi}}_2$ and $\tilde{\tilde{\xi}}_3$ are Ra-Fu variables defined as

$$\tilde{\tilde{\xi}}_1 \sim \mathcal{N}(\tilde{\rho}_1, 1), \text{ with } \tilde{\rho}_1 = (1, 2, 3),$$

$$\tilde{\tilde{\xi}}_2 \sim \mathcal{N}(\tilde{\rho}_2, 2), \text{ with } \tilde{\rho}_2 = (2, 3, 4),$$

$$\tilde{\tilde{\xi}}_3 \sim \mathcal{N}(\tilde{\rho}_3, 1), \text{ with } \tilde{\rho}_3 = (3, 4, 5).$$

where $\tilde{\rho}_i (i = 1, 2, 3)$ are triangular fuzzy numbers. We perform the Ra-Fu simulation with 1000 cycles and obtain that $\bar{f} = 2.5604$.

4.4.3.2 Spanning Tree-Based Genetic Algorithm

Spanning tree-based genetic algorithms (abbr. st-GA) paly an important role within many fields. It generally arises in one of two ways, directly or indirectly. In some direct applications, we wish to connect a set of points using the least cost or least length collection of arcs. Frequently, the points represent physical entities such as components of a computer chip, or users of a system who need to be connected to each other or to a central service such as central processor in a computer system [2]. In indirect applications, we either (1) wish to connect some set of points using a measure of performance that on the surface bears little resemblance to the minimum spanning tree objective (sum of arc costs), or (2) the problem itself bears little resemblance to an "optimal tree" problem – in these instances, we often need to be creative in modelling the problem so that it becomes a minimum spanning tree problem. Next, we introduce the steps of spanning tree-based genetic algorithm in detail, and interested readers can refer to related literatures [5, 108, 109, 309, 341].

Representation and initialization. The genetic representation is a kind of data structure which represents the candidate solutions of the problem in coding space. Usually, different problems have different data structures or genetic representations. Here, we employ the sub-tree $I - J$ and the sub-tree $J - k$ to represent the transport pattern from plants to DCs, and from DCs to customers respectively. Each chromosome in this problem consists of three parts. The first part is J binary digits to represent the opened/closed DCs. The last two parts are two Prüfer numbers representing the distribution pattern from plants to DCs, and from DCs to customers, respectively.

In 1889, Cayley proved that there are p^{p-2} distinct labeled trees for a complete graph with p nodes. Prüfer presented the simplest proof of Cayley's formula by establishing a one-to-one correspondence between the set of spanning tree and a set of $p - 2$ digit with an integer between 1 and p inclusive [109]. For the sub-tree $I - J$, denote the plants $1, 2, \ldots, I$ as the component of set $I = \{1, 2, \ldots, I\}$ and define DCs $1, 2, \ldots, J$ as the component of the set $D = \{I + 1, I + 2, \ldots, I + J\}$. Obviously, this distribution graph has $I + J$ nodes, which means that we need $I + J - 2$ digit Prüfer numbers in the range $[1, I + J]$ to uniquely represent the sub-tree $I - J$. By using the similar ways, we produce another $J + K - 2$ digit Prüfer numbers representing sub-trees $J - K$. An illustration of a feasible chromosome representation is given in Fig. 4.9. The first sub-string is 4 binary digits, representing opened/closed DCs. The last two sub-strings are Prüfer numbers consist of 5 and 7 digits to represent distribution patterns.

Then we take the third sub-string for example to explain the encoding, feasibility check and decoding algorithm.

Encoding. The transport tree illustrated in Fig. 4.10, converted into the corresponding Prüfer number, is shown in Fig. 4.11.

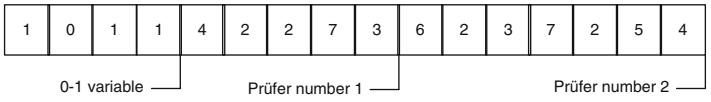


Fig. 4.9 An illustration of chromosome

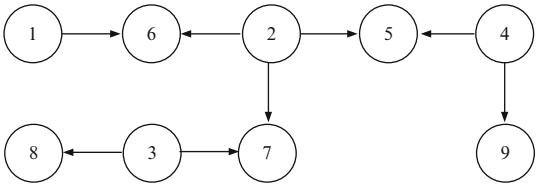


Fig. 4.10 Spanning tree

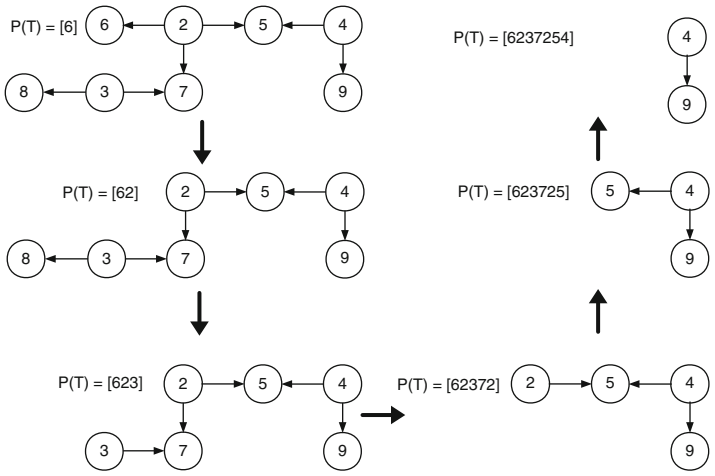


Fig. 4.11 Encoding procedure

The encoding procedure is as follows.

Procedure `Encoding a tree to a Prüfer number`

Input: tree data set
Output: A Prüfer number $P(T)$
Step 1. Let j be the lowest-numbered leaf node in the tree. Let k be the node that is the incident to node j. Then k becomes the leftmost digit of the number $P(T)$. $P(T)$ is built up by appending digits to the right; thus $P(T)$ is built and read from left to right;
Step 2. Remove j and edge (j, k) from further consideration. Thus, j is no longer considered at all and if j is the only successor of k, then j becomes a leaf node;
Step 3. If only two nodes remain to be considered, $P(T)$ has been formed with $J + K - 2$ digits between 1 and $J + K$ inclusive, so stop; otherwise, return to step 1.

Feasibility check for Prüfer number. The process of initializing a chromosome (a Prüfer number) is by choosing $J + K - 2$ digits from the range $[1, J + K]$ at random. Thus it is possible to generate some infeasible chromosomes that cannot be adapted into the transport network graph. Due to this reason, feasibility should be checked before decoding the Prüfer number into the spanning tree. As we know, Prüfer number encoding, explicitly contains information of a node degree that any node with degree d will appear exactly $d - 1$ times in the encoding. Thus, when a node appears d times in Prüfer number, the node has exactly $d + 1$ connections with other nodes.

Then we create the handling for the feasibility of the chromosome with following criterion: Denote that L_J^j and L_K^k are the number of appearance of nodes j and k

which are included in the set J and K respectively from $P(T)$. And we denote that $\overline{L_J^j}$ and $\overline{L_K^k}$ are the number of appearances of nodes j and k in $\overline{P}(T)$which are included in set J and K respectively. If $\sum_{j\in J}(L_J^j + 1) + \sum_{j\in J}\overline{L_J^j} = \sum_{k\in K}(L_K^k + 1) + \sum_{k\in K}\overline{L_K^k}$, then $P(T)$ is feasible; otherwise, $P(T)$ is infeasible. Here,we design the feasibility check and repairing procedure for the Prüfer number to be decoded into spanning tree as follows:

Produce: feasibility check by using the Ra-Fu simulation and repairing procedure for the Prüfer number. Repeat the following steps, until $\sum_{j\in J}(L_J^j + 1) + \sum_{j\in J}\overline{L_J^j} = \sum_{k\in K}(L_K^k + 1) + \sum_{k\in K}\overline{L_K^k}$.

Step 1. Determine L_J^j and L_K^k from $P(T)$, $\overline{L_J^j}$ and $\overline{L_K^k}$ from $\overline{P}(T)$.

Step 2. If $\sum_{j\in J}(L_J^j + 1) + \sum_{j\in J}\overline{L_J^j} > \sum_{k\in K}(L_K^k + 1) + \sum_{k\in K}\overline{L_K^k}$, then select one digit in $P(T)$ which contains node $j(j \in J)$ and replace it with the number $k(k \in K)$. Otherwise, select one digit in $P(T)$ which contains node $k(k \in K)$ and replace it with the number $j(j \in J)$.

Decoding. After checking the feasibility of the chromosome, the chromosome of this problem can be decoded into spanning trees in order to determine the transport pattern. Considering that the total capacity of DCs which will be opened has to satisfy the total demanded by customers, the chromosome is decoded in the backward direction. Firstly, the transport tree between opened DCs and customers is obtained by changing the capacity of the closed DCs to be zero and decoding of the last segment of chromosome. After that, the total amount required for a product on each DC is determined. Lastly, the transport tree between suppliers and opened DCs is obtained by decoding the second segment of chromosome.

The decoding procedure of the second Prüfer number shown in Fig. 4.9 and its trace table are given in Fig. 4.12 and Table 4.3, respectively. And we also give the step-by-step procedure for decoding the second Prüfer number as follow: Firstly, Let $P(T) = [\,6\ 2\ 3\ 7\ 2\ 5\ 4\,]$ be the original Prüfer number, and we have $\overline{P}(T) = [\,1\ 8\ 9\,]$ as being the set of all nodes that are not part of $P(T)$ and are designed as eligible for consideration. Node 1 is the lowest-numbered eligible node in $\overline{P}(T)$ and node 6 is the leftmost digit of $P(T)$. However, since these two nodes are not in the same set, we add an edge (1, 6) to the tree, remove node 1 from $\overline{P}(T)$ and node 6 from $P(T)$ leaving $P(T) = [\,2\ 3\ 7\ 2\ 5\ 4\,]$, and since node 6 no longer appears in the remaining part of $P(T)$, add it to $\overline{P}(T)$, so $\overline{P}(T) = [\,6\ 8\ 9\,]$. Assign the available amount of units to $x_{16} = \min\{m_1, b_6\} = 4{,}900$ which satisfies the defined constraints. Update availability $m_1 = m_1 - x_{16} = 3851$ and $b_6 = b_6 - x_{16} = 0$. Secondly, node 6 lowest-numbered eligible node in $\overline{P}(T)$ and node 2 is the leftmost digit of $P(T)$, and these two nodes are not in the same set, so we add (2, 6) to the tree, remove node 6 from $\overline{P}(T)$ and node 2 from $P(T)$ leaving $P(T) = [\,3\ 7\ 2\ 5\ 4\,]$ and $\overline{P}(T) = [\,8\ 9\,]$. Assign $x_{26} = \min\{m_2, b_6\} = 0$. Update $m_2 = m_2 - x_{26} = 0$ and $b_6 = b_6 - x_{26} = 0$. Repeat this process until finally, $P(T)$ is empty and are left with only node 4 and 9 in $\overline{P}(T)$. Since there is still an available source in node 4 and demand in node 9, we add the edge (4, 9) to the tree. The decoding procedures of the first Prüfer number is similar.

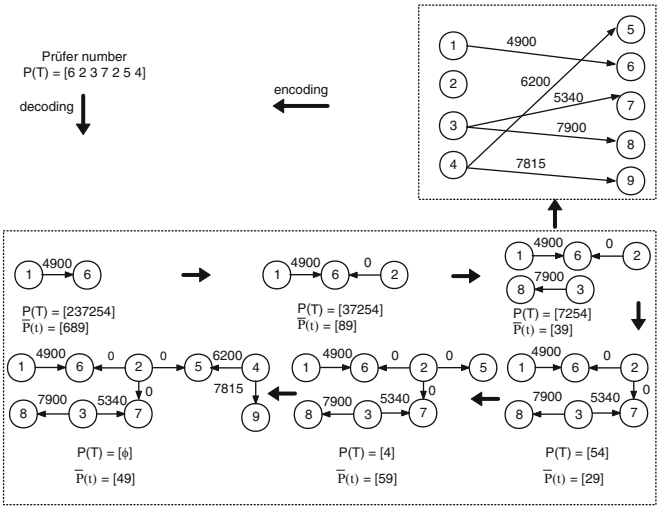


Fig. 4.12 Decoding procedure

Procedure Decoding a Prüfer number to a spanning tree

Input: A Prüfer number $P(T)$
Output: Tree data set
Step 1. Let $P(T)$ be the original Prüfer number and $\bar{P}(T)$ be the set of all nodes that are not part of $P(T)$ and are designed as eligible for consideration;
Step 2. Repeat the following substep (2.1)–(2.5) until no digit left in $P(T)$: (2.1) Let i be the lowest numbered eligible node in $\bar{P}(T)$ and j be the leftmost digit of $P(T)$. (2.2) If i and j are not in the same set S or D, add the edge (i, j) to tree T. Otherwise, select the next digit k from $P(T)$ that not included in the same set with i, exchange j with k, add the edge (j, k) to the tree T. (2.3) Remove j (or k) from $P(T)$ and i from $\bar{P}(T)$. If j (or k) does not occur anywhere in the remaining part of $P(T)$, put it into. Designate i as no longer eligible. (2.4) Assign the available amount of units to $x_{ij} = \min\{a_i, b_j\}$ (or $x_{ik} = \min\{a_i, b_k\}$) to edge (i, j) [or (j, k)], where $i \in S$ and $j, k \in D$. (2.5) Update availability $a_i = a_i - x_{ij}$ and $b_i = b_i - x_{ij}$;
Step 3. If no digits remain in $P(T)$ then there are exactly two nodes, i and j, still eligible in $\bar{P}(T)$ for consideration. Add edge (i, j) to tree T and form a tree with $m + n - 1$ edges;
Step 4. Repeat this process until $P(T)$ is empty

Genetic operators. The genetic operators mimic the process of hereditary of genes to create new offspring in each generation. The operators are used to alter the genetic composition of individuals during representation. There are two common genetic operators: crossover and mutation.

Crossover is the main genetic operator which is done to explore a new solution space. It operates on two chromosomes at a time and generates offspring by

Table 4.3 Trace of decoding procedure

m_j	d_k	j	k	y_{jk}
(8751, 0, 17600, 15800)	(6200, 4900, 5340, 7900, 7815)	1	6	4900
(3851, 0, 17600, 15800)	(6200, 0, 5340, 7900, 7815)	2	6	0
(3851, 0, 17600, 15800)	(6200, 0, 5340, 7900, 7815)	3	8	7900
(3851, 0, 9700, 15800)	(6200, 0, 5340, 0, 7815)	3	7	5340
(3851, 0, 4360, 15800)	(6200, 0, 0, 0, 7815)	2	7	0
(3851, 0, 4360, 15800)	(6200, 0, 0, 0, 7815)	2	5	0
(3851, 0, 4360, 15800)	(6200, 0, 0, 0, 7815)	4	5	6200
(3851, 0, 4360, 9600)	(0, 0, 0, 0, 7815)	4	9	7815
(3851, 0, 4360, 1785)	(0, 0, 0, 0, 0)	–	–	–

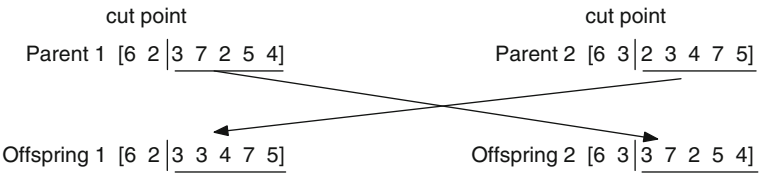
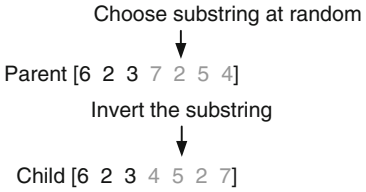


Fig. 4.13 One-point crossover process

Fig. 4.14 Inversion mutation process



combining both chromosomes' features. As a simple way to achieve crossover, a one-cut point crossover operation is used to choose a random cut-point and to generate the offspring by combining the segment of one parent to the left of the cut-point with the segment of the other parent to the right of the cut-point shown as Fig. 4.13.

Mutation is a background operator which produces spontaneous random changes in various chromosomes to explore a new solution space. A simple way to achieve mutation would be to alter one or more genes. In this paper, we adopt inversion mutation by selecting two positions within a chromosome at random and then invert the substring between these two positions illustrated in Fig. 4.14. The Prüfer numbers resulting by this mutation operation are always feasible in the sense that they can be decoded into a corresponding transport tree due to the feasibility criteria $L_s + \overline{L_s} = L_D + \overline{L_D}$ are unchanged after these operations.

Fitness evaluation and selection. The fitness evaluation is to check the solution value of the objective function subjected to the problem constraints. The single-objective problem can be easily manipulated by calculating the fitness value of each chromosome according to the objective function. However, with the multi-objective

problem, we can only calculate each objective value and cannot simply evaluate its fitness value when in practice the objective functions conflict with each other. In other words, we cannot obtain the absolute optimal solution, but can only get the Pareto optimal solution.

As to the fitness function for evaluating chromosomes, we employ the weighted sums method to contract the fitness function. Then we use the following evaluation to combine the multi-objective functions into one overall fitness function and evaluate each chromosome.

Step 1. Calculate the objective values ($F_i, i = 1, 2$).

Step 2. Chose the solution points which contain the maximum F_1 (or the minimum F_2) corresponding to each objective function value, and then compare them with the stored solution points in the previous generation and select the best points to save again.

$$\begin{aligned} F_i^{max(t)} &= \max_k \left\{ F_i^{max(t-1)}, F_i^{(t)}(X_k) | k = 1, 2, \ldots, i_{size} \right\} \\ F_i^{min(t)} &= \min_k \left\{ F_i^{min(t-1)}, F_i^{(t)}(X_k) | k = 1, 2, \ldots, i_{size} \right\} \end{aligned} \tag{4.59}$$

where $F_i^{max(t)}$, $F_i^{min(t)}$ are the maximum and minimum values of the ith objective function at generation t, respectively, $F_i^{(t)}(X_k))$ is the ith objective function value of the kth chromosome at generation t, and i_{size} is equal to the pop_{size} plus the offsprings generated after genetic operations.

Step 3. Solve the following equation to get weight for the fitness function:

$$\varepsilon_i = F_i^{max(t)} - F_i^{min(t)}, \ \omega_i = \frac{\varepsilon_i}{\sum_{i=1}^{2} \varepsilon_i}, i = 1, 2 \tag{4.60}$$

Step 4. Fitness function is obtained by combining the objective functions as follows:

$$eval(X_k) = \sum_{i=1}^{2} \omega_i F_i(X_k), k = 1, 2, \ldots, i_{size} \tag{4.61}$$

Selection provides the driving force in a GA. The selection directs the genetic search toward promising regions in the search space. During the past two decades, many selection methods have been proposed, examined and compared. Roulette wheel selection, proposed by Holland, is the best known selection type. The basic idea is to determine selection probability or survival probability for each chromosome proportional to the fitness value. Then a model roulette wheel can be made displaying these probabilities. The selection process is based on spinning the wheel the number of times equal to population size, each selecting a single chromosome for the new procedure.

Procedure `Procedure for the roulette wheel selection`

Input: i_{size} chromosomes
Output: Selected parament chromosomes
Step 1. Calculate a cumulative probability q_k for each chromosome P_k, (k=1, $2, \ldots, i_{size}$);
Step 2. Generate a random real number $r \in [0, 1]$;
Step 3. If $r \leq q_1$, then select the first chromosome P_1; otherwise select the kth chromosome $P_k (2 \leq k \leq i_{size})$ such that $q_{k-1} < r \leq q_k$;
Step 4. Repeat steps 2 and 3 for i_{size} times and obtain i_{size} copies of chromosomes;
Step 5. If the best chromosome is not selected in the next generation, replace one from the new population randomly by the best one;

Using this selection process, we can keep the best chromosome from the current generation for the next generation.

Overall procedure of the proposed method. The steps of our algorithm for solving the problem are outlined as follows.

Procedure `Spanning tree-based GA`

Input: initial data and GA parameters
Output: minimum total cost
begin
  $t \leftarrow 1$;
  initialization $P(T)$ by spanning tree-based encoding;
  fitness $eval(P)$;
  while *(not termination condition)* **do**
    crossover $P(T)$ to yield $C(T)$ by one point crossover;
    mutation $P(T)$ to yield $C(T)$ by inversion mutation;
    check the feasibility of the offspring and repair the infeasible off spring;
    fitness $eval(C)$;
    select $P(T|1)$ from $P(T)$ and $C(T)$ by roullete wheel selection;
    $t \leftarrow t + 1$;
  end
end

4.4.4 Numerical Examples

Example 4.19. Consider the following linear multi-objective problem with Ra-Fu coefficients and use the ε constraint method to resolve it.

$$\begin{cases} \max f_1(\boldsymbol{x},\boldsymbol{\xi}) = \tilde{\tilde{\xi}}_1 x_1 - \tilde{\tilde{\xi}}_2 x_2 + \tilde{\tilde{\xi}}_3 x_3 - \tilde{\tilde{\xi}}_4 x_4 + \tilde{\tilde{\xi}}_5 x_5 \\ \max f_2(\boldsymbol{x},\boldsymbol{\xi}) = c_1\tilde{\tilde{\xi}}_6 x_1 - c_2\tilde{\tilde{\xi}}_7 x_2 + c_3\tilde{\tilde{\xi}}_8 x_3 - c_4\tilde{\tilde{\xi}}_9 x_4 + c_5\tilde{\tilde{\xi}}_{10} x_5 \\ \text{s.t.} \begin{cases} x_1 + x_2 + x_3 + x_4 + x_5 \le 450 \\ x_1 - x_2 + x_3 - x_4 + x_5 \ge 110 \\ 4x_1 - 2x_2 + 1.5x_3 - x_4 + 2x_5 \le 800 \\ x_1 + 4x_2 + 2x_3 + 5x_4 + 3x_5 \le 460 \\ x_1 \ge 10, x_2 \ge 10, x_3 \ge 10, x_4 \ge 10, x_5 \ge 10 \end{cases} \end{cases} \tag{4.62}$$

where $c = (c_1, c_2, c_3, c_4, c_5) = (1.2, 0.5, 1.3, 0.8, 0.9)$,

$$\begin{array}{llll} \tilde{\tilde{\xi}}_1 \sim \mathcal{N}(\tilde{d}_1, 1), & \text{with } \tilde{d}_1 = (101, 113, 116), & \tilde{\tilde{\xi}}_2 \sim \mathcal{N}(\tilde{d}_2, 4), & \text{with } \tilde{d}_2 = (238, 241, 246), \\ \tilde{\tilde{\xi}}_3 \sim \mathcal{N}(\tilde{d}_3, 1), & \text{with } \tilde{d}_3 = (84, 87, 90), & \tilde{\tilde{\xi}}_4 \sim \mathcal{N}(\tilde{d}_4, 2), & \text{with } \tilde{d}_4 = (55, 56, 58), \\ \tilde{\tilde{\xi}}_5 \sim \mathcal{N}(\tilde{d}_5, 1), & \text{with } \tilde{d}_5 = (90, 92, 93), & \tilde{\tilde{\xi}}_6 \sim \mathcal{N}(\tilde{d}_6, 1), & \text{with } \tilde{d}_6 = (152, 156, 158), \\ \tilde{\tilde{\xi}}_7 = \mathcal{N}(\tilde{d}_7, 2), & \text{with } \tilde{d}_7 = (140, 143, 144), & \tilde{\tilde{\xi}}_8 \sim \mathcal{N}(\tilde{d}_8, 2), & \text{with } \tilde{d}_8 = (210, 214, 216), \\ \tilde{\tilde{\xi}}_9 \sim \mathcal{N}(\tilde{d}_9, 2), & \text{with } \tilde{d}_9 = (153, 157, 159), & \tilde{\tilde{\xi}}_{10} = \mathcal{N}(\tilde{d}_{10}, 2), & \text{with } \tilde{d}_{10} = (168, 172, 178). \end{array}$$

and $\tilde{d}_i (i = 1, 2, \ldots, 10)$ are triangular fuzzy variables. Then we get the chance constraint model as follows,

$$\begin{cases} \max \left[\bar{f}_1, \bar{f}_2\right] \\ \text{s.t.} \begin{cases} Pos\{\theta | Pr\{\tilde{\tilde{\xi}}_1 x_1 - \tilde{\tilde{\xi}}_2 x_2 + \tilde{\tilde{\xi}}_3 x_3 - \tilde{\tilde{\xi}}_4 x_4 + \tilde{\tilde{\xi}}_5 x_5 \ge \bar{f}_1\} \ge \beta_1\} \ge \alpha_1 \\ Pos\{\theta | Pr\{1.2\tilde{\tilde{\xi}}_6 x_1 - 1.5\tilde{\tilde{\xi}}_7 x_2 + 0.3\tilde{\tilde{\xi}}_8 x_3 - 0.8\tilde{\tilde{\xi}}_9 x_4 + 0.9\tilde{\tilde{\xi}}_{10} x_5 \ge \bar{f}_2\} \ge \beta_2\} \ge \alpha_2 \\ x_1 + x_2 + x_3 + x_4 + x_5 \le 450 \\ x_1 - x_2 + x_3 - x_4 + x_5 \ge 110 \\ 4x_1 - 2x_2 + 1.5x_3 - x_4 + 2x_5 \le 800 \\ x_1 + 4x_2 + 2x_3 + 5x_4 + 3x_5 \le 460 \\ x_1 \ge 10, x_2 \ge 10, x_3 \ge 10, x_4 \ge 10, x_5 \ge 10 \end{cases} \end{cases} \tag{4.63}$$

Take $\alpha_1 = \beta_1 = \alpha_2 = \beta_2 = 0.9$, and it follows from Theorem 4.6 that problem (4.63) is equivalent to

$$\begin{cases} \max H_1(\boldsymbol{x}) = 113.3x_1 - 241.5x_2 + 87.3x_3 - 56.2x_4 + 92.1x_5 \\ \qquad\qquad + \Phi^{-1}(0.1)\sqrt{x_1^2 + 4x_2^2 + x_3^2 + 2x_4^2 + x_5^2} \\ \max H_2(\boldsymbol{x}) = 156.2x_1 - 143.1x_2 + 214.2x_3 - 157.2x_4 + 172.6x_5 \\ \qquad\qquad + \Phi^{-1}(0.1)\sqrt{x_1^2 + 2x_2^2 + 2x_3^2 + 2x_4^2 + 2x_5^2} \\ \text{s.t.} \begin{cases} x_1 + x_2 + x_3 + x_4 + x_5 \le 450 \\ x_1 - x_2 + x_3 - x_4 + x_5 \ge 110 \\ 4x_1 - 2x_2 + 1.5x_3 - x_4 + 2x_5 \le 800 \\ x_1 + 4x_2 + 2x_3 + 5x_4 + 3x_5 \le 460 \\ x_1 \ge 10, x_2 \ge 10, x_3 \ge 10, x_4 \ge 10, x_5 \ge 10 \end{cases} \end{cases} \tag{4.64}$$

where $\Phi^{-1}(x)$ is the standard normally distributed function. Next, we use the ε constraint method to resolve the crisp problem (4.64).

Step 1. Let $H_1(\boldsymbol{x})$ be the reference constraint and $H_2(\boldsymbol{x})$ be the objective function. Then compute $\max_{x\in X} H_1(\boldsymbol{x})$ and we have $\varepsilon_0 = 24432.79$.

Step 2. Let $H_2(\boldsymbol{x})$ be the objective function and construct the following single objective problem,

$$\begin{cases} \max H_2(\boldsymbol{x}) = 156.2x_1 - 143.1x_2 + 214.2x_3 - 157.2x_4 + 172.6x_5 \\ \qquad\qquad + \Phi^{-1}(0.1)\sqrt{x_1^2 + 2x_2^2 + 2x_3^2 + 2x_4^2 + 2x_5^2} \\ \text{s.t.} \begin{cases} 113.3x_1 - 241.5x_2 + 87.3x_3 - 56.2x_4 + 92.1x_5 \\ +\Phi^{-1}(0.1)\sqrt{x_1^2 + 4x_2^2 + x_3^2 + 2x_4^2 + x_5^2} \le 24432.79 \\ x_1 + x_2 + x_3 + x_4 + x_5 \le 450 \\ x_1 - x_2 + x_3 - x_4 + x_5 \ge 110 \\ 4x_1 - 2x_2 + 1.5x_3 - x_4 + 2x_5 \le 800 \\ x_1 + 4x_2 + 2x_3 + 5x_4 + 3x_5 \le 460 \\ x_1 \ge 10, x_2 \ge 10, x_3 \ge 10, x_4 \ge 10, x_5 \ge 10 \end{cases} \end{cases} \tag{4.65}$$

Then we obtain the optimal solution $\boldsymbol{x} = (170.77, 10.00, 84.62, 10.00, 10.00)^T$.

Example 4.20. Consider the following nonlinear multi-objective problem with Ra-Fu coefficients and use the Ra-Fu simulation-based genetic algorithm to resolve it.

$$\begin{cases} \max [\bar{f}_1, \bar{f}_2] \\ \text{s.t.} \begin{cases} Pos\{\theta | Pr\{3\tilde{\tilde{\xi}}_1^2 x_1^2 - 2\tilde{\tilde{\xi}}_1\tilde{\tilde{\xi}}_2 x_1 x_2 + 1.3\tilde{\tilde{\xi}}_2^2 x_2^2 \ge \bar{f}_1\} \ge 0.9\} \ge 0.9 \\ Pos\{\theta | Pr\{2.5\tilde{\tilde{\xi}}_3^2 x_1^2 + 3\tilde{\tilde{\xi}}_3\tilde{\tilde{\xi}}_4 x_1 x_2 + 5\tilde{\tilde{\xi}}_4^2 x_2^2 \ge \bar{f}_2\} \ge 0.9\} \ge 0.9 \\ x_1 + x_2 \le 10 \\ 5x_1 - 2x_2 \ge 2 \\ x_1, x_2 \ge 0 \end{cases} \end{cases} \tag{4.66}$$

where $\xi_i (i = 1, \ldots, 4)$ are all independently Ra-Fu variables as follows

$$\tilde{\tilde{\xi}}_1 \sim \mathcal{N}(\tilde{\mu}_1, 2), \quad \text{with } \tilde{\mu}_1 = (5, 6, 7), \tilde{\tilde{\xi}}_2 \sim \mathcal{N}(\tilde{\mu}_2, 1), \text{ with } \tilde{\mu}_2 = (6.5, 8, 10),$$
$$\tilde{\tilde{\xi}}_3 \sim \mathcal{N}(\tilde{\mu}_3, 1.5), \text{ with } \tilde{\mu}_3 = (4, 5, 6), \tilde{\tilde{\xi}}_4 \sim \mathcal{N}(\tilde{\mu}_4, 2), \text{ with } \tilde{\mu}_4 = (5, 7, 8).$$

where $\tilde{\mu}_i$ are all triangular fuzzy numbers, $i = 1, \ldots, 4$.

Let the probability of crossover be 0.4 and the probability of mutation be 0.2, and after running 2200 times, you can find optimal results in Table 4.4 and Fig. 4.15.

4.5 Ra-Fu DCM

Uncertain environment, event, and the chance function are key elements in DCM. Let us redefine them in Ra-Fu decision systems, and introduce the principle of uncertainty. By uncertain environment (in this case the Ra-Fu environment) we

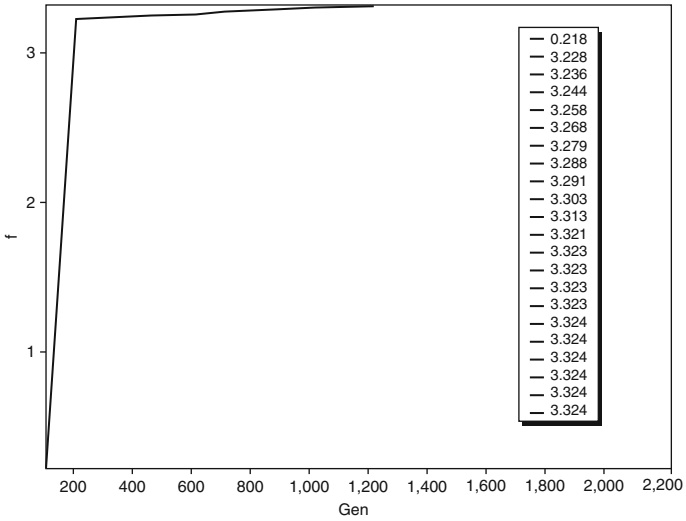


Fig. 4.15 The process of Ra-Fu simulation-based st-GA

Table 4.4 The optimal solution by Ra-Fu simulation-based st-GA

w_1	w_2	x_1	x_2	$\bar{f}_1$	$\bar{f}_2$	$\bar{f}$	Gen
0.1	0.9	2.2823	0.0925	1.1123	6.0069	5.5175	2200
0.2	0.8	1.1542	2.7342	1.4170	5.8217	4.9407	2200
0.3	0.7	0.2140	1.9297	1.5559	5.8935	4.5922	2200
0.4	0.6	1.5625	0.2912	0.9358	5.8060	3.8579	2200
0.5	0.5	2.6555	1.1940	0.8505	5.7968	3.3236	2200

mean the Ra-Fu constraints represented by

$$g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, j = 1, 2, \ldots, p \tag{4.67}$$

where $\boldsymbol{x}$ is a decision vector, and $\boldsymbol{\xi}$ is a Ra-Fu vector. By the event we mean the system of inequalities

$$h_k(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, k = 1, 2, \ldots, q \tag{4.68}$$

The chance function of an event ε characterized by (4.68) is defined as the chance measure of the event ε, i.e.,

$$f(\boldsymbol{x}) = Ch\{h_k(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, k = 1, 2, \ldots, q\} \tag{4.69}$$

subject to the uncertain environment (4.67).

For each decision $\boldsymbol{x}$ and realization $\boldsymbol{\xi}$, an event ε is said to be consistent in the uncertain environment if the following two conditions hold: (a) $h_k(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0$,

$k = 1, 2, \ldots, q$; and (b) $g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, j \in J$, where J is the index set of all dependent constraints.

Assume that there are m events ε_i characterized by $h_{ik}(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, k = 1, 2, \ldots, q_i$ for $i = 1, 2, \ldots, m$ in the uncertain environment $g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, j = 1, 2, \ldots, p$. The principle of uncertainty implies that the chance function of the ith event ε_i in the uncertain environment is

$$f_i(\boldsymbol{x}) = Ch \left\{ \begin{array}{l} h_{ik}(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, k = 1, 2, \ldots, q \\ g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, j \in J_i \end{array} \right\} \tag{4.70}$$

where J_i are defined by

$$J_i = \{j \in \{1, 2, \ldots, p\} | g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0 \text{ is a dependent constraint of } \varepsilon_i\}$$

for $i = 1, 2, \ldots, m$.

4.5.1 General Model for Ra-Fu DCM

When α-chance measure is used, we may formulate a random fuzzy DCM as follows:

$$\left\{ \begin{array}{l} \max Ch\{h_k(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, k = 1, 2, \ldots, q\}(\alpha) \\ \text{s.t. } g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, j = 1, 2, \ldots, p \end{array} \right. \tag{4.71}$$

where $\boldsymbol{x}$ is an n-dimensional decision vector, $\boldsymbol{\xi}$ is a random fuzzy vector, the event ε is characterized by $h_k(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, k = 1, 2, \ldots, q$, α is a given possibility level, and the uncertain environment is described by the random fuzzy constraints $g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, j = 1, 2, \ldots, p$.

Remark 4.13. If the Ra-Fu vector ξ degenerates to a random vector, then for any given $\alpha > 0$,

$$Ch\{h_k(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, k = 1, 2, \ldots, q\}(\alpha) \equiv Pr\{h_k(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, k = 1, 2, \ldots, q\}.$$

Thus the model (4.71) becomes

$$\left\{ \begin{array}{l} \max Pr\{h_k(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, k = 1, 2, \ldots, q\} \\ \text{s.t. } g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, j = 1, 2, \ldots, p \end{array} \right. \tag{4.72}$$

which is a standard stochastic DCM.

Remark 4.14. If the Ra-Fu vector $\boldsymbol{\xi}$ degenerates to a fuzzy vector, then for any given $\alpha > 0$,

$$Ch\{h_k(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, k = 1, 2, \ldots, q\}(\alpha) = 1$$

if $Pos\{h_k(x,\xi) \leq 0, k = 1,2,\ldots,q\} \geq \alpha$, and 0 otherwise. Roughly speaking, maximizing the chance $Ch\{h_k(\boldsymbol{x},\boldsymbol{\xi}) \leq 0, k = 1,2,\ldots,q\}(\alpha)$ implies maximizing the possibility $Pos\{h_k(\boldsymbol{x},\boldsymbol{\xi}) \leq 0, k = 1,2,\ldots,q\}$. Thus the model (4.71) becomes

$$\begin{cases} \max Pos\{h_k(\boldsymbol{x},\boldsymbol{\xi}) \leq 0, k = 1,2,\ldots,q\} \\ \text{s.t. } g_j(\boldsymbol{x},\boldsymbol{\xi}) \leq 0, j = 1,2,\ldots,p \end{cases} \tag{4.73}$$

which is a standard fuzzy DCM.

If there are multiple events in the uncertain environment, then we have the following Ra-Fu dependent-chance multiobjective decision making model,

$$\begin{cases} \max \begin{bmatrix} Ch\{f_1(\boldsymbol{x},\boldsymbol{\xi}) \leq 0\}(\alpha_1) \\ Ch\{f_2(\boldsymbol{x},\boldsymbol{\xi}) \leq 0\}(\alpha_2) \\ \cdots \\ Ch\{f_m(x,\xi) \leq 0\}(\alpha_m) \end{bmatrix} \\ \text{s.t. } g_j(\boldsymbol{x},\boldsymbol{\xi}) \leq 0, j = 1,2,\ldots,p \end{cases} \tag{4.74}$$

where the events ε_i are characterized by $f_i(\boldsymbol{x},\boldsymbol{\xi}) \leq 0$ and α_i are given possibility levels, $i = 1,2,\ldots,m$, respectively.

Ra-Fu dependent-chance goal programming is employed to formulate Ra-Fu decision systems according to the priority structure and target levels set by the decision-maker,

$$\begin{cases} \min \sum_{j=1}^{l} P_j \sum_{i=1}^{m} (u_{ij} d_i^+ + v_{ij} d_i^-) \\ \text{s.t.} \begin{cases} Ch\left\{ f_i(\boldsymbol{x},\boldsymbol{\xi}) \leq 0 \right\}(\alpha_i) + d_i^- - d_i^+ = b_i, & i = 1,2,\ldots,m \\ g_j(\boldsymbol{x},\boldsymbol{\xi}) \leq 0, & j = 1,2,\ldots,p \\ d_i^-, d_i^+ \geq 0, & i = 1,2,\ldots,m \end{cases} \end{cases} \tag{4.75}$$

where P_j is the preemptive priority factor which express the relative importance of various goals, $P_j >> P_{j+1}$, for all j, u_{ij} is the weighting factor corresponding to positive deviation for goal i with priority j assigned, u_{ij} is the weighting factor corresponding to negative deviation for goal i with priority j assigned, d_i^+ is the positive deviation from the target of goal i, d_i^- is the negative deviation from the target of goal i, α_i is the given possibility level, g_j is a function in system constraints, b_i is the target value according to goal i, l is the number of priorities, m is the number of goal constraints, and p is the number of system constraints.

4.5.2 Linear Ra-Fu DCM and Goal Programming Method

In this section, we restrict our attention on the linear multiobjective programming problem with Ra-Fu parameters. We firstly introduce a class of Ra-Fu DCMs

which can be directly transformed into crisp ones and secondly apply the goal programming method to solve them.

4.5.2.1 Crisp Equivalent Model

Consider the following model,

$$\begin{cases} \max\left[\tilde{\bar{c}}_1^T \boldsymbol{x}, \tilde{\bar{c}}_2^T \boldsymbol{x}, \ldots, \tilde{\bar{c}}_m^T \boldsymbol{x}\right] \\ \text{s.t.} \begin{cases} \tilde{\bar{e}}_r^T \boldsymbol{x} \le \tilde{\bar{b}}_r, r = 1, 2, \ldots, p \\ \boldsymbol{x} \ge 0 \end{cases} \end{cases} \tag{4.76}$$

where $\boldsymbol{x} \in X \subset \mathbf{R}^n$, $\tilde{\bar{c}}_i = (\tilde{\bar{c}}_{i1}, \tilde{\bar{c}}_{i2}, \ldots, \tilde{\bar{c}}_{in})^T$, $\tilde{\bar{e}}_r = (\tilde{\bar{e}}_{r1}, \tilde{\bar{e}}_{r2}, \ldots, \tilde{\bar{e}}_{rn})^T$ are Ra-Fu vectors, $i = 1, 2, \ldots, m$, and $\tilde{\bar{b}}_r$ are fuzzy variables, $r = 1, 2, \ldots, p$. Then by the definition of chance measure of Ra-Fu variables, we have the following dependent-chance model of (4.76),

$$\begin{cases} \max\left[Ch\{\tilde{\bar{c}}_i^T \boldsymbol{x} \ge \bar{f}_i\}(\alpha_i), i = 1, 2, \ldots, m\right] \\ \text{s.t.} \begin{cases} Ch\{\tilde{\bar{e}}_r^T \boldsymbol{x} \le \tilde{\bar{b}}_r\}(\eta_r) \ge \theta_r, r = 1, 2, \ldots, p \\ \boldsymbol{x} \ge 0 \end{cases} \end{cases} \tag{4.77}$$

where $\alpha_i, \eta_r, \theta_r$ are given confidence levels, f_i are predetermined objective value, $i = 1, 2, \ldots, m, r = 1, 2, \ldots, p$. Then by (4.22), problem (4.77) can be rewritten as

$$\begin{cases} \max\left[\sup\{\beta_i | Pos\{\theta | Pr\{\tilde{\bar{c}}_i^T \boldsymbol{x} \ge \bar{f}_i\} \ge \beta_i\} \ge \alpha_i\}, i = 1, 2, \ldots, m\right] \\ \text{s.t.} \begin{cases} Pos\{\theta | Pr\{\tilde{\bar{e}}_r^T \boldsymbol{x} \le \tilde{\bar{b}}_r\} \ge \theta_r\} \ge \eta_r, r = 1, 2, \ldots, p \\ \boldsymbol{x} \ge 0 \end{cases} \end{cases} \tag{4.78}$$

One way of solving the dependent-chance multiobjective programming model is to convert the objectives and constraints of problem (4.78) into their respective crisp equivalents. As we know, this process is usually a hard work and only successful for some special cases. Next, we will consider a special case and present the result in this section.

Theorem 4.13. *Assume that the Ra-Fu vector $\tilde{\bar{c}}_i = (\tilde{\bar{c}}_{i1}, \tilde{\bar{c}}_{i2}, \ldots, \tilde{\bar{c}}_{in})^T$ is normally distributed with mean vector $\tilde{d}_i^c(\theta) = (\tilde{d}_{i1}^c(\theta), \tilde{d}_{i2}^c(\theta), \ldots, \tilde{d}_{in}^c(\theta))^T$ and positive definite covariance matrix V_i^c, written as $\tilde{\bar{c}}_i \sim \mathcal{N}(d_i^c(\theta), V_i^c)$, where $\tilde{d}_{ij}^c(\theta)$ is a fuzzy variable characterized by the following membership function,*

$$\mu_{\tilde{d}_{ij}^c(\theta)}(t) = \begin{cases} L\left(\dfrac{d_{ij}^c - t}{\delta_{ij}^c}\right), t \le d_{ij}^c, \delta_{ij}^c > 0 \\ R\left(\dfrac{t - d_{ij}^c}{\gamma_{ij}^c}\right), t \ge d_{ij}^c, \gamma_{ij}^c > 0 \end{cases} \quad \theta \in \Theta \tag{4.79}$$

where $\delta_{ij}^c, \gamma_{ij}^c$ *are positive numbers expressing the left and right spreads of* $\tilde{d}_{ij}(\theta)$, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$, *and reference functions* $L, R : [0, 1] \to [0, 1]$ *with* $L(1) = R(1) = 0$ *and* $L(0) = R(0) = 1$ *are non-increasing, continuous functions. Then we have that*

$$\sup\{\beta_i | Pos\{\theta | Pr\{\tilde{\tilde{c}}(\omega)_i^T \boldsymbol{x} \geq \bar{f}_i\} \geq \beta_i\} \geq \alpha_i\} = \Phi\left(\frac{R^{-1}(\alpha_i)\gamma_{ij}^c + d_i^{cT}\boldsymbol{x} - \bar{f}_i}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right).$$

Proof. From Theorem 4.9, we have that

$$\begin{aligned} &Pos\{\theta | Pr\{\tilde{\tilde{c}}(\theta)_i^T \boldsymbol{x} \geq \bar{f}_i\} \geq \beta_i\} \geq \alpha_i \\ &\quad \Leftrightarrow \bar{f}_i \leq \Phi^{-1}(1 - \beta_i)\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}} + R^{-1}(\alpha_i)\gamma_i^c \boldsymbol{x} + d_i^{cT}\boldsymbol{x} \end{aligned}$$

Then it follows that $\beta_i \leq 1 - \Phi\left(\frac{\bar{f}_i - (R^{-1}(\alpha_i)\gamma_i^c x + d_i^{cT} x)}{\sqrt{x^T V_i^c x}}\right)$. Since $\Phi(-x) = 1 - \Phi(x)$, then we have

$$\beta_i \leq \Phi\left(\frac{R^{-1}(\alpha_i)\gamma_i^c \boldsymbol{x} + d_i^{cT}\boldsymbol{x} - \bar{f}_i}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right).$$

Thus

$$\sup\{\beta_i | Pos\{\theta | Pr\{\tilde{\tilde{c}}(\theta)_i^T \boldsymbol{x} \geq \bar{f}_i\} \geq \beta_i\} \geq \alpha_i\} = \Phi\left(\frac{R^{-1}(\alpha_i)\gamma_i^c \boldsymbol{x} + d_i^{cT}\boldsymbol{x} - \bar{f}_i}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right).$$

This completes the proof. □

By Theorem 4.10, we have that $Pos\{\theta | Pr\{\tilde{\tilde{e}}_r^T \boldsymbol{x} \leq \bar{\tilde{b}}_r\} \geq \theta_r\} \geq \eta_r$ is equivalent to

$$\Phi^{-1}(\theta_r)\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2} - (d_r^b - d_r^e \boldsymbol{x}) - R^{-1}(\eta_r)(\gamma_r^b + \delta_r^e \boldsymbol{x}) \leq 0,$$

then (4.78) can be rewritten as

$$\begin{cases} \max \left[\Phi\left(\dfrac{R^{-1}(\alpha_i)\gamma_i^c \boldsymbol{x} + d_i^{cT}\boldsymbol{x} - \bar{f}_i}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right), i = 1, 2, \ldots, m \right] \\ \text{s.t. } \boldsymbol{x} \in X \end{cases} \tag{4.80}$$

where $X = \{\boldsymbol{x} \in R^n | \Phi^{-1}(\theta_r)\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2} - (d_r^b - d_r^e \boldsymbol{x}) - R^{-1}(\eta_r)(\gamma_r^b + \delta_r^e \boldsymbol{x}) \leq 0, \boldsymbol{x} \geq 0\}$. Then (4.80) is a typical nonlinear multi-objective programming problem without uncertain parameters, which can be easily solved by traditional method.

4.5.2.2 Goal Programming Method

The goal programming method is initialized by Charnes and Cooper [46] in 1961. After that, Ijiri [140], Lee [189], Kendall and Lee [156], and Ignizio [139] deeply researched and widely developed it. When dealing with many multi-objective decision making problems, it is widely applied since it could provide with a technique which is accepted by many decision makers, that is, it could point out the preference information and harmoniously inosculate it into the model.

The basic idea of goal programming method is that, for the objective function $\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))^T$, decision makers give a goal value $\boldsymbol{f}^o = (f_1^o, f_2^o, \ldots, f_m^o)^T$ such that every objective function $f_i(\boldsymbol{x})$ approximates the goal value f_i^o as closely as possible. Let $d_p(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{f}^o) \in \mathbf{R}^m$ be the deviation between $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{f}^o$, then consider the following problem,

$$\min_{\boldsymbol{x} \in X} d_p(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{f}^o) \tag{4.81}$$

where the goal value $\boldsymbol{f}^o$ and the weight vector $\boldsymbol{w}$ is predetermined by the decision maker. The weight w_i expresses the importance factor that the objective function $f_i(\boldsymbol{x})$ $(i = 1, 2, \ldots, m)$ approximates the goal value f_i^o, $1 \leq p \leq \infty$.

When $p = 1$, it is recalled the simple goal programming method which is most widely used. Then we have,

$$d_p(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{f}^o) = \sum_{i=1}^{m} w_i |\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}^o|.$$

Since there is the notation $|\cdot|$ in $d_p(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{f}^o)$, it isn't a differentiable function any more. Therefore, denote that

$$d_i^+ = \frac{1}{2}(|f_i(\boldsymbol{x}) - f_i^o| + (f_i(\boldsymbol{x}) - f_i^o)),$$
$$d_i^- = \frac{1}{2}(|f_i(\boldsymbol{x}) - f_i^o| - (f_i(\boldsymbol{x}) - f_i^o)).$$

where d_i^+ expresses the quantity that $f_i(\boldsymbol{x})$ exceeds f_i^o and d_i^- expresses the quantity that $f_i(\boldsymbol{x})$ is less than f_i^o. It is easy to prove that, for any $i = 1, 2, \ldots, m$,

$$\begin{aligned} &d_i^+ + d_i^- = |f_i(\boldsymbol{x}) - f_i^o| \\ &d_i^+ - d_i^- = f_i(\boldsymbol{x}) - f_i^o \\ &d_i^+ d_i^- = 0,\ d_i^+, d_i^- \geq 0 \end{aligned} \tag{4.82}$$

When $p = 1$, problem (4.81) can be rewritten as,

$$\begin{cases} \min \sum_{i=1}^{m} w_i (d_i^+ + d_i^-) \\ \text{s.t.} \begin{cases} f_i(\boldsymbol{x}) + d_i^+ - d_i^- = f_i^o, \ i = 1, 2, \ldots, m \\ d_i^+ d_i^- = 0, d_i^+, d_i^- \geq 0, \ i = 1, 2, \ldots, m \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{4.83}$$

In order to easily solve the problem (4.83), abandon the constraint $d_i^+ d_i^- = 0$ $(i = 1, 2, \ldots, m)$ and we have

$$\begin{cases} \min \sum_{i=1}^{m} w_i (d_i^+ + d_i^-) \\ \text{s.t.} \begin{cases} f_i(\boldsymbol{x}) + d_i^+ - d_i^- = f_i^o, \ i = 1, 2, \ldots, m \\ d_i^+, d_i^- \geq 0, \qquad\qquad i = 1, 2, \ldots, m \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{4.84}$$

Theorem 4.14. *If $(\boldsymbol{x}, \bar{\boldsymbol{d}}^+, \bar{\boldsymbol{d}}^-)$ is the optimal solution of problem (4.84), $\bar{\boldsymbol{x}}$ is doubtlessly the optimal solution of problem (4.81), where $\bar{\boldsymbol{d}}^+ = (\bar{d}_1^+, \bar{d}_2^+, \ldots, \bar{d}_m^+)$ and $\bar{\boldsymbol{d}}^- = (\bar{d}_1^-, \bar{d}_2^-, \ldots, \bar{d}_m^-)$*

Proof. Since $(\boldsymbol{x}, \bar{\boldsymbol{d}}^+, \bar{\boldsymbol{d}}^-)$ is the optimal solution of problem (4.84), we have $\boldsymbol{x} \in X$, $\bar{\boldsymbol{d}}^+ \geq 0$, $\bar{\boldsymbol{d}}^- \geq 0$ and

$$f_i(\boldsymbol{x}) + \bar{d}_i^+ - \bar{d}_i^- = f_i^o, \ i = 1, 2, \ldots, m. \tag{4.85}$$

1. If $\bar{d}_i^+ = \bar{d}_i^- = 0$, we have $f_i(\boldsymbol{x}) = f_i^o$, which means $\boldsymbol{x}$ is the optimal solution problem (4.81).

2. If there exists $i_0 \in \{1, 2, \ldots, m\}$ such that $f_i(\boldsymbol{x}) \neq f_i^o$, $\bar{d}_i^+ \bar{d}_i^- = 0$ doubtlessly holds. If not, we have $\bar{d}_i^+ > 0$ and $\bar{d}_i^- > 0$. We respectively discuss them as follows.

i. If $\bar{d}_i^+ - \bar{d}_i^- > 0$, for $i \in \{1, 2, \ldots, m\}$, let

$$\tilde{d}_i^+ = \begin{cases} \bar{d}_i^+ - \bar{d}_i^-, \ i = i_0 \\ \bar{d}_i^+, \qquad i \neq i_0 \end{cases} \quad \tilde{d}_i^- = \begin{cases} 0, \quad i = i_0 \\ \bar{d}_i^-, \ i \neq i_0 \end{cases} \tag{4.86}$$

Thus, $\tilde{d}_{i_0}^+ < \bar{d}_{i_0}^+$ and $\tilde{d}_{i_0}^- < \bar{d}_{i_0}^-$ both hold. It follows from (4.85) and (4.86) that,

$$f_i(\boldsymbol{x}) + \tilde{d}_i^+ - \tilde{d}_i^- = \begin{cases} f_i(\boldsymbol{x}) + 0 - (\bar{d}_i^+ - \bar{d}_i^-) = f_i^o, \ i = i_0 \\ f_i(\boldsymbol{x}) + \bar{d}_i^+ - \bar{d}_i^- = f_i^o, \qquad i \neq i_0 \end{cases}$$

We also know $\boldsymbol{x} \in X$, $\tilde{d}_i^+ \geq 0$ and $\tilde{d}_i^- \geq 0$. Denote $\tilde{\boldsymbol{d}}^+ = (\tilde{d}_1^+, \tilde{d}_2^+, \ldots, \tilde{d}_m^+)$ and $\tilde{\boldsymbol{d}}^- = (\tilde{d}_1^-, \tilde{d}_2^-, \ldots, \tilde{d}_m^-)$, then we have $(\boldsymbol{x}, \tilde{\boldsymbol{d}}^+, \tilde{\boldsymbol{d}}^-)$ is a feasible solution of problem (4.84). If follows from $\tilde{d}_{i_0}^+ < \bar{d}_{i_0}^+$ and $\tilde{d}_{i_0}^- < \bar{d}_{i_0}^-$ that,

$$\sum_{i=1}^{m}(\tilde{d}_{i_0}^+ + \tilde{d}_{i_0}) < \sum_{i=1}^{m}(\bar{d}_{i_0}^+ + \bar{d}_{i_0}) \tag{4.87}$$

this conflict with the assumption that $(\boldsymbol{x}, \bar{\boldsymbol{d}}^+, \bar{\boldsymbol{d}}^-)$ is the optimal solution of problem (4.84).

(ii) If $\bar{d}_i^+ - \bar{d}_i^- < 0$, for $i \in \{1, 2, \ldots, m\}$, let

$$\tilde{d}_i^+ = \begin{cases} 0, & i = i_0 \\ \bar{d}_i^+, & i \neq i_0 \end{cases} \quad \tilde{d}_i^- = \begin{cases} -(\bar{d}_i^+ - \bar{d}_i^-), & i = i_0 \\ \bar{d}_i^-, & i \neq i_0 \end{cases} \tag{4.88}$$

We can similarly prove that it conflicts with the assumption that $(\boldsymbol{x}, \bar{\boldsymbol{d}}^+, \bar{\boldsymbol{d}}^-)$ is the optimal solution of problem (4.84).

So far, we have proved that $(\boldsymbol{x}, \bar{\boldsymbol{d}}^+, \bar{\boldsymbol{d}}^-)$ is the optimal solution of problem (4.83). Since the feasible region of problem (4.83) is included in the one of problem (4.84), $(\boldsymbol{x}, \bar{\boldsymbol{d}}^+, \bar{\boldsymbol{d}}^-)$ is the optimal solution of problem (4.84). Next, we will prove that $(\boldsymbol{x}, \bar{\boldsymbol{d}}^+, \bar{\boldsymbol{d}}^-)$ is the optimal solution of problem (4.81). For any feasible solution $(\boldsymbol{x}, \boldsymbol{d}^+, \boldsymbol{d}^-)$, it follows from (4.82) that,

$$|f_i(\boldsymbol{x}) - f_i^o| = d_i^+ + d_i^-, \ |f_i(\bar{\boldsymbol{x}}) - f_i^o| = \bar{d}_i^+ + \bar{d}_i^-, \ i = 1, 2, \ldots, m.$$

For any $\boldsymbol{x} \in X$, since

$$\sum_{i=1}^{m}|f_i(\bar{\boldsymbol{x}}) - f_i^o| = \sum_{i=1}^{m}(\bar{d}_i^+ + \bar{d}_i^-) \leq \sum_{i=1}^{m}(d_i^+ + d_i^-) = \sum_{i=1}^{m}|f_i(\boldsymbol{x}) - f_i^o|,$$

this means that $\bar{\boldsymbol{x}}$ is the optimal solution of problem (4.81). □

4.5.3 *Nonlinear Ra-Fu DCM and Ra-Fu Simulation-Based rw-GA*

Consider the following model,

$$\begin{cases} \max[\bar{f}_1, \bar{f}_2, \ldots, \bar{f}_m] \\ \text{s.t.} \begin{cases} Pos\{\theta | Pr\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}_i\} \geq \beta_i\} \geq \alpha_i, \ i = 1, 2, \ldots, m \\ Pos\{\theta | Pr\{g_r(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \geq \theta_r\} \geq \eta_r, \ \ r = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases}$$

where α_i, β_i, η_r and θ_r are predetermined confidence levels, $i = 1, 2, \ldots, m, r = 1, 2, \ldots, p$. If $f_i(\boldsymbol{x}, \boldsymbol{\xi})$ or $g_r(\boldsymbol{x}, \boldsymbol{\xi})$ or both of them are all nonlinear functions with respect to $\boldsymbol{\xi}$, we cannot directly convert it into crisp model, then another method is introduced to solve it.

4.5.3.1 Ra-Fu Simulation for DCM

Assume that ξ is an n-dimensional Ra-Fu vector defined on the possibility space $(\Theta, \mathscr{P}(\Theta), Pos)$, and $f : \mathbf{R}^n \to \mathbf{R}$ is a measurable function. For any confidence level α, we design a Ra-Fu simulation to compute the α-chance $Ch\{f(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}\}(\alpha)$. Equivalently, we should find the supremum $\bar{\beta}$ such that

$$Pos\{\theta | Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}(\theta)) \geq \bar{f}\} \geq \bar{\beta}\} \geq \alpha.$$

We randomly generate θ_k from Θ such that $Pos\{\theta_k\} \geq \varepsilon$, and write $v_k = Pos\{\theta_k\}, k = 1, 2, \ldots, N$, respectively, where ε is a sufficiently small number. For any number θ_k, by using stochastic simulation, we can estimate the probability $g(\theta_k) = Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}(\theta_k)) \geq \bar{f}\}$. For any number r, we set

$$L(r) = \frac{1}{2}\left(\max_{1 \leq k \leq N}\{v_k | g(\theta_k) \geq r\} + \min_{1 \leq k \leq N}\{1 - v_k | g(\theta_k) < r\}\right)$$

If follows from monotonicity that we may employ bisection search to find the maximal value r such that $L(r) \geq \alpha$. This value is an estimation of L. We summarize this process as follows.

Then the procedure simulating the α-chance $Ch\{f(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}\}(\alpha)$ can be summarized as follows:

Procedure `Ra-Fu simulation for DCM`

Input: The decision vector $\boldsymbol{x}$
Output: The α-chance $Ch\{f(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}\}(\alpha)$
Step 1. Randomly sample θ_k from Θ such that $Pos\{\theta_k\} \geq \varepsilon$ for $k = 1, 2, \ldots, N$, where ε is a sufficiently small number;
Step 2. Find the maximal value r such that $L(r) \geq \alpha$ holds;
Step 3. Return r.

Example 4.21. We employ the Ra-Fu simulation to calculate the chance measure $Ch\left\{\sqrt{\tilde{\tilde{\xi}}_1^2 + \tilde{\tilde{\xi}}_2^2 + \tilde{\tilde{\xi}}_3^2} \geq 3\right\}(0.9)$, where $\tilde{\tilde{\xi}}_1$, $\tilde{\tilde{\xi}}_2$ and $\tilde{\tilde{\xi}}_3$ are Ra-Fu variables defined as

$$\begin{aligned}
&\tilde{\tilde{\xi}}_1 \sim \mathscr{N}(\tilde{\rho}_1, 1), \text{ with } \tilde{\rho}_1 = (1, 2, 3), \\
&\tilde{\tilde{\xi}}_2 \sim \mathscr{N}(\tilde{\rho}_2, 2), \text{ with } \tilde{\rho}_2 = (2, 3, 4), \\
&\tilde{\tilde{\xi}}_3 \sim \mathscr{N}(\tilde{\rho}_3, 1), \text{ with } \tilde{\rho}_3 = (3, 4, 5).
\end{aligned}$$

A run of Ra-Fu simulation with 1000 cycles shows that

$$Ch\left\{\sqrt{\tilde{\xi}_1^2+\tilde{\xi}_2^2+\tilde{\xi}_3^2}\geq 3.2\right\}(0.9)=0.9870.$$

4.5.3.2 Random Weight-Based GA

Ishibuchi et al. [144] proposed a weight-sum based fitness assignment method, called random-wight Genetic Algorithm (rw-GA) to obtain a variable search direction toward the Pareto frontier. Weighted-sum approach can be viewed as an extension of methods used in the conventional approach for the multiobjective optimizations to the GA. It assigns weights to each objective function and combines the weighted objectives into a single objective function. Typically, there are two types of search behavior in the objective space: fixed-direction search and multiple-direction search, as demonstrated in Figs. 4.16 and 4.17. The random-weight approach gives the genetic algorithms a tendency to demonstrate a variable search direction, therefore, able to sample the area uniformly over the entire frontier.

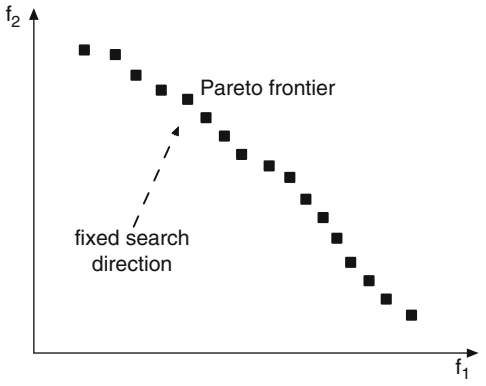


Fig. 4.16 Search in a fixed direction in criterion space

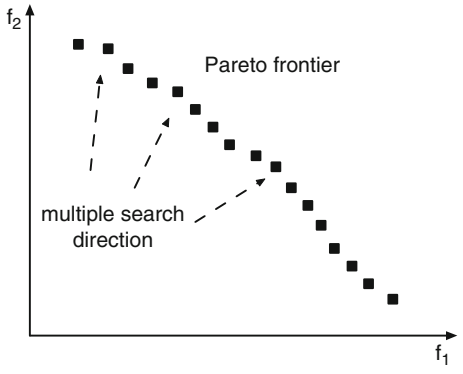


Fig. 4.17 Search in multiple directions in criterion space

Suppose that we are going to maximize q objective functions. The weighted-sum objective is given as follows:

$$z = \sum_{k=1}^{q} w_k f_k(\boldsymbol{x}) \tag{4.89}$$

The random weights w_k are calculated by the equation

$$w_k = \frac{r_k}{\sum_{j=1}^{q} r_j}, k = 1, 2, \ldots, q \tag{4.90}$$

where r_i are nonnegative random numbers.

Before selecting a pair of parents for crossover operation, a new set of random weights is specified by (4.90), and the fitness values for each individual are calculated by (4.89). The selection probability p_i for individual i is then defined by the following linear scaling function:

$$p_i = \frac{z_i - z_{\min}}{\sum_{j=1}^{pop-size}(z_i - z_{\min})} \tag{4.91}$$

where $z_{\min}$ is the worst fitness value in the current population.

A tentative set of Pareto solutions is stored and updated at each generation. For a problem with q objectives, there are q extreme points int he Pareto solutions, each of which maximizes one objective. An elite preserving strategy is suggested for putting the n extreme points plus some randomly selected Pareto solutions into the next population. Let N_{pop} denote the population size and N_{elite} denote the number of elite solutions to preserve. The overall structure of their implementation of genetic algorithms is given as follows:

Step 1. Initialization. Randomly generate an initial population containing N_{pop} individuals.

Step 2. Evaluation. Calculation the values of q objective functions for each individual. Update a tentative set of Pareto solutions.

Step 3. Selection. Repeat the following steps to select $(N_{pop} - N_{elite})$ pairs of parent: Specify random weighted by (4.90), calculate fitness function by (4.89), calculate selection probability by (4.91), and select a pair of parent individuals for a crossover operation.

Step 4. Crossover. For each pair selected, apply a crossover operation to generate offspring.

Step 5. Mutation. Apply a mutation operation to each offspring generated by the crossover operation.

Step 6. Elitist strategy. Randomly select N_{pop} individuals from the tentative set of Pareto solutions. Add the selected solutions N_{elite} to $(N_{pop} - N_{elite})$ individuals generated in the foregoing steps to construct a population of N_{pop} of individuals.

Step 7. Termination test. If a pre-specified stopping condition is satisfied, stop the run; otherwise return stop 1.

The procedure is given as follow:

Procedure `Random weight-based genetic algorithm(rw-GA)`

Input: the objective $f_i(\boldsymbol{x})$ of each chromosome $\boldsymbol{x}^i$, $i = 1, 2, \ldots, q$, $\forall i \in popSize$

Output: fitness value $eval(\boldsymbol{x}^i)$, $\forall i \in popSize$

begin

$r_j \leftarrow$ random[0,1], $j = 1, 2, \ldots, q$;//non-negative random number;

$w_k \leftarrow r_k / \sum_{i=1}^{q} r_i, k = 1, 2, \ldots, q$;

$eval(\boldsymbol{x}^i) \leftarrow \sum_{i=1}^{q} w_k (f_i(\boldsymbol{x}^i) - z_k^{min})$, $\forall i$;

Output: $eval(\boldsymbol{x}^i)$, $\forall i$

end

4.5.4 Numerical Examples

Example 4.22. Consider the following linear dependent chance multi-objective model with Ra-Fu coefficients,

$$\begin{cases} \max f_1(\boldsymbol{x}, \boldsymbol{\xi}) = Ch\left\{\tilde{\tilde{\xi}}_1 x_1 + \tilde{\tilde{\xi}}_2 x_2 + \tilde{\tilde{\xi}}_3 x_3 + \tilde{\tilde{\xi}}_4 x_4 + \tilde{\tilde{\xi}}_5 x_5 \geq \bar{f_1}\right\} (0.9) \\ \max f_2(\boldsymbol{x}, \boldsymbol{\xi}) = Ch\left\{c_1\tilde{\tilde{\xi}}_6 x_1 + c_2\tilde{\tilde{\xi}}_7 x_2 + c_3\tilde{\tilde{\xi}}_8 x_3 + c_4\tilde{\tilde{\xi}}_9 x_4 + c_5\tilde{\tilde{\xi}}_{10} x_5 \geq \bar{f_2}\right\} (0.9) \\ \text{s.t.} \begin{cases} x_1 + x_2 + x_3 + x_4 + x_5 \leq 350 \\ x_1 + x_2 + x_3 + x_4 + x_5 \geq 300 \\ 4x_1 + 2x_2 + 1.5x_3 + x_4 + 2x_5 \leq 1085 \\ x_1 + 4x_2 + 2x_3 + 5x_4 + 3x_5 \leq 660 \\ x_1 \geq 20, x_2 \geq 20, x_3 \geq 20, x_4 \geq 20, x_5 \geq 20 \end{cases} \end{cases} \tag{4.92}$$

where $c = (c_1, c_2, c_3, c_4, c_5) = (1.2, 0.5, 1.3, 0.8, 0.9)$, and

$$\begin{aligned} &\tilde{\tilde{\xi}}_1 \sim \mathcal{N}(\tilde{u}_1, 1), \text{ with } \tilde{u}_1 \sim (113, 4, 4)_{LR}, \ \tilde{\tilde{\xi}}_2 \sim \mathcal{N}(\tilde{u}_2, 4), \quad \text{with } \tilde{u}_2 \sim (241, 7, 7)_{LR}, \\ &\tilde{\tilde{\xi}}_3 \sim \mathcal{N}(\tilde{u}_3, 1), \text{ with } \tilde{u}_3 \sim (87, 2, 2)_{LR}, \ \tilde{\tilde{\xi}}_4 \sim \mathcal{N}(\tilde{u}_4, 2), \quad \text{with } \tilde{u}_4 \sim (56, 2, 2)_{LR}, \\ &\tilde{\tilde{\xi}}_5 \sim \mathcal{N}(\tilde{u}_5, 1), \text{ with } \tilde{u}_5 \sim (92, 3, 3)_{LR}, \ \tilde{\tilde{\xi}}_6 \sim \mathcal{N}(\tilde{u}_6, 1), \quad \text{with } \tilde{u}_6 \sim (628, 8, 8)_{LR}, \\ &\tilde{\tilde{\xi}}_7 \sim \mathcal{N}(\tilde{u}_7, 2), \text{ with } \tilde{u}_7 \sim (143, 4, 4)_{LR}, \ \tilde{\tilde{\xi}}_8 \sim \mathcal{N}(\tilde{u}_8, 2), \quad \text{with } \tilde{u}_8 \sim (476, 5, 5)_{LR}, \\ &\tilde{\tilde{\xi}}_9 \sim \mathcal{N}(\tilde{u}_9, 2), \text{ with } \tilde{u}_9 \sim (324, 4, 4)_{LR}, \ \tilde{\tilde{\xi}}_{10} \sim \mathcal{N}(\tilde{u}_{10}, 2), \text{ with } \tilde{u}_{10} \sim (539, 7, 7)_{LR}. \end{aligned}$$

and $\tilde{u}_i (i = 1, 2, \ldots, 10)$ are all assumed as triangular fuzzy variables which are a class of L-R fuzzy variables. Wherein, the risk tolerance given by decision maker is $\bar{f_1} = 35{,}000$, $\bar{f_2} = 170{,}000$.

Denote the feasible region of model (4.92) is X. According to Theorem 4.9 and (4.80), problem (4.92) is equivalent to the following model,

$$\max_{\boldsymbol{x}\in X}[H_1(\boldsymbol{x}), H_2(\boldsymbol{x})] \tag{4.93}$$

where

$$H_1(\boldsymbol{x}) = \Phi\left(\frac{113.4x_1 + 241.7x_2 + 87.2x_3 + 56.2x_4 + 92.3x_5 - 35000}{\sqrt{x_1^2 + 4x_2^2 + x_3^2 + 2x_4^2 + x_5^2}}\right),$$

$$H_2(\boldsymbol{x}) = \Phi\left(\frac{628.8x_1 + 143.4x_2 + 476.5x_3 + 324.4x_4 + 539.7x_5 - 170000}{\sqrt{x_1^2 + 2x_2^2 + 2x_3^2 + 2x_4^2 + 2x_5^2}}\right).$$

Next, we use the goal programming method to solve the problem (4.93). Let $H_1^0 = 0.90$ and $H_2^0 = 0.99$ be the decision maker's reference value. According to (4.84), we get the following single objective problem,

$$\begin{cases} \min \sum_{i=1}^{2} w_i(d_i^+ + d_i^-) \\ \text{s.t.} \begin{cases} \Phi\left(\dfrac{113.4x_1 + 241.7x_2 + 87.2x_3 + 56.2x_4 + 92.3x_5 - 35000}{\sqrt{x_1^2 + 4x_2^2 + x_3^2 + 2x_4^2 + x_5^2}}\right) + d_1^+ - d_1^- = 0.90 \\ \Phi\left(\dfrac{628.8x_1 + 143.4x_2 + 476.5x_3 + 324.4x_4 + 539.7x_5 - 170000}{\sqrt{x_1^2 + 2x_2^2 + 2x_3^2 + 2x_4^2 + 2x_5^2}}\right) + d_2^+ - d_2^- = 0.99 \\ x_1 + x_2 + x_3 + x_4 + x_5 \le 350 \\ x_1 + x_2 + x_3 + x_4 + x_5 \ge 300 \\ 4x_1 + 2x_2 + 1.5x_3 + x_4 + 2x_5 \le 1085 \\ x_1 + 4x_2 + 2x_3 + 5x_4 + 3x_5 \le 660 \\ x_1 \ge 20, x_2 \ge 20, x_3 \ge 20, x_4 \ge 20, x_5 \ge 20 \\ d_1^+, d_1^-, d_2^+, d_2^- \ge 0 \end{cases} \end{cases} \tag{4.94}$$

Take $w_1 = w_2 = 0.5$ and we get the efficient solution,

$$\boldsymbol{x} = (223.21, 21.80, 20.00, 20.00, 20.00)^T$$

and $H_1(\boldsymbol{x}) = 0.90$, $H_2(\boldsymbol{x}) = 0.99$.

Example 4.23. Let's consider the following problem,

$$\begin{cases} \max f_1(\boldsymbol{x}, \boldsymbol{\xi}) = Ch\left\{3\tilde{\tilde{\xi}}_1^2 x_1^2 - 2\tilde{\tilde{\xi}}_1\tilde{\tilde{\xi}}_2 x_1 x_2 + 1.3\tilde{\tilde{\xi}}_2^2 x_2^2 \geq 5\right\}(0.9) \\ \max f_2(\boldsymbol{x}, \boldsymbol{\xi}) = Ch\left\{2.5\tilde{\tilde{\xi}}_3^2 x_1^2 + 3\tilde{\tilde{\xi}}_3\tilde{\tilde{\xi}}_4 x_1 x_2 + 5\tilde{\tilde{\xi}}_4^2 x_2^2 \geq 12\right\}(0.9) \\ \text{s.t.} \begin{cases} x_1 + x_2 \leq 10 \\ 5x_1 - 2x_2 \geq 2 \\ x_1, x_2 \geq 0 \end{cases} \end{cases} \tag{4.95}$$

where $\xi_i (i = 1, \ldots, 4)$ are all independently Ra-Fu variables as follows,

$$\tilde{\tilde{\xi}}_1 \sim \mathcal{N}(\tilde{\mu}_1, 2), \quad \text{with } \tilde{\mu}_1 = (5, 6, 7), \tilde{\tilde{\xi}}_2 \sim \mathcal{N}(\tilde{\mu}_2, 1), \text{ with } \tilde{\mu}_2 = (6.5, 8, 10),$$
$$\tilde{\tilde{\xi}}_3 \sim \mathcal{N}(\tilde{\mu}_3, 1.5), \text{ with } \tilde{\mu}_3 = (4, 5, 6), \tilde{\tilde{\xi}}_4 \sim \mathcal{N}(\tilde{\mu}_4, 2), \text{ with } \tilde{\mu}_4 = (5, 7, 8).$$

where $\tilde{\mu}_i$ are all triangular fuzzy numbers, $i = 1, \ldots, 4$. It follows that

$$\begin{cases} \max H_1(x) = \sup\{\beta_1 | Pos\{Pr\{3\tilde{\tilde{\xi}}_1^2 x_1^2 - 2\tilde{\tilde{\xi}}_1\tilde{\tilde{\xi}}_2 x_1 x_2 + 1.3\tilde{\tilde{\xi}}_2^2 x_2^2 \geq 5\} \geq 0.9\}\} \\ \max H_2(x) = \sup\{\beta_1 | Pos\{Pr\{2.5\tilde{\tilde{\xi}}_3^2 x_1^2 + 3\tilde{\tilde{\xi}}_3\tilde{\tilde{\xi}}_4 x_1 x_2 + 5\tilde{\tilde{\xi}}_4^2 x_2^2 \geq 12\} \geq 0.9\}\} \\ \text{s.t.} \begin{cases} x_1 + x_2 \leq 10 \\ 5x_1 - 2x_2 \geq 2 \\ x_1, x_2 \geq 0 \end{cases} \end{cases} \tag{4.96}$$

The satisfactory solutions to problem (4.96) is listed in Table 4.5 and shown in Figs. 4.18 and 4.19.

4.6 Application to Chinese Liquor

The problem considered in this section comes from Luzhou Co., Ltd which is one of the producers of Chinese liquor in China and its product is typical of Chinese strong aromatic spirits. At present, the company is planning to produce fruit beverages. The company wishes to design a SCN for the product to determine not only the subset of plants and DCs to be opened, but also the distribution strategy that will satisfy the demand imposed by customers in a cost-effective and timely manner under an all

Table 4.5 The optimal solution by Ra-Fu simulation-based rw-GA

w_1	w_2	x_1	x_2	H_1	H_2	H	Gen
0.1	0.9	9.988	0.012	0.8812	0.5673	0.5987	2000
0.2	0.8	9.973	0.027	0.8967	0.5534	0.6221	2000
0.3	0.7	9.992	0.008	0.9135	0.5327	0.6469	2000
0.4	0.6	9.982	0.018	0.9356	0.5219	0.6874	2000
0.5	0.5	9.997	0.003	0.9621	0.5174	0.7398	2000

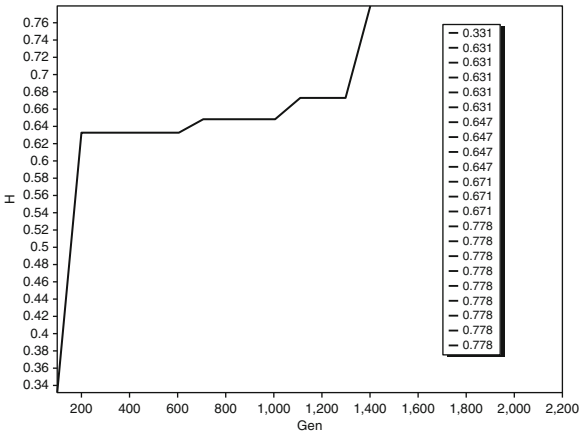


Fig. 4.18 The search process of Ra-Fu simulation-based rw-GA

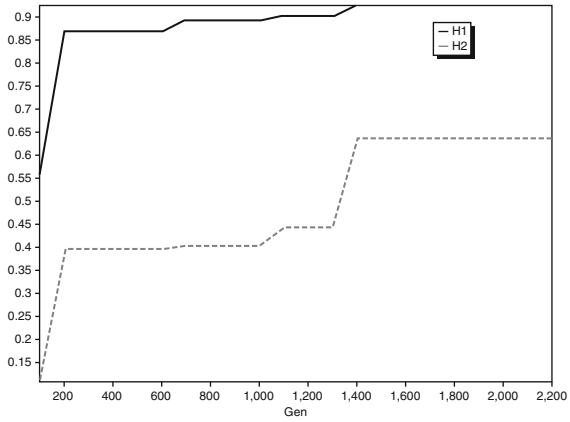


Fig. 4.19 Two objective values by Ra-Fu simulation-based rw-GA

capacities constraint. However, in the supply chain network design problem of this company, it is hard to describe these problem parameters as known variables because there are not sufficient enough data to analyze. For instance, since the changing gasoline price often results in scarcity of precise data, the shipping cost from one supplier to one plant (or from one DC to a customer)is usually a normal distributed variable with an unknown expected value μ. Furthermore, the time from one supplier to one plant (or from one DC to a customer) is fuzzy because of the uncertainty of country/urban traffic in China, hence expected value μ can be "between 150 and 250 yuan". And the amount of demand imposed by customers is a normal distributed variable with an unknown expected value μ due to seasonal effect. Meanwhile, since

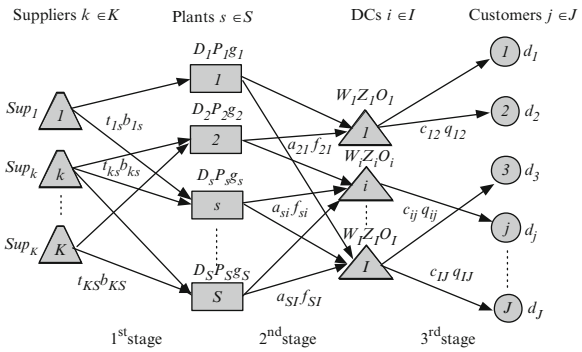


Fig. 4.20 Three-stage supply chain network of Luzhou Co., Ltd

the customers' expected price for the product is fuzzy, the expected value μ can be "around 20 ton". Therefore, a situation exists whereby shipping costs and the demand imposed by customers may be random variables taking fuzzy parameters. In this situation, we can use Ra-Fu variables to deal with these uncertain parameters of combining randomness and fuzziness. Considering the company managers' objectives, we minimize the total cost of the supply chain, and maximize customer service levels in terms of acceptable delivery times (coverage). Hence, we formulate a single-product, multi-stage, multi-objective SCN design problem with Ra-Fu market demands and shipment costs. Figure 4.20 presents a three-stage supply chain network of Luzhou Co., Ltd.

In mixed random and fuzzy environments, to model the single-product, multi-stage, multi-objective SCN design problem of the company, the following assumptions are made:

1. Shipping costs and customer demand are regarded as random fuzzy variables.
2. The number of customers and suppliers and capacities are known.
3. The number of potential plants and DCs and their maximum capacities are known.
4. Since the manager of this company wants to reduce cost and find an optimal network strategy, and a most demand can be satisfied in one day according to the current distribution capacity of this company, inventory is considered as zero based on market conditions and product nature.
5. Customers are supplied products from a single DC.

Based the assumptions above, we propose a Ra-Fu multi-objective mixed-integer non-linear programming model for the problem. In the model, the objectives of minimization of total costs are comprised of the fixed costs of vehicles, variable costs, waiting costs, and penalty costs, and maximization of customer services can be specified by the percent of rendering to all customers of acceptable delivery times.

4.6.1 Notations

The mathematical notation and formulations are as follows:

Indices: i is an index for DCs ($i \in I$), j is an index for customers ($j \in J$), k is an index for suppliers ($k \in K$), s is an index for manufacturing plants ($s \in S$).

Model variables: b_{ks} is the quantity of raw material shipped from supplier k to plant s. f_{si} is the quantity of the product shipped from plant s to DC i. q_{ij} is the quantity of the product shipped from DC i to customer j.

$$z_i = \begin{cases} 1, \text{ if DC } i \text{ is open} \\ 0, \text{ otherwise.} \end{cases}$$
$$p_s = \begin{cases} 1, \text{ if plant } s \text{ is open} \\ 0, \text{ otherwise.} \end{cases}$$
$$y_{ij} = \begin{cases} 1, \text{ if DC } i \text{ servers customer } j \\ 0, \text{ otherwise.} \end{cases}$$

Model parameters: D_s is the capacity of plant s. w_i is the annual throughput at DC i. sup_k is the capacity of supplier k for raw material. $\tilde{d}_j$, which is a Ra-Fu variable, is the demand for the product at customer j. W_i is the maximum number of DC i. P is the maximum number of plants. v_i is the annual fixed cost for operating a DC i. g_s is the annual fixed cost for operating a plant s. $\tilde{c}_{ij}$, which is a Ra-Fu variable, is the unit transportation cost for the product from DC i to customer j. $\tilde{a}_{si}$, which is a Ra-Fu variable, is the unit transportation cost for the product from plant s to DC i. $\tilde{t}_{ks}$, which is a Ra-Fu variable, is the unit transportation and purchasing cost for raw material from supplier k to plant s. r is the utilization rate of raw material per unit of the product. h_{ij} is delivery time (in hours) from DC i to customer j. τ is the maximum allowable delivery time (hours) from DC i to customers j. C_i is the set of customers that can be reached from DC i in τ hours, or $C(i) = \{i | h_{ij} \leq \tau\}$. Υ is the set of opened DCs, $\mathbf{R}$ is the set of opened plants. β is the penalty if DCs can not reach customers in τ time.

4.6.2 Modelling

Based on the requirement of the SCN design problem of this company, we propose a three-stage Ra-Fu programming model to tackle it. The first-stage consists of deciding the plant decisions p_s and the amount of raw material from suppliers to plants based on the uncertain shipping cost $\tilde{t}_{ks}$, the second-stage consists of the DCs decisions z_i and the amount of transporting products from plants to DCs in an optimal manner based upon the DCs and the uncertain shipping cost $\tilde{a}_{si}$, and third-stage includes server decisions y_{ij} and the amount of delivered products from DCs to customers in an optimal schedule based upon the uncertain shipping cost $\tilde{c}_{ij}$. In addition, for the sake of improving the supply

chain's responsiveness to demand, we consider the penalty cost arising from DCs being unable to service customers in τ time into total cost, where h_{ij} can be calculated by the distance from DC i to j and past experience in the transportation Chinese liquor in this paper. The objective f_1 is to minimize total costs comprised by fixed costs of facilities $\sum_{s\in S} g_s p_s + \sum_{i\in I} v_i z_i$, the transporting costs $\sum_{s\in S}\sum_{k\in K} \tilde{t}_{ks} b_{ks} + \sum_{s\in S}\sum_{i\in I} \tilde{a}_{si} f_{si} + \sum_{j\in J}\sum_{i\in I} \tilde{c}_{ij} q_{ij}$ and the penalty cost $\sum_{j\in J}\sum_{i\in I} \beta(h_{ij} - \tau)^+$. Furthermore, the objective f_2 is to maximize the customer service level. To measure the customer service level of a SCN, we employ the products rendered to customers from DCs within the stipulated access time τ, which can be denoted by $(\sum_{i\in \aleph}\sum_{j\in \mathbf{R}} q_{ij})/(\sum_{j\in J} \tilde{d}_j)$. From the discussion above, we develop the mathematical formulations of objectives as follow:

$$\begin{aligned}\min f_1 = &\sum_{s\in S} g_s p_s + \sum_{i\in I} v_i z_i + \sum_{s\in S}\sum_{k\in K} \tilde{t}_{ks} b_{ks} + \sum_{s\in S}\sum_{i\in I} \tilde{a}_{si} f_{si} \\ &+ \sum_{j\in J}\sum_{i\in I} \tilde{c}_{ij} q_{ij} + \sum_{j\in J}\sum_{i\in I} \beta(h_{ij} - \tau)^+ \\ \max f_2 = &\left(\sum_{i\in \aleph}\sum_{j\in \mathbf{R}} q_{ij}\right) \Big/ \left(\sum_{j\in J} \tilde{d}_j\right)\end{aligned}$$

The constraints for the Ra-Fu multi-objective programming model are divided into technique constraints and capacity constraints. Since one customer can be serviced by only one DC, thus we employ the constraint

$$\sum_{i\in I} y_{ij} = 1, \quad \forall i \tag{4.97}$$

to represent such an assignment. Only if the delivery time h_{ij} from DC i to customer j is more than the stipulated access time τ, will the penalty cost occur. Therefore, we use the constraint

$$(h_{ij} - \tau)^+ = \{0, 1\}, \quad \forall i, j \tag{4.98}$$

to specify it. With a similar process of dealing with the binary variable, we can obtain

$$z_i = \{0, 1\}, \quad \forall i \tag{4.99}$$

$$p_s = \{0, 1\}, \quad \forall s \tag{4.100}$$

$$y_{ij} = \{0, 1\}, \quad \forall i, j \tag{4.101}$$

In addition, the annual throughput w_i at DC i in unit time (1 year) must be not larger than the maximum number W_i for the DC. Thus, the constraint

$$\sum_{i \in I} z_i \leq W_i \tag{4.102}$$

is employed. Similarly, we also obtain the constraints

$$\begin{aligned} &\sum_{s \in S} p_s \leq P \\ &\sum_{s \in S} b_{ks} \leq sup_k, \quad \forall\, s. \end{aligned}$$

Since the assumption of no inventory, it means that the quantity of inbound is equal to the number of output at DC i, and equal to the quantity of outbound. Then the quantity of customers product demand must be not larger than the quantity of annual throughput of the opened DC. Thus,

$$\sum_{s \in S} f_{si} = \sum_{j \in J} q_{ij}, \quad \forall i \tag{4.103}$$

$$q_{ij} = \tilde{d}_j y_{ij}, \quad \forall i, j \tag{4.104}$$

$$\sum_{j \in J} \tilde{d}_j y_{ij} \leq w_i z_i, \quad \forall i \tag{4.105}$$

For suppliers capacity constraints and plant production capacity constraints we can use two equations

$$r \sum_{i \in I} f_{si} \leq \sum_k b_{ks}, \quad \forall s \tag{4.106}$$

$$r \sum_{i \in I} f_{si} \leq D_s p_s, \quad \forall s \tag{4.107}$$

From the discussions above, by integration of the (4.97)–(4.107), we can formulate a random fuzzy multi-objective mixed-integer non-linear programming model as follows:

$$
\begin{cases}
\min f_1 = \sum_{s\in S} g_s p_s + \sum_{i\in I} v_i z_i + \sum_{s\in S}\sum_{k\in K} \tilde{t}_{ks} b_{ks} + \sum_{s\in S}\sum_{i\in I} \tilde{a}_{si} f_{si} \\
\qquad + \sum_{j\in J}\sum_{i\in I} \tilde{c}_{ij} q_{ij} + \sum_{j\in J}\sum_{i\in I} \beta(h_{ij}-\tau)^+ \\
\max f_2 = \left(\sum_{i\in \aleph}\sum_{j\in \mathbf{R}} q_{ij}\right) / \left(\sum_{j\in J} \tilde{d}_j\right) \\
\text{s.t.} \begin{cases}
\sum_{i\in I} y_{ij} = 1, \quad \forall i \\
\sum_{j\in J} \tilde{d}_j y_{ij} \le w_i z_i, \quad \forall i \\
\sum_{i\in I} z_i \le W \\
q_{ij} = \tilde{d}_j y_{ij}, \quad \forall i, j \\
\sum_{s\in S} f_{si} = \sum_{j\in J} q_{ij}, \quad \forall i \\
\sum_{s\in S} p_s \le P \\
\sum_{s\in S} b_{ks} \le sup_k, \quad \forall s \\
r\sum_{i\in I} f_{si} \le \sum_k b_{ks}, \quad \forall s \\
r\sum_{i\in I} f_{si} \le D_s p_s, \quad \forall s \\
z_i = \{0,1\}, \quad \forall i \\
p_s = \{0,1\}, \quad \forall s \\
y_{ij} = \{0,1\}, \quad \forall i, j \\
(h_{ij}-\tau)^+ = \{0,1\}, \quad \forall i, j \\
b_{ks} \ge 0, \quad \forall k, s \\
f_{si} \ge 0, \quad \forall s, i \\
q_{ij} \ge 0, \quad \forall i, j \\
h_{ij} \ge 0, \quad \forall i, j \\
\tau \ge 0, \quad \forall i, j \\
\beta \ge 0, \quad \forall i, j
\end{cases}
\end{cases}
\tag{4.108}
$$

Generally, in order to solve the model above, we have to transform these Ra-Fu variables into deterministic parameters using the above proposed technique. Considering the decision maker's objective, the model (4.108) can be solved by minimizing the expected value of total costs and maximizing the expected value of customer service under (α, β) chance constraints.

Here, we will restrict application to the problem described in the first section, and use the proposed model and technique to solve it. Based on the data and related material investigated by Xu and Liu [341], we will discuss the expected value model, the chance-constrained model and the dependent-chance model of the supply chain network of Luzhou Co. Ltd. This company is one of the producers of Chinese liquor and is planning to take advantage of the local orange growing to extend production into fruit berverages. Market research shows that the company can capture a portion of the national fruit beverage market by using their famous brand name.

The raw material to produce fruit beverages can be supplied easily from Naxi, Yub, Neij, and Panzh. The company intends to establish new plants at five potential locations. These locations were selected based on some specific considerations. The Longmt, is considered as a plant because of the richness in raw material and the relatively convenient traffic, is the fist location. The second location is Jiangyq where all other facilities of the company are located. Hechuan is consider as the third location because of the cheap land. The fourth is Luxian because of the richness of raw materials. The last is Xuyong where the cost of land to establish a plant is the lowest.

The company is planning to open at most six DCs. The location of DCs was determined according to demand densities of 150 customer zones to be served and access time from DCs to customer zones. The locations of DCs are Beij, Shangh, Wuhan, Guanz, Xian, and Chengd. The company intends to establish a supply chain network that will satisfy company objectives for the product. The company objectives, as given in mathematical models, are the minimization of overall supply chain cost, and the maximization of customer service, i.e. the percentage of customer demand that can be delivered within the stipulated access time τ and the maximization of capacity utilization balance for DCs (i.e. equity on utilization ratios).

Table 4.6 gives information about the capacities of suppliers, plants, and DCs, and the fixed costs of these facilities. As seen in Table 4.6, capacity and fixed costs in these facilities are different from each other because of the different conditions in each potential location. Tables 4.7 and 4.8 show transportation costs of from supplies to plants and from plants to DCs. Where r_1, r_2, and r_3 is the expected shipping costs. At first, we begin with small size problem of 6 customers. The demand and shipping cost is presented by Tables 4.9 and 4.10, respectively. Where d_1, d_2, d_3 is expected customer demand, $\sigma = 100$. Fig. 4.21 shows the best distribution pattern for this problem by a spanning tree-based GA.

Under probability $\alpha = 0.9$ and credibility $\beta = 0.95$, and when the utilization rate of raw material per unit of the product is $r = 0.8$, we use a spanning tree-based genetic algorithm to solve this problem. We partly show the transportation plans of this company at the first stage and the second stage (Tables 4.11 and 4.12). We can not give the allocation of 150 customers to DCs, since it will consume more space. From those results, it can be seen that the optimal solution is reached by opening

Table 4.6 Capacities and fixed costs for suppliers, plants, and distribution centers

Suppliers	Capacity (ton/year)	Plants	Capacity (ton/year)	Fixed cost (thousand yuan/year)	DCs	Capacity (ton/year)	Fixed cost (thousand yuan/year)
Naxi	14640	Longmt	8751	4093	Beij	7140	441
Yub	8731	Jiangyq	6280	2641	Shangh	7604	467
Neij	4405	Hechuan	7094	2553	Wuhan	9528	534
Panzh	14447	Luxian	7760	3519	Guanz	6904	309
		Xuyong	7755	3560	Xian	8949	505
					Chengd	9111	518

Table 4.7 Shipping cost value for 1st stage (unit ton/thousand yuan)

Suppliers	Transportation cost	Plants				
		Longmt	Jiangyq	Hechuan	Luxian	Xuyong
Naxi	r_1	0.10	0.10	0.15	0.15	0.20
	r_2	0.15	0.15	0.20	0.20	0.25
	r_3	0.20	0.20	0.25	0.25	0.30
Yub	r_1	0.20	0.20	0.15	0.20	0.20
	r_2	0.25	0.25	0.20	0.25	0.25
	r_3	0.30	0.30	0.25	0.30	0.30
Neij	r_1	0.10	0.10	0.15	0.10	0.20
	r_2	0.15	0.15	0.20	0.15	0.25
	r_3	0.20	0.20	0.25	0.20	0.30
Panzh	r_1	0.15	0.15	0.15	0.15	0.20
	r_1	0.20	0.20	0.20	0.20	0.25
	r_1	0.25	0.25	0.25	0.25	0.30

Table 4.8 Shipping cost value for 2nd stage (unit ton/thousand yuan)

Plants	Shipping cost	DCs					
		Beij	Shangh	Wuhan	Guanz	Xian	Chengd
Longmt	r_1	0.20	0.20	0.10	0.15	0.15	0.10
	r_2	0.25	0.25	0.15	0.20	0.20	0.15
	r_3	0.30	0.30	0.20	0.25	0.25	0.20
Jiangyq	r_1	0.20	0.20	0.10	0.15	0.15	0.10
	r_2	0.25	0.25	0.15	0.20	0.20	0.15
	r_3	0.30	0.30	0.20	0.25	0.25	0.20
Hechuan	r_1	0.20	0.20	0.10	0.15	0.15	0.10
	r_2	0.25	0.25	0.15	0.20	0.20	0.15
	r_3	0.30	0.30	0.20	0.25	0.25	0.20
Luxian	r_1	0.20	0.20	0.10	0.15	0.15	0.10
	r_2	0.25	0.25	0.15	0.20	0.20	0.15
	r_3	0.30	0.30	0.20	0.25	0.25	0.20
Xuyong	r_1	0.20	0.20	0.10	0.15	0.15	0.10
	r_2	0.25	0.25	0.15	0.20	0.20	0.15
	r_3	0.30	0.30	0.20	0.25	0.25	0.20

only 4 plants (Longmt, Jiangyq, Hechuan, Luxian) and 3 DCs (Wuhan, Xian, and Chengd).

The company is planning to satisfy customer demand from DCs within one day, i.e., 24 h. The scatter diagram of the annual customer demand versus access time

Table 4.9 Customer demand (ton/year)

Customers demand	Customers					
	1	2	3	4	5	6
d_1	3800	4800	4800	2800	2300	4300
d_2	4000	5000	5000	3000	2500	4500
d_3	4200	5200	5200	3200	2700	4700

Table 4.10 Shipping cost value from DCs to customer (unit ton/thousand yuan)

DCs	Shipping cost	Customers					
		1	2	3	4	5	6
Beij	r_1	0.20	0.20	0.20	0.25	0.10	0.10
	r_2	0.25	0.25	0.25	0.30	0.15	0.15
	r_3	0.30	0.30	0.30	0.35	0.20	0.20
Shangh	r_1	0.20	0.20	0.20	0.15	0.10	0.10
	r_2	0.25	0.25	0.25	0.30	0.20	0.15
	r_3	0.30	0.30	0.30	0.35	0.25	0.20
Wuhan	r_1	0.10	0.10	0.10	0.15	0.10	0.10
	r_2	0.15	0.15	0.15	0.20	0.15	0.15
	r_3	0.20	0.20	0.20	0.25	0.20	0.20
Guanz	r_1	0.20	0.20	0.10	0.15	0.10	0.10
	r_2	0.25	0.25	0.15	0.20	0.15	0.15
	r_3	0.30	0.30	0.20	0.25	0.20	0.20
Xian	r_1	0.10	0.15	0.10	0.15	0.10	0.10
	r_2	0.15	0.20	0.15	0.20	0.15	0.15
	r_3	0.20	0.25	0.20	0.25	0.20	0.20
Chengd	r_1	0.20	0.10	0.10	0.15	0.10	0.10
	r_2	0.25	0.25	0.15	0.20	0.15	0.15
	r_3	0.30	0.30	0.20	0.25	0.20	0.20

from the closest DCs is plotted to obtain information about how large customer demand is, and how far away they are located from DCs. When Fig. 4.22, which is the service level of the distribution center Chengd is examined, it is seen that 83.3% of customers have demands smaller than 600 tons per year. Also, when the capacities of DCs are not taken in the consideration, it is possible to reach the 93.3% of customers within 24 h. Actually, as to the stipulated access time τ, it is an efficient tool for the evaluation of the SCN designed. If the stipulated access time τ, is not considered SCN's response to customers demand will decrease, and the performance of SCN will also decrease. In addition, how to determine the access time of all customer depend on not only a SCN's overall performance in the minimization of total cost and maximization of customers service level in all solutions for this

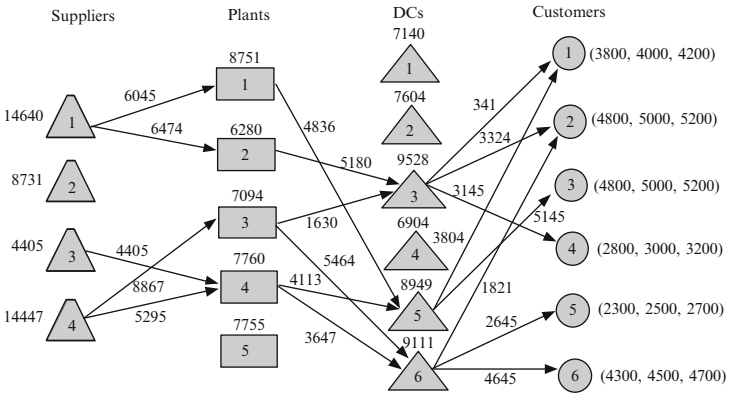


Fig. 4.21 Illustration of the optimal distribution pattern

Table 4.11 Shipping strategy for 1st stage (ton)

Suppliers	Plants				
	Longmt	Jiangyq	Hechuan	Luxian	Xuyong
Naxi	6790	7850	–	–	–
Yub	–	–	1195	–	–
Neij	1480	–	–	2925	–
Panzh	–	–	7672	6775	–

Table 4.12 Shipping strategy for 2nd stage (ton)

Plants	DCs					
	Beij	Shangh	Wuhan	Guanz	Xian	Chengd
Longmt	–	–	3248	–	3206	–
Jiangyq	–	–	6280	–	–	–
Hechuan	–	–	–	–	–	7094
Luxian	–	–	–	–	5743	2017
Xuyong	–	–	–	–	–	–

problem, but also on the distance from a DC to a customer. In the problem, if only the access time for a given customer is smaller than the stipulated access time τ in operation, we consider that the customer can be effectively serviced by the DC.

If we consider a given budget, with regard to the number of DCs opened, we make the following observations. First, when the location cost factor increases, i.e., the DC location costs increase relative to other costs, the number of opened DCs decreases. Second, when the transportation costs between facilities increases, the number of opened DCs also decreases. However, since the total cost and customer service levels are often conflicting, how to handle multi-objective programming depends on the decision-maker's objectives. Generally, the solution to this problem is often a balance of multiple objectives.

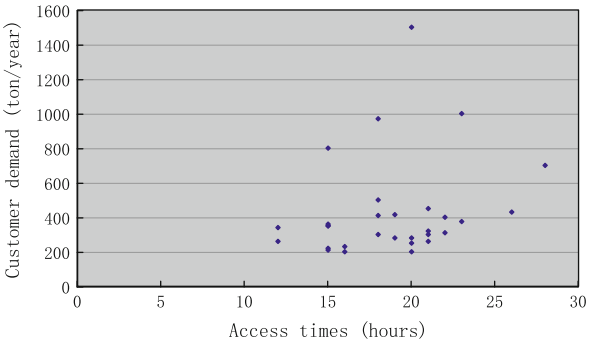


Fig. 4.22 Access time-demand distribution

Table 4.13 The size of tested problems

Problem	Number of suppliers, S	Number of plants, K	Number of DCs, I	Number of customers, L	Number maximum opened plants, P	Number maximum opened DCs, W
1	3	5	5	4	4	4
2	10	10	10	21	6	6
3	20	15	12	50	9	7
4	10	6	8	100	4	5

By expected value and chance constraint programming techniques, the Ra-Fu multi-objective mixed-integer non-linear programming model of this company's SCN design problem has been reduced to a deterministic model, and a genetic algorithm is proposed to solve the model. Till now, no one has formulated or attacked an SCN design problem in the above manner. Here, the techniques illustrated can easily be applied to other SCN problems. Therefore, these techniques are the appropriate tools to tackle other supply chain network problems in realistic environments.

4.6.3 Comparison of st-GA and m-GA

To see the efficiency and effectiveness of the st-GA in the Ra-Fu environment, we also present, under deterministic conditions, the st-GA and matrix-based genetic algorithm (m-GA) based on Michalewicz's approach [219] by using the same program language. Both st-GA and m-GA were run on Pentium 4 PC with the same GA paraments $p_c = 0.4$ and $p_m = 0.2$. To get more information about these algorithms, we divided each test problem into three numerical experiments by giving different sizes of pop_size and max_gen. Each of numerical experiment is run for 10 times (See Table 4.14).

Table 4.14 Computation time and memory used by m-GA and st-GA (ACT, average computation time in second; memory, required unit memory space to represent the chromosome)

Problem	*pop_* size	max_ gen	m-GA		(dc)st-GA		(rf)st-GA	
			ACT	Memory	ACT	Memory	ACT	Memory
1	25	750	1.7	61	1.6	22	1.6	22
	50	1500	5.9	61	5.8	22	5.8	22
2	200	2000	75.9	410	68.6	76	68.6	76
	300	3000	216.4	410	168.3	76	168.3	76
3	200	3000	468.9	1081	260.4	119	260.4	119
	300	6000	948.6	1081	547.4	119	547.4	119
4	200	4000	607.8	909	385.3	133	385.3	133
	300	10000	2897.5	909	1469.4	133	1469.4	133

The performance of the st-GA is tested by using four different sized test problems, and the results from these are in Table 4.13 and [309]. Where (dc)st-GA denotes the spanning tree-based genetic algorithm under deterministic condition, (rf)st-GA denotes the spanning tree-based genetic algorithm in a random fuzzy environment. It is shown in Table 4.14 that, for a small size problem, st-GA method in a Ra-Fu environment and with deterministic conditions has equal computational time and memory but has a better result than that of m-GA. However, since the search space for this kind of problem is so large, it is very important to set the experiment with a reasonable population size and maximum generation to ensure a good result.

Chapter 5
Random Rough Multiple Objective Decision Making

Since the rough set was initialized by Pawlak[295], it has been applied to many fields. Later, many scholars proposed the concept of two-fold uncertain variables combined rough variables with fuzzy and random variables. The concept of random rough variables (abbr. Ra-Ro variable) has proposed by many scholars [207, 344, 345], and it has been widely extended to many fields. Xu and Yao [344–347] discussed the basic definition and properties of the Ra-Ro variables, and introduced the expected value model, chance constrained model, dependent chance model and bi-level model, respectively. Some crisp equivalent models are given, and relative algorithms are proposed. Finally, these models and algorithms are applied to some realistic problems, such as, queuing problems, inventory problems, production-inventory systems and so on.

This chapter mainly introduces random multi-objective decision making problems under rough approximation. We further propose two kinds of Ra-Ro variables, i.e., discrete Ra-Ro variables and continuous Ra-Ro variables, and introduces some special examples. After introducing those basic concepts and properties, three parts are presented respectively from different viewpoints:

1. Ra-Ro expected value model (Ra-Ro EVM). Usually, decision makers find it difficult to make a decision when they encounter an uncertain parameter. A clear criteria must be introduced to assist the decision. The expected value operator of random rough variables is introduced and the crisp equivalent model is deduced when the distribution is clear.
2. Ra-Ro chance-constraint model (Ra-Ro CCM). Sometimes, decision makers don't strictly require the objective value to be maximal benefit, but only need to obtain the maximum benefit under a predetermined confidence level. Then the chance constrained model is proposed and the crisp equivalent model is deduced when the distribution is clear.
3. Ra-Ro dependent-chance model (Ra-Ro DCM). When decision makers predetermine an objective value and require the maximal probability that objective values exceed the predetermined one.

Finally, the application to an inventory problem is presented to show the effectiveness of the above three models. Readers can refer to the following content to know more details.

J. Xu and L. Yao, *Random-Like Multiple Objective Decision Making*, Lecture Notes in Economics and Mathematical Systems 647, DOI 10.1007/978-3-642-18000-2_5,

5.1 Inventory Problem with Ra-Ro Phenomena

The inventory problem, known as a classical and complex problem, has been paid considerable attention. Nevertheless, most quantitative analysis on the inventory problem is mainly concerned on single item and deterministic parameters, such as crisp yield, crisp demand , crisp cost and so on. Riezebos and Gaalman [261] discussed a class of single-item inventory problems for expected inventory order crossovers and showed that the improved policy was still heuristic in nature. In [14, 15, 260], researchers respectively introduce an inventory model with deterministic order or demand. Classical multi-item inventory models are well studied in well known books [123, 193, 229, 291] and others. Different programming methods have been applied to solve multi-item inventory problems by many scholars like Prem and Vart [255], Worell and Hall [337] and others.

The uncertain inventory problem is also difficult and deserves to be researched (Fig. 5.1). Some scholars have well researched some inventory problems with crisp and vague parameters. Order, or demand, or the planning horizon have been considered as fuzzy or random variables in some literatures [153, 213]. In [213], the stochastic planning horizon had been considered in a two storage inventory model and the region reducing genetic algorithm (RRGA) was proposed to solve the model. In [153], Kao and Hsu considered the demand in the inventory system as a fuzzy number and then converted it into a deterministic one. Recently, demand has been considered as a fuzzy variable by some scholars [153, 198]. Ishii and Konno [145] considered shortage cost as a fuzzy number and demand as a random variable in the classical newsbody problem. Then a single-period inventory problem with fuzziness and randomness simultaneously was researched by Dutta, Chakraborty and Roy [92].

However, there has been no attempt to research another mixed environment, where randomness and roughness both appear simultaneously. For some seasonal items (Ice cream, Christmas trees, woolen materials), the demand may vary year to year. According to historical data or abundance of information, we can know

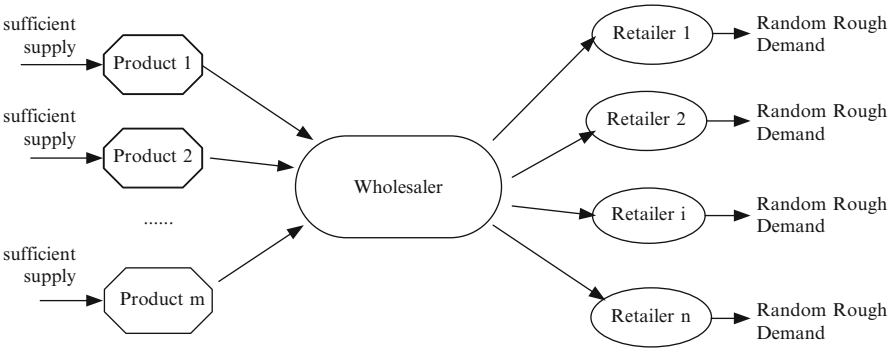


Fig. 5.1 Inventory system

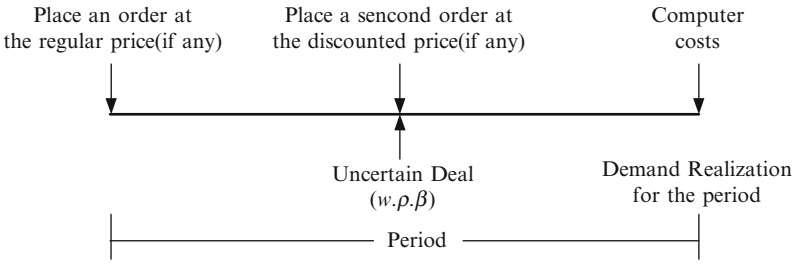


Fig. 5.2 Sequence of events in a period

demand in one year is subject to the stochastic distribution. However, the expected value of stochastic distribution is vague and varies year to year. Thus, it is difficult for decision makers to achieve a better decision. Hence, we have to consider it as an uncertain variable. Rough variable can be applied to depict it well if the average sold amount is clear from the statistical data of each year. Thus, demand for some seasonal items can be described as a Ra-Ro variable to help the decision develop better strategies (Fig. 5.2).

For example, the newsboy problem is a classical inventory problem which has been well studied. In this problem a boy operating a news stall has to determine the number x of newspapers to order in advance from the publisher at a cost of $\$c$ per one newspaper every day. The selling price $\$a$ per one newspaper is also known. If the newspaper is not sold at the end of the day, then the newspapers have a small value of $\$b$ per one newspaper at the recycling center. However, the demand D is imprecise and is forecasted only by previous selling amounts. The newsboy problem with considering shortage cost c as a fuzzy number and demand D as a random variable has been redefined [145]. Some fuzzy multi-product constraints were considered in the newsboy problem by Shao and Ji [284]. Recently, the demand D has been considered by some scholars [92] as a fuzzy random variable which is a two-fold uncertain variable. There is still a case which is ignored by many scholars. In the last 100 days, the demand D doesn't follows the random distribution, but it follows the normal distribution every 10 days when we cut the 100 days into 10, denoted by $D \sim \mathcal{N}(\mu_i, \sigma^2)$. However μ_i is still an uncertain variable and we can describe it with the rough variable. Then the demand D is a Ra-Ro variable and we can use it to compute the order number next day. Similarly, other parameters such as shortage costs can be dealt with by the Ra-Ro variable in order to precisely compute the order number. Thus the newsboy's profit should be

$$f(x) = \begin{cases} (a-c)x, & \text{if } x \leq \tilde{\tilde{D}} \\ (b-c)x + (a-b)\tilde{\tilde{D}}, & \text{if } x \geq \tilde{\tilde{D}} \end{cases}$$

where $\tilde{\tilde{D}}$ is a Ra-Ro variable. Then we can apply the expected value operator and chance operator to compute the maximal profit according to the newsboy's different aims.

In realistic decision making situations, there are cases in which a decision must be made on the basis of uncertain data. For dealing with such decision making problems including uncertainty, the probabilistic approach and the rough-theoretic approach are typical. Stochastic theory and rough set theory have been well developed and applied in a wide variety of uncertainty surrounding real problems since their introduction, such as, Lempel and Moran introduced SALSA (Stochastic Approach for Link-Structure Analysis) algorithm [192] in 2000 and Slowinski [294] applied the method of rough sets to the medical domain. Different types of stochastic programming and rough programming models have been introduced to suit the different purposes of management, such as the expectation model [206], chance constrained programming [45, 206], dependent-chance programming [204, 207], etc. In these models, randomness and roughness are considered as separate aspects. However, in a decision-making process, we may face a hybrid uncertain environment where randomness and roughness coexist. In such cases, the concept of the Ra-Ro variable is a useful tool in dealing with the two types of uncertainty simultaneously.

5.2 Random Rough Variable

Before introducing the concept of Ra-Ro variables, let's recall some definitions and properties of rough sets.

5.2.1 Rough Set

The rough sets theory introduced by Pawlak [295, 295] has often proved to be an excellent mathematical tool for the analysis of a vague description of objects (called actions in decision problems). The adjective vague, referring to the quality of information, means inconsistency or ambiguity which follows from information granulation. The rough sets philosophy is based on the assumption that with every object of the universe there is associated a certain amount of information (data, knowledge), expressed by means of some attributes used for object description. Objects having the same description are indiscernible (similar) with respect to the available information. The indiscernibility relation thus generated constitutes a mathematical basis of the rough sets theory; it induces a partition of the universe into blocks of indiscernible objects, called elementary sets, that can be used to build knowledge about a real or abstract world. The use of the indiscernibility relation results in information granulation. The rough sets theory, dealing with representation and processing of vague information, presents a series of intersections and complements with respect to many other theories and mathematical techniques handling imperfect information, like probability theory, evidence theory of DempsterShafer, fuzzy sets theory, discriminant analysis and mereology [88, 177, 249, 250, 253, 292, 293, 296].

For algorithmic reasons, the information regarding the objects is supplied in the form of a data table, whose separate rows refer to distinct objects (actions), and

whose columns refer to di.erent attributes considered. Each cell of this table indicates an evaluation (quantitative or qualitative) of the object placed in that row by means of the attribute in the corresponding column.

Formally, a data table is the 4-tuple $\mathbf{S} = (U, Q, V, f)$, where U is a finite set of objects (universe), $Q = q_1, q_2, \ldots, q_n$is a finite set of attributes, V_q is the domain of the attribute, $V = \bigcup_{q \in Q} V_q$ and $f : U \times Q \to V$ is a total function such that $f(x, q) \in V_q$ for each $x \in U, q \in Q$, called *information function.*

Therefore, each object x of U is described by a vector (string) $Des_Q(x) = (f(x, q_1), f(x, q_2), \ldots, f(x, q_m)$, called description of x in terms of the evaluations of the attributes from Q; it represents the available information about x.

To every (non-empty) subset of attributes P is associated an indiscernibility relation on U, denoted by I_P:

$$I_p = \{(x, y)| \in U \times U : f(x, q) = f(y, q) \forall q\}$$

If $(x, y) \in I_p$, it is said that the objects x and y are P-indiscernible. Clearly, the indiscernibility relation thus de.ned is an equivalence relation (reflexive, symmetric and transitive). The family of all the equivalence classes of the relation I_P is denoted by $U|I_P$ and the equivalence class containing an element $x \in U$ by $I_p(x)$. The equivalence classes of the relation I_P are called *P-elementary sets*. If $P = Q$, the Q-elementary sets are called *atoms*.

Let $\mathbf{S}$ be a data table, X a non-empty subset of U and $\Phi \neq P \subseteq Q$. The *P-lower* approximation and the *P-upper* approximation of X in $\mathbf{S}$ are defined, respectively, by:

$$\underline{P}(X) = \{x \in U : I_p(x) \subseteq X\}$$
$$\bar{P}(X) = \bigcup_{x \in X} I_P(X)$$

The elements of $\underline{P}(X)$ are all and only those objects $x \in U$ which belong to the equivalence classes generated by the indiscernibility relation I_P, contained in X; the elements of $\bar{P}(X)$ are all and only those objects $x \in U$ which belong to the equivalence classes generated by the indiscernibility relation I_P, containing at least one object x belonging to X. In other words, $\underline{P}X$ is the largest union of the P-elementary sets included in X, while $\bar{P}(X)$ is the smallest union of the P-elementary sets containing X.

1. The *P-boundary* of X in S, denoted by $Bn_P(X)$, is $Bn_P(X) = \bar{P}(X) - \underline{P}(X)$.
2. The following relation holds: $\underline{P}(X) \subseteq X \subseteq \bar{P}(X)$.

Therefore, if an object x belongs to $\underline{P}(X)$, it is certainly also an element of X, while if x belongs to $\bar{P}(X)$, it may belong to the set X. $Bn_P(X)$ constitutes the "doubtful region" of X: nothing can be said with certainty about the belonging of its elements to the set X.

The following relation, called *complementarity property*, is satisfied: $\underline{P}(X) = U - \bar{P}(U - X)$.

If the P-boundary of X is empty, $Bn_P(X) = \Phi$, then the set X is an ordinary (exact) set with respect to P, that is, it may be expressed as the union of a certain number of P-elementary sets; otherwise, if $Bn_P(X) \neq \Phi$, the set X is an approximate (rough) set with respect to P and may be characterized by means of the approximations $\underline{P}(X)$ and $\bar{P}(X)$. The family of all the sets $X \subseteq U$ having the same P-lower and P-upper approximations is called a *rough set*.

The following ratio defines an accuracy of the approximation of X, $X \neq \Phi$ by means of the attributes from P:

$$\alpha_P(X) = \frac{|\underline{P}(X)|}{|\bar{P}(X)|}$$

where $|Y|$ indicates the cardinality of a (finite) set Y. Obviously, $0 \leq \alpha_P(X) \leq 1$; if $\alpha_P(X) = 1$, X is an ordinary (exact) set with respect to P; if $\alpha_P(X) = 1$, X is a rough (vague) set with respect to P.

Another ratio defines a quality of the approximation of X by means of the attributes from P:

$$\gamma_P(X) = \frac{|\underline{P}(X)|}{|X|}$$

The quality $\gamma_P(X)$ represents the relative frequency of the objects correctly classified by means of the attributes from P. Moreover, $0 \leq \alpha_P(X) \leq \gamma_P(X) \leq 1$, and $\gamma_P(X) = 0$ iff $\alpha_P(X) = 0$, while $\gamma_P(X) = 1$ iff $\alpha_P(X) = 1$.

The definition of approximations of a subset $X \subseteq U$ can be extended to a classi.cation, i.e. a partition $Y = \{Y_1, Y_2, \ldots, Y_n\}$ of U. Subsets $Y_i, i = 1, 2, \ldots, n$ are disjunctive classes of Y. By P-lower (P-upper) approximation of Y in S, we mean sets $\underline{P}(Y) = \{\underline{P}(Y_1), \underline{P}(Y_2), \ldots, \underline{P}(Y_n)\}$ and $\bar{P}(Y) = \{\bar{P}(Y_1), \bar{P}(Y_2), \ldots, \bar{P}(Y_n)\}$, respectively. The coefficient

$$\gamma_P(X) = \frac{\left| \sum_{1=1}^{n} \underline{P}(X) \right|}{|U|}$$

is called quality of the approximation of classication Y by set of attributes P, or in short, quality of classification. It expresses the ratio of all P-correctly classified objects to all objects in the system.

The main preoccupation of the rough sets theory is approximation of subsets or partitions of U, representing a knowledge about U, with other sets or partitions built up using available information about U. From the viewpoint of a particular object $x \in U$, it may be interesting, however, to use the available information to assess the degree of its membership to a subset X of U. The subset X can be identified with a concept of knowledge to be approximated. Using the rough set approach one can calculate the membership function $\mu_X^P(x)$ (rough membership function) as

$$\mu_X^P(x) = \frac{X \cap I_p(x)}{I_p(x)}$$

The value of $\mu_X^P(x)$ may be interpreted analogously to conditional probability and may be understood as the degree of certainty (credibility) to which x belongs to X. Observe that the value of the membership function is calculated from the available data, and not subjectively assumed, as it is the case of membership functions of fuzzy sets.

Between the rough membership function and the approximations of X the following relationships hold (Pawlak [295]):

$$\underline{P}(X) = \{x \in U : \mu_X^P(x) = 1\}, \bar{P}(X) = \{x \in U : \mu_X^P(x) > 0\}$$
$$Bn_P(X) = \{x \in U : 0 < \mu_X^P(x) < 1\}, \underline{P}(U - X) = \{x \in U : \mu_X^P(x) = 0\}$$

In the rough sets theory there is, therefore, a close link between vagueness (granularity) connected with rough approximation of sets and uncertainty connected with rough membership of objects to sets.

After the rough set was initialized by Pawlak [295], it has been applied to many fields to deal with vague description of objectives. He asserted that any vague information can be approximated by other crisp information. In this section, we will recall these fundamental concepts and introduce its application to the statistical field and programming problem.

Definition 5.1. (Slowinski and Vanderpooten [297]) Let U be a universe, and X a set representing a concept. Then its lower approximation is defined by

$$\underline{X} = \{x \in U | R^{-1}(x) \subset X\}, \tag{5.1}$$

and the upper approximation is defined by

$$\overline{X} = \bigcup_{x \in X} R(x), \tag{5.2}$$

where R is the similarity relationship on U. Obviously, we have $\underline{X} \subseteq X \subseteq \overline{X}$.

Definition 5.2. (Pawlak [295]) The collection of all sets having the same lower and upper approximations is called a rough set, denoted by $(\underline{X}, \overline{X})$. Its boundary is defined as follows,

$$Bn_R(X) = \overline{X} - \underline{X}. \tag{5.3}$$

In order to know the degree of the upper and lower approximation describing the set X, the concept of the *accuracy* of approximation is proposed by Greco et al. [119],

$$\alpha_R(X) = \frac{|\underline{X}|}{|\overline{X}|} \tag{5.4}$$

where $X \neq \Phi$, $|X|$ expresses the cardinal number of the set X when X is a finite set, otherwise it expresses the Lebesgue measure.

Another ratio defines a *quality* of the approximation of X by means of the attributes from R according to Greco et al. [119],

$$\gamma_R(X) = \frac{|\underline{X}|}{|X|} \tag{5.5}$$

The quality $\gamma_R(X)$ represents the relative *frequency* of the objects correctly classified by means of the attributes from R.

Remark 5.1. For any set A we can represents its frequency of the objects correctly approximated by $(\underline{X}, \overline{X})$ as follows,

$$\beta_R(A) = \frac{|\underline{X} \cap A|}{|\overline{X} \cap A|}.$$

If $\underline{X} \subseteq A \subseteq \overline{X}$, namely, A has the upper approximation $\overline{X}$ and the lower approximation $\underline{X}$, we have that $\beta_R(A)$ degenerates to the *quality* $\gamma_R(A)$ of the approximation.

As we know, the *quality* $\gamma_R(A)$ of the approximation describes the frequency of A, and when $\gamma_R(A) = 1$, we only have $|A| = |\underline{X}|$, namely, the set A is well approximated by the lower approximation. If we want to make A be a definable set, there must be $\gamma_R(A) = 1$ and $\alpha_R(X) = 1$ both holds. Then we could make use the following definition to combine them into together.

Definition 5.3. Let $(\underline{X}, \overline{X})$ be a rough set under the similarity relationship R and A be any set satisfying $\underline{X} \subseteq A \subseteq \overline{X}$. Then we define the approximation function as follows expressing the relative frequency of the objects of A correctly classified into $(\underline{X}, \overline{X})$,

$$\mathrm{Appr}_R(A) = 1 - \eta\left(1 - \frac{|A|}{|\overline{X}|}\right) \tag{5.6}$$

where η is a predetermined by the decision maker's preference.

From Definition 5.3, we know that $\frac{|A|}{|\overline{X}|}$ which keeps accord with $\gamma_R(A)$ describes the relative frequency of the objects correctly classified by R from the view of the upper approximation $\overline{X}$. Obviously, $\mathrm{Appr}_R(A)$ is a number between 0 and 1, and is increasing along with the increase of $|A|$. The extreme case $\mathrm{Appr}_R(A) = 1$ means that $|A| = |\overline{X}|$, namely, A is completely described by $\overline{X}$.

Lemma 5.1. *Let $(\underline{X}, \overline{X})$ be a rough set under the similarity relationship R and A be any set satisfying $\underline{X} \subseteq A \subseteq \overline{X}$. Then we have*

$$Appr_R(A) = \frac{\eta\alpha_R(A) + (1-\eta)\gamma_R(A)}{\gamma_R(A)}.$$

Proof. Since $\underline{X} \subseteq A \subseteq \overline{X}$, it means that A has the lower approximation $\underline{X}$ and the upper approximation $\overline{X}$, and it follows from Greco et al. [119] that

$$\alpha_R(A) = \frac{|\underline{X}|}{|\overline{X}|}, \quad \gamma_R(A) = \frac{|\underline{X}|}{|A|}.$$

Thus,

$$\frac{|A|}{|\overline{X}|} = \frac{\alpha_R(A)}{\gamma_R(A)}.$$

It follows that

$$\begin{aligned} \mathrm{Appr}_R(A) &= 1 - \eta\left(1 - \frac{|A|}{|\overline{X}|}\right) \\ &= 1 - \eta\left(1 - \frac{\alpha_R(A)}{\gamma_R(A)}\right) \\ &= \frac{\eta\alpha_R(A) + (1-\eta)\gamma_R(A)}{\gamma_R(A)} \end{aligned}$$

This completes the proof. □

Lemma 5.2. *Let $(\underline{X}, \overline{X})$ be a rough set on the finite universe under the equivalence relationship R, A be any set satisfying $\underline{X} \subseteq A \subseteq \overline{X}$ and $\eta \in (0, 1)$. Then $Appr_R(A) = 1$ holds if and only if $\underline{X} = A = \overline{X}$.*

Proof. If $\underline{X} = A = \overline{X}$ holds, it is obvious that $\mathrm{Appr}_R(A) = 1$ according to Definition 5.4. Let's proved the necessity of the condition.

If $\mathrm{Appr}_R(A) = 1$ holds for any A satisfying $\underline{X} \subseteq A \subseteq \overline{X}$, it follows from Lemma 5.1 that, for $0 < \eta \leq 1$,

$$\frac{\eta\alpha_R(A) + (1-\eta)\gamma_R(A)}{\gamma_R(A)} = 1 \Rightarrow \alpha_R(A) = \gamma_R(A) \Rightarrow |\overline{X}| = |A|.$$

Since $A \subseteq \overline{X}$ and the universe is finite, we have that $A = \overline{X}$. Because A is any set satisfying $\underline{X} \subseteq A \subseteq \overline{X}$, let $A = X$, then we have $X = \overline{X}$. It follows from the property proposed by Pawlak [295] that $\underline{X} = X = \overline{X}$. Thus, we have $\underline{X} = A = \overline{X}$. □

Lemma 5.1 shows that the approximation function Appr inherits the accuracy and quality of the approximation, and extends it to the relationship between any set A and the rough set $(\underline{X}, \overline{X})$. Lemma 5.2 shows that the approximation function is complete and well describes the property in traditional rough set theory, and describe the property only by one index.

Lemma 5.3. *Let $(\underline{X}, \overline{X})$ be a rough set on the infinite universe under the similarity relationship R, A be any set satisfying $\underline{X} \subseteq A \subseteq \overline{X}$ and $\eta \in (0, 1)$. If $Appr_R(A) = 1$ holds, then there exist the similarity relationship R^* such that $|\underline{X}| = |A| = |\overline{X}|$, where $|\cdot|$ expresses the Lebesgue measure.*

Fig. 5.3 Apartment

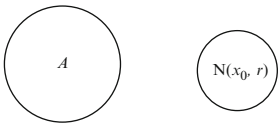


Fig. 5.4 Tangent

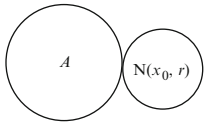
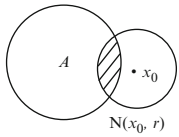


Fig. 5.5 Intersection



Proof. According to Lemma 5.2, we know that $|A| = |\overline{X}|$ must hold. Let $\underline{X} = \overline{X}/\partial\overline{X}$ under the similarity relationship R^*, where $\partial\overline{X}$ is composed by all the elements such that $|\partial\overline{X}| = 0$, namely, the measure of $\partial\overline{X}$ is 0. Next, we will prove that $\overline{X}/\partial\overline{X} \subseteq A$.

1. If $|\overline{X}| = 0$, then $\overline{X}/\partial\overline{X} = \Phi$. Thus, $|\underline{X}| = |A| = |\overline{X}| = 0$.

2. If $|\overline{X}| \neq 0$, we only need to prove that for any $x^0 \in \overline{X}/\partial\overline{X}$, $x^0 \in A$. In fact, when $x^0 \in \overline{X}/\partial\overline{X}$, then $x^0 \in int(\overline{X})$ holds, where $int(\overline{X})$ is the internal part of $\overline{X}$. It follows that there exists $r > 0$ such that $N(x^0, r) \subset int(\overline{X})$ and $|N(x^0, r)| > 0$. There exist four cases describing the relationship between A and $N(x^0, r)$.

Case 1. $A \cap N(x^0, r) = \Phi$ (see Fig. 5.3) . Since $N(x^0, r) \subset int(\overline{X}) \subset \overline{X}$ and $A \subseteq \overline{X}$, we have that

$$|\overline{X}| \geq |N(x^0, r) \cup A| = |N(x^0, r)| + |A|.$$

This conflicts with $|A| = |\overline{X}|$.

Case 2. $A \cap N(x^0, r) = P$, where the set P includes countable points (see Fig. 5.4). Obviously, we have $|P| = 0$, thus $|N(x_0, r)/P| = |N(x_0, r)| > 0$. Then we have

$$|\overline{X}| \geq |N(x^0, r) \cup A| = |N(x^0, r)/P| + |A|.$$

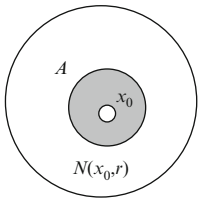
This also conflicts with $|A| = |\overline{X}|$.

Case 3. $A \cap N(x^0, r) = P'$, where $P' \subset N(x^0, r)/\{x_0\}$. As Fig. 5.5 shows, we can divide it into three parts, namely, $(N(x^0, r)/P) = P' \cup \{x_0\} \cup T$, where P', T and $\{x_0\}$ don't have the same element with each other. Then $|T| > 0$, it follows that

$$|\overline{X}| \geq |N(x^0, r) \cup A| = |T| + |A|.$$

This also conflicts with $|A| = |\overline{X}|$.

Fig. 5.6 Inclusion



Case 4. $A \supset (N(x^0, r)/x_0)$ (see Fig. 5.6). This means that for any $x_0 \in int(A)$, $x_0 \notin A$. It follows that $A \cap int(A) = \Phi$, then we have

$$|\overline{X}| \geq |int(A) \cup A| = |int(A)| + |A|.$$

This also conflicts with $|A| = |\overline{X}|$. In above, we can get $\overline{X}/\partial\overline{X} \subseteq A$. Thus, there exists the lower approximation $\underline{X} = \overline{X}/\partial\overline{X}$ such that $\underline{X} \subseteq A \subseteq \overline{X}$ under the similarity relationship $R*$. □

Remark 5.2. In fact, we can extend Definition 5.3 to more general set. when $\underline{X} \subseteq A \subseteq \overline{X}$, we have the following equivalent formula,

$$\begin{aligned}\mathrm{Appr}_R(A) &= 1 - \eta\left(1 - \frac{|A|}{|\overline{X}|}\right) \\ &= \frac{|A \cap \underline{X}|}{|\underline{X}|}\left(1 - \eta\left(1 - \frac{|A \cap \overline{X}|}{|\overline{X}|}\right)\right) \\ &= \frac{|A \cap \underline{X}|}{|\underline{X}|} + \eta\left(\frac{|A \cap \overline{X}|}{|\overline{X}|} - \frac{|A \cap \underline{X}|}{|\underline{X}|}\right)\end{aligned}$$

Furthermore, we get the definition of the approximation function for any set A.

Definition 5.4. Let $(\underline{X}, \overline{X})$ be the rough set generated by X under the similarity relationship R, for any set A, the approximation function of event A by $(\underline{X}, \overline{X})$ is defined as follows

$$\mathrm{Appr}_R(A) = \frac{|A \cap \underline{X}|}{|\underline{X}|} + \eta\left(\frac{|A \cap \overline{X}|}{|\overline{X}|} - \frac{|A \cap \underline{X}|}{|\underline{X}|}\right),$$

where η is a given parameter predetermined by the decision maker's preference.

From Definition 5.4, we know that $\mathrm{Appr}_R(A)$ expresses the relationship between the set A and the set $(\underline{X}, \overline{X})$ generated by X, that is, the frequency of A correctly classified into $(\underline{X}, \overline{X})$ according to the similarity relationship R. It has the internal link with the *accuracy* α_R of the approximation and the *quality* γ_R of the approximation in some extent. α_R expresses the degree of the upper and lower approximation describing the set X. $\gamma_R(X)$ represents the relative frequency of the objects correctly classified by means of the attributes from R. Then Appr_R combines both of them together and considers the level which A has the attributes correctly classified by $(\underline{X}, \overline{X})$ for any A.

Lemma 5.4. *Let* $(\underline{X}, \overline{X})$ *be a rough set, for any set* A*, we have the following conclusion,*

$$Appr_R = \begin{cases} 1, & \text{if } A \supseteq \overline{X} \\ 1 - \eta\big(1 - \dfrac{\alpha_R(A)}{\gamma_R(A)}\big), & \text{if } \underline{X} \subset A \subset \overline{X} \\ \dfrac{1 - \eta(1 - \alpha_R(A))}{\gamma_R(A)}, & \text{if } A \subseteq \underline{X} \\ 0, & \text{if } A \cap \overline{X} = \Phi \\ \dfrac{|A \cap \overline{X}|}{|\overline{X}|}\left(\dfrac{\beta_R(A)}{\alpha_R(A)} + \eta\big(1 - \dfrac{\beta_R(A)}{\alpha_R(A)}\big)\right), & \text{otherwise} \end{cases} \tag{5.7}$$

Proof. 1. If $A \supseteq \overline{X}$, we have that $A \cap \underline{X} = \underline{X}$ and $A \cap \overline{X} = \overline{X}$. Then $\text{Appr}_R = 1$.
2. If $\underline{X} \subseteq A \subseteq \overline{X}$, we have that $A \cap \underline{X} = \underline{X}$ and $A \cap \overline{X} = A$. It follows that $\text{Appr}_R = 1 - \eta(1 - \frac{|A|}{|\overline{X}|})$.
3. If $A \subset \underline{X}$, we have that $A \cap \underline{X} = A$ and $A \cap \overline{X} = A$. It follows that $\text{Appr}_R = \frac{1-\eta(1-\alpha_R(A))}{\gamma_R(A)}$.
4. If $A \cap \overline{X} = \Phi$, we have that $A \cap \underline{X} = \Phi$ and $A \cap \overline{X} = \Phi$. It follows that $\text{Appr}_R = 0$.
5. For the others, we have

$$\begin{aligned} \text{Appr}_R(A) &= \frac{|A \cap \underline{X}|}{|\underline{X}|} + \eta\left(\frac{|A \cap \overline{X}|}{|\overline{X}|} - \frac{|A \cap \underline{X}|}{|\underline{X}|}\right) \\ &= \frac{|A \cap \overline{X}|}{|\overline{X}|}\left(\frac{|A \cap \underline{X}|}{|\underline{X}|} \cdot \frac{|\overline{X}|}{|A \cap \overline{X}|} + \eta\left(1 - \frac{|A \cap \underline{X}|}{|\underline{X}|} \cdot \frac{|\overline{X}|}{|A \cap \overline{X}|}\right)\right) \\ &= \frac{|A \cap \overline{X}|}{|\overline{X}|}\left(\frac{\beta_R(A)}{\alpha_R(A)} + \eta\left(1 - \frac{\beta_R(A)}{\alpha_R(A)}\right)\right) \end{aligned}$$

This completes the proof. □

For the different purposes, we can respectively discuss the extreme case as follows.

Remark 5.3. When $\eta = 1$, we have $\text{Appr}_R(A) = \frac{|A \cap \underline{X}|}{|\underline{X}|}$. It means that the decision maker only consider the level that A includes the frequency of A correctly classified into $\underline{X}$ according to the similarity relationship R.

Remark 5.4. When $\eta = 0$, we have $\text{Appr}_R(A) = \frac{|A \cap \overline{X}|}{|\overline{X}|}$. It means that the decision maker only consider the level that A includes the frequency of A correctly classified into $\overline{X}$ according to the similarity relationship R.

In fact, the rough set theory is increasingly developed by many scholars and applied to many fields, for example, data mining, decision reduction, system analysis and so on. Figure 5.7 shows that the rough approximation. The curves including the internal points is X. The two thick curves including their internal points are the upper and lower approximation.

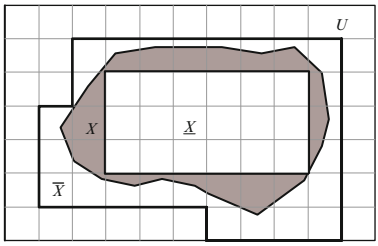


Fig. 5.7 Rough approximation

5.2.2 Ra-Ro Variable

Let's focus on the continuous set in the one dimension real space **R**. There are still some vague sets which cannot be directly fixed and need to be described by the rough approximation. For example, set **R** be the universe, a similarity relation $\simeq$ is defined as $a \simeq b$ if and only if $|a - b| \leq 10$. We have that for the set $[20, 50]$, its lower approximation $\underline{[20, 50]} = [30, 40]$ and its upper approximation $\overline{[20, 50]} = [10, 60]$. Then the upper and lower approximation of the set [20, 50] make up a rough set ([30, 40],[10, 60]) which is the collection of all sets having the same lower approximation [30, 40] and upper approximation [10, 60].

Especially, when we consider a random event with uncertain parameter such as the probability, expected value and variance, the rough approximation can be applied to find the more accurate distribution. Let's firstly consider the following example. There are 10,000 spare parts which are divided into 100 groups. After carefully examination, we found that the lifetime ξ of all the spare parts in each group follows the exponential distribution as follows,

$$\phi \sim \begin{cases} \frac{1}{\theta_i} e^{-x/\theta_i}, & \text{if } 0 \leq x < +\infty \\ 0, & \text{otherwise} \end{cases} \tag{5.8}$$

where $\theta_i (i = 1, 2, \ldots, 10)$ is the expected value expressing the average lifetime of the spare parts in each group. As we know, they are usually different. Let's assume that the minimal and maximal values are 105 and 90, respectively. Then we can denote $\theta_i \in [90, 105]$. Now, for the new 100 ones, decision makers require the random error doesn't exceed 5. Thus, the similarity relationship is fixed, that is, $|x - y| \leq 5$, then the rough set ([95,100], [85,110]) can well describe the average lifetime in every group. Then we define the concept of the Ra-Ro variable as follows.

Definition 5.5. A Ra-Ro variable ξ is a random variable with uncertain parameter $\tilde{\theta} \in X$, where X is approximated by $(\underline{X}, \overline{X})$ according to the similarity relation R, namely, $\underline{X} \subseteq X \subseteq \overline{X}$.

For convenience, we usually denote $\tilde{\theta} \vdash (\underline{X}, \overline{X})_R$ expressing that $\tilde{\theta}$ is in some set A which is approximated by $(\underline{X}, \overline{X})$ according to the similarity relation R, namely, $\underline{X} \subseteq A \subseteq \overline{X}$.

Example 5.1. Let's consider the normally distributed random variable ξ with the following density function,

$$\rho(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-\bar{\mu})^2}{2}}, -\infty < x < +\infty \tag{5.9}$$

where $\tilde{\mu} \vdash ([1,2],[0,3])_R$. Then ξ is a random rough variable.

Example 5.2. If $\xi \sim \mathscr{U}(1+\tilde{\theta}, 3+\tilde{\theta})$ is an uniformly distributed random variable, where $\tilde{\theta} \vdash ([2,3],[1,4])_R$, then ξ is a Ra-Ro variable.

5.2.2.1 Discrete Ra-Ro Variable

Random variables are usually divided into two kinds. One is the discrete random variable, the other is the continuous random variable. By the definition of random rough variable, we know a Ra-Ro variable is essentially a random variable taking rough value. Hence, Ra-Ro variables also can be divided into the discrete Ra-Ro variable and the continuous one. Interested readers can also consult in the paper wrote by Xu and Yao [347].

Definition 5.6. (Xu and Yao [347])(Discrete Ra-Ro variables) A discrete Ra-Ro variable is a function ξ from a rough space $(\Lambda, \Delta, \mathscr{A}, \pi)$ to a collection Ω of discrete random variables such that for any element λ, $\xi(\lambda)$ is a discrete random variable, that is,

$$\xi(\lambda)(\omega) = \begin{cases} f_1(\lambda\omega_1) \text{ if } & \omega = \omega_1 \\ f_2(\lambda\omega_2) \text{ if } & \omega = \omega_2 \\ \cdots & \\ f_n(\lambda\omega_n) \text{ if } & \omega = \omega_n \end{cases} \tag{5.10}$$

where f_i is any continuous function, $\lambda \vdash ([a,b],[c,d])_R$, $(c \le a < b \le d)$ and $\sum_{i=1}^{n} Pr\{\omega = \omega_i\} = 1$.

Obviously, ξ is a discrete random variables with the uncertain parameter $\lambda \in X$ which is approximated by the rough set $([a,b],[c,d])$ under the similarity relationship R. $\xi(\lambda)$ itself is a discrete Ra-Ro variable, and it can be combined by different discrete random variables defined on the probability space Ω. Now, let's look at two intuitive example of discrete Ra-Ro variables.

Example 5.3. Assume $([a,b],[c,d])$ be a rough set under the similarity relationship R, where $c \le a < b \le d$. Let $\Omega = \{\omega_1, \omega_2, \omega_3\}$ be the probability space with the probability 0.5 of ω_1, probability 0.2 of ω_2, and probability 0.3 of ω_3. For each $\lambda \vdash ([a,b],[c,d])_R$, let the value of $\xi(\lambda)$ be a random variable defined Ω by

$$\xi(\lambda)(\omega) = \begin{cases} 2^{\lambda\omega_1}, \text{ if } \omega = \omega_1 \\ 2^{\lambda\omega_2}, \text{ if } \omega = \omega_2 \\ 2^{\lambda\omega_3}, \text{ if } \omega = \omega_3 \end{cases} \tag{5.11}$$

Then ξ is a Ra-Ro variable, and in fact a discrete random rough variable.

Example 5.4. Let $([a,b],[c,d])$ be a rough set provided in Example 5.3, $\xi(\lambda)$ is defined by the following function,

$$\xi(\lambda)(\omega) = \begin{cases} \lambda\xi_1, \text{ if } c \leq \lambda \leq a \\ \lambda\xi_2, \text{ if } b \leq \lambda \leq d \end{cases} \tag{5.12}$$

where $\xi_i, i = 1,2$ is respectively a discrete random variable defined on the probability space Ω. Apparently, ξ is a discrete Ra-Ro variable.

The expected value and variance of Ra-Ro variable is another important property we must pay attention to. Then the basic definition of discrete Ra-Ro variable is shown as follows.

The expected value and variance of Ra-Ro variable is another important property we must pay attention to. Then the basic definition of discrete Ra-Ro variable is shown as follows.

Definition 5.7. (Xu and Yao [347]) Let ξ be an Ra-Ro variable with the uncertain parameter λ, where $\lambda \vdash (\underline{X}, \overline{X})_R$, then its expected value is defined by

$$E[\xi] = \int_0^\infty \text{Appr}\{E[\xi(\lambda)] \geq r\}dr - \int_{-\infty}^0 \text{Appr}\{E[\xi(\lambda)] \leq r\}dr \tag{5.13}$$

Definition 5.8. Let ξ be a Ra-Ro variable defined in Definition 5.6, and the probability of $\omega = \omega_i (i = 1,2,\ldots)$ be p_i, where $\sum_{i=1}^\infty p_i = 1$. if the series

$$\sum_{i=1}^\infty \omega_i f_i(\lambda\omega_i) \tag{5.14}$$

is absolutely convergent, then we call (5.14) as the expected value of discrete random variable $\xi(\lambda)$, denoted by $E[\xi(\lambda)]$.

Definition 5.9. Let ξ be a Ra-Ro variable defined in Definition 5.6, and the probability of $\omega = \omega_i (i = 1,2,\ldots)$ be p_i, where $\sum_{i=1}^\infty p_i = 1$. If the following equation exists,

$$V[\xi] = E[\xi - E[\xi]]^2 \tag{5.15}$$

we call $V[\xi]$ as the variance of ξ.

Lemma 5.5. *(Xu and Yao [347]) Suppose ξ be a Ra-Ro variable defined on $(\Lambda, \Delta, \mathscr{A}, \pi)$. Then*

$$V[\xi] = E[\xi^2] - (E[\xi])^2 \tag{5.16}$$

Proof. For any $\lambda \in \Lambda$, $\xi(\lambda)$ is a random rough variable, we have

$$\begin{aligned}
V[\xi] &= E(\xi^2) - [E(\xi)]^2 \\
&= \int_0^\infty \mathrm{Appr}\{E[\xi(\lambda) - E(\xi(\lambda))]^2 \geq r\}dr \\
&\quad - \int_{-\infty}^0 \mathrm{Appr}\{E[\xi(\lambda) - E(\xi(\lambda))]^2 \leq r\}dr \\
&= \int_0^\infty \mathrm{Appr}\{E(\xi^2(\lambda)) - [E(\xi(\lambda))]^2 \geq r\}dr \\
&\quad - \int_{-\infty}^0 \mathrm{Appr}\{E(\xi^2(\lambda)) - [E(\xi(\lambda))]^2 \leq r\}dr \\
&= \int_0^\infty \mathrm{Appr}\{E(\xi^2(\lambda)) \geq r\}dr - \int_0^\infty \mathrm{Appr}\{[E(\xi(\lambda))]^2 \geq r\}dr \\
&\quad - \left(\int_{-\infty}^0 \mathrm{Appr}\{E(\xi^2(\lambda)) \geq r\}dr - \int_{-\infty}^0 \mathrm{Appr}\{[E(\xi(\lambda))]^2 \geq r\}dr\right) \\
&= E[\xi^2] - (E[\xi])^2
\end{aligned}$$

This completes the proof. □

Lemma 5.6. *(Xu and Yao [347]) If ξ is a Ra-Ro variable whose variance exists, a and b are real numbers, then $V[a\xi + b] = a^2 V[\xi]$.*

Proof. It follows from the definition of variance that

$$V[a\xi + b] = E[(a\xi + b - aE[\xi] - b)^2] = a^2 E[(\xi - E[\xi])^2] = a^2 V[\xi] \quad (5.17)$$

□

Next, let's restrict our attention to three kinds of special discrete Ra-Ro variables and induce their properties.

In the traditional experiment, we all consider the probability of success or failure of one event as a certain value. However, not all the probability are crisp, we always find that it is a uncertain value as the statistical data increase. In the following section, we fasten on a kind of 0–1 distribution whose probability of success is a rough variable.

Definition 5.10. (Xu and Yao [347])(Ra-Ro 0-1 distribution) In an experiment, we assume that $\tilde{p} \vdash (\underline{X}, \overline{X})_R$ expresses the probability of success of the event ξ such that,

$$\overline{X} = \{\lambda | 0 \leq \lambda \leq 1\}; \qquad \underline{X} = \{\lambda | a \leq \lambda \leq b\}$$

where $0 \leq a < b \leq 1$. We call the event ξ is subject to Ra-Ro 0–1 distribution.

Obviously, the probability of failure $\tilde{q}$ of the event ξ is defined on the rough set $(\underline{X'}, \overline{X'})$ as follows,

$$\overline{X'} = \{\lambda | 0 \leq \lambda \leq 1\}; \qquad \underline{X'} = \{\lambda | 1 - b \leq \lambda \leq 1 - a\} \tag{5.18}$$

By Definition 5.6 and Lemma 5.5, the expected value and variance of ξ can be obtained as follows.

Theorem 5.1. *(Xu and Yao [347]) Let ξ follow Ra-Ro 0–1 distribution, and $\tilde{p} \vdash (\underline{X}, \overline{X})_R$ express the probability of success of the event ξ such that,*

$$\overline{X} = \{\lambda | 0 \leq \lambda \leq 1\}; \qquad \underline{X} = \{\lambda | a \leq \lambda \leq b\},$$

where $0 \leq a < b \leq 1$. Then

$$E[\xi] = \tfrac{1-\eta}{2}(a+b) + \tfrac{\eta}{2},$$
$$V[\xi] = [\tfrac{1-\eta}{2}(a^2+b^2) + \tfrac{\eta}{2}] - [\tfrac{1-\eta}{2}(a+b) + \tfrac{\eta}{2}]^2.$$

Proof. Since $\xi(\lambda)$ is a random variable subject to 0–1 distribution as follows,

$$\xi(\lambda) \sim \begin{bmatrix} 0 & 1 \\ \tilde{q} & \tilde{p} \end{bmatrix}$$

then

$$E[\xi(\lambda)] = 0 \cdot \tilde{q} + 1 \cdot \tilde{p} = \tilde{p}$$

Namely, $E[\xi(\lambda)]$ is an uncertain parameter approximated by the rough set $(\underline{X}, \overline{X})$. It follows that,

$$\mathrm{Appr}\{E[\xi(\lambda)] \geq r\} = \begin{cases} 0 & \text{if} \quad r \geq 1 \\ \eta(1-r) & \text{if} \quad b \leq r \leq 1 \\ \eta(1-r) + (1-\eta)\dfrac{b-r}{b-a} & \text{if} \quad a \leq r \leq b \\ 1 - \eta r & \text{if} \quad 0 \leq r \leq a \\ 1 & \text{if} \quad r \leq 0 \end{cases}$$

Then

$$\begin{aligned} E[\xi] &= \int_0^{+\infty} \mathrm{Appr}\{E[\xi(\lambda)] \geq r\} dr - \int_{-\infty}^{0} \mathrm{Appr}\{E[\xi(\lambda)] \leq r\} dr \\ &= \int_0^a (1 - \eta r) dr + \int_a^b \frac{1}{2}\left(\eta(1-r) + (1-\eta)\frac{b-r}{b-a}\right) dr + \int_b^1 \eta(1-r) dr \\ &= \frac{1-\eta}{2}(a+b) + \frac{\eta}{2} \end{aligned}$$

Next, we will compute the variance of ξ. Obviously, $\xi^2(\lambda)$ is a random variable subject to 0–1 distribution as follows,

$$\xi^2(\lambda) \sim \begin{bmatrix} 0 & 1 \\ 1-\tilde{p}^2 & \tilde{p}^2 \end{bmatrix}$$

Then

$$E[\xi^2(\lambda)] = 0 \cdot (1-\tilde{p}^2) + 1 \cdot \tilde{p}^2 = \tilde{p}^2$$

Obviously, $\tilde{p}^2 \vdash (\underline{X'}, \overline{X'})_R$ is defined as follows,

$$\overline{X'} = \{\lambda | 0 \le \lambda \le 1\}; \qquad \underline{X'} = \{\lambda | a^2 \le \lambda \le b^2\}.$$

Then we have

$$E[\xi^2] = \frac{1-\eta}{2}(a^2+b^2) + \frac{\eta}{2}$$

It follows from the property of random variables that

$$V[\xi] = E[\xi^2] - (E[\xi])^2 = \left[\frac{1-\eta}{2}(a^2+b^2) + \frac{\eta}{2}\right] - \left[\frac{1-\eta}{2}(a+b) + \frac{\eta}{2}\right]^2.$$

The proof is complete. □

As we know, in an experiment, if the probability of success is p, the probability $P(r)$ of r successes in m independent trials of the experiment is subject to binomial distribution. However, in some cases the probability p of success is not precise, which needs to be estimated or obtain from expert opinion. How can we deal with this kind of events? There is a technique of Ra-Ro binomial to solve this problem.

Definition 5.11. (Xu and Yao [347]) (Ra-Ro binomial distribution) Assume that $\tilde{p} \vdash (\underline{X}, \overline{X})_R$ expresses the probability of success of the event ξ such that,

$$\overline{X} = \{\lambda | 0 \le \lambda \le 1\}; \qquad \underline{X} = \{\lambda | a \le \lambda \le b\}$$

where $0 \le a < b \le 1$. We call the probability $\tilde{P}(r)$ of r successes in m independent trials of the experiment as random rough binomial.

With the rough arithmetic, we obtain

$$\tilde{P}(r) = \binom{m}{r} \tilde{p}^r \tilde{q}^{m-r} \tag{5.19}$$

where $\tilde{q} \vdash (\underline{X'}, \overline{X'})$ is the probability of the failed events, and

$$\overline{X'} = \{\lambda | 0 \le \lambda \le 1\}; \qquad \underline{X'} = \{\lambda | 1-b \le \lambda \le 1-a\}.$$

Theorem 5.2. *(Xu and Yao [347]) Let a Ra-Ro variable be subject to the binomial distribution. Assume that $\tilde{p} \vdash (\underline{X}, \overline{X})_R$ expresses the probability of the success in m independent trials of the experiment, where $\overline{X} = \{\lambda | 0 \leq \lambda \leq 1\}, \underline{X} = \{\lambda | a \leq \lambda \leq b\} (0 \leq a < b \leq 1)$. Then*

$$E[\xi] = \frac{1-\eta}{2}(b-a)m + \frac{\eta m}{2},$$
$$V[\xi] = m\left(\left[\frac{1-\eta}{2}(a^2+b^2) + \frac{\eta}{2}\right] - \left[\frac{1-\eta}{2}(a+b) + \frac{\eta}{2}\right]^2\right).$$

Proof. Because every trial is independent of each other, we can only compute the expected value of one trial in the experiment. Obviously, for the ith trial ξ_i, it is a Ra-Ro variable subject to the 0–1 distribution, that is,

$$\xi_i(\lambda) \sim \begin{bmatrix} 0 & 1 \\ \tilde{q} & \tilde{p} \end{bmatrix}$$

where $\tilde{q} \vdash (\underline{X'}, \overline{X'})$. Thus,

$$E[\xi_i(\lambda)] = 0 \cdot \tilde{q} + 1 \cdot \tilde{p} = \tilde{p}$$

Since ξ is independent with each other, it follows that,

$$E[\xi(\lambda)] = \sum_{i=1}^{m} E[\xi_i(\lambda)] = m\tilde{p}.$$

Since $\tilde{p}$ is an uncertain probability of the success in m independent trials of the experiment defined on the rough set $(\underline{X}, \overline{X})$, $E[\xi(\lambda)] = m\tilde{p}$ is an uncertain parameter defined on the rough set $(\underline{X}^*, \overline{X}^*)$ as follows

$$\overline{X}^* = \{\lambda | 0 \leq \lambda \leq m\}; \qquad \underline{X}^* = \{\lambda | am \leq \lambda \leq bm\}$$

By the definition of trust measure and Definition 5.7, we have

$$\mathrm{Appr}\{E[\xi(\lambda)] \geq r\} = \begin{cases} 0 & \text{if} \quad r \geq m \\ \dfrac{\eta(m-r)}{m} & \text{if} \quad bm \leq r \leq m \\ \eta\dfrac{m-r}{m} + (1-\eta)\dfrac{bm-r}{(b-a)m} & \text{if} \quad am \leq r \leq bm \\ \eta\dfrac{m-r}{m} + 1 - \eta & \text{if} \quad 0 \leq r \leq am \\ 1 & \text{if} \quad r \leq 0 \end{cases}$$

It follows that

$$\begin{aligned} E[\xi] &= \int_0^{+\infty} \mathrm{Appr}\{E[\xi(\lambda)] \geq r\}dr - \int_{-\infty}^{0} \mathrm{Appr}\{E[\xi(\lambda)] \leq r\}dr \\ &= \int_0^{am} \left(\eta\frac{m-r}{m} + 1 - \eta\right)dr + \int_{am}^{bm} \left(\eta\frac{m-r}{m} + (1-\eta)\frac{bm-r}{(b-a)m}\right)dr \\ &\quad + \int_{bm}^{m} \frac{\eta(m-r)}{m}dr = \frac{1-\eta}{2}(b-a)m + \frac{\eta m}{2} \end{aligned}$$

It follows from the proof of Theorem 5.1 that

$$E[\xi_i^2] = \frac{1-\eta}{2}(a^2+b^2) + \frac{\eta}{2}$$

Then

$$\begin{aligned} V[\xi_i] &= E[\xi_i - E[\xi_i]]^2 = E[\xi_i^2] - (E[\xi_i])^2 \\ &= \left[\frac{1-\eta}{2}(a^2+b^2) + \frac{\eta}{2}\right] - \left[\frac{1-\eta}{2}(a+b) + \frac{\eta}{2}\right]^2 \end{aligned}$$

It follows that

$$V[\xi] = \sum_{i=1}^{m} V[\xi_i] = m\left(\left[\frac{1-\eta}{2}(a^2+b^2) + \frac{\eta}{2}\right] - \left[\frac{1-\eta}{2}(a+b) + \frac{\eta}{2}\right]^2\right).$$

This completes the proof. □

In real-life problems, there are a lot of examples about poisson distribution. However, not all the parameters τ are certain. Some of them need to be estimated or obtain from expert opinion, so we apply the technique that we substitute a rough variable for τ to solve this problem. Before giving the definition of random rough poisson, we introduce the following lemmas.

Lemma 5.7. *(Xu and Yao [344]) Let ζ be an uncertain parameter defined on the rough set $(\underline{X}, \overline{X})$, where $\overline{X} = \{\lambda | c \leq \lambda \leq d\}, \underline{X} = \{\lambda | a \leq \lambda \leq b\}(0 < c \leq a < b \leq d)$. It follows that*

$$\begin{aligned} \zeta^k &\vdash ([a^k, b^k], [c^k, d^k])_R \\ e^{\zeta} &\vdash ([e^a, e^b], [e^c, e^d])_R \end{aligned}$$

where k is any positive integer.

Proof. According to the concept of interval number defined by Alefeld and Herzberger [3], a rough variable ξ could be considered as a kind of interval number. Since $\zeta \vdash ([a, b], [c, d])_R$, it follows from the interval arithmetic defined by Alefeld and Herzberger [3] and Hansen [129, 130] that, for $0 < c \leq a < b \leq d$,

$$([a,b],[c,d]) \times ([a,b],[c,d]) \times \cdots \times ([a,b],[c,d]) = ([a^k, b^k], [c^k, d^k])$$

Similarly,

$$\begin{aligned}\sum_{k=0}^{\infty} \frac{([a^k, b^k], [c^k, d^k])}{k!} &= \left(\left[\sum_{k=0}^{\infty} \frac{a^k}{k!}, \sum_{k=0}^{\infty} \frac{b^k}{k!} \right], \left[\sum_{k=0}^{\infty} \frac{c^k}{k!}, \sum_{k=0}^{\infty} \frac{d^k}{k!} \right] \right) \\ &= ([e^a, e^b], [e^c, e^d])\end{aligned}$$

Then we have

$$\begin{aligned}\zeta^k &\vdash ([a^k, b^k], [c^k, d^k])_R \\ e^{\zeta} &\vdash ([e^a, e^b], [e^c, e^d])_R\end{aligned}$$

This completes the proof. □

Lemma 5.8. *(Xu and Yao [347]) Let $\tau = m\tilde{p}$, where $\tilde{p}$ is an uncertain probability of the success defined on the rough set $([a,b],[0,1])$, $m = 1, 2, \ldots$. Then for any nonnegative integer r, we have*

$$\lim_{m\to\infty} \binom{m}{r} \tilde{p}^r \tilde{q}^{m-r} = \frac{\tau^r e^{-\tau}}{r!}$$

Proof. Obviously, τ is also an uncertain parameter defined on $([\tau_1, \tau_2], [\tau_3, \tau_4])$, $\tilde{p} \vdash ([a,b],[0,1])$, and $\tilde{q} \vdash ([1-b, 1-a], [0,1])$. Then $a = \frac{\tau_1}{m}, b = \frac{\tau_2}{m}, \tau_3 = 0, \tau_4 = m$. By Lemma 5.7, we have

$$\begin{aligned}\tilde{p}^r \tilde{q}^{m-r} &= \frac{m(m-1)\cdots(m-r+1)}{r!} ([a,b],[0,1])^r ([1-b,1-a],[0,1])^{m-r} \\ &= \frac{m(m-1)\cdots(m-r+1)}{r!} \cdot ([a^r, b^r], [0,1]) \cdot ([(1-b)^r, (1-a)^r], [0,1]) \\ &= \frac{m(m-1)\cdots(m-r+1)}{r!} \cdot \left(\left[\left(\frac{\tau_1}{m}\right)^r, \left(\frac{\tau_2}{m}\right)^r \right], \left[0, \left(\frac{\tau_4}{m}\right)^r \right] \right) \\ &\quad \cdot \left(\left[\left(1 - \frac{\tau_2}{m}\right)^{m-r}, \left(1 - \frac{\tau_1}{m}\right)^{m-r} \right], [0,1] \right) \\ &= \frac{1}{r!} \cdot ([\tau_1^r, \tau_2^r], [0, \tau_4^r]) \cdot \left(\left[\left(1 - \frac{\tau_2}{m}\right)^{m-r}, \left(1 - \frac{\tau_1}{m}\right)^{m-r} \right], [0,1] \right) \\ &\quad \cdot \left[1 \cdot \left(1 - \frac{1}{m}\right) \cdots \left(1 - \frac{r-1}{m}\right) \right]\end{aligned}$$

For every fixed r, when $m \to \infty$,

$$\begin{aligned}&\lim_{m\to\infty} \left(1 - \frac{\tau_2}{m}\right)^{m-r} = e^{-\tau_2}, \quad \lim_{m\to\infty} \left(1 - \frac{\tau_1}{m}\right)^{m-r} = e^{-\tau_1}, \\ &\lim_{m\to\infty} \left[1 \cdot \left(1 - \frac{1}{m}\right) \cdots \left(1 - \frac{r-1}{m}\right) \right] = 1.\end{aligned}$$

Since $\tau_4 = m$, $\lim_{m\to\infty} e^{-\tau_4} = 0$. Then

$$\lim_{m\to\infty} \binom{m}{r} \tilde{p}^r \tilde{q}^{m-r} = \frac{1}{r!} \cdot ([\tau_1^r, \tau_2^r], [0, \tau_4^r]) \cdot ([e^{-\tau_2}, e^{-\tau_1}], [0, 1]) = \frac{\tilde{\tau}^r e^{-\tilde{\tau}}}{r!}$$

This completes this proof. □

Definition 5.12. (Xu and Yao [347]) (Ra-Ro poisson distribution) Suppose the probability that random variable $\xi(\lambda)$ takes the value $0, 1, 2, \ldots$ is as follows

$$Pr\{\xi(\lambda) = r\} = \frac{\tilde{\tau}^r e^{-\tilde{\tau}}}{r!} \tag{5.20}$$

where τ is an uncertain parameter defined on the rough set $(\underline{X}, \overline{X})(\underline{X} = \{\lambda | 0 \leq \lambda \leq \infty\}, \overline{X} = \{\lambda | \tau_1 \leq \lambda \leq \tau_2\}, 0 \leq \tau_1 < \tau_2 \leq \infty)$, then we can say that ξ is a Poisson distributed Ra-Ro variable.

Theorem 5.3. *(Xu and Yao [347]) (The expected value of Ra-Ro poisson) Assume that ξ follows Ra-Ro poisson distribution. The probability P_r of $\xi(\lambda) = r$ is $\frac{\tau^r}{r!} e^{-\tau} (k = 0, 1, 2, \ldots, \infty)$, $\tau \vdash ([a, b], [c, d])_R$ is an uncertain parameter, $0 < c \leq a < b \leq d$. Then*

$$\begin{aligned} E[\xi] &= \frac{1-\eta}{2}(a+b) + \frac{\eta}{2}(c+d), \\ V[\xi] &= \left[\frac{1-\eta}{2}(a+b+a^2+b^2) + \frac{\eta}{2}(c+d+c^2+d^2)\right] \\ &\quad - \left[\frac{1-\eta}{2}(a+b) + \frac{\eta}{2}(c+d)\right]^2. \end{aligned}$$

Proof. According to the definition of discrete Ra-Ro variable, we can get that $\xi(\lambda)$ is a random variable. We must compute the expected value of random event $\xi(\lambda)$ by Definition 5.7. Because random event $\xi(\lambda)$ satisfies poisson distribution, from Lemmas 5.7 and 5.8, it follows that

$$E[\xi(\lambda)] = \sum_{r=0}^{\infty} r \cdot P_r = \sum_{r=0}^{\infty} r \cdot \frac{\tau^r}{r!} e^{-\tau} = \tau$$

Obviously, $E[\xi(\lambda)]$ is uncertain because the uncertainty of τ. By the definition of expected value of Ra-Ro variable, we can calculate the expected value of ξ.

$$\begin{aligned} E[\xi] &= \int_0^{+\infty} \text{Appr}\{E[\xi(\lambda)] \geq r\} dr - \int_{-\infty}^{0} \text{Appr}\{E[\xi(\lambda)] \leq r\} dr \\ &= \frac{1-\eta}{2}(a+b) + \frac{\eta}{2}(c+d) \end{aligned}$$

Then

$$
\begin{aligned}
E[\xi^2(\lambda)] &= \sum_{r=0}^{\infty} r^2 \cdot p_r = \sum_{r=0}^{\infty} r^2 \cdot \frac{\tau^r}{r!} e^{-\tau} = \sum_{r=1}^{\infty} (r-1+1) \cdot \frac{\tau^r}{(r-1)!} e^{-\tau} \\
&= \tau^2 \sum_{r=2}^{\infty} \frac{\tau^{r-2}}{(r-2)!} e^{-\tau} + \tau \sum_{r=1}^{\infty} \frac{\tau^{r-1}}{(r-1)!} e^{-\tau} \\
&= \tau^2 + \tau
\end{aligned}
$$

It follows that $E[\xi^2(\lambda)] \vdash (\underline{X}^+, \overline{X}^+)_R$, where $\overline{X}^+ = \{\lambda | c^2 + c \le \lambda \le d^2 + d\}$, $\underline{X}^+ = \{\lambda | a^2 + a \le \lambda \le b^2 + b\}$. Thus,

$$
\begin{aligned}
E[\xi^2] &= \int_0^{+\infty} \mathrm{Appr}\{E[\xi^2(\lambda)] \ge r\} dr - \int_{-\infty}^{0} \mathrm{Appr}\{E[\xi^2(\lambda)] \le r\} dr \\
&= \frac{1-\eta}{2}(a + b + a^2 + b^2) + \frac{\eta}{2}(c + d + c^2 + d^2)
\end{aligned}
$$

We have

$$
\begin{aligned}
V[\xi] &= E[\xi^2] - [E(\xi)]^2 \\
&= \left[\frac{1-\eta}{2}(a + b + a^2 + b^2) + \frac{\eta}{2}(c + d + c^2 + d^2) \right] \\
&\quad - \left[\frac{1-\eta}{2}(a + b) + \frac{\eta}{2}(c + d) \right]^2.
\end{aligned}
$$

This completes the proof. □

5.2.2.2 Continuous Ra-Ro Variable

As random variable has continuous distribution function, we can define the continuous Ra-Ro variable.

Definition 5.13. (Xu and Yao [344]) (Continuous Ra-Ro variable) For a Ra-Ro variable ξ, if $\xi(\lambda)$ is a random variable with a continuous distributive function with a parameter approximated by the rough set $(\underline{X}, \overline{X})$. Then we call ξ a continuous Ra-Ro variable.

Example 5.5. The lifetime ξ of a modern engine is an exponentially distributed variable with an unknown expected value θ, and has the following form of probability density function,

$$
\phi(x) = \begin{cases} \frac{1}{\theta} e^{-x/\theta}, & \text{if} \quad 0 \le x < \infty \\ 0, & \text{otherwise} \end{cases} \tag{5.21}
$$

where $\theta \vdash ([a, b], [c, d])_R$ and $([a, b], [c, d])$ is a rough set under the similarity relationship R. Then $\xi(\lambda)$ is a continuous random variable for each θ. Hence, ξ is a continuous Ra-Ro variable.

Definition 5.14. (Xu and Yao [344]) (The differentiation of Ra-Ro variable) If there exists differentiation for a function $F(\xi, f(x))$ with Ra-Ro coefficients, we can define it as following

$$\frac{\partial F(\xi, f(x))}{\partial x} = F\left(\xi, \frac{\partial f(x)}{\partial x}\right) \tag{5.22}$$

Definition 5.15. (Xu and Yao [344]) Suppose ξ is a Ra-Ro variable, then $\xi(\lambda)$ is a random variable. If the density function of $\xi(\lambda)$ is $f(x, \xi)$, and

$$E[\xi(\lambda)] = \int_{x\in\Omega} xf(x)dx = ([a, b], [c, d]) \tag{5.23}$$

where a, b, c, d are finite real numbers. Then we call $f(x, \xi)$ as density function of Ra-Ro variable ξ.

Definition 5.16. (Xu and Yao [344]) For the continuous Ra-Ro variable ξ, if its density function is $f(x)$, then we can define the expected value of ξ as follows

$$E[\xi] = \int_0^\infty \text{Appr}\left\{\int_{x\in\Omega} xf(x)dx \geq r\right\} dr - \int_{-\infty}^0 \text{Appr}\left\{\int_{x\in\Omega} xf(x)dx \leq r\right\} dr$$

Let's pay more attention to three special kinds of continuous Ra-Ro variables and research their properties.

Assume a random variable follows the uniform distribution and has the distribution function $U(a, b)$. However, in some real problems, all the parameters a and b are both uncertain, and need to be estimated through experience and historical data. Here, we assume that they are both approximated by different rough sets $(\underline{X}, \overline{X})$ and $(\underline{X}', \overline{X}')$, respectively.

Definition 5.17. (Xu and Yao [347]) (Ra-Ro uniform distribution) Let ξ be a random variable which follows uniform distribution and has the following density function,

$$f(\lambda)(x) = \begin{cases} \dfrac{1}{\tilde{b} - \tilde{a}} & \text{if} \quad \tilde{a} \leq x \leq \tilde{b} \\ 0 & \text{otherwise} \end{cases}$$

where $\tilde{a} \vdash (\underline{X}, \overline{X})$ and $\tilde{b} \vdash (\underline{X}', \overline{X}')$ are two uncertain parameters, and

$$\begin{aligned} &\overline{X} = \{\lambda | s_1 \leq \lambda \leq s_4\}, \underline{X} = \{\lambda | s_2 \leq \lambda \leq s_3\} \\ &\overline{X}' = \{\lambda | t_1 \leq \lambda \leq t_4\}, \underline{X}' = \{\lambda | t_2 \leq \lambda \leq t_3\} \end{aligned} \tag{5.24}$$

then ξ is subject to Ra-Ro uniform distribution, denoted as $\xi \sim U(\tilde{a}, \tilde{b})$.

According to the definition of the expected value and variance of continuous Ra-Ro, we can get the following theorem.

Theorem 5.4. *(Xu and Yao [347]) (The expected value of Ra-Ro uniform) Assume that* $\xi \sim U(\tilde{a}, \tilde{b})$ *is an uniformly distributed Ra-Ro variable, where* $\tilde{a} \vdash ([s_2, s_3], [s_1, s_4])_R$ $(0 < s_1 < s_2 < s_3 < s_4)$ *and* $\tilde{b} \vdash ([t_2, t_3], [t_1, t_4])$ $(0 < t_1 < t_2 < t_3 < t_4, s_i < t_i)$. *Then we have*

$$\begin{aligned} E[\xi] &= \frac{1-\eta}{4}(s_2+t_2+s_3+t_3)+\frac{\eta}{4}(s_1+t_1+s_4+t_4), \\ V[\xi] &= \left[\frac{1-\eta}{12}(t_2^2+s_2t_2+s_2^2+t_3^2+s_3t_3+s_3^2)\right. \\ &\quad \left.+\frac{\eta}{12}(t_1^2+s_1t_1+s_1^2+t_4^2+s_4t_4+s_4^2)\right] \\ &\quad -\left[\frac{1-\eta}{4}(s_2+t_2+s_3+t_3)+\frac{\eta}{4}(s_1+t_1+s_4+t_4)\right]^2. \end{aligned}$$

Proof. According to Definition 5.16, we have

$$\begin{aligned} E[\xi] &= \int_0^{+\infty} \operatorname{Appr}\left\{\int_\Omega xf(\lambda)(x)dx \geq r\right\} dr \\ &\quad -\int_{-\infty}^0 \operatorname{Appr}\left\{\int_\Omega xf(\lambda)(x)dx \leq r\right\} dr \\ &= \int_0^{+\infty} \operatorname{Appr}\left\{\int_0^{+\infty} x\frac{1}{\tilde{b}-\tilde{a}}dx \geq r\right\} dr \\ &\quad -\int_{-\infty}^0 \operatorname{Appr}\left\{\int_0^{+\infty} x\frac{1}{\tilde{b}-\tilde{a}}dx \leq r\right\} dr \\ &= \int_0^{+\infty} \operatorname{Appr}\left\{\frac{\tilde{a}+\tilde{b}}{2} \geq r\right\} dr - \int_{-\infty}^0 \operatorname{Appr}\left\{\frac{\tilde{a}+\tilde{b}}{2} \leq r\right\} dr \end{aligned}$$

According to rough arithmetic operators, it follows that,

$$\frac{\tilde{a}+\tilde{b}}{2} \vdash \left(\left[\frac{s_2+t_2}{2}, \frac{s_3+t_3}{2}\right], \left[\frac{s_1+t_1}{2}, \frac{s_4+t_4}{2}\right]\right)_R$$

Then we have

$$E[\xi] = \frac{1-\eta}{4}(s_2+t_2+s_3+t_3)+\frac{\eta}{4}(s_1+t_1+s_4+t_4).$$

Similarly, we can compute the expected value of ξ^2 as follows

$$\begin{aligned} E[\xi^2] &= \frac{1-\eta}{12}(t_2^2+s_2t_2+s_2^2+t_3^2+s_3t_3+s_3^2) \\ &\quad +\frac{\eta}{12}(t_1^2+s_1t_1+s_1^2+t_4^2+s_4t_4+s_4^2) \end{aligned}$$

Then we can get its variance by $V[\xi] = E[\xi - E[\xi]]^2$. The proof is completed. □

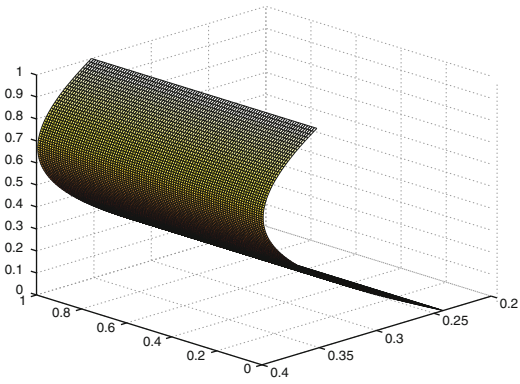


Fig. 5.8 Ra-Ro normal distribution

The normal distribution of random variables is often found in many real-life problems, for example, errors of some equipments are subject to normal distribution. However, not all means or variances are precise, they usually need to be estimated by historical data or experience. Therefore, we propose the Ra-Ro normal distribution to describe the uncertain mean or variance.

Definition 5.18. (Xu and Yao [347]) (Ra-Ro normal distribution) Suppose ξ is a Ra-Ro variable. If the random variable $\xi(\lambda)$ follows normal distribution with uncertain expected value $\tilde{\mu} \vdash (\underline{X}, \overline{X})_R$ or variance $\tilde{\sigma}^2 \vdash (\underline{X}, \overline{X})_R$ or both of them, then ξ is subject to Ra-Ro normal distribution, denoted as $\xi \sim \mathcal{N}(\tilde{\mu}, \tilde{\sigma}^2)$ (see Fig. 5.8).

By the definition, we know that there are two kinds of basic random rough variables which are subject to normal distribution. One is a random variable which is subject to normal distribution and whose expected value is an uncertain variable approximated by the rough set $(\underline{X}, \overline{X})$. The other is a random variable with normal distribution whose variance an uncertain variable approximated by the rough set $(\underline{X}, \overline{X})$. Next, let's discuss the expected value and variance of the two kinds of Ra-Ro normal variables, respectively. See Examples 5.6 and 5.7.

Example 5.6. Suppose $\xi \sim \mathcal{N}(\tilde{\mu}, \sigma^2)$ is a random rough variable, where $\tilde{\mu} \vdash ([a, b], [c, d])_R$. Then ξ is Ra-Ro variable subject to normal distribution.

Example 5.7. Suppose $\xi \sim \mathcal{N}(\mu, \tilde{\sigma}^2)$ is a random rough variable, where $\tilde{\sigma}$ is approximated by the rough set $(\underline{X}', \bar{X}')$ defined as follows,

$$\underline{X}' = \{x | a' \leq x \leq b'\}, \bar{X}' = \{x | c' \leq x \leq d'\} \tag{5.25}$$

Then ξ is Ra-Ro variable subject to normal distribution.

Next, let's discuss the expected value and variance of the two kinds of Ra-Ro normal variables.

Theorem 5.5. *(Xu and Yao [347]) Suppose $\xi \sim \mathcal{N}(\tilde{\mu}, \sigma^2)$ is a normally distributed Ra-Ro variable, where $\tilde{\mu} \vdash (\underline{X}, \overline{X})_R$ and $\overline{X} = \{\lambda | c \leq \lambda \leq d\}, \underline{X} = \{\lambda | a \leq \lambda \leq b\}, 0 < c \leq a < b \leq d$. Then*

$$E[\xi] = \frac{1-\eta}{2}(a+b) + \frac{\eta}{2}(c+d),$$
$$V[\xi] = \left[\frac{1-\eta}{2}(a^2+b^2) + \frac{\eta}{2}(c^2+d^2) + \sigma^2\right] - \left[\frac{1-\eta}{2}(a+b) + \frac{\eta}{2}(c+d)\right]^2.$$

Proof. Since $\xi \sim \mathcal{N}(\tilde{\mu}, \sigma^2)$, we have

$$E[\xi(\lambda)] = \tilde{\mu}.$$

It follows that

$$\begin{aligned} E[\xi] &= \int_0^{+\infty} \mathrm{Appr}\{\lambda \in \Lambda | E[\xi(\lambda)] \geq r\} dr - \int_{-\infty}^{0} \mathrm{Appr}\{\lambda \in \Lambda | E[\xi(\lambda)] \leq r\} dr \\ &= \frac{1-\eta}{2}(a+b) + \frac{\eta}{2}(c+d) \end{aligned}$$

Since

$$E[\xi^2(\lambda)] = \int_{-\infty}^{+\infty} x^2 \cdot \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\tilde{\mu})^2}{2\sigma^2}} dx = \sigma^2 + \tilde{\mu}^2$$

It follows from expected value operator that

$$E[\xi^2] = \frac{1-\eta}{2}(a^2+b^2) + \frac{\eta}{2}(c^2+d^2) + \sigma^2.$$

Then we have

$$\begin{aligned} V[\xi] &= E[\xi^2] - (E[\xi])^2 \\ &= \left[\frac{1-\eta}{2}(a^2+b^2) + \frac{\eta}{2}(c^2+d^2) + \sigma^2\right] - \left[\frac{1-\eta}{2}(a+b) + \frac{\eta}{2}(c+d)\right]^2. \end{aligned}$$

The theorem is proved. □

Theorem 5.6. *(Xu and Yao [347]) Suppose $\xi \sim \mathcal{N}(\mu, \tilde{\sigma}^2)$ is a normally distributed Ra-Ro variable, where $\tilde{\sigma} \vdash (\underline{X}', \overline{X}')_R$ and $\overline{X}' = \{\lambda | c' \leq \lambda \leq d'\}, \underline{X}' = \{\lambda | a' \leq \lambda \leq b'\}, 0 < c' \leq a' < b' \leq d'$. Then*

$$E[\xi] = \mu, V[\xi] = \frac{1-\eta}{2}(a'^2 + b'^2) + \frac{\eta}{2}(c'^2 + d'^2)$$

Proof. Since $\xi(\lambda) \sim N(\mu, \tilde{\sigma}^2)$, we have

$$E[\xi] = \mu.$$

Next, we compute the variance of ξ.

$$E[\xi^2(\lambda)] = \int_{-\infty}^{+\infty} x^2 \cdot \frac{1}{\sqrt{2\pi}\tilde{\sigma}} e^{-\frac{(x-\mu)^2}{2\tilde{\sigma}^2}} dx = \tilde{\sigma}^2 + \mu^2$$

It follow from the expected value operator of the rough variable that

$$E[\xi^2] = \mu^2 + \frac{1-\eta}{2}(a'^2 + b'^2) + \frac{\eta}{2}(c'^2 + d'^2)$$

It can be easily obtained as follows

$$V[\xi] = E[\xi^2] - (E[\xi])^2 = \frac{1-\eta}{2}(a'^2 + b'^2) + \frac{\eta}{2}(c'^2 + d'^2).$$

The theorem is proved. □

Similarly, we only consider the continuous Ra-Ro variable following exponential distribution which is the most universal in the realistic world. The Ra-Ro exponential variable can be applied in describing many uncertain events and solving these uncertain problems.

Definition 5.19. (Xu and Yao [347]) (Ra-Ro exponential distribution) If ξ is a random variable subject to exponential distribution with uncertain parameter approximated by the rough set $(\underline{X}, \overline{X})$ under the similarity relationship R. Then ξ follows Ra-Ro exponential distribution, denoted as $\xi \sim exp(\lambda)$ (see Fig. 5.9).

The same with Ra-Ro normal distribution, we can also define its density function as follows,

$$p(x) = \begin{cases} \tilde{\eta} e^{-\tilde{\eta} x}, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

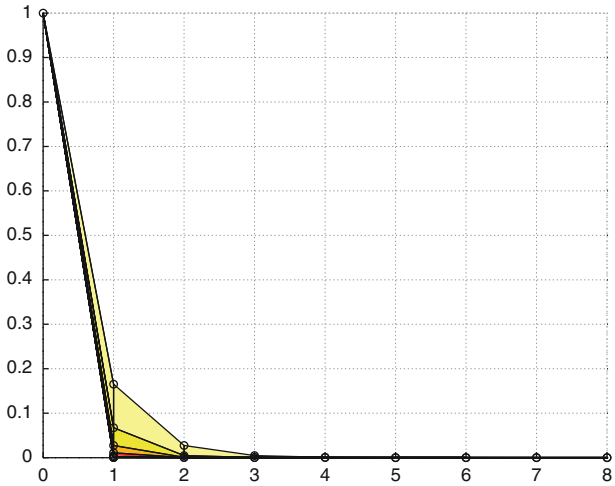


Fig. 5.9 Ra-Ro exponential distribution

where $\tilde{\eta}$ is approximated by the rough set $(\underline{X}, \overline{X})$ under the similarity relationship R. Then it is easy to obtain the following theorem.

Theorem 5.7. *(Xu and Yao [347]) Assume that the density function of Ra-Ro variable ξ is $p(x)$, which is characterized as follows,*

$$p(x) = \begin{cases} \tilde{\kappa} e^{-\tilde{\kappa} x}, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

where $\tilde{\kappa} \vdash (\underline{X}, \overline{X})_R$ is an uncertain parameter and $\overline{X} = \{\lambda | c \leq \lambda \leq d\}, \underline{X} = \{\lambda | a \leq \lambda \leq b\}, 0 < c \leq a < b \leq d$. Then

$$\begin{aligned} E[\xi] &= \frac{1-\eta}{2}\left(\frac{1}{a}+\frac{1}{b}\right)+\frac{\eta}{2}\left(\frac{1}{c}+\frac{1}{d}\right), \\ V[\xi] &= \left[\frac{1-\eta}{2}\left(\frac{1}{a^2}+\frac{1}{b^2}\right)+\frac{\eta}{2}\left(\frac{1}{c^2}+\frac{1}{d^2}\right)\right] \\ &\quad -\left[\frac{1-\eta}{2}\left(\frac{1}{a}+\frac{1}{b}\right)+\frac{\eta}{2}\left(\frac{1}{c}+\frac{1}{d}\right)\right]^2. \end{aligned}$$

Proof. According to Definition 5.16, we have

$$\begin{aligned} E[\xi] &= \int_0^{+\infty} \operatorname{Appr}\left\{\int_\Omega xp(x)dx \geq r\right\} dr - \int_{-\infty}^0 \operatorname{Appr}\left\{\int_\Omega xp(x)dx \leq r\right\} dr \\ &= \int_0^{+\infty} \operatorname{Appr}\left\{\frac{1}{\tilde{\kappa}} \geq r\right\} dr - \int_{-\infty}^0 \operatorname{Appr}\left\{\frac{1}{\tilde{\kappa}} \leq r\right\} dr \end{aligned}$$

Since $\tilde{\kappa} \vdash ([a,b],[c,d])(0 < c \leq a < b \leq d)$, it follows from the interval number operator that

$$\frac{1}{\tilde{\kappa}} \vdash \left(\left[\frac{1}{b}, \frac{1}{a}\right], \left[\frac{1}{d}, \frac{1}{c}\right]\right)_R$$

Then we have

$$E[\xi] = \frac{1-\eta}{2}\left(\frac{1}{a}+\frac{1}{b}\right)+\frac{\eta}{2}\left(\frac{1}{c}+\frac{1}{d}\right).$$

Since

$$E[\xi^2(\lambda)] = \frac{1}{\tilde{\kappa}^2},$$

we have

$$E[\xi^2] = \frac{1-\eta}{2}\left(\frac{1}{a^2}+\frac{1}{b^2}\right)+\frac{\eta}{2}\left(\frac{1}{c^2}+\frac{1}{d^2}\right).$$

Thus,

$$V[\xi] = \left[\frac{1-\eta}{2}\left(\frac{1}{a^2}+\frac{1}{b^2}\right)+\frac{\eta}{2}\left(\frac{1}{c^2}+\frac{1}{d^2}\right)\right] - \left[\frac{1-\eta}{2}\left(\frac{1}{a}+\frac{1}{b}\right)+\frac{\eta}{2}\left(\frac{1}{c}+\frac{1}{d}\right)\right]^2$$

This completes the proof. □

5.3 Ra-Ro EVM

Optimization is usually to seek solution over a set of possible by certain criteria. If the decision maker takes only one criterion into consideration, it is a single objective optimization problem, which have been well studied for the past 50 years. If DM considers more than one criterion simultaneously, the multi-objective optimization problems are presented and widely researched. Recently, many scholars are studying this kind of problem which arises in many areas of real world. For example, Bhatia [24] introduced the higher order strong convexity to multi-objective optimization problem and found the equivalence of mixed saddle points. Zhang [361] designed a new multi-objective optimization technique, multi-objective optimization immune algorithm to get the Pareto optimal solution. However, in the real-life optimization and decision making problems, the multi-objective programming problems are very complex and there must be many uncertain factors to be considered. Dietz et al. [79] considered demands as fuzzy variables in the problem of optimal design of batch plants. Slowinski [294] applied the method of rough sets to solve the uncertain problem in the medical domain. Greco, Matarazzo and Slowinski [120] then applied rough sets method to sort data according multiple attributes and criteria. Sakawa et al [273] applied the interactive fuzzy satisfying method to solve a class of multi-objective linear programming problems with random variable coefficients. Recently, many scholars pay their attention to the programming with two-fold uncertain parameters such as fuzzy random programming and birandom programming. Xu and Yao [344] introduced the expected value multi-objective programming problems with Ra-Ro coefficients. In this section, we will recall its basic properties and extend them. Consider the following multi-objective programming problem with Ra-Ro coefficients

$$\begin{cases} \max[f_1(\boldsymbol{x},\boldsymbol{\xi}), f_2(\boldsymbol{x},\boldsymbol{\xi}),\ldots,f_m(\boldsymbol{x},\boldsymbol{\xi})] \\ \text{s.t.} \quad g_j(\boldsymbol{x},\boldsymbol{\xi}) \le 0, j = 1,2,\ldots,p \end{cases} \tag{5.26}$$

where $\boldsymbol{x} \in X \subset \mathbf{R}$ is a n-dimensional decision vector, $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_n)$ is a Ra-Ro vector, $f_i(\boldsymbol{x},\boldsymbol{\xi})$ are objective functions, $i = 1,2,\ldots,p$. Because of the existence of Ra-Ro vector $\boldsymbol{\xi}$, problem (5.26) is not well-defined. That is, the meaning of maximizing $f_i(\boldsymbol{x},\boldsymbol{\xi}), i = 1,2,\ldots,m$ is not clear and constraints $g_r(\boldsymbol{x},\boldsymbol{\xi}) \le 0$,

$r = 1, 2, \ldots, p$ do not define a deterministic feasible set. To deal with the Ra-Ro events ξ, the expected value programming, chance-constraint programming, and dependent-chance programming are brought forward.

We can also formulate a Ra-Ro decision system as a random rough goal programming model according to the priority structure and target levels set by the decision-maker,

$$\begin{cases} \min \sum_{j=1}^{l} P_j \sum_{j=1}^{m} (u_{ij} d_i^+ + v_{ij} d_i^-) \\ \text{s.t.} \begin{cases} f_i(\boldsymbol{x}, \boldsymbol{\xi}) + d_i^- - d_i^+ = b_i \quad i = 1, 2, \ldots, m \\ g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, j = 1, 2, \ldots, p \end{cases} \end{cases}$$

where P_j is the preemptive priority factor which expresses the relative importance of various goals, $P_j \gg P_{j+1}$, for all j, u_{ij} is the weighting factor corresponding to positive deviation for goal i with priority j assigned, v_{ij} is the weighting factor corresponding to negative deviation for goal i with priority j assigned, d_i^+ is the positive deviation from the target of goal i, defined as

$$d_i^+ = [f_i(\boldsymbol{x}, \boldsymbol{\xi}) - b_i] \vee 0,$$

d_i^- is the negative deviation from the target of goal i, defined as

$$d_i^- = [b_i - f_i(\boldsymbol{x}, \boldsymbol{\xi})] \vee 0,$$

f_i is a function in goal constraints, g_j is a function in real constraints, b_i is the target value according to goal i, l is the number of priorities, m is the number of goal constraints, and p is the number of real constraints.

5.3.1 General Model for Ra-Ro EVM

Based on the definition of the expected value of random rough events f_j and g_j, the maximum expected value multi-objective model (EVM) is proposed as follows,

$$\begin{cases} \max[E[f_1(\boldsymbol{x}, \boldsymbol{\xi})], E[f_2(\boldsymbol{x}, \boldsymbol{\xi})], \ldots, E[f_m(\boldsymbol{x}, \boldsymbol{\xi})]] \\ \text{s.t.} \quad E[g_j(\boldsymbol{x}, \boldsymbol{\xi})] \leq 0, j = 1, 2, \ldots, p \end{cases} \tag{5.27}$$

Definition 5.20. (Xu and Yao [344]) If $\boldsymbol{x}^*$ is an efficient solution of problem (5.27), we call it as a Ra-Ro expected efficient solution.

Clearly, the problem (5.27) is a multi-objective with crisp parameters. Then we can convert it into a single-objective programming by traditional method of weight sum,

$$\begin{cases} \max \sum_{i=1}^{m} \omega_i E[f_i(\boldsymbol{x}, \boldsymbol{\xi})] \\ \text{s.t.} \begin{cases} E[g_j(\boldsymbol{x}, \boldsymbol{\xi})] \leq 0, j = 1, 2, \ldots, p \\ \omega_1 + \omega_2 + \cdots + \omega_m = 1 \end{cases} \end{cases} \tag{5.28}$$

Theorem 5.8. *(Xu and Yao [344]) Problem (5.28) is equivalent to problem (5.27), i.e., the efficient solution of problem (5.27) is the optimal solution of problem (5.28) and the optimal solution of problem (5.28) is the efficient solution of problem (5.27).*

Proof. Apparently, the efficient solution of problem (5.27) is the optimal solution of problem (5.28). Let 's consider wether the optimal solution of problem (5.28) is the efficient solution of problem (5.27).

Suppose $\boldsymbol{x}^*$ is an optimal solution of problem (5.26). If $\boldsymbol{x}^*$ is not an efficient solution of problem (5.27), then there exists $\boldsymbol{x}_0$ such that $E[f_i(\boldsymbol{x}^0, \boldsymbol{\xi})] \geq E[f_i(\boldsymbol{x}^*, \boldsymbol{\xi})](i = 1, 2, \ldots, m)$, and there at least exists a k such that $E[f_k(\boldsymbol{x}^0, \boldsymbol{\xi})] > E[f_k(\boldsymbol{x}^*, \boldsymbol{\xi})]$. Then,

$$\sum_{i=1}^{m} \omega_i E[f_i(\boldsymbol{x}^0, \boldsymbol{\xi})] > \sum_{i=1}^{m} \omega_i E[f_i(\boldsymbol{x}^*, \boldsymbol{\xi})].$$

This conflicts with the assumption that $\boldsymbol{x}^*$ is an optimal solution of problem (5.28). This completes the proof. □

Theorem 5.9. *(Xu and Yao [344]) Let $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_n)$ be a random rough vector on the rough space$(\Lambda, \Delta, \mathscr{A}, \pi)$, and f_i and $g_j : \mathscr{A}^n \to \mathscr{A}$ be convex continuous functions with respect to $\boldsymbol{x}$, $i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, p$. Then the expected value programming problem (5.28) is a convex programming.*

Proof. Let $\boldsymbol{x}^1$ and $\boldsymbol{x}^2$ be two feasible solutions. Because $g_j(\boldsymbol{x}, \boldsymbol{\xi})$ is a convex continuous function with respect to $\boldsymbol{x}$, then

$$g_j(\rho \boldsymbol{x}^1 + (1 - \rho)\boldsymbol{x}^2, \boldsymbol{\xi}) \leq \rho g_j(\boldsymbol{x}^1, \boldsymbol{\xi}) + (1 - \rho) g_j(\boldsymbol{x}^2, \boldsymbol{\xi}),$$

where $0 \leq \rho \leq 1$, $j = 1, 2, \ldots, p$. We can have

$$E[g_j(\rho \boldsymbol{x}^1 + (1 - \rho)\boldsymbol{x}^2, \boldsymbol{\xi})] \leq E[\rho g_j(\boldsymbol{x}^1, \boldsymbol{\xi}) + (1 - \rho) g_j(\boldsymbol{x}^2, \boldsymbol{\xi})].$$

Because for any $\lambda \in \Lambda$, $\boldsymbol{\xi}(\lambda)$ is a random vector. Then by the linearity of expected value of random variable, we have

$$E[g_j(\boldsymbol{x}^1, \boldsymbol{\xi}(\lambda)) + (1 - \rho) g_j(\boldsymbol{x}^2, \boldsymbol{\xi}(\lambda))] = \rho E[g_j(\boldsymbol{x}^1, \boldsymbol{\xi}(\lambda))] + (1 - \rho) E[g_j(\boldsymbol{x}^2, \boldsymbol{\xi}(\lambda))]$$

Following the linearity of expected value operator of rough variable, we can obtain

$$
\begin{aligned}
&E[\rho g_j(\boldsymbol{x}^1, \boldsymbol{\xi}) + (1-\rho) g_j(\boldsymbol{x}^2, \boldsymbol{\xi})] \\
&\quad = E[\rho E[g_j(\boldsymbol{x}^1, \boldsymbol{\xi}(\lambda))] + (1-\rho) E[g_j(\boldsymbol{x}^2, \boldsymbol{\xi}(\lambda))]] \\
&\quad = \rho E[E[g_j(\boldsymbol{x}^1, \boldsymbol{\xi}(\lambda))]] + 1 - \rho) E[E[g_j(\boldsymbol{x}^2, \boldsymbol{\xi}(\lambda))]] \\
&\quad = E[g_j(\boldsymbol{x}^1, \boldsymbol{\xi})] + (1-\rho) E[g_j(\boldsymbol{x}^2, \boldsymbol{\xi})].
\end{aligned}
$$

Then $E[g_j(\rho \boldsymbol{x}^1 + (1-\rho)\boldsymbol{x}^2, \boldsymbol{\xi})] \leq \rho E[g_j(\boldsymbol{x}^1, \boldsymbol{\xi})] + (1-\rho) E[g_j(\boldsymbol{x}^2, \boldsymbol{\xi})] \leq 0$. This means that $\rho \boldsymbol{x}^1 + (1-\rho)\boldsymbol{x}^2$ is also a feasible solution. Then $X(\boldsymbol{x} \in X)$ is a convex feasible set.

For every i, $f_i(\boldsymbol{x}, \boldsymbol{\xi})$ is a convex continuous function with respect to $\boldsymbol{x}$, it follows that

$$
f_i(\rho \boldsymbol{x}^1 + (1-\rho)\boldsymbol{x}^2, \boldsymbol{\xi}) \leq \rho f_i(\boldsymbol{x}^1, \boldsymbol{\xi}) + (1-\rho) f_i(\boldsymbol{x}^2, \boldsymbol{\xi}),
$$

then

$$
E[f_i(\rho \boldsymbol{x}^1 + (1-\rho)\boldsymbol{x}^2, \boldsymbol{\xi})] \leq \rho E[f_i(\boldsymbol{x}^1, \boldsymbol{\xi})] + (1-\rho) E[f_i(\boldsymbol{x}^2, \boldsymbol{\xi})],
$$

then

$$
\begin{aligned}
\sum_{i=1}^{m} \omega_i E[f_i(\rho \boldsymbol{x}^1 + (1-\rho)\boldsymbol{x}^2, \boldsymbol{\xi})] \leq \rho \sum_{i=1}^{m} \omega_i E[f_j(\boldsymbol{x}^1, \boldsymbol{\xi})] \\
+ (1-\rho) \sum_{i=1}^{r} \omega_i E[f_j(\boldsymbol{x}^2, \boldsymbol{\xi})].
\end{aligned}
$$

This means function $\sum_{i=1}^{m} \omega_i E[f_i(\boldsymbol{x}, \boldsymbol{\xi})]$ is convex. Above all, we can conclude that the expected value programming problem (5.28) is a convex programming. □

We can also obtain the Ra-Ro expected value goal programming as follows,

$$
\begin{cases}
\min \left(\sum_{i=1}^{m} P_i (u_i d_i^+ + v_i d_i^-) + \sum_{r=1}^{m} P_j (u_j d_j^+ + v_j d_j^-) \right) \\
\text{s.t.} \begin{cases}
E[f_i(\boldsymbol{x}, \boldsymbol{\xi})] + d_i^- - d_i^+ = q_i, i = 1, 2, \ldots, m \\
E[g_j(\boldsymbol{x}, \boldsymbol{\xi})] + d_j^- - d_j^+ = 0, j = 1, 2, \ldots, p \\
d_i^-, d_i^+, d_j^-, d_j^+ \geq 0 \\
u_i, v_i, u_j, v_j = 0 \quad \text{or} \quad 1
\end{cases}
\end{cases}
\tag{5.29}
$$

where P_i, P_j are the priority coefficients that express the importance of goals.

5.3.2 Linear Ra-Ro EVM and the Ideal Point Method

Since there are two kinds of Ra-Ro variables, one is discrete Ra-Ro variable, the other is continuous Ra-Ro variable, we will propose two kinds of crisp equivalent models with respectively discrete or continuous Ra-Ro coefficients. Consider the following model,

$$\begin{cases} \max\left[\tilde{\bar{c}}_1^T\boldsymbol{x}, \tilde{\bar{c}}_2^T\boldsymbol{x}, \ldots, \tilde{\bar{c}}_m^T\boldsymbol{x}\right] \\ \text{s.t.} \begin{cases} \tilde{\bar{e}}_r^T\boldsymbol{x} \le \tilde{\bar{b}}_r, r = 1, 2, \ldots, p \\ \boldsymbol{x} \ge 0 \end{cases} \end{cases} \tag{5.30}$$

where $\tilde{\bar{c}}_i = (\tilde{\bar{c}}_{i1}, \tilde{\bar{c}}_{i2}, \ldots, \tilde{\bar{c}}_{in})^T$, $\tilde{\bar{e}}_r = (\tilde{\bar{e}}_{r1}, \tilde{\bar{e}}_{r2}, \ldots, \tilde{\bar{e}}_{rn})^T$ are random rough vectors and $\tilde{\bar{b}}_r$ are Ra-Ro variables, $i = 1, 2, \ldots, m, r = 1, 2, \ldots, p$. By expected value operator we can get the programming problem as follows,

$$\begin{cases} \max\left[E[\tilde{\bar{c}}_1^T\boldsymbol{x}], E[\tilde{\bar{c}}_2^T\boldsymbol{x}], \ldots, E[\tilde{\bar{c}}_m^T\boldsymbol{x}]\right] \\ \text{s.t.} \begin{cases} E[\tilde{\bar{e}}_r^T\boldsymbol{x}] \le E[\tilde{\bar{b}}_r], r = 1, 2, \ldots, p \\ \boldsymbol{x} \ge 0 \end{cases} \end{cases} \tag{5.31}$$

5.3.2.1 Crisp Equivalent Model

For the expected value programming with discrete and some continuous Ra-Ro variables, we can obtain the crisp equivalent models by Theorems 5.1, 5.2, 5.3, 5.4, 5.5, 5.6 and 5.7. For the important case, we will give it by the following theorem.

Theorem 5.10. *(Xu and Yao [347]) Assume that Ra-Ro vector $\tilde{\bar{c}}_i$ is characterized by $\tilde{\bar{c}}_i \sim \mathcal{N}(\bar{c}_i(\lambda), V_i^c)$, where $\bar{c}_i(\lambda) = (\bar{c}_{i1}(\lambda), \bar{c}_{i2}(\lambda), \ldots, \bar{c}_{in}(\lambda))^T)$ is a rough vector such $\bar{c}_{ij}(\lambda)$ is approximated by a rough set and V_i^c is a positive definite covariance matrix. The random rough vector $\tilde{\bar{e}}_r$ and Ra-Ro variable $\tilde{\bar{b}}_r$ are respectively characterized by $\tilde{\bar{e}}_r \sim \mathcal{N}(\bar{e}_r(\lambda), V_r^e)$ and $\tilde{\bar{b}}_r \sim \mathcal{N}(\bar{b}_r(\lambda), V_r^b)$, where $\bar{e}_r(\lambda) = (\bar{e}_{r1}(\lambda), \bar{e}_{r2}(\lambda), \ldots, \bar{e}_{rn}(\lambda))^T)$ and $\bar{e}_r(\lambda)$ are variables approximated by some rough sets. Let $\bar{c}_{ij}(\lambda) \vdash ([a_{ij}^c, b_{ij}^c], [c_{ij}^c, d_{ij}^c])_R$, $\bar{e}_{rj}(\lambda) \vdash ([a_{rj}^e, b_{rj}^e], [c_{rj}^e, d_{rj}^e])_R$ and $\bar{b}_r(\lambda) \vdash ([a_r^b, b_r^b], [c_r^b, d_r^b])_R$ (where $0 < c_{ij}^c \le a_{ij}^c < b_{ij}^c \le d_{ij}^c$, $0 < c_{rj}^e \le a_{rj}^e < b_{rj}^e \le d_{rj}^e$ and $0 < c_{rj}^b \le a_{rj}^b < b_{rj}^b \le d_{rj}^b$). Then, we have the following equivalent model of problem (5.31),*

$$\begin{cases} \max[\Psi_1^c\boldsymbol{x}, \Psi_2^c\boldsymbol{x}, \ldots, \Psi_m^c\boldsymbol{x}] \\ s.t. \begin{cases} \Psi_r^e\boldsymbol{x} \le \Psi_r^b, r = 1, 2, \ldots, p \\ \boldsymbol{x} \ge 0 \end{cases} \end{cases} \tag{5.32}$$

where $\Psi_i^c = \frac{1-\eta}{2}(a_{ij}^c + b_{ij}^c) + \frac{\eta}{2}(c_{ij}^c + d_{ij}^c)$, $\Psi_i^e = \frac{1-\eta}{2}(a_i^e + b_i^e) + \frac{\eta}{2}(c_i^e + d_i^e)$, $\Psi_i^b = \frac{1-\eta}{2}(a_i^b + b_i^b) + \frac{\eta}{2}(c_i^b + d_i^b)$, $\chi_i^{\Xi}(\chi = a, b, c, d, \Xi = e, b)$ *respectively denote the vectors.*

Proof. Firstly, let's analyze the objective function and obtain the equivalent function. Since $\tilde{\tilde{c}}_i \sim \mathcal{N}(\bar{c}_i(\lambda), V_i^c)$, it follows from Theorem 5.5 that

$$E[\tilde{\tilde{c}}_{ij}] = \frac{1-\eta}{2}(a_{ij}^c + b_{ij}^c) + \frac{\eta}{2}(c_{ij}^c + d_{ij}^c)$$

Then

$$E[\tilde{\tilde{c}}_i^T \boldsymbol{x}] = E\left[\sum_{j=1}^{n} \tilde{\tilde{c}}_{ij} x_j\right] = \sum_{j=1}^{n} E[\tilde{\tilde{c}}_{ij}] x_j = \left[\frac{1-\eta}{2}(a_{ij}^c + b_{ij}^c) + \frac{\eta}{2}(c_{ij}^c + d_{ij}^c)\right]^T \boldsymbol{x}$$

where $a_i^c, b_i^c, c_i^c, d_i^c, i = 1, 2, \ldots, m.$ respectively express their vectors. Denote $\Psi_i^c = \frac{1-\eta}{2}(a_{ij}^c + b_{ij}^c) + \frac{\eta}{2}(c_{ij}^c + d_{ij}^c)$. Then the objective function can be converted into

$$\max\{\Psi_1^c x, \Psi_2^c, \ldots, \Psi_m^c\}$$

Similarly to deal with the constraints, we have

$$\Psi_r^e x \leq \Psi_r^b$$

where $\Psi_i^e = \frac{1-\eta}{2}(a_i^e + b_i^e) + \frac{\eta}{2}(c_i^e + d_i^e), \Psi_i^b = \frac{1-\eta}{2}(a_i^b + b_i^b) + \frac{\eta}{2}(c_i^b + d_i^b)$. Then problem (5.31) has been transformed into

$$\begin{cases} \max\{\Psi_1^c \boldsymbol{x}, \Psi_2^c \boldsymbol{x}, \ldots, \Psi_m^c \boldsymbol{x}\} \\ s.t. \begin{cases} \Psi_r^e \boldsymbol{x} \leq \Psi_r^b, r = 1, 2, \ldots, p \\ \boldsymbol{x} \geq 0 \end{cases} \end{cases}$$

This completes the proof. □

5.3.2.2 The Ideal Point Method

In this section, we make use of the ideal point method proposed in [115, 311, 340] to resolve the multiobjective problem (5.32) with crisp parameters. If the decision maker can firstly propose an estimated value $\bar{F}_i$ for each objective function $\Psi_i^c \boldsymbol{x}$ such that

$$\bar{F}_i \geq \max_{x \in X'} \Psi_1^c \boldsymbol{x}, i = 1, 2, \ldots, m \tag{5.33}$$

where $X' = \{\boldsymbol{x} \in X | \Psi_r^e \boldsymbol{x} \leq \Psi_r^b, r = 1, 2, \ldots, p, \boldsymbol{x} \geq 0\}$, then $\bar{\mathbf{F}}_i = (\bar{F}_1, \bar{F}_2, \ldots, \bar{F}_m)^T$ is called the ideal point, especially, if $\bar{F}_i \geq \max_{x \in X'} \Psi_1^c \boldsymbol{x}$ for all i, we call $\bar{\mathbf{F}}$ the most ideal point.

The basic theory of the ideal point method is to take a especial norm in the objective space $\mathbf{R}^m$ and obtain the feasible solution $\boldsymbol{x}$ that the objective value approaches the ideal point $\bar{\mathbf{F}} = (\bar{F}_1, \bar{F}_2, \ldots, \bar{F}_m)^T$ under the norm distance, that is, to seek the feasible solution $\boldsymbol{x}$ satisfying

$$\min_{x \in X'} u(\Psi^c(\boldsymbol{x})) = \min_{x \in X'} ||\Psi^c(\boldsymbol{x}) - \bar{F}||.$$

Usually, the following norm functions are used to describe the distance:

1. p-mode function

$$d_p(\Psi^c(\boldsymbol{x}), \bar{F}; \omega) = \left[\sum_{i=1}^{m} \omega_i |\Psi_i^c \boldsymbol{x} - \bar{F}_i|^p\right]^{\frac{1}{p}}, \ 1 \leq p < +\infty \tag{5.34}$$

2. The maximal deviation function

$$d_{+\infty}(\Psi^c(\boldsymbol{x}), \bar{F}; \omega) = \max_{1 \leq i \leq m} \omega_i |\Psi_i^c \boldsymbol{x} - \bar{F}_i| \tag{5.35}$$

3. Geometric mean function

$$d(\Psi^c(\boldsymbol{x}), \bar{F}) = \left[\prod_{i=1}^{m} |\Psi_i^c \boldsymbol{x} - \bar{F}_i|^p\right]^{\frac{1}{m}}. \tag{5.36}$$

The weight parameter vector $\omega = (\omega_1, \omega_2, \ldots, \omega_m)^T > 0$ needs to be predetermined.

Theorem 5.11. *Assume that* $\bar{F}_i > \max_{x \in X'} \Psi_1^c \boldsymbol{x} (i = 1, 2, \ldots, m)$. *If* $\boldsymbol{x}^*$ *is the optimal solution of the following problem*

$$\min_{x \in X'} d_p(\Psi^c(\boldsymbol{x}), \bar{F}; \omega) = \left[\sum_{i=1}^{m} \omega_i |\Psi_i^c \boldsymbol{x} - \bar{F}_i|^p\right]^{\frac{1}{p}} \tag{5.37}$$

then $\boldsymbol{x}^*$ *is an efficient solution of problem (5.32). On the contrary, if* $\boldsymbol{x}^*$ *is an efficient solution of problem (5.32), then there exists a weight vector* ω *such that* $\boldsymbol{x}^*$ *is the optimal solution of problem (5.37).*

Proof. This result can be easily obtained, and we hereby don't prove it. □

Next, we take the p-mode function to describe the procedure of solving the problem (5.32).

Step 1. Find the ideal point. If the decision maker can give the ideal objective value satisfying the condition (5.33), the value will be considered as the ideal point. However, decision makers themselves don't know how to give the objective value, then we can get the ideal point by solving the following programming problem,

$$\begin{cases} \max \Psi_i^c \boldsymbol{x} \\ \text{s.t.} \begin{cases} \Psi_r^e x \leq \Psi_r^b, r = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{5.38}$$

Then the ideal point $\bar{\mathbf{F}} = (\bar{F}_1, \bar{F}_2, \ldots, \bar{F}_m)^T$ can be fixed by $\bar{F}_i = \Psi_i^c \boldsymbol{x}^*$, where $\boldsymbol{x}^*$ is the optimal solution of problem (5.38).

Step 2. Fix the weight. The method of selecting the weight can be referred to many literatures, interested readers can consult them. We usually use the following function to fix the weight,

$$\omega_i = \frac{\bar{F}_i}{\sum\limits_{i=1}^{m} \bar{F}_i}.$$

Step 3. Construct the minimal distance problem. Solve the following single objective programming problem to obtain the efficient solution of problem (5.32),

$$\begin{cases} \min \left[\sum\limits_{i=1}^{m} \omega_i |\Psi_i^c \boldsymbol{x} - \bar{F}_i|^t \right]^{\frac{1}{t}} \\ \text{s.t.} \begin{cases} \Psi_r^e \boldsymbol{x} \leq \Psi_r^b, r = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{5.39}$$

Usually, we take $t = 2$ to compute it.

5.3.3 *Non-Linear Ra-Ro EVM and Ra-Ro Simulation-Based ECTS*

Let's consider the following Ra-Ro EVM,

$$\begin{cases} \max [E[f_1(\boldsymbol{x}, \boldsymbol{\xi})], E[f_2(\boldsymbol{x}, \boldsymbol{\xi})], \ldots, E[f_m(\boldsymbol{x}, \boldsymbol{\xi})]] \\ \text{s.t. } E[g_j(\boldsymbol{x}, \boldsymbol{\xi})] \leq 0, j = 1, 2, \ldots, p \end{cases} \tag{5.40}$$

where $f_i(\boldsymbol{x}, \boldsymbol{\xi})$, or $g_r(\boldsymbol{x}, \boldsymbol{\xi})$ or both of them are nonlinear functions with respect to the Ra-Ro vector $\boldsymbol{\xi}$. For the problem (5.40), we cannot usually convert it into the crisp one because of the existence of the nonlinear function. Thus, an efficient intelligent algorithm needs to be applied to find its approximate solution. In this section, the Ra-Ro simulation-base TS will be proposed to solve the problem (5.40).

5.3.3.1 Ra-Ro Simulation for EVM

Let's consider the objective function $f(\boldsymbol{x}, \boldsymbol{\xi})$ and introduce how to compute its expected value for given $\boldsymbol{x}$ by Ra-Ro simulation. We uniformly sample $\underline{\lambda}_1, \underline{\lambda}_2, \ldots, \underline{\lambda}_N$ from the lower approximation $\underline{X}$ and $\bar{\lambda}_1, \bar{\lambda}_2, \ldots, \bar{\lambda}_N$ from the upper approximation $\overline{X}$. For each $\underline{\lambda}_i$ and $\bar{\lambda}_i (i = 1, 2, \ldots, N)$, $\boldsymbol{\xi}(\underline{\lambda}_i)$ and $\boldsymbol{\xi}(\bar{\lambda}_i)$ are both random vectors. Then we can apply random simulation to get their expected values. Randomly generate $\omega_i^1, \omega_i^2, \ldots, \omega_i^M$ from Ω according to the probability measure Pr for each $\underline{\lambda}_i (i = 1, 2, \ldots, N)$. Then

$$E[f(\boldsymbol{x}, \boldsymbol{\xi}(\underline{\lambda}_i))] = \frac{\sum_{j=1}^{M} f(\boldsymbol{x}, \boldsymbol{\xi}(\underline{\lambda}_i)(\omega_i^j))}{M}.$$

Similarly, randomly generate $\theta_i^1, \theta_i^2, \ldots, \theta_i^M$ from Ω according to the probability measure Pr for each $\bar{\lambda}_i (i = 1, 2, \ldots, N)$. Then

$$E[f(\boldsymbol{x}, \boldsymbol{\xi}(\bar{\lambda}_i))] = \frac{\sum_{j=1}^{M} f(\boldsymbol{x}, \boldsymbol{\xi}(\bar{\lambda}_i)(\theta_i^j))}{M}.$$

In the end, we can get the expected value of $f_j(\boldsymbol{x}, \xi)$ as follows

$$E[f_j(\boldsymbol{x}, \boldsymbol{\xi})] = \frac{\sum_{i=1}^{M} (\eta E[f_j(\boldsymbol{x}, \boldsymbol{\xi}(\bar{\lambda}_i))] + (1 - \eta) E[f_j(\boldsymbol{x}, \boldsymbol{\xi}(\underline{\lambda}_i))])}{M}.$$

Then the procedure simulating the expected value of the function $f(\boldsymbol{x}, \boldsymbol{\xi})$ can be summarized as follows:

Procedure `Ra-Ro simulation for EVM`

Input: The decision vector $\boldsymbol{x}$
Output: The expected value $E[f(\boldsymbol{x}, \boldsymbol{\xi})]$
Step 1. Set $r = s = t = 0$;
Step 2. Uniformly sample $\underline{\lambda}_1, \underline{\lambda}_2, \ldots, \underline{\lambda}_N$ from the lower approximation $\underline{X}$ and $\bar{\lambda}_1, \bar{\lambda}_2, \ldots, \bar{\lambda}_N$ from the upper approximation $\overline{X}$;
Step 3. Randomly generate ω_i and $\theta_i (i = 1)$ from Ω according to the probability measure Pr for each $\bar{\lambda}_i$ and $\underline{\lambda}_i$;
Step 4. Compute $f(\boldsymbol{x}, \boldsymbol{\xi}(\underline{\lambda}_i)(\omega_i^j))$ and $f(\boldsymbol{x}, \boldsymbol{\xi}(\bar{\lambda}_i)(\theta_i^j))$, $j = 1, 2, \ldots, M$;
Step 5. $r \leftarrow r + f(\boldsymbol{x}, \boldsymbol{\xi}(\underline{\lambda}_i)(\omega_i^j))$ and $s \leftarrow s + f(\boldsymbol{x}, \boldsymbol{\xi}(\bar{\lambda}_i)(\theta_i^j))$;
Step 6. Repeat the third to fifth steps for N times;
Step 7. $t \leftarrow t + \eta r + (1 - \eta)s$ and $i + +$;
Step 8. If $i > M$, return $\frac{t}{MN}$.

Example 5.8. Let $\tilde{\tilde{\xi}}_1$, $\tilde{\tilde{\xi}}_2$ and $\tilde{\tilde{\xi}}_3$ are three Ra-Ro variables as follows,

$$\begin{aligned}
&\tilde{\tilde{\xi}}_1 \sim \mathscr{U}(\tilde{\rho}_1, \tilde{\rho}_1 + 2), \text{ with } \tilde{\rho}_1 = ([1,2],[1,3]),\\
&\tilde{\tilde{\xi}}_2 \sim \mathscr{N}(\tilde{\rho}_2, 1), \qquad \text{with } \tilde{\rho}_2 = ([0,1],[0,3]),\\
&\tilde{\tilde{\xi}}_3 \sim exp(\tilde{\rho}_3), \qquad \text{with } \tilde{\rho}_3 = ([1,2],[0,3]),
\end{aligned}$$

A run of Ra-Ro simulation with 1000 cycles shows that

$$E\left[\sqrt{\tilde{\tilde{\xi}}_1^2 + \tilde{\tilde{\xi}}_2^2 + \tilde{\tilde{\xi}}_3^2}\right] = 3.2279.$$

5.3.3.2 Enhanced Continuous Tabu Search (ECTS)

Local search employs the idea that a given solution $\boldsymbol{x}$ may be improved by making small changes. Those solutions obtained by modifying solution $\boldsymbol{x}$ are called neighbors of $\boldsymbol{x}$. The local search algorithm starts with some initial solution and moves from neighbor to neighbor as long as possible while decreasing the objective function value. The main problem with this strategy is to escape from local minima where the search cannot find any further neighborhood solution that decreases the objective function value. Different strategies have been proposed to solve this problem. One of the most efficient strategies is tabu search. Tabu search allows the search to explore solutions that do not decrease the objective function value only in those cases where these solutions are not forbidden. This is usually obtained by keeping track of the last solutions in term of the action used to transform one solution to the next. When an action is performed it is considered tabu for the next T iterations, where T is the tabu status length. A solution is forbidden if it is obtained by applying a tabu action to the current solution. The Tabu Search metaheuristic has been defined by Glover[114]. The basic ideas of TS have also been sketched by Hansen [131]. After that, TS has achieved widespread success in solving practical optimization problems in different domains(such as resource management, process design, logistic and telecommunications).

A *tabu list* is a set of solutions determined by historical information from the last t iterations of the algorithm, where t is fixed or is a variable that depends on the state of the search, or a particular problem. At each iteration, given the current solution $\boldsymbol{x}$ and its corresponding neighborhood $N(\boldsymbol{x})$, the procedure moves to the solution in the neighborhood $N(\boldsymbol{x})$ that most improves the objective function. However, moves that lead to solutions on the tabu list are forbidden, or are tabu. If there are no improving moves, TS chooses the move which least changes the objective function value. The tabu list avoids returning to the local optimum from which the procedure has recently escaped. A basic element of tabu search is the *aspiration criterion*, which determines when a move is admissible despite being on the tabu list. One *termination* criterion for the tabu procedure is a limit in the number of consecutive moves for which no improvement occurs. Given an objective function

$f(\boldsymbol{x})$ over a feasible domain D, a generic tabu search for finding an approximation of the global minimum of $f(\boldsymbol{x})$ is given as follows:

Procedure `Layout of Tabu Search`

Input: A problem instance
Output: A (sub-optimal) solution
Step 1. Initialization: (a) Generate an initial solution $(\boldsymbol{x})$ and set $\boldsymbol{x}^* = \boldsymbol{x}$, (b) Initialize the tabu list $T = \Phi$, (c) Set iteration counters $k = 0$ and $l = 0$;
Step 2. While $(N(\boldsymbol{x})\ T \neq \Phi)$, do (a) $k = k + 1, l = l + 1$, (b) Select $\boldsymbol{x}$ as the best solution from the set $N(\boldsymbol{x})\ T$, (c) If $f(\boldsymbol{x}) < f(\boldsymbol{x}^*)$, then update $\boldsymbol{x}^* = \boldsymbol{x}$ and set $l = 0$, (d) If $k = \bar{k}$ or if $l = \bar{l}$ go to step 3;
Step 3. Output the best solution found $\boldsymbol{x}^*$.

In the following part, we will introduce the detail steps about how to apply the an special TS algorithm–Enhanced Continuous Tabu Search(ECTS) proposed by Chelouah and Siarry [47] based on the Ra-Ro simulation to solve a multi-objective expected value model with Ra-Ro parameters.

Setting of parameters. Two of parameters must be set before any execution of ECTS:

1. Initialization
2. Control parameters

For each of these categories, some parameter values must be chosen by the user and some parameter values must be calculated. These four subsets of parameters are listed in Table 5.1.

Initialization. In this stage, we will list the representation of the solution. We have resumed and adapted the method described in detail in [289]. Randomly generate a solution $\boldsymbol{x}$ and check its feasibility by the Ra-Ro simulation such that $E[g_r(\boldsymbol{x}, \boldsymbol{\xi})] \leq 0 (r = 1, 2, \ldots, p)$. Then generate its neighborhood by the concept of 'ball' defined in [289]. A ball $B(\boldsymbol{x}, r)$ is centered on $\boldsymbol{x}$ with radius r, which contains all points $\boldsymbol{x}'$ such that $||\boldsymbol{x}' - \boldsymbol{x}|| \leq 4$ (the symbol $||\cdot||$ denotes the Euclidean norm). To obtain a homogeneous exploration of the space, we consider a set of balls centered on the current solution $\boldsymbol{x}$, with $h_0, h_1, \ldots, h_\eta$. Hence the space is partitioned into concentric 'crowns' $C_i(\boldsymbol{x}, h_{i-1}, h_i)$, such that

$$C_i(\boldsymbol{x}, h_{i-1}, h_i) = \{\boldsymbol{x}' | h_{i-1} \leq ||\boldsymbol{x}' - \boldsymbol{x}|| \leq h_i\}.$$

The η neighbors of s are obtained by random selection of one point inside each crown C_i, for i varying from 1 to η. Finally, we select the best neighbor of $\boldsymbol{x}$ among these η neighbors, even if it is worse than $\boldsymbol{x}$. In ECTS, we replace the balls by hyper-rectangles for the partition of the current solution neighborhood (see Fig. 5.10), and we generate neighbors in the same way. The reason for using a hyperrectangular neighborhood instead of crown 'balls' is the following: it is mathematically much easier to select a point inside a specified hyperrectangular zone than to select a point inside a specified crown ball. Therefore in the first case, we only have to

Table 5.1 Listing of the ECTS parameters

A. *Initialization parameters chosen by the user*
Search domain of each function variable
Starting point
Content of the tabu list
Content of the promising list
B. *Initialization parameters calculated*
Length δ of the smallest edge of the initial hyperrectangular search domain
Initial threshold for the acceptance of a promising area
Initial best point
Number η of neighbors of the current solution investigated at each iteration
Maximum number of successive iterations without any detection of a promising area
Maximum number of successive iterations without any improvement of the objective function value
Maximum number of successive reductions of the hyperrectangular neighborhood and of the radius of tabu balls with out any improvement
Maximum number of iterations
C. *Control parameters chosen by the user*
Length N_t of the tabu list
Length N_p of the promising list
Parameter ρ_t allowing to calculate the initial radius of tabu balls
Parameter ρ_{neigh} allowing to calculate the initial size of the hyperrectangular neighborhood
D. *Control parameters calculated*
Initial radius ε_t of tabu balls
Initial radius ε_p of promising balls
Initial size of the hyperrectangular neighborhood

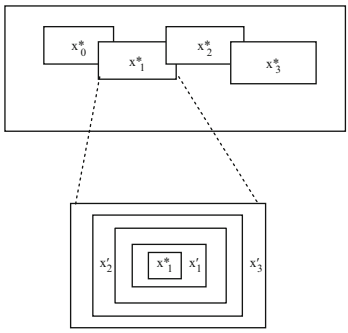


Fig. 5.10 Partition of current solution neighborhood

compare the coordinates of the randomly selected point with the bounds that define the hyperrectangular zone at hand.

Next, we will describe the initialization of some parameters and the tuning of the control parameters. In other words, we give the 'definition' of all the parameters of ECTS. The parameters in part A of Table 5.1 are automatically built by using the

parameters fixed at the beginning. The parameters in part B of Table 5.1 are valued in the following way:

1. The search domain of analytical test functions is set as prescribed in the literature, the initial solution $\boldsymbol{x}^*$ is randomly chosen and checked if it is feasible by the Ra-Ro simulation.
2. The tabu list is initially empty.
3. To complete the promising list, the algorithm randomly draw a point. This point is accepted as the center of an initial promising ball, if it does not belong to an already generated ball. In this way the algorithm generates N_p sample points which are uniformly dispersed in the whole space solution S.
4. The initial threshold for the acceptance of a promising area is taken equal to the average of the objective function values over the previous N_p sample points.
5. The best point found is taken equal to the best point among the previous N_p.
6. The number η of neighbors of the current solution investigated at each iteration is set to twice the number of variables, if this number is equal or smaller than five, otherwise η is set to 10.
7. The maximum number of successive iterations without any detection of a new promising area is equal to twice the number of variables.
8. The maximum number of successive iterations without any improvement of the objective function value is equal to five times the number of variables.
9. The maximum number of successive reductions of the hyperrectangular neighborhood and of the radius of tabu balls without any improvement of the objective function value is set to twice the number of variables.
10. The maximum number of iterations is equal to 50 times the number of variables.

There exist two types of control parameters. Some parameters are chosen by the user. Other ones are deduced from the chosen parameters. The fixed parameters are the length of the tabu list (set to 7, which is the usual tuning advocated by Glover), the length of the promising list (set to 10, like in [68]) and the parameters ρ_t, ρ_p and ρ_{neigh} (set to 100, 50, and 5, respectively). The expressions of ε_t and ε_p are δ/ρ_t and δ/ρ_p respectively, and the initial size of the hyperrectangular neighborhood of the current solution (the more external hyperrectangle) is obtained by dividing δ by the factor ρ_{neigh}.

Diversification. At this stage, the process starts with the initial solution, used as the current one. ECTS generates a specified number of neighbors: one point is selected inside each hyperrectangular zone around the current solution. Each neighbor is accepted only if it does not belong to the tabu list. The best of these neighbors becomes the new current solution, even if it is worse than the previous one. A new promising solution is detected and generated according to the procedure described above. This promising solution defines a new promising area if it does not already belong to a promising ball. If a new promising area is accepted, the worst area of the promising list is replaced by the newly accepted promising area. The use of the promising and tabu lists stimulates the search for solutions far from the starting one and the identified promising areas. The diversification process stops after a given number of successive iterations without any detection of a new promising

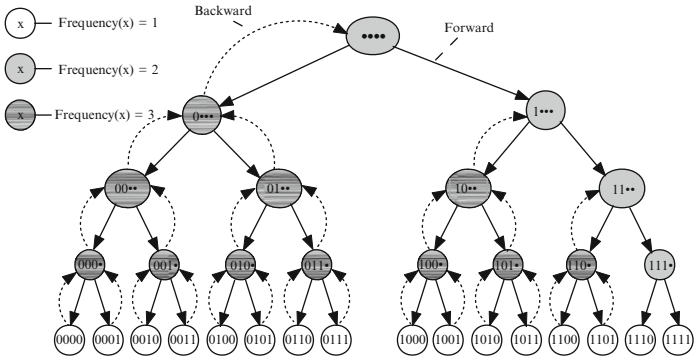


Fig. 5.11 A standard "backtracking" (depth first) branch-and-bound approach

area. Then the algorithm determines the most promising area among those present in the promising list (Fig. 5.11).

Search for the most promising area. In order to determine the most promising area, we proceed in three steps. First, we calculate the average value of the objective function over all the solutions present in the promising list. Secondly, we eliminate all the solutions for which the function value is higher than this average value. Thirdly, we deal with the thus reduced list in the following way. We halve the radius of the tabu balls and the size of the hyperrectangular neighborhood. For each remaining promising solution, we perform the generation of the neighbors and selection of the best. We replace the promising solution by the best neighbor located, yet only if this neighbor is better than that solution. After having scanned the whole promising list, the algorithm removes the least promising solution. This process is reiterated after halving again the above two parameters. It stops when just one promising area remains.

Intensification. The first step of the intensification stages is the resetting of the tabu list. The remaining promising area allows the definition of a new search domain. The center of this area is taken as the current point, and the tabu search starts again: generation of neighbors not belong to the tabu list, selection of the best, and insertion of the best solution into the tabu list. This selected neighbor becomes the new current solution, even if it is worse than the previous one. After a predetermined number of successive iterations without any improvement of the objective function value (e.g. quadratic error between two successive solutions less than 10^{-3}), the size of the hyperrectangular neighborhood and the radius of the tabu balls are halved, tabu list is reset, and we restart the procedure from the best point found until now. To stop the algorithm, we use two criteria: a specified number of successive reductions of the two parameters above without any significant improvement of the objective function value and a specified maximum number of iterations.

5.3.4 Numerical Examples

Example 5.9. Let us consider a multi-objective programming with Ra-Ro coefficients which are subject to binomial distribution.

$$\begin{cases} \max F_1(\boldsymbol{x}, \boldsymbol{\xi}) = 2\xi_1 x_1^2 + 3\xi_2 x_2 - \xi_3 x_3 \\ \max F_2(\boldsymbol{x}, \boldsymbol{\xi}) = 5\xi_4 x_2 - 2\xi_5 x_1 + 2\xi_6 x_3 \\ \text{s.t.} \begin{cases} 5x_1 - 3x_2^2 + 6\sqrt{x_3} \leq 50 \\ 4\sqrt{x_1} + 6x_2 - 4.5x_3 \leq 20 \\ x_1 + x_2 + x_3 \leq 15 \\ x_1, x_2, x_3 \geq 0 \end{cases} \end{cases} \tag{5.41}$$

where $\xi_i (i = 1, 2, \ldots, 6)$ are discrete Ra-Ro variables subject to binomial distribution and their probabilities are respectively as follows,

$$\begin{array}{ll} \tilde{p}_1 \vdash ([0.2, 0.5], [0, 1]), & \tilde{p}_2 \vdash ([0.6, 0.8], [0, 1]), \\ \tilde{p}_3 \vdash ([0.45, 0.95], [0, 1]), & \tilde{p}_4 \vdash ([0.4, 0.5], [0, 1]), \\ \tilde{p}_5 \vdash ([0.36, 0.64], [0, 1]), & \tilde{p}_6 \vdash ([0.55, 0.65], [0, 1]). \end{array}$$

$\xi_j (j = 1, 2, \ldots, 12)$ are continuous Ra-Ro variables subject to the following distribution,

$$\begin{array}{ll} \xi_7 \sim exp(\lambda_7), & \lambda_7 \vdash ([1, 2], [0, 4]) \\ \xi_{10} \sim exp(\lambda_{10}), & \lambda_{10} \vdash ([3, 4], [2, 8]) \\ \xi_8 \sim \mathcal{N}(\mu_8, 1), & \lambda_{10} \vdash ([2, 3], [1, 5]) \\ \xi_{11} \sim \mathcal{N}(\mu_{11}, 1), & \lambda_{11} \vdash ([1, 3], [0, 4]) \\ \xi_9 \sim \mathcal{U}(\tilde{a}_9, \tilde{b}_9), & \xi_{12} \sim \mathcal{U}(\tilde{a}_{12}, \tilde{b}_{12}) \\ \tilde{a}_9 \vdash ([1, 2], [0, 2]), & \tilde{b}_9 \vdash ([6, 8], [2, 10]) \\ \tilde{a}_{12} \vdash ([1, 2], [1, 3]), & \tilde{b}_{12} \vdash ([2, 4], [2, 8]) \end{array}$$

From the mathematical view, the problem (5.41) is not well defined because of the uncertain parameters. Then we apply the expected value technique to deal with this uncertain programming. Assume that the total number of trials is 20. By Theorem 5.2, we have and

$$\begin{array}{ll} E[\xi_1] = \frac{1}{4}(0.2 + 0.5 + 1) * 20 = 8.5, & E[\xi_2] = \frac{1}{4}(0.6 + 0.8 + 1) * 20 = 12, \\ E[\xi_3] = \frac{1}{4}(0.45 + 0.95 + 1) * 20 = 12, & E[\xi_4] = \frac{1}{4}(0.4 + 0.5 + 1) * 20 = 9.5, \\ E[\xi_5] = \frac{1}{4}(0.36 + 0.64 + 1) * 20 = 10, & E[\xi_6] = \frac{1}{4}(0.55 + 0.65 + 1) * 20 = 11, \end{array}$$

Then the problem (5.41) can be converted into

$$\begin{cases} \max f_1(\boldsymbol{x}) = 17x_1^2 + 36x_2 - 12x_3 \\ \max f_2(\boldsymbol{x}) = 47.5x_2 - 20x_1 + 22x_3 \\ \text{s.t.} \begin{cases} 5x_1 - 3x_2^2 + 6\sqrt{x_3} \le 50 \\ 4\sqrt{x_1} + 6x_2 - 4.5x_3 \le 20 \\ x_1 + x_2 + x_3 \le 15 \\ x_1, x_2, x_3 \ge 0 \end{cases} \end{cases} \tag{5.42}$$

Then we apply the ideal method to get the optimal solution of problem (5.42). Firstly, we get the ideal point $f^* = (f_1^*, f_2^*)^T = (2263.03, 542.50)^T$ by solving the following problem,

$$\begin{cases} \max f_i(\boldsymbol{x}) \\ \text{s.t.} \begin{cases} 5x_1 - 3x_2^2 + 6\sqrt{x_3} \le 50 \\ 4\sqrt{x_1} + 6x_2 - 4.5x_3 \le 20 \\ x_1 + x_2 + x_3 \le 15 \\ x_1, x_2, x_3 \ge 0 \end{cases} \end{cases} \tag{5.43}$$

Secondly, we fix the weight by the following method,

$$w_1 = \frac{f_1}{f_1 + f2} = 0.807, w_2 = \frac{f_2}{f_1 + f2} = 0.193.$$

Thirdly, construct the new objective function,

$$f(\boldsymbol{x}) = \sqrt{0.807|f_1(\boldsymbol{x}) - 2263.03|^2 + 0.197|f_2(\boldsymbol{x}) - 542.50|^2},$$

then we get the following single objective programming problem,

$$\begin{cases} \min f(\boldsymbol{x}) = \sqrt{0.807|f_1(\boldsymbol{x}) - 2263.03|^2 + 0.197|f_2(\boldsymbol{x}) - 542.50|^2} \\ \text{s.t.} \begin{cases} 5x_1 - 3x_2^2 + 6\sqrt{x_3} \le 50 \\ 4\sqrt{x_1} + 6x_2 - 4.5x_3 \le 20 \\ x_1 + x_2 + x_3 \le 15 \\ x_1, x_2, x_3 \ge 0 \end{cases} \end{cases} \tag{5.44}$$

Finally, we get the optimal solution $\boldsymbol{x}^* = (11.32, 2.20, 1.48)^T$.

Example 5.10. Let us consider a multi-objective programming with Ra-Ro coefficients,

$$\begin{cases} \max F_1(\boldsymbol{x}, \boldsymbol{\xi}) = x_1^2 + x_2^2 \\ \max F_2(\boldsymbol{x}, \boldsymbol{\xi}) = 3x_1 - 2x_2 + x_1x_2 \\ \text{s.t.} \begin{cases} \sqrt{(x_1 - \xi)^2 + (x_2 - \xi)^2} \le 7 \\ x_1 + x_2 \le 5 \\ x_1, x_2 \ge 0 \end{cases} \end{cases} \tag{5.45}$$

Table 5.2 The result by Ra-Ro simulation-based by ECTS

w_1	w_2	x_1	x_2	$F_1(\boldsymbol{x})$	$F_2(\boldsymbol{x})$	$f(\boldsymbol{x})$	Gen
0.3	0.7	4.9537	0.0349	24.5395	14.8626	17.7657	1000
0.4	0.6	4.9688	0.0239	24.5621	14.9810	19.7716	1000
0.5	0.5	4.9511	0.0331	24.5149	14.9841	19.7495	1000

where ξ is a normally distributed Ra-Ro variable, written as $\xi \sim \mathcal{N}(\tilde{\rho}, 1)$, where $\tilde{\rho} \vdash ([1,2],[0,3])$ is a rough variable.

Next, we apply the tabu search algorithm based on the Ra-Ro simulation to solve the above problem.

Step 1. Set the move step $h = 0.5$ and the h neighbor $N(\boldsymbol{x}, h)$ for the present point $\boldsymbol{x}$ is defined as follows,

$$N(\boldsymbol{x}, h) = \left\{ \boldsymbol{y} | \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \leq h \right\}.$$

The random move of point $\boldsymbol{x}$ to point $\boldsymbol{y}$ in its h neighbor along direction s is given by

$$\boldsymbol{y}_s = \boldsymbol{x}_s + rh,$$

where r is a random number that belongs to [0,1], $s = 1, 2, 3$.

Step 2. Denote $X = \{(x_1, x_2, x_3) | \sqrt{(x_1 - \xi)^2 + (x_2 - \xi)^2} \leq 7; x_1 + x_2 \leq 5; x_i \geq 0, i = 1, 2\}$. Give the step set $H = \{h_1, h_2, \ldots, h_r\}$ and randomly generate a feasible point $\boldsymbol{x}_0 \in X$ by Ra-Ro simulation. One should empty the Tabu list T (the list of inactive steps) at the beginning.

Step 3. For each active neighbor $N(\boldsymbol{x}, h)$ of the present point $\boldsymbol{x}$, where $h \in H - T$, a feasible random move that satisfies all the constraints in problem (5.45) is to be generated.

Step 4. Construct the single objective function as follows,

$$f(\boldsymbol{x}, \boldsymbol{\xi}) = w_1 F_1(\boldsymbol{x}, \boldsymbol{\xi}) + w_2 F_2(\boldsymbol{x}, \boldsymbol{\xi}),$$

where $w_1 + w_2 = 1$. Compare the $f(\boldsymbol{x}, \boldsymbol{\xi})$ of the feasible moves with that of the current solution by the Ra-Ro simulation. If an augmenter in new objective function of the feasible moves exists, one should save this feasible move as the updated current one by adding the corresponding step to the Tabu list T and go to the next step; otherwise, go to the next step directly.

Step 5. Stop if the termination criteria are satisfied; other wise, empty T if it is full; then go to Step 3. Here, we set the computation is determined if the better solution doesn't change again.

Some results can be found in Table 5.2 and Figs. 5.12 and 5.13.

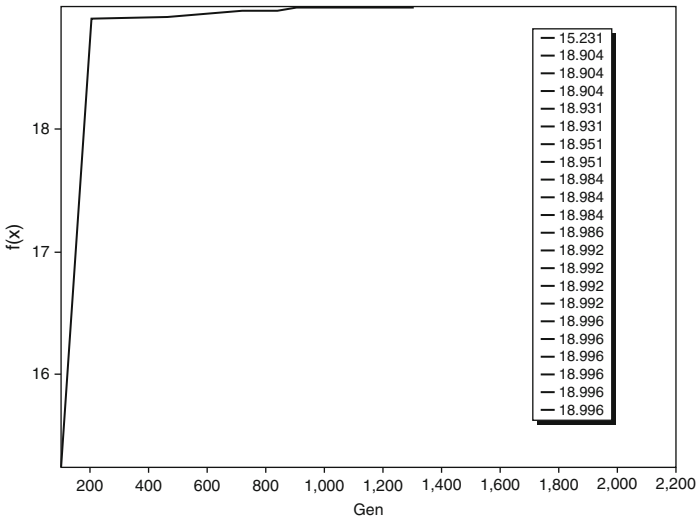


Fig. 5.12 The result computed by ECTS when $w_1 = 0.6$ and $w_2 = 0.4$

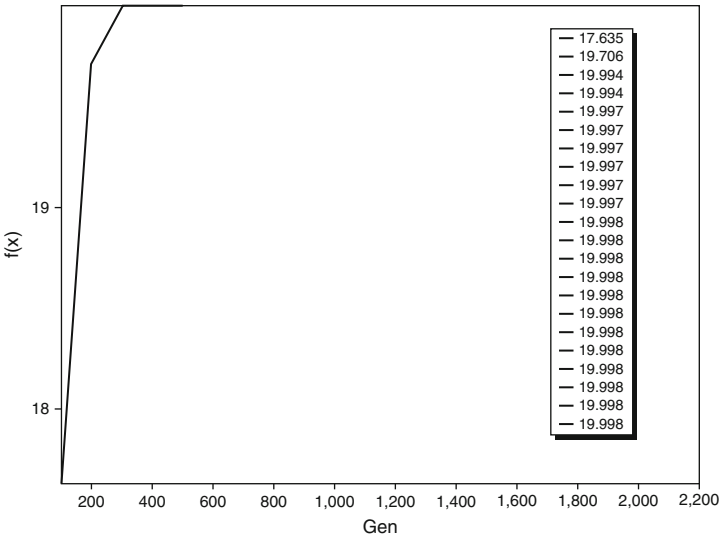


Fig. 5.13 The result computed by ECTS when $w_1 = 0.5$ and $w_2 = 0.5$

5.4 Ra-Ro CCM

Similarly, the chance measure of Ra-Ro variable is also used to be an efficient tool to deal with the multiobjective decision making problems with Ra-Ro parameters. In this section, we will introduce its basic definition and property and present the general Ra-Ro CCM. Then a class of linear Ra-Ro CCMs will be deal with by some

mathematical technique. For those which cannot be directly transformed into crisp one, we apply the Ra-Ro simulation-based tabu search algorithm to deal with it.

5.4.1 General Model

The following part will introduce some basic definitions and properties of the chance measure of the Ra-Ro variable.

Definition 5.21. Let $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_n)$ be a Ra-Ro vector with $\xi_i \vdash (\underline{X}_i, \overline{X}_i)(i = 1, 2, \ldots, n)$, and $f_j : \mathscr{A}^n \to \mathscr{A}$ be continuous functions, $j = 1, 2, \ldots, m$. Then the primitive chance of Ra-Ro event characterized by $f_j(\boldsymbol{\xi}) \leq 0, j = 1, 2, \ldots, m$ is a function from [0,1] to [0,1], defined as

$$\begin{aligned} &Ch\{f_j(\boldsymbol{\xi}) \leq 0, j = 1, 2, \ldots, m\}(\alpha) \\ &\quad = sup\left\{\beta | \text{Appr}\left\{\lambda | Pr\left\{\begin{array}{l} f_j(\boldsymbol{\xi}(\lambda)) \leq 0, \\ j = 1, 2, \ldots, m \end{array}\right\} \geq \beta\right\} \geq \alpha\right\} \end{aligned} \tag{5.46}$$

If we want to maximize the optimistic value to the return function subject to some chance constraints, then we have the following Ra-Ro CCM,

$$\left\{\begin{array}{l} \max[\bar{f}_1, \bar{f}_2, \ldots, \bar{f}_m] \\ \text{s.t.} \left\{\begin{array}{l} Ch\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}_i\}(\gamma_i) \geq \delta_i \quad i = 1, 2, \ldots, m \\ Ch\{g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\}(\alpha_j) \geq \beta_j, \quad j = 1, 2, \ldots, p \\ \boldsymbol{x} \geq 0 \end{array}\right. \end{array}\right. \tag{5.47}$$

where α_j and β_j are specified confidence levels for $j = 1, 2, \ldots, p$, and $\max \bar{f}_i$ is the (γ_i, δ_i)-optimistic value to the return function $f_i(\boldsymbol{x}, \boldsymbol{\xi})$ with respect to primitive chance Ch.

If the priority structure and target levels are set by the decision maker, then we may formulate a Ra-Ro decision system as a chance-constraint goal programming model,

$$\left\{\begin{array}{l} \min \sum\limits_{j=1}^{l} P_j \sum\limits_{j=1}^{m} (u_{ij} d_i^+ + v_{ij} d_i^-) \\ \text{s.t.} \left\{\begin{array}{l} Ch\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) - b_i \leq d_i^+\}(\gamma_i^+) \geq \delta_i^+ \quad i = 1, 2, \ldots, m \\ Ch\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) - b_i \leq d_i^-\}(\gamma_i^-) \geq \delta_i^- \quad i = 1, 2, \ldots, m \\ Ch\{g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\}(\alpha_j) \geq \beta_j, j = 1, 2, \ldots, p \\ d_i^-, d_i^+, d_j^-, d_j^+ \geq 0 \\ u_i, v_i, u_j, v_j = 0 \quad \text{or} \quad 1 \end{array}\right. \end{array}\right. \tag{5.48}$$

where P_j is the preemptive priority factor which expresses the relative importance of various goals, $P_j \gg P_{j+1}$, for all j, u_{ij} is the weighting factor corresponding to positive deviation for goal i with priority j assigned, v_{ij} is the weighting factor corresponding to negative deviation for goal i with priority j assigned, d_i^+ is the (γ_i^+, δ_i^+)-optimistic positive deviation from the target of goal i, defined as

$$\min\{d \vee 0 | Ch\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) - b_i \leq d\}(\gamma_i^+) \geq \delta_i^+\}$$

d_i^- is the (γ_i^-, δ_i^-)-optimistic negative deviation from the target of goal i, defined as

$$\min\{d \vee 0 | Ch\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) - b_i \leq d\}(\gamma_i^-) \geq \delta_i^-\}.$$

From Definition 5.21, we know that

$$\begin{aligned} &Ch\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}_i\}(\gamma_i) \geq \delta_i \\ &\quad \Longleftrightarrow \text{Appr}\{\lambda | Pr\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}_i\} \geq \delta_i\} \geq \gamma_i, i = 1, 2, \ldots, m, \end{aligned}$$

and

$$\begin{aligned} &Ch\{g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\}(\alpha_j) \geq \beta_j \\ &\quad \Longleftrightarrow \text{Appr}\{\lambda | Pr\{g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \geq \beta_j\} \geq \alpha_j, j = 1, 2, \ldots, p. \end{aligned}$$

Then problem (5.47) can be written as

$$\begin{cases} \max[\bar{f}_1, \bar{f}_2, \ldots, \bar{f}_m] \\ \text{s.t.} \begin{cases} \text{Appr}\{\lambda | Pr\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}_i\} \geq \delta_i\} \geq \gamma_i, \ i = 1, 2, \ldots, m \\ \text{Appr}\{\lambda | Pr\{g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \geq \beta_j\} \geq \alpha_j, \ j = 1, 2, \ldots, p \\ \boldsymbol{x} \geq 0 \end{cases} \end{cases} \tag{5.49}$$

where $\delta_i, \gamma_i, \alpha_j, \beta_j$ are predetermined confidence levels, $i = 1, 2, \ldots, m; j = 1, 2, \ldots, p$. Appr$\{\cdot\}$ denotes the approximation level of the event in $\{\cdot\}$, and $Pr\{\cdot\}$ denotes the probability of the event in $\{\cdot\}$.

5.4.2 Linear Ra-Ro CCM and Two-Stage Method

Let's still consider the following linear multi-objective programming model,

$$\begin{cases} \max\left[\tilde{\tilde{c}}_1^T \boldsymbol{x}, \tilde{\tilde{c}}_2^T \boldsymbol{x}, \ldots, \tilde{\tilde{c}}_m^T \boldsymbol{x}\right] \\ \text{s.t.} \begin{cases} \tilde{\tilde{e}}_r^T \boldsymbol{x} \leq \tilde{\tilde{b}}_r, r = 1, 2, \ldots, p \\ \boldsymbol{x} \geq 0 \end{cases} \end{cases}$$

where $\tilde{\tilde{c}}_i = (\tilde{\tilde{c}}_{i1}, \tilde{\tilde{c}}_{i2}, \ldots, \tilde{\tilde{c}}_{in})^T$, $\tilde{\tilde{e}}_r = (\tilde{\tilde{e}}_{r1}, \tilde{\tilde{e}}_{r2}, \ldots, \tilde{\tilde{e}}_{rn})^T$ are random rough vectors and $\tilde{\tilde{b}}_r$ are random variables, $i = 1, 2, \ldots, m, r = 1, 2, \ldots, p$. The chance constrained multi-objective programming model is as follows,

$$\begin{cases} \max[\bar{f}_1, \bar{f}_2, \ldots, \bar{f}_m] \\ \text{s.t.} \begin{cases} \text{Appr}\{\lambda | Pr\{\tilde{\tilde{c}}_i^T \boldsymbol{x} \geq \bar{f}_i\} \geq \delta_i\} \geq \gamma_i, i = 1, 2, \ldots, m \\ \text{Appr}\{\lambda | Pr\{\tilde{\tilde{e}}_r^T \boldsymbol{x} \leq \tilde{\tilde{b}}_r\} \geq \alpha_j, r = 1, 2, \ldots, p \\ \boldsymbol{x} \geq 0 \end{cases} \end{cases} \tag{5.50}$$

where $\delta_i, \gamma_i, \alpha_j, \beta_j$ are predetermined confidence levels, $i = 1, 2, \ldots, m; j = 1, 2, \ldots, p$. We can usually compute their chance measure by the definition for some discrete and Ra-Ro variables and obtain the crisp equivalent.

5.4.2.1 Crisp Equivalent Model

Next, we will introduce a special multiobjective programming with Ra-Ro normal parameters and its equivalent model.

Theorem 5.12. *(Xu and Yao [345]) Assume that Ra-Ro vector $\tilde{\bar{c}}_i$ is characterized by $\tilde{\bar{c}}_i \sim \mathcal{N}(\bar{c}_i(\lambda), V_i^c)$, where $\bar{c}_i = (\bar{c}_{i1}(\lambda), \bar{c}_{i2}(\lambda), \ldots, \bar{c}_{in}(\lambda))^T)$ is a vector by the rough set and V_i^c is a positive definite covariance matrix. Assume that $\bar{c}_i(\lambda)^T \boldsymbol{x} \vdash ([a, b], [c, d])$ (where $0 < c \le a \le b \le d$) is characterized by the following approximation function,*

$$Appr\{\bar{c}_i(\lambda)^T \boldsymbol{x} \ge t\} = \begin{cases} 0 & \text{if} \quad d \le t \\ \dfrac{d-t}{2(d-c)} & \text{if} \quad b \le t \le d \\ \dfrac{1}{2}\left(\dfrac{d-t}{d-c} + \dfrac{b-t}{b-a}\right) & \text{if} \quad a \le t \le b \\ \dfrac{1}{2}\left(\dfrac{d-t}{d-c} + 1\right) & \text{if} \quad c \le t \le a \\ 1 & \text{if} \quad t \le c \end{cases}$$

Then, we have $Appr\{\lambda | Pr\{\tilde{\bar{c}}_i^T \boldsymbol{x} \ge \bar{f}_i\} \ge \delta_i\} \ge \gamma_i$ if and only if

$$\begin{cases} b + R \le \bar{f}_i \le d - 2\gamma_i(d-c) + R & \text{if} \quad b \le M \le d \\ a + R \le \bar{f}_i \le \dfrac{d(b-a) + b(d-c) - 2\gamma_i(d-c)(b-a)}{d-c+b-a} + R & \text{if} \quad a \le M \le b \\ c + R \le \bar{f}_i \le d - (d-c)(2\gamma_i - 1) + R & \text{if} \quad c \le M \le a \\ \bar{f}_i \le c + R & \text{if} \quad M \le c \end{cases}$$

where $M = \bar{f}_i - \Phi^{-1}(1-\delta_i)\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}$ and $R = \Phi^{-1}(1-\delta_i)\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}$ and Φ is the standardized normal distribution and $\delta_i, \gamma_i \in [0, 1]$ are predetermined confidence levels.

Proof. Let's formulate why $\bar{c}_i(\lambda)^T \boldsymbol{x}$ is a rough variable. From the assumption we know that $\bar{c}_{ij}(\lambda)$ is a rough variable and $\bar{c}_i(\lambda) = (\bar{c}_{i1}(\lambda), \bar{c}_{i2}(\lambda), \ldots, \bar{c}_{in}(\lambda))^T$. We set

$$\bar{c}_{ij}(\lambda) \vdash ([a_{ij}, b_{ij}], [c_{ij}, d_{ij}]),$$
$$\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T.$$

It follows that $x_j \bar{c}_{ij}(\lambda) = ([x_j a_{ij}, x_j b_{ij}], [x_j c_{ij}, x_j d_{ij}])$,

$$\begin{aligned}\bar{c}_i(\lambda)^T\boldsymbol{x} &= \sum_{j=1}^{n} c_{ij}(\lambda)x_j \vdash \sum_{j=1}^{n}([x_j a_{ij}, x_j b_{ij}], [x_j c_{ij}, x_j d_{ij}]) \\ &= \left(\left[\sum_{j=1}^{n} a_{ij}x_j, \sum_{j=1}^{n} a_{ij}x_j\right], \left[\sum_{j=1}^{n} c_{ij}x_j, \sum_{j=1}^{n} d_{ij}x_j\right]\right)\end{aligned}$$

So $\bar{c}_i(\lambda)^T\boldsymbol{x}$ is also a rough variable. Now we set

$$\begin{aligned}a &= \sum_{j=1}^{n} a_{ij}x_j, b = \sum_{j=1}^{n} a_{ij}x_j, \\ c &= \sum_{j=1}^{n} c_{ij}x_j, d = \sum_{j=1}^{n} d_{ij}x_j.\end{aligned}$$

then $\bar{c}_i(\lambda)^T\boldsymbol{x} \vdash ([a,b],[c,d])$. In addition, $\tilde{\bar{c}}_i$ is a Ra-Ro vector which is distributed with mean vector $\bar{c}_i(\lambda)$ and positive definite covariance matrix V_i^c, written as $\tilde{\bar{c}}_i \sim \mathcal{N}(\bar{c}_i(\lambda), V_i^c)$. It follows that $\tilde{\bar{c}}_i^T\boldsymbol{x} \sim \mathcal{N}(\bar{c}_i(\lambda)^T\boldsymbol{x}, \boldsymbol{x}^T V_i^c \boldsymbol{x})$. Then, we have that

$$\begin{aligned}&Pr\{\tilde{\bar{c}}_i^T\boldsymbol{x} \geq \bar{f}_i\} \geq \delta_i \\ &\Leftrightarrow Pr\left\{\frac{\tilde{\bar{c}}_i^T\boldsymbol{x} - \bar{c}_i(\lambda)^T\boldsymbol{x}}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}} \geq \frac{\bar{f}_i - \bar{c}_i(\lambda)^T\boldsymbol{x}}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right\} \geq \delta_i \\ &\Leftrightarrow \bar{c}_i(\lambda)^T\boldsymbol{x} + \Phi^{-1}(1-\delta_i)\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}} \geq \bar{f}_i.\end{aligned}$$

It follows that, for given confidence levels $\delta_i, \gamma_i \in [0,1]$,

$$\begin{aligned}&\mathrm{Appr}\{\lambda | Pr\{\tilde{\bar{c}}_i^T\boldsymbol{x} \geq \bar{f}_i\} \geq \delta_i\} \geq \gamma_i \\ &\Leftrightarrow \mathrm{Appr}\left\{\lambda | \bar{c}_i(\lambda)^T\boldsymbol{x} + \Phi^{-1}(1-\delta_i)\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}} \geq \bar{f}_i\right\} \geq \gamma_i \\ &\Leftrightarrow \gamma_i \leq \begin{cases} \dfrac{d-M}{2(d-c)} & \text{if} \quad b \leq M \leq d \\ \dfrac{1}{2}\left(\dfrac{d-M}{d-c} + \dfrac{b-M}{b-a}\right) & \text{if} \quad a \leq M \leq b \\ \dfrac{1}{2}\left(\dfrac{d-M}{d-c} + 1\right) & if \quad c \leq M \leq a \\ 1 & \text{if} \quad M \leq c \end{cases} \\ &\Leftrightarrow \begin{cases} b + R \leq \bar{f}_i \leq d - 2\gamma_i(d-c) + R & \text{if} \quad b \leq M \leq d \\ a + R \leq \bar{f}_i \leq \dfrac{d(b-a) + b(d-c) - 2\gamma_i(d-c)(b-a)}{d-c+b-a} + R & \text{if} \quad a \leq M \leq b \\ c + R \leq \bar{f}_i \leq d - (d-c)(2\gamma_i - 1) + R & \text{if} \quad c \leq M \leq a \\ \bar{f}_i \leq c + R & \text{if} \quad M \leq c \end{cases}\end{aligned}$$

where $M = \bar{f}_i - \Phi^{-1}(1-\delta_i)\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}$, $R = \Phi^{-1}(1-\delta_i)\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}$. This completes the proof. □

Theorem 5.13. *(Xu and Yao [345]) Suppose that Ra-Ro vectors $\tilde{\bar{e}}_r$ are characterized by $\tilde{\bar{e}}_r \sim \mathcal{N}(\bar{e}_r(\lambda), V_r^e)$, and the Ra-Ro variables $\tilde{\bar{b}}_r$ are characterized by $\tilde{\bar{b}}_r \sim \mathcal{N}(\bar{b}_r(\lambda), (\sigma_r^b)^2)$; where $\bar{e}_{rj}(\lambda)$, $\bar{b}_r(\lambda)$ are rough variables, and V_r^e, $(\sigma_r^b)^2$ are positive definite covariances. By Theorem 5.12, we have $\bar{e}_r(\lambda)^T \boldsymbol{x}$, $\bar{b}_r(\lambda)$ are rough variables , then $\bar{e}_r(\lambda)^T \boldsymbol{x} - \bar{b}_r(\lambda) = [(a,b),(c,d)](c \le a \le b \le d)$ is also a rough variable. We assume that it is characterized by the following approximation function,*

$$Appr\{\bar{e}_r(\lambda)^T \boldsymbol{x} - \bar{b}_r(\lambda) \le t\} = \begin{cases} 0 & \text{if} \quad t \le c \\ \dfrac{t-c}{2(d-c)} & \text{if} \quad c \le t \le a \\ \dfrac{1}{2}\left(\dfrac{t-c}{d-c} + \dfrac{t-a}{b-a}\right) & \text{if} \quad a \le t \le b \\ \dfrac{1}{2}\left(\dfrac{t-c}{d-c} + 1\right) & \text{if} \quad b \le t \le d \\ 1 & \text{if} \quad d \le t \end{cases}$$

where $\bar{e}_r = (\bar{e}_{r1}(\lambda), \bar{e}_{r2}(\lambda), \ldots, \bar{e}_{rn}(\lambda))^T$. Then, we have that $Appr\{\lambda | Pr\{\tilde{\bar{e}}_r^T \boldsymbol{x} \le \tilde{\bar{b}}_r\} \ge \theta_r\} \ge \tilde{\eta}_r$ if and only if

$$\begin{cases} a \ge M \ge c + 2(d-c)\tilde{\eta}_r & \text{if} \quad c \le M \le a \\ b \ge M \ge \dfrac{2\tilde{\eta}_r(d-c)(b-a) + c(b-a) + a(d-c)}{d-c+b-a} & \text{if} \quad a \le M \le b \\ d \ge M \ge (2\tilde{\eta}_r - 1)(d-c) + c & \text{if} \quad b \le M \le d \\ M \ge d & \text{if} \quad M \ge d \end{cases}$$

where $M = -\Phi^{-1}(\theta_r)\sqrt{x^T V_r^e x + (\sigma_r^b)^2}$.

Proof. From the assumption, $\tilde{\bar{e}}_r \sim \mathcal{N}(\bar{e}_r(\lambda), V_r^e)$, $\tilde{\bar{b}}_r \sim \mathcal{N}(\bar{b}_r(\lambda), (\sigma_r^b)^2)$, it follows that $\tilde{\bar{e}}_r^T \boldsymbol{x} \sim \mathcal{N}(\bar{e}_r(\lambda)^T \boldsymbol{x}, \boldsymbol{x}^T V_r^e \boldsymbol{x})$, $\tilde{\bar{b}}_r \sim \mathcal{N}(\bar{b}_r(\lambda), (\sigma_r^b)^2)$. Then, $\tilde{\bar{e}}_r^T \boldsymbol{x} - \tilde{\bar{b}}_r \sim \mathcal{N}(\bar{e}_r(\lambda)^T \boldsymbol{x} - b_r, \boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2)$, we have that

$$\begin{aligned} & Pr\{\tilde{\bar{e}}_r^T \boldsymbol{x} \le \tilde{\bar{b}}_r\} \ge \theta_r \\ & \Leftrightarrow Pr\left\{\frac{\tilde{\bar{e}}_r^T \boldsymbol{x} - \tilde{\bar{b}}_r - (\bar{e}_r(\lambda)^T \boldsymbol{x} - \bar{b}_r(\lambda))}{\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2}} \le \frac{\bar{b}_r(\lambda) - \bar{e}_r(\lambda)^T \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2}}\right\} \ge \theta r \\ & \Leftrightarrow \bar{e}_r(\lambda)^T \boldsymbol{x} - \bar{b}_r(\lambda) \le -\Phi^{-1}(\theta_r)\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2} \end{aligned}$$

Since $\tilde{\bar{e}}_r^T \boldsymbol{x} - \tilde{\bar{b}}_r = [(a,b),(c,d)]$, for given confidence levels $\theta_r, \tilde{\eta}_r \in [0,1]$, we have that,

$$\begin{aligned} & \text{Appr}\{\lambda | Pr\{\tilde{\bar{e}}_r^T \boldsymbol{x} \le \tilde{\bar{b}}_r\} \ge \theta_r\} \ge \tilde{\eta}_r \\ & \Leftrightarrow \text{Appr}\{\lambda | \bar{e}_r^T \boldsymbol{x} - b_r \le M\} \ge \eta_r \end{aligned}$$

$$\Leftrightarrow \eta_r \leq \begin{cases} \dfrac{M-c}{2(d-c)} & \text{if} \quad c \leq M \leq a. \\ \dfrac{1}{2}\left(\dfrac{M-c}{d-c}+\dfrac{M-a}{b-a}\right) & \text{if} \quad a \leq M \leq b \\ \dfrac{1}{2}\left(\dfrac{M-c}{d-c}+1\right) & \text{if} \quad b \leq M \leq d \\ 1 & \text{if} \quad M \geq d \end{cases}$$

$$\Leftrightarrow \begin{cases} a \geq M \geq c+2(d-c)\eta_r & \text{if} \quad c \leq M \leq a \\ b \geq M \geq \dfrac{2\eta_r(d-c)(b-a)+c(b-a)+a(d-c)}{d-c+b-a} & \text{if} \quad a \leq M \leq b \\ d \geq M \geq (2\eta_r-1)(d-c)+c & \text{if} \quad b \leq M \leq d \\ M \geq d & \text{if} \quad M \geq d \end{cases}$$

where $M = -\Phi^{-1}(\theta_r)\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2}$. This completes the proof. □

By Theorems 5.12 and 5.13, we have the following equivalent model of chance constraint programming with Ra-Ro coefficients,

$$\begin{cases} \max[\bar{f}_1, \bar{f}_2, \ldots, \bar{f}_m] \\ \text{s.t.} \begin{cases} b+R \leq \bar{f}_i \leq d-2\gamma_i(d-c)+R & \text{if} \quad b \leq M \leq d \\ a+R \leq \bar{f}_i \leq \dfrac{d(b-a)+b(d-c)-2\gamma_i(d-c)(b-a)}{d-c+b-a}+R & \text{if} \quad a \leq M \leq b \\ c+R \leq \bar{f}_i \leq d-(d-c)(2\gamma_i-1)+R & \text{if} \quad c \leq M \leq a \\ \bar{f}_i \leq c+R & \text{if} \quad M \leq c \\ \boldsymbol{x} \geq 0 \end{cases} \end{cases} \tag{5.51}$$

where $M = \bar{f}_i - \Phi^{-1}(1-\delta_i)\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}$ and $R = \Phi^{-1}(1-\delta_i)\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}$. In fact, the above model could be rewritten as

$$\begin{cases} \max[H_1(\boldsymbol{x}), H_2(\boldsymbol{x}), \ldots, H_m(\boldsymbol{x})] \\ \text{s.t. } \boldsymbol{x} \in X' \end{cases} \tag{5.52}$$

where $H_i(\boldsymbol{x}) = d - 2\gamma_i(d-c) + R$, $M = \bar{f}_i - \Phi^{-1}(1-\delta_i)\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}$, $R = \Phi^{-1}(1-\delta_i)\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}$ and $X' = \{\boldsymbol{x} \in X | M \leq d\}$.

Theorem 5.14. *Let* $H(\boldsymbol{x}) = \sum_{i=1}^m \lambda_i H_i(\boldsymbol{x})$, $\lambda_i \in [0,1]$, $H_i(\boldsymbol{x}) = \Phi^{-1}(1-\delta_i)\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}} + \mu_i + \sigma_i \Phi^{-1}(1-\gamma_i), i = 1,2,\ldots,m$, *and* $X = \{\boldsymbol{x} \in R^n | Appr\{\lambda | Pr\{\tilde{\tilde{e}}_r^T \boldsymbol{x} \leq \tilde{\tilde{b}}_r\} \geq \theta_r\} \geq \eta_r, r = 1,2,\ldots,p; \boldsymbol{x} \geq 0\}$. *If* $\gamma_i \geq 0.5$, $\delta_i \geq 0.5$, $\theta_i \geq 0.5$ *and* $\eta_i \geq 0.5$, *problem (5.52) is convex.*

Proof. According to [301], $\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}$ is a convex function. In addition, since $\gamma_i \geq 0.5$ and $\delta_i \geq 0.5$, it follows that $\Phi^{-1}(1-\gamma_i) \leq 0$ and $\Phi^{-1}(1-\delta_i) \leq 0$, we know $H_i(\boldsymbol{x})$ are concave functions, $i = 1, 2, \ldots, m$. So let $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ be any two points in feasible set, and $\alpha \in [0, 1]$, we have that

$$H_i[\alpha \boldsymbol{x}_1 + (1-\alpha)\boldsymbol{x}_2] \geq \alpha H_i(\boldsymbol{x}_1) + (1-\alpha) H_i(\boldsymbol{x}_2)$$

for $\lambda_i \in [0, 1]$,

$$\lambda_i H_i[\alpha \boldsymbol{x}_1 + (1-\alpha)\boldsymbol{x}_2] \geq \alpha\lambda_i H_i(\boldsymbol{x}_1) + (1-\alpha)\lambda_i H_i(\boldsymbol{x}_2)$$

moreover

$$\sum_{i=1}^{m} \lambda_i H_i[\alpha \boldsymbol{x}_1 + (1-\alpha)\boldsymbol{x}_2] \geq \alpha \sum_{i=1}^{m} \lambda_i H_i(\boldsymbol{x}_1) + (1-\alpha) \sum_{i=1}^{m} \lambda_i H_i(\boldsymbol{x}_2)$$

that is

$$H[\alpha \boldsymbol{x}_1 + (1-\alpha)\boldsymbol{x}_2] \geq \alpha H(\boldsymbol{x}_1) + (1-\alpha) H(\boldsymbol{x}_2)$$

So the objective function $H(\boldsymbol{x})$ is concave. Next, we prove that X is convex. From Theorem 5.13, we know that

$$\begin{aligned} &\mathrm{Appr}\{\lambda | Pr\{\tilde{\tilde{e}}_r^T \boldsymbol{x} \leq \tilde{\tilde{b}}_r\} \geq \theta_r\} \geq \eta_r \\ &\quad \Leftrightarrow \Phi^{-1}(\theta_r)\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2} + \mu_r + \sigma_r \Phi^{-1}(\eta_r) \leq 0 \end{aligned}$$

Let $g_r(\boldsymbol{x}) = \Phi^{-1}(\theta_r)\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2} + \mu_r + \sigma_r \Phi^{-1}(\eta_r)$. Then $X = \{\boldsymbol{x} \in R^n | g_r(\boldsymbol{x}) \leq 0, r = 1, 2, \ldots, p; \boldsymbol{x} \geq 0\}$. Since $\theta_r \geq 0.5$ and $\eta_r \geq 0.5$, it follows that $\Phi^{-1}(\theta_r) \geq 0$ and $\Phi^{-1}(\eta_r) \geq 0$, and $g_r(\boldsymbol{x})$ are convex functions, $r = 1, 2, \ldots, p$. Let $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ be two feasible solutions, then

$$g_r(\boldsymbol{x}_1) \leq 0, g_r(\boldsymbol{x}_2) \leq 0$$

according to g_r's convexity, we have

$$g_r[\alpha \boldsymbol{x}_1 + (1-\alpha)\boldsymbol{x}_2] \leq \alpha g_r(\boldsymbol{x}_1) + (1-\alpha) g_r(\boldsymbol{x}_2) \leq 0$$

where $0 \leq \alpha \leq 1$, $r = 1, 2, \ldots, p$. This means that $\alpha \boldsymbol{x}_1 + (1-\alpha)\boldsymbol{x}_2$ is also a feasible solution. So X is a convex set. Above all, we can conclude that problem $\max_{x \in X} H(\boldsymbol{x})$ is a convex programming and its global optimal solution can be obtained easily. □

5.4.2.2 Two-Stage Method

In this section, we will use the two-stage method to seek the efficient solution of the crisp multi-objective programming problem (5.52). The two-stage method is proposed by Li and Lee [199] on the basis of the maximin method proposed by Zimmermann [365].

The first stage: apply Zimmermann's minimum operator to obtain the maximal satisfying degree α^0 of the objective set and the related feasible solution $\boldsymbol{x}^0$, i.e.,

$$\begin{cases} \max \alpha \\ \text{s.t.} \begin{cases} \mu_k(\boldsymbol{x}) = \dfrac{H_k(\boldsymbol{x}) - H_k^{'}}{H_k^{*} - H_k^{'}} \geq \alpha, k = 1, 2, \ldots, m \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{5.53}$$

Assume that the optimal solution of problem (5.53) is $(\boldsymbol{x}^0, \alpha^0)$, where α^0 is the optimal satisfying degree of the whole objective sets. If the optimal solution of problem (5.53) is unique, $\boldsymbol{x}^0$ is the efficient of the problem (5.52). However, we cannot usually know if the optimal solution of problem (5.53) is unique, then the efficiency of $\boldsymbol{x}^0$ must be checked by the following stage.

The second stage: check the efficiency of efficiency of $\boldsymbol{x}^0$ or seek the new efficient solution $\boldsymbol{x}^1$. Construct a new model whose objective function is to maximize the average satisfying degree of all objects subject to the additional constraint $\alpha_k \geq \alpha^0 (k = 1, 2, \ldots, m)$. Since the compensatory of the arithmetic mean operator, the solution obtained in the second stage is efficient. The existence of the constraint $\alpha_k \geq \alpha^0 (k = 1, 2, \ldots, m)$ guarantees the mutual equilibrium of every objective functions.

$$\begin{cases} \max \frac{1}{m} \sum\limits_{k=1}^{m} \alpha_k \\ \text{s.t.} \begin{cases} \alpha^0 \leq \alpha_k \leq \mu_k(\boldsymbol{x}), k = 1, 2, \ldots, , m \\ 0 \leq \alpha_k \leq 1, \quad \boldsymbol{x} \in X \end{cases} \end{cases} \tag{5.54}$$

Assume that the optimal solution of problem (5.54) is $\boldsymbol{x}^1$. It's easy to prove that $\boldsymbol{x}^1$ is also the solution of problem (5.53), thus we have $\boldsymbol{x}^1 = \boldsymbol{x}^0$ if the solution of problem (5.53) is unique. But if the solution of problem (5.53) is not unique, $\boldsymbol{x}^0$ may be efficient solution or not and we can guarantee $\boldsymbol{x}^1$ is definitely efficient. Thus in any case, the two-stage method can provide an efficient solution in the second stage.

5.4.3 Nonlinear Ra-Ro CCM and Ra-Ro Simulation-Based PTS

Let's still consider the following nonlinear multi-objective programming problem,

$$\begin{cases} \max\ [\bar{f}_1, \bar{f}_2, \ldots, \bar{f}_m] \\ \text{s.t.} \begin{cases} \text{Appr}\{\lambda | Pr\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}_i\} \geq \delta_i\} \geq \gamma_i,\ i = 1, 2, \ldots, m \\ \text{Appr}\{\lambda | Pr\{g_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \geq \beta_j\} \geq \alpha_j,\ j = 1, 2, \ldots, p \\ \boldsymbol{x} \geq 0, \end{cases} \end{cases}$$

where $f_i(\boldsymbol{x}, \boldsymbol{\xi})$, or $g_j(\boldsymbol{x}, \boldsymbol{\xi})$ or both of them are nonlinear functions with respect to $\boldsymbol{\xi}$, $\delta_i, \gamma_i, \alpha_j, \beta_j$ are predetermined confidence levels, $i = 1, 2, \ldots, m; j = 1, 2, \ldots, p$. Appr$\{\cdot\}$ denotes the approximation level of the event in $\{\cdot\}$, and $Pr\{\cdot\}$ denotes the probability of the event in $\{\cdot\}$. If their distribution is unclear and it is difficult to convert it into a deterministic one, we have to apply the parallel tabu search algorithm based on the Ra-Ro simulation for CCM to obtain its optimal solution. Next, we will introduce the Ra-Ro simulation-based parallel tabu search algorithm for chance constrained model.

5.4.3.1 Ra-Ro Simulation for CCM

Let's consider to simulate the critical value $\bar{f}$ of $\text{Appr}\{\lambda | Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}\} \geq \beta\} \geq \alpha$ by Ra-Ro simulation. Generate $\underline{\lambda}_1, \underline{\lambda}_2, \ldots, \underline{\lambda}_N$ from the lower approximation $\underline{X}$ and $\bar{\lambda}_1, \bar{\lambda}_2, \ldots, \bar{\lambda}_N$ from the upper approximation $\overline{X}$ according to the approximation function Appr. For each $\underline{\lambda}_i$ and $\bar{\lambda}_i (i = 1, 2, \ldots, N)$, $\boldsymbol{\xi}(\underline{\lambda}_i)$ and $\boldsymbol{\xi}(\bar{\lambda}_i)$ are both random vectors. For any number t, let $\underline{N}(t)$ denote the number of $\underline{\lambda}_k$ satisfying $Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}(\underline{\lambda}_k)) \geq t\} \geq \beta$ for $k = 1, 2, \ldots, N$ and $\bar{N}(t)$ denote the number of $\bar{\lambda}_k$ satisfying $Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}(\bar{\lambda}_k)) \geq t\} \geq \beta$ for $k = 1, 2, \ldots, N$, where $Pr\{\cdot\}$ may be estimated by random simulation. Then we may find the maximal value t such that

$$\frac{\underline{N}(t) + \bar{N}(t)}{2N} \geq \alpha$$

This value is an estimation of $\bar{f}$. Then the procedure simulating the critical value $\bar{f}$ of $\text{Appr}\{\lambda | Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}\} \geq \beta\} \geq \alpha$ can be summarized as follows:

Procedure `Ra-Ro simulation for CCM`

Input: The decision vector $\boldsymbol{x}$
Output: The critical value $\bar{f}$ of $\text{Appr}\{\lambda | Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}\} \geq \beta\} \geq \alpha$
Step 1. Generate $\underline{\lambda}_1, \underline{\lambda}_2, \ldots, \underline{\lambda}_N$ from the lower approximation $\underline{X}$ according to the approximation function Appr;
Step 2. Generate $\bar{\lambda}_1, \bar{\lambda}_2, \ldots, \bar{\lambda}_N$ from the upper approximation $\overline{X}$ according to the approximation function Appr;
Step 3. Find the maximal value t such that $\frac{\underline{N}(t) + \bar{N}(t)}{2N} \geq \alpha$ holds;
Step 4. Return t.

Example 5.11. In order to compute $Ch\left\{\sqrt{\tilde{\xi}_1^2 + \tilde{\xi}_2^2 + \tilde{\xi}_3^2} \geq \bar{f}\right\}(0.9) \geq 0.9$, where $\tilde{\xi}_1, \tilde{\xi}_2$ and $\tilde{\xi}_3$ are three Ra-Ro variables as follows,

$$\begin{aligned}
\tilde{\tilde{\xi}}_1 &\sim \mathscr{U}(\tilde{\rho}_1, \tilde{\rho}_1 + 2), \text{ with } \tilde{\rho}_1 = ([1,2],[1,3]),\\
\tilde{\tilde{\xi}}_2 &\sim \mathscr{N}(\tilde{\rho}_2, 1), \qquad \text{with } \tilde{\rho}_2 = ([0,1],[0,3]),\\
\tilde{\tilde{\xi}}_3 &\sim exp(\tilde{\rho}_3), \qquad \text{with } \tilde{\rho}_3 = ([1,2],[0,3]),
\end{aligned}$$

A run of Ra-Ro simulation with 1000 cycles shows that $\bar{f} = 2.1$.

5.4.3.2 Parallel Tabu Search Algorithm

In this section, we will combine the Ra-Ro simulation with the parallel tabu search(abbr. PTS) algorithm to solve the multiobjective problem. TS is an efficient tool to solve the multiobjective problems. However, as the problem size gets larger, TS has some drawbacks:

(a) TS needs to compute the objective function for solution candidates in the neighborhood around a solution at each iteration. The calculation is very time consuming in large-scale problems. The large size problem often gives large neighborhood even though the neighborhood is defined as a set of solution candidate with the Hamming distance equal to 1.
(b) The complicated non-linear optimal problem has many local minima in large scale problems. That implies that one-point search does not give satisfactory solutions due to the huge search space. Complicated optimal problems require the solution diversity.

In this section, the decomposition of the neighborhood accommodates drawback. The neighborhood is decomposed into several sub-neighborhoods. A processor may be assigned to each sub-neighborhood so that the best solution candidate is selected independently in each sub-neighborhood. After selecting the best solution in each sub-neighborhood, the best solution is eventually selected from the best solutions in the sub-neighborhood. Also, the multiple Tabu lengths is proposed to deal with the multiobjective problem with Ra-Ro parameters. TS itself has only one Tabu length. Moreover, it is important to find out better solutions from different directions rather than from only one direction for a longer period. Namely it is effective to make the solution search process more diverse.

Many classifications of PTS algorithms have been proposed [13, 40, 66, 67, 99, 110, 111, 227, 258, 328]. They are based on many criteria: number of initial solutions, identical or different parameter settings, control and communications strategies. In this section, we mainly introduce the parallel adaptive tabu search algorithm introduced by Talbi, Hafidi and Geib [314] and readers can refer to the related literature. They have identified two main categories (Fig. 5.14).

Domain decomposition: Parallelism in this class of algorithms relies exclusively on:

(a) The decomposition of the search space: the main problem is decomposed into a number of smaller subproblems, each subproblem being solved by a different TS algorithm [313].

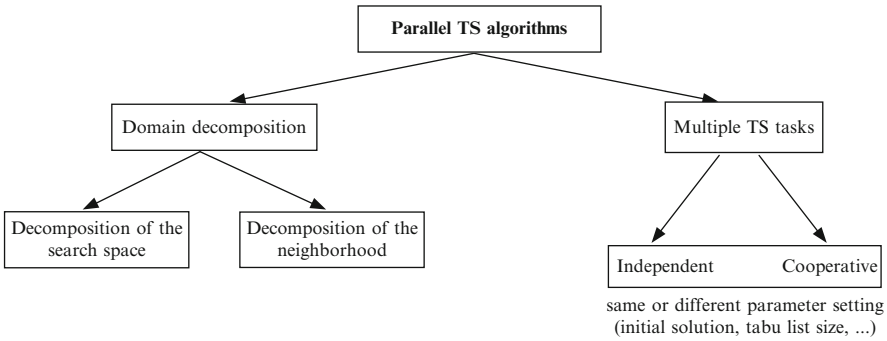


Fig. 5.14 Hierarchical classification of PTS strategies

Table 5.3 Another taxonomy dimension for PTS algorithm

	Tasks or Data	
	Number	Location
Non-adaptive	Static	Static
Semi-adaptive	Static	Dynamic
Adaptive	Dynamic	Dynamic

(b) The decomposition of the neighborhood: the search for the best neighbor at each iteration is performed in parallel, and each task evaluates a different subset of the partitioned neighborhood [40, 312].

A high degree of synchronisation is required to implement this class of algorithms.

Multiple tabu search tasks: This class of algorithms consists in executing multiple TS algorithms in parallel. The di.erent TS tasks start with the same or different parameter values (initial solution, tabu list size, maximum number of iterations, etc.). Tabu tasks may be independent (without communication) [214,258] or cooperative. A cooperative algorithm has been proposed in [66], where each task performs a given number of iterations, then broadcasts the best solution. The best of all solutions becomes the initial solution for the next phase.

Parallelizing the exploration of the search space or the neighborhood is problem-dependent. This assumption is strong and is met only for few problems. The second class of algorithms is less restrictive and then more general. A parallel algorithm that combines the two approaches (two-level parallel organization) has been proposed in [13].

We can extend this classification by introducing a new taxonomy dimension: the way scheduling of tasks over processors is done. PTS algorithms fall into three categories depending on whether the number and/or the location of work (tasks, data) depend or not on the load state of the parallel machine (Table 5.3):

Non-adaptive: This category represents PTS in which both the number of tasks of the application and the location of work (tasks or data) are generated at compile

time (static scheduling). The allocation of processors to tasks (or data) remains unchanged during the execution of the application regardless of the current state of the parallel machine. Most of the proposed algorithms belong to this class.

An example of such an approach is presented in [254]. The neighborhood is partitioned in equal size partitions depending on the number of workers, which is equal to the number of processors of the parallel machine. In [40], the number of tasks generated depends on the size of the problem and is equal to n^2, where n is the problem size.

When there are noticeable load or power differences between processors, the search time of the non-adaptive approach presented is derived by the maximum execution time over all processors (highly loaded processor or the least powerful processor). A significant number of tasks are often idle waiting for other tasks to complete their work.

Semi-adaptive: To improve the performance of the parallel non adaptive TS algorithms, dynamic load balancing must be introduced [13, 254]. This class represents applications for which the number of tasks is fixed at compile-time, but the locations of work (tasks, data) are determined and/or changed at run-time (as seen in Table 5.3). Load balancing requirements are met in [254] by a dynamic redistribution of work between processors. During the search, each time a task finishes its work, it proceeds to a work-demand. Dynamic load balancing through partition of the neighborhood is done by migrating data.

However, the parallelism degree in this class of algorithms is not related to load variation in the parallel system: when the number of tasks exceeds the number of idle nodes, multiple tasks are assigned to the same node. Moreover, when there are more idle nodes than tasks, some of them will not be used.

Adaptive: A parallel adaptive program refers to a parallel computation with a dynamically changing set of tasks. Tasks may be created or killed function of the load state of the parallel machine. Different types of load state dessimination schemes may be used [37]. A task is created automatically when a processor becomes idle. When a processor becomes busy, the task is killed. Next, let's introduce the design about the parallel adaptive TS introduced by Talbi, Hafidi and GeibTalbi [314].

The programming style used is the master/workers paradigm. The master task generates work to be processed by the workers. Each worker task receives a work from the master, computes a result and sends it back to the master. The master/workers paradigm works well in adaptive dynamic environments because:

(a) When a new node becomes available, a worker task can be started there.
(b) When a node becomes busy, the master task gets back the pending work which was being computed on this node, to be computed on the next available node.

The master implements a central memory through which passes all communication, and that captures the global knowledge acquired during the search. The number of workers created initially by the master is equal to the number of idle nodes in the parallel platform. Each worker implements a sequential TS task. The initial solution is generated randomly and the tabu list is empty. The parallel adaptive TS algorithm reacts to two events (Fig. 5.15):

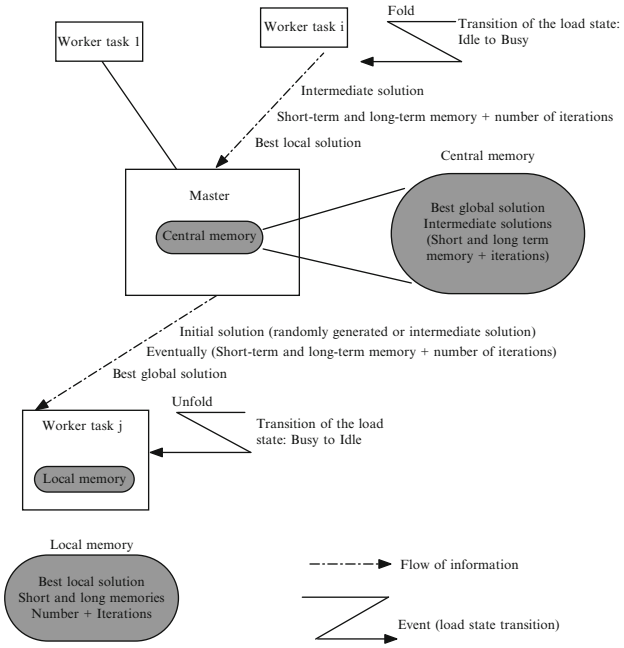


Fig. 5.15 Architecture of the parallel adaptive TS

Transition of the load state of a node from idle to busy: If a node hosting a worker becomes loaded, the master folds up the application by withdrawing the worker. The concerned worker puts back all pending work to the master and dies. The pending work is composed of the current solution, the best local solution found, the short-term memory, the long-term memory and the number of iterations done without improving the best solution. The master updates the best global solution if it's worst than the best local solution received.

Transition of the load state of a node from busy to idle: When a node becomes available, the master unfolds the application by staring a new worker on it. Before starting a sequential TS, the worker task gets the values of the different parameters from the master: the best global solution and an initial solution which may be an intermediate solution found by a folded TS task, which constitute a "good" initial solution. In this case, the worker receives also the state of the short-term memory, the long-term memory and the number of iterations done without improving the best solution.

The local memory of each TS task which defines the pending work is composed of (Fig. 5.15): the best solution found by the task, the number of iterations applied, the intermediate solution and the adaptive memory of the search (short-term and long-term memories). The central memory in the master is then composed of (Fig. 5.15): the best global solution found by all TS tasks, the different intermediate solutions with the associated number of iterations and adaptive memory.

5.4.4 Numerical Examples

Example 5.12. Let' consider the following CCM problem,

$$\begin{cases} \max[f_1, f_2] \\ \text{s.t.} \begin{cases} \text{Appr}\{\lambda | Pr\{\tilde{\tilde{\xi}}_1 x_1 + \tilde{\tilde{\xi}}_2 x_2 + \tilde{\tilde{\xi}}_3 x_3 \geq f_1\} \geq \delta_1\} \geq \gamma_1, \\ \text{Appr}\{\lambda | Pr\{c_1 \tilde{\tilde{\xi}}_4 x_1 + c_2 \tilde{\tilde{\xi}}_5 x_2 + c_3 \tilde{\tilde{\xi}}_6 x_3 \geq f_2\} \geq \delta_2\} \geq \gamma_2, \\ x_1 + x_2 + x_3 \leq 15, \\ x_1 + x_2 + x_3 \geq 10, \\ x_1 + 4x_2 + 2x_3 \leq 30, \\ 2 \leq x1, x2, x3 \leq 6, \end{cases} \end{cases} \tag{5.55}$$

where $c = (c_1, c_2, c_3) = (1.2, 0.8, 1.5)$,

$\tilde{\tilde{\xi}}_1 \sim \mathcal{N}(\rho_1, 1)$, with $\rho_1 \vdash ([1,2],[0,3])$, $\tilde{\tilde{\xi}}_2 \sim \mathcal{N}(\rho_2, 4)$, with $\rho_2 \vdash ([2,3],[1,4])$,
$\tilde{\tilde{\xi}}_3 \sim \mathcal{N}(\rho_3, 1)$, with $\rho_3 \vdash ([3,4],[2,5])$, $\tilde{\tilde{\xi}}_4 \sim \mathcal{N}(\rho_4, 2)$, with $\rho_4 \vdash ([0,1],[0,3])$,
$\tilde{\tilde{\xi}}_5 \sim \mathcal{N}(\rho_5, 1)$, with $\rho_5 \vdash ([1,2],[0,3])$, $\tilde{\tilde{\xi}}_6 \sim \mathcal{N}(\rho_6, 1)$, with $\rho_6 \vdash ([2,3],[0,3])$,

and $\rho_i (i = 1, 2, \ldots, 6)$ are rough variables. We set $\delta_i = \gamma_i = 0.9$, then $\Phi^{-1}(1 - \delta_i) = -1.28, i = 1, 2$. It follows that problem (5.55) is equivalent to

$$\begin{cases} \max H_1(\boldsymbol{x}) = -(2.4x_1 + 1.4x_2 + 2.2x_3 + 1.28\sqrt{x_1^2 + 4x_2^2 + x_3^2}) \\ \max H_2(\boldsymbol{x}) = -(2.88x_1 + 1.92x_2 + 3.6x_3 + 1.28\sqrt{2x_1^2 + x_2^2 + x_3^2}) \\ \text{s.t.} \begin{cases} x_1 + x_2 + x_3 \leq 15 \\ x_1 + x_2 + x_3 \geq 10 \\ x_1 + 4x_2 + 2x_3 \leq 30 \\ 2 \leq x_1, x_2, x_3 \leq 6 \end{cases} \end{cases} \tag{5.56}$$

Then we use the two-stage method to solve the above problem. Construct the membership function as follows,

$$\mu_i = \frac{H_i(\boldsymbol{x}) - H_i^0}{H_i^1 - H_i^0}, \ i = 1, 2,$$

where $H_i^1 = \max_{\boldsymbol{x} \in X} H_i(\boldsymbol{x})$ and $H_i^0 = \min_{\boldsymbol{x} \in X} H_i(\boldsymbol{x})$. Then we get

$$H_1^1 = -30.00, \ H_1^0 = -45.10, \ H_2^1 = -33.35, \ H_2^0 = -58.49.$$

According to the stage method, the problem (5.56) can be written as,

$$\begin{cases} \max \alpha \\ \text{s.t.} \begin{cases} \dfrac{2.4x_1 + 1.4x_2 + 2.2x_3 + 1.28\sqrt{x_1^2 + 4x_2^2 + x_3^2} - 30.00}{14.9} \geq \alpha \\ \dfrac{2.88x_1 + 1.92x_2 + 3.6x_3 + 1.28\sqrt{2x_1^2 + x_2^2 + x_3^2} - 33.35}{25.14} \geq \alpha \\ x_1 + x_2 + x_3 \leq 15 \\ x_1 + x_2 + x_3 \geq 10 \\ x_1 + 4x_2 + 2x_3 \leq 30 \\ 2 \leq x_1, x_2, x_3 \leq 6 \end{cases} \end{cases} \tag{5.57}$$

Then we obtained the optimal solution of problem $\boldsymbol{x}^* = (6.00, 3.00, 6.00)^T$ and $\alpha = 0.9998$. At the second stage, construct the following programming to check the efficiency of $\boldsymbol{x}^*$,

$$\begin{cases} \max \frac{1}{2}(\alpha_1 + \alpha_2) \\ \text{s.t.} \begin{cases} \dfrac{2.4x_1 + 1.4x_2 + 2.2x_3 + 1.28\sqrt{x_1^2 + 4x_2^2 + x_3^2} - 30.00}{14.9} \geq \alpha_1 \\ \dfrac{2.88x_1 + 1.92x_2 + 3.6x_3 + 1.28\sqrt{2x_1^2 + x_2^2 + x_3^2} - 33.35}{25.14} \geq \alpha_2 \\ x_1 + x_2 + x_3 \leq 15 \\ x_1 + x_2 + x_3 \geq 10 \\ x_1 + 4x_2 + 2x_3 \leq 30 \\ 2 \leq x_1, x_2, x_3 \leq 6 \\ 0.9998 \leq \alpha_1, \alpha_2 \leq 1 \end{cases} \end{cases} \tag{5.58}$$

Resolve it and we get $\boldsymbol{x}'^* = \boldsymbol{x}^* = (6.00, 3.00, 6.00)^T$. This means the optimal solution of problem (5.56) is unique.

Example 5.13. Let us consider a multi-objective programming with Ra-Ro coefficients,

$$\begin{cases} \max[f_1, f_2] \\ \text{s.t.} \begin{cases} \text{Appr}\{\lambda | Pr\{\sqrt{(x_1 - \xi)^2 + (x_2 - \xi)^2} \geq f_1\} \geq 0.8\} \geq 0.8 \\ \text{Appr}\{\lambda | Pr\{\sqrt{(x_1 + \xi)^2 + (x_2 + \xi)^2} \geq f_2\} \geq 0.8\} \geq 0.8 \\ x_1 + x_2 \leq 5 \\ x_1, x_2 \geq 0 \end{cases} \end{cases} \tag{5.59}$$

where ξ is a normally distributed Ra-Ro variable, written as $\xi \sim \mathcal{N}(\tilde{\rho}, 1)$, where $\tilde{\rho} \vdash ([1, 2], [0, 3])$ is a rough variable.

Next, we apply the parallel tabu search algorithm based on the Ra-Ro simulation to solve the nonlinear programming problem (5.59) with the Ra-Ro parameters.

Step 1. Set the move step $h = 0.5$ and the h neighbor $N(\boldsymbol{x}, h)$ for the present point $\boldsymbol{x}$ is defined as follows,

$$N(\boldsymbol{x}, h) = \left\{ \boldsymbol{y} | \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \leq h \right\}.$$

The random move of point $\boldsymbol{x}$ to point $\boldsymbol{y}$ in its h neighbor along direction s is given by

$$\boldsymbol{y}_s = \boldsymbol{x}_s + rh,$$

where r is a random number that belongs to [0,1], $s = 1, 2$.

Step 2. Give the step set $H = \{h_1, h_2, \ldots, h_r\}$ and randomly generate a feasible point $\boldsymbol{x}_0$ checked by the random rough simulation. One should empty the Tabu list T (the list of inactive steps) at the beginning.

Step 3. For each active neighbor $N(\boldsymbol{x}, h)$ of the present point $\boldsymbol{x}$, where $h \in H - T$, a feasible random move that satisfies all the constraints in problem (5.59) is to be generated.

Step 4. Construct the single objective function as follows,

$$\boldsymbol{f}(\boldsymbol{x}) = w_1 f_1 + w_2 f_2$$

where $w_1 + w_2 = 1$ and $w_i (i = 1, 2)$ is predetermined by the decision maker. Compare the $\boldsymbol{f}(\boldsymbol{x})$ of the feasible moves with that of the current solution by the Ra-Ro simulation. If an augmenter in new objective function of the feasible moves exists, one should save this feasible move as the updated current one by adding the corresponding step to the Tabu list T and go to the next step; otherwise, go to the next step directly.

Step 5. Stop if the termination criteria are satisfied; other wise, empty T if it is full; then go to Step 3. Here, we set the computation is determined if the better solution doesn't change again.

We can solve the programming problem (5.59) by the parallel tabu search algorithm and genetic algorithm, respectively. Table 5.4 shows the result computed by parallel TS algorithm setting the Tabu length is 5 and the candidate number is 3.

Table 5.4 The result computed by PTS algorithm at different weights

w_1	w_2	x_1	x_2	x_3	H	Gen
0.1	0.9	90.68	25.19	84.13	−2304.55	270
0.2	0.8	90.25	25.08	84.66	−2287.08	240
0.3	0.7	89.82	24.99	85.19	−2269.57	256
0.4	0.6	89.39	24.89	85.72	−2252.01	269
0.5	0.5	88.99	24.79	86.25	−2234.40	294
0.6	0.4	88.53	24.70	86.78	−2216.74	291
0.7	0.3	88.10	24.60	87.30	−2199.03	268
0.8	0.2	87.67	24.50	87.83	−2181.29	281
0.9	0.1	87.24	24.40	88.36	−2163.48	276

Table 5.5 The result computed by GA at different weights

w_1	w_2	x_1	x_2	x_3	H	Gen
0.1	0.9	88.98	25.97	85.06	−2304.84	5000
0.2	0.8	91.64	25.29	83.07	−2287.3	5000
0.3	0.7	91.53	24.67	83.77	−2269.82	5000
0.4	0.6	91.1	24.54	84.36	−2252.26	5000
0.5	0.5	87.78	24.99	87.23	−2234.52	5000
0.6	0.4	88.61	24.77	86.62	−2216.42	5000
0.7	0.3	87.00	25.08	87.92	−2199.16	5000
0.8	0.2	87.06	24.53	88.41	−2181.31	5000
0.9	0.1	86.23	24.44	89.33	−2163.57	5000

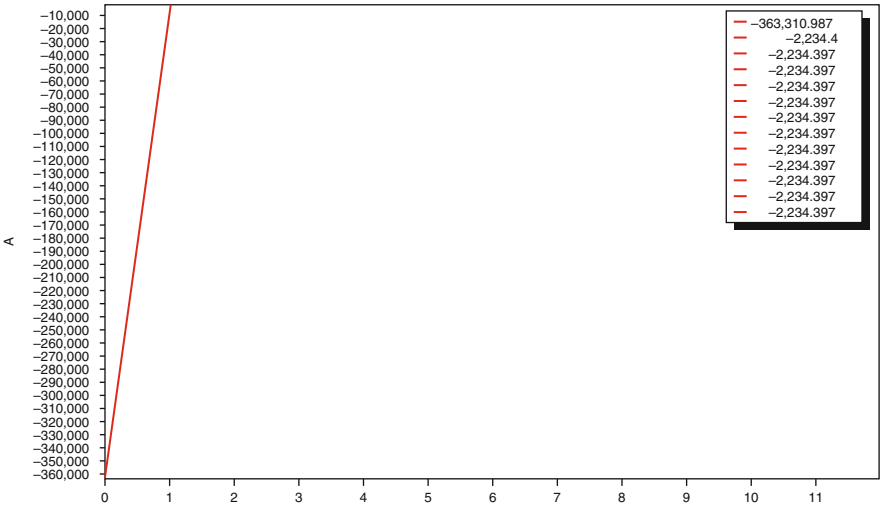


Fig. 5.16 The iteration process by PTS at $w_1 = 0.5$ and $w_2 = 0.5$

Table 5.5 shows the result computed by GA setting the chromosome population number is 20, the crossover probability is 0.85 and the mutation probability is 0.01. Figures 5.16 and 5.17 show the computing process by the two algorithm at $w_1 = 0.5$ and $w_2 = 0.5$, respectively.

5.5 Ra-Ro DCM

In 1959, Charnes and Cooper [45] proposed the probability maximization model (P-model) for the random programming problems, then Liu [207] introduced the dependent-chance programming according to the definition of the chance measure

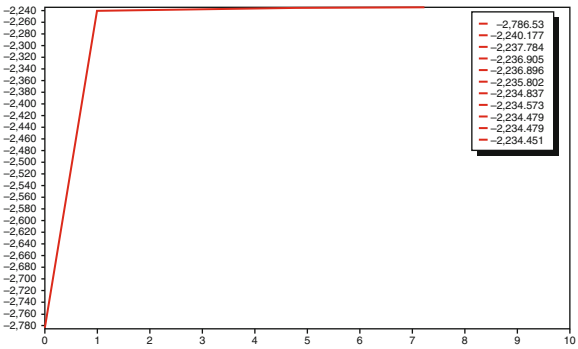


Fig. 5.17 The iteration process by GA at $w_1 = 0.5$ and $w_2 = 0.5$

on the basis of the P-model. Its meaning lies in arriving at the maximal probability which decision makers anticipate.

5.5.1 General Model

Let's consider the following model

$$\begin{cases} \max[Ch\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) \leq f_i\}(\alpha_i), i = 1, 2, \ldots, m] \\ \text{s.t.} \begin{cases} g_r(\boldsymbol{x}) \leq 0, r = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{5.60}$$

where x is a n-dimensional decision vector, $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_n)$ is a Ra-Ro vector, and α_i is the given confidence level. Here, the constraints are all certain. For uncertain constraints, we can deal with them by the technique of chance-constrained programming. When the Ra-Ro variable degenerates to the single uncertain variable, we obtain the following results.

Remark 5.5. If the Ra-Ro vector ξ degenerates to a random vector, for any given α_i,

$$Ch\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) \leq f_i\}(\alpha_i) = Pr\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) \leq f_i\}(\alpha_i), i = 1, 2, \ldots, n.$$

Thus, the problem (5.60) is equivalent to

$$\begin{cases} \max [Pr\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) \leq f_i\}(\alpha_i), i = 1, 2, \ldots, n] \\ \text{s.t.} \begin{cases} g_r(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0, r = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{5.61}$$

where $\boldsymbol{\xi}$ is a random vector, and this model is a standard stochastic DCM.

Remark 5.6. When the Ra-Ro variable $\boldsymbol{\xi}$ degenerates to a rough vector, for any given α_i, $Pr\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) \leq f_i\} = 1$. This means

$$Ch\{f_i(\boldsymbol{x},\boldsymbol{\xi}) \le f_i\}(\alpha_i) = \mathrm{Appr}\{f_i(\boldsymbol{x},\boldsymbol{\xi}) \le f_i\}(\alpha_i), i = 1,2,\ldots,n.$$

Thus, the problem (5.60) is converted into

$$\begin{cases} \max\left[\mathrm{Appr}\{f_i(\boldsymbol{x},\boldsymbol{\xi}) \le f_i\}(\alpha_i), i = 1,2,\ldots,n\right] \\ \text{s.t.} \begin{cases} g_r(\boldsymbol{x},\boldsymbol{\xi}) \le 0, r = 1,2,\ldots,p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{5.62}$$

where ξ_i is a rough variable, and this model is identical with the dependent-chance programming model in rough environment.

Remark 5.7. Because there are many priority factors which need to be considered in real life, the goal programming problem is applied to formulate DM's goal level. Then the goal programming problem of (5.60) can be got as follows,

$$\begin{cases} \max \sum_{j=1}^{l} P_j \sum_{i=1}^{n} (u_{ij} d_i^- + v_{ij} d_i^+) \\ \text{s.t.} \begin{cases} Ch\{f_i(\boldsymbol{x},\boldsymbol{\xi}) \le f_i\}(\alpha_i) + d_i^- - d_i^+ = \beta_i \\ d_i^-, d_i^+ \ge 0, i = 1,2,\ldots,n \\ g_r(\boldsymbol{x},\boldsymbol{\xi}) \le 0, r = 1,2,\ldots,p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{5.63}$$

where P_j is the preemptive priority factor which DM considers the relative importance of item i. For all j, u_{ij}, v_{ij} are respectively the weighting factor corresponding to negative or positive deviation for goal i with priority j assigned. l is the number of priorities. α_i is the given level value, β_i is the goal value according to goal i.

5.5.2 Linear Ra-Ro DCM and the Fuzzy Goal Method

Let's still consider the following linear multi-objective model,

$$\begin{cases} \max\left[\tilde{\tilde{c}}_1^T \boldsymbol{x}, \tilde{\tilde{c}}_2^T \boldsymbol{x}, \ldots, \tilde{\tilde{c}}_m^T \boldsymbol{x}\right] \\ \text{s.t.} \begin{cases} e_r^T \boldsymbol{x} \le b_r, r = 1,2,\ldots,p \\ \boldsymbol{x} \in X \end{cases} \end{cases}$$

where $\tilde{\tilde{c}}_i = (\tilde{\tilde{c}}_{i1}, \tilde{\tilde{c}}_{i2}, \ldots, \tilde{\tilde{c}}_{in})$ is the random rough vector, $i = 1,2,\ldots,m$. Its dependent-chance multi-objective model is as follows,

$$\begin{cases} \max[\beta_1, \beta_2, \ldots, \beta_m] \\ \text{s.t.} \begin{cases} \mathrm{Appr}\{\lambda | Pr\{\tilde{\tilde{c}}_i^T \boldsymbol{x} \le f_i\} \ge \beta_i\} \ge \alpha_i, i = 1,2,\ldots,m \\ e_r^T \boldsymbol{x} \le b_r, \qquad r = 1,2,\ldots,p \\ \boldsymbol{x} \in X \end{cases} \end{cases}$$

where $\tilde{\bar{c}}_i = (\tilde{\bar{c}}_{i1}, \tilde{\bar{c}}_{i2}, \ldots, \tilde{\bar{c}}_{in})$ is a Ra-Ro vector, α_i is the given confidence level and f_i is the predetermined value.

5.5.2.1 Crisp Equivalent Model

Similarly, when the random rough parameters are subject to normal distribution, we still have the following results.

Theorem 5.15. *(Xu and Yao [347]) Assume that Ra-Ro vector $\tilde{\bar{c}}_i$ is characterized by $\tilde{\bar{c}}_i \sim \mathcal{N}(\bar{c}_i(\lambda), V_i^c)$, where $c_i = (c_{i1}(\lambda), c_{i2}(\lambda), \ldots, c_{in}(\lambda))^T)$ is a vector approximated by rough sets and V_i^c is a positive definite covariance matrix. Let $\bar{c}_i(\lambda)^T \boldsymbol{x} \vdash ([a,b],[c,d])_R$ (where $0 \le c \le a < b \le d$), then we have the following four equivalent models*

$$\begin{cases} \max[\beta_1, \beta_2, \ldots, \beta_m] \\ \text{s.t.} \begin{cases} \Phi\left(\dfrac{f_i - a}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right) \le \beta_i, & i = 1,2,\ldots,m \\ \Phi\left(\dfrac{f_i - c - 2\alpha_i(d-c)}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right) \ge \beta_i, & i = 1,2,\ldots,m \\ e_r^T \boldsymbol{x} \le b_r, & r = 1,2,\ldots,p \\ \boldsymbol{x} \in X \end{cases} \end{cases}$$

and

$$\begin{cases} \max[\beta_1, \beta_2, \ldots, \beta_m] \\ \text{s.t.} \begin{cases} \Phi\left(\dfrac{f_i - b}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right) \le \beta_i, & i = 1,2,\ldots,m \\ \Phi\left(\dfrac{f_i - l}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right) \ge \beta_i, & i = 1,2,\ldots,m, \\ e_r^T \boldsymbol{x} \le b_r, & r = 1,2,\ldots,p \\ \boldsymbol{x} \in X \end{cases} \end{cases}$$

where $l = \max\{a, \frac{c(b-a)+a(d-c)-2\alpha_i(d-c)(b-a)}{b-a+d-c}\}$, *and*

$$\begin{cases} \max[\beta_1, \beta_2, \ldots, \beta_m] \\ s.t. \begin{cases} \Phi\left(\dfrac{f_i - d}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right) \le \beta_i,\ i = 1, 2, \ldots, m \\ \Phi\left(\dfrac{f_i - t}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right) \ge \beta_i,\ i = 1, 2, \ldots, m \\ e_r^T \boldsymbol{x} \le b_r, \qquad r = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases}$$

where $t = \max\{b, (2\alpha_i - 1)(d - c) + c\}$, *and*

$$\begin{cases} \max[\beta_1, \beta_2, \ldots, \beta_m] \\ s.t. \begin{cases} \Phi\left(\dfrac{f_i - d}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right) \ge \beta_i,\ i = 1, 2, \ldots, m \\ e_r^T \boldsymbol{x} \le b_r, \qquad r = 1, 2, \ldots, p \\ \boldsymbol{x} \in X. \end{cases} \end{cases}$$

Proof. Since $\tilde{\bar{c}}_i \sim \mathcal{N}(\bar{c}_i(\lambda), V_i^c)$ is a Ra-Ro vector, it follows that $\tilde{\bar{c}}_i^T \boldsymbol{x} \sim \mathcal{N}(\bar{c}_i^T(\lambda)\boldsymbol{x}, \boldsymbol{x}^T V_i^c \boldsymbol{x})$. Then we have

$$\begin{aligned} & Pr\{\tilde{\bar{c}}_{ij}^T \boldsymbol{x} \le f_i\} \ge \beta_i \\ & \Leftrightarrow Pr\left\{\frac{\tilde{\bar{c}}_i(\lambda)^T \boldsymbol{x} - \bar{c}_i(\lambda)^T \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}} \le \frac{f_i - \bar{c}_i(\lambda)^T \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right\} \ge \beta_i \\ & \Leftrightarrow \beta_i \le \Phi\left(\frac{f_i - \bar{c}_i(\lambda)^T \boldsymbol{x}}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right) \\ & \Leftrightarrow \bar{c}_i(\lambda)^T \boldsymbol{x} \le f_i - \Phi^{-1}(\beta_i)\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}} \end{aligned}$$

Let's formulate why $\bar{c}_i(\lambda)^T \boldsymbol{x}$ is a rough variable. From the assumption we know that $\bar{c}_{ij}(\lambda)$ is a rough variable and $\bar{c}_i(\lambda) = (\bar{c}_{i1}(\lambda), \bar{c}_{i2}(\lambda), \ldots, \bar{c}_{in}(\lambda))^T$. We set

$$\begin{aligned} \bar{c}_{ij}(\lambda) &\vdash ([a_{ij}, b_{ij}], [c_{ij}, d_{ij}]), \\ \boldsymbol{x} &= (x_1, x_2, \ldots, x_n)^T. \end{aligned}$$

It follows that $x_j \bar{c}_{ij}(\lambda) \vdash ([x_j a_{ij}, x_j b_{ij}], [x_j c_{ij}, x_j d_{ij}])$,

$$\begin{aligned}\bar{c}_i(\lambda)^T\boldsymbol{x} &= \sum_{j=1}^{n}\bar{c}_{ij}(\lambda)x_j \vdash \sum_{j=1}^{n}([x_j a_{ij}, x_j b_{ij}], [x_j c_{ij}, x_j d_{ij}]) \\ &= \left(\left[\sum_{j=1}^{n} a_{ij}x_j, \sum_{j=1}^{n} a_{ij}x_j\right], \left[\sum_{j=1}^{n} c_{ij}x_j, \sum_{j=1}^{n} d_{ij}x_j\right]\right)\end{aligned}$$

So $\bar{c}_i(\lambda)^T\boldsymbol{x}$ is also a rough variable. Now we set

$$\begin{aligned}a &= \sum_{j=1}^{n} a_{ij}x_j, b = \sum_{j=1}^{n} a_{ij}x_j, \\ c &= \sum_{j=1}^{n} c_{ij}x_j, d = \sum_{j=1}^{n} d_{ij}x_j,\end{aligned}$$

then $\bar{c}_i(\lambda)^T\boldsymbol{x} = ([a,b],[c,d])$. It follows that, for given confidence level α_i and predetermined value f_i,

$$\begin{aligned}&\text{Appr}\{\lambda | Pr\{\tilde{\bar{c}}_i^T\boldsymbol{x} \le f_i\} \ge \beta_i\} \ge \alpha_i \\ &\Leftrightarrow \text{Appr}\left\{\lambda | \bar{c}_i(\lambda)^T\boldsymbol{x} \le f_i - \Phi^{-1}(\beta_i)\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}\right\} \ge \alpha_i \\ &\Leftrightarrow \alpha_i \le \begin{cases} 0, & \text{if} \quad L \le c \\ \dfrac{L-c}{2(d-c)} & \text{if} \quad c \le L \le a \\ \dfrac{1}{2}\left(\dfrac{L-c}{d-c} + \dfrac{L-a}{b-a}\right) & \text{if} \quad a \le L \le b \\ \dfrac{1}{2}\left(\dfrac{L-c}{d-c} + 1\right) & \text{if} \quad b \le L \le d \\ 1 & \text{if} \quad L \le d \end{cases}\end{aligned}$$

where $L = f_i - \Phi^{-1}(\beta_i)\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}$. Because α_i is a given confidence level between 0 and 1, there is no optimal solution for $L \le c$. We can only discuss the following four cases.

(Case 1) $c \le L \le a$. We have

$$\alpha_i \le \frac{L-c}{2(d-c)} \Leftrightarrow \frac{d_i - H_i}{2(d_i - c_i)} \ge \alpha_i \Leftrightarrow 2\alpha_i(d-c) + c \le L.$$

Clearly, $2\alpha_i(d-c) + c \ge c$, thus, the constraint is converted into

$$2\alpha_i(d-c) + c \le L \le a \Leftrightarrow \Phi\left(\frac{f_i - a}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right) \le \beta_i \le \Phi\left(\frac{f_i - c - 2\alpha_i(d-c)}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right).$$

Then the problem (5.50) can be changed into

$$\begin{cases} \max[\beta_1, \beta_2, \ldots, \beta_m] \\ \text{s.t.} \begin{cases} \Phi\left(\dfrac{f_i - a}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right) \le \beta_i, & i = 1, 2, \ldots, m \\ \Phi\left(\dfrac{f_i - c - 2\alpha_i(d-c)}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right) \ge \beta_i, & i = 1, 2, \ldots, m \\ e_r^T \boldsymbol{x} \le b_r, & r = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{5.64}$$

(Case 2) $a \le L \le b$. We have

$$\alpha_i \le \frac{1}{2}\left(\frac{L-c}{d-c} + \frac{L-a}{b-a}\right) \Leftrightarrow L \ge \frac{c(b-a) + a(d-c) - 2\alpha_i(d-c)(b-a)}{b-a+d-c}$$

Let $l = \max\left\{a, \frac{c(b-a)+a(d-c)-2\alpha_i(d-c)(b-a)}{b-a+d-c}\right\}$, then the constraint can be convert into

$$l \le L \le b \Leftrightarrow \Phi\left(\frac{f_i - b}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right) \le \beta_i \le \Phi\left(\frac{f_i - l}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right).$$

Then problem (5.50) can be changed into

$$\begin{cases} \max[\beta_1, \beta_2, \ldots, \beta_m] \\ \text{s.t.} \begin{cases} \Phi\left(\dfrac{f_i - b}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right) \le \beta_i, & i = 1, 2, \ldots, m \\ \Phi\left(\dfrac{f_i - l}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right) \ge \beta_i, & i = 1, 2, \ldots, m \\ e_r^T \boldsymbol{x} \le b_r, & r = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{5.65}$$

where $l = \max\left\{a, \frac{c(b-a)+a(d-c)-2\alpha_i(d-c)(b-a)}{b-a+d-c}\right\}$.

(Case 3) $b \le L \le d$. We have

$$\alpha_i \le \frac{1}{2}\left(\frac{L-c}{d-c} + 1\right) \Leftrightarrow L \ge (2\alpha_i - 1)(d-c) + c.$$

Let $t = \max\{b, (2\alpha_i - 1)(d - c) + c$, then the constraint can be convert into

$$t \le L \le d \Leftrightarrow \Phi\left(\frac{f_i - d}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right) \le \beta_i \le \Phi\left(\frac{f_i - t}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right).$$

Then problem (5.50) can be changed into

$$\begin{cases} \max[\beta_1, \beta_2, \ldots, \beta_m] \\ \text{s.t.} \begin{cases} \Phi\left(\dfrac{f_i - d}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right) \le \beta_i, \ i = 1, 2, \ldots, m \\ \Phi\left(\dfrac{f_i - t}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right) \ge \beta_i, \ i = 1, 2, \ldots, m \\ e_r^T \boldsymbol{x} \le b_r, \qquad r = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{5.66}$$

where $t = \max\{b, (2\alpha_i - 1)(d - c) + c\}$.

(Case 4) $d \le L$. It's obviously correct that $\alpha_i \le 1$ for $\alpha_i \in [0, 1]$. Thus, the constraint is only as follows

$$d \le L \Leftrightarrow \beta_i \le \Phi\left(\frac{f_i - d}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right).$$

Then problem (5.50) can be changed into

$$\begin{cases} \max[\beta_1, \beta_2, \ldots, \beta_m] \\ \text{s.t.} \begin{cases} \Phi\left(\dfrac{f_i - d}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right) \ge \beta_i, \ i = 1, 2, \ldots, m \\ e_r^T \boldsymbol{x} \le b_r, \qquad r = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{5.67}$$

This completes the proof. □

5.5.2.2 The Fuzzy Goal Method

In this section, we take the problem (5.67) as an example to introduce how to use the fuzzy goal method to solve the multi-objective programming problems. As we know, the standard distribution function $\Phi(x)$ is a nonlinear function, so it is difficult to

solve it by usual technique. Hereby we mainly introduce the fuzzy goal method proposed by Sakawa [270] to solve this kind of nonlinear multi-objective programming problems.

Let $H_i(\boldsymbol{x}) = \Phi\left(\frac{f_i - d}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right)$, then the problem (5.67) can be rewritten as,

$$\begin{cases} \max[H_1(\boldsymbol{x}), H_2(\boldsymbol{x}), \ldots, H_m(\boldsymbol{x})] \\ \text{s.t.} \begin{cases} e_r^T \boldsymbol{x} \leq b_r, \ r = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{5.68}$$

Assume that DM (decision maker) have fixed the membership function $\mu_k(H_k(\boldsymbol{x}))$ and given the goal membership function value $\bar{\mu}_k$ $(k = 1, 2, \ldots, m)$. Let's consider the following programming problem,

$$\begin{cases} \max \sum\limits_{k=1}^{m} d_k^- \\ \text{s.t.} \begin{cases} \mu_k(H_k(\boldsymbol{x})) + d_k^- - d_k^+ = \bar{\mu}_k, & k = 1, 2, \ldots, m \\ e_r^T \boldsymbol{x} \leq b_r, & r = 1, 2, \ldots, p \\ d_k^+ d_k^- = 0, d_k^+, d_k^- \geq 0, & k = 1, 2, \ldots, m \\ \boldsymbol{x} \in X \end{cases} \end{cases} \tag{5.69}$$

where d_k^+, d_k^- is the positive and negative deviation. Then we have the following result between the optimal solution of the problem (5.69) and the efficient solution of the problem (5.68).

Theorem 5.16. *(Sakawa [270]) 1. If $\boldsymbol{x}^*$ is the optimal solution of the problem (5.69), and $0 < \mu_k(H_k(\boldsymbol{x}^*)) < 1$, $d_k^+ = 0 (k = 1, 2, \ldots, m)$ holds, then $\boldsymbol{x}^*$ is an efficient solution of the problem (5.68).*

2. If $\boldsymbol{x}^*$ is an efficient solution of the problem (5.68), and $0 < \mu_k(H_k(\boldsymbol{x}^*)) < 1 (k = 1, 2, \ldots, m)$, then $\boldsymbol{x}^*$ is an efficient solution of the problem (5.69) and $d_k^+ = 0 (k = 1, 2, \ldots, m)$ holds.

Proof. The proof process can be found in [270], and we don't introduce it here. □

5.5.3 Nonlinear Ra-Ro DCM and Ra-Ro Simulation-Based RTS

Consider the following nonlinear multi-objective dependent chance model,

$$\begin{cases} \max[\beta_1, \beta_2, \ldots, \beta_m] \\ \text{s.t.} \begin{cases} \text{Appr}\{\lambda | Pr\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) \leq f_i\} \geq \beta_i\} \geq \alpha_i, & i = 1, 2, \ldots, m \\ e_r^T \boldsymbol{x} \leq b_r, & r = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases}$$

where $f_i(\boldsymbol{x}, \boldsymbol{\xi})$ are nonlinear functions with respect to $\boldsymbol{\xi}$, $\boldsymbol{\xi}$ is a Ra-Ro vector, α_i is the given confidence level and f_i is the predetermined value. We cannot usually convert it into a crisp one because of the existence of the nonlinear function $f_i(\boldsymbol{x}, \boldsymbol{\xi})$, so the parametric TS based on the Ra-Ro simulation can be used to deal with the dependent chance model to obtain optimal solution. Next, let's firstly introduce the Ra-Ro simulation for DCM.

5.5.3.1 Ra-Ro Simulation for DCM

Suppose that $\boldsymbol{\xi}$ is an n-dimensional Ra-Ro vector defined on the rough set $(\underline{X}, \bar{X})$ under the similarity relationship, and $f : \mathbf{R}^n \to \mathbf{R}$ is a measurable function. For any real number $\alpha \in (0, 1]$, we design a Ra-Ro simulation to compute the α-chance $Ch\{f(\boldsymbol{x}, \boldsymbol{\xi}) \leq f\}(\alpha)$ for given $\boldsymbol{x}$. That is, we should find the supremum $\bar{\beta}$ such that

$$\text{Appr}\{\lambda | Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}) \leq f\} \geq \bar{\beta}\} \geq \alpha \tag{5.70}$$

Generate $\underline{\lambda}_1, \underline{\lambda}_2, \ldots, \underline{\lambda}_N$ from the lower approximation $\underline{X}$ and $\bar{\lambda}_1, \bar{\lambda}_2, \ldots, \bar{\lambda}_N$ from the upper approximation $\overline{X}$ according to the approximation function Appr. For any number t, let $\underline{N}(t)$ denote the number of $\underline{\lambda}_k$ satisfying $Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}(\underline{\lambda}_k)) \leq f\} \geq t$ for $k = 1, 2, \ldots, N$, and $\bar{N}(t)$ denote the number of $\bar{\lambda}_k$ satisfying $Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}(\bar{\lambda}_k)) \leq f\} \geq t$ for $k = 1, 2, \ldots, N$, where $Pr\{\cdot\}$ may be estimated by random simulation. Then we may find the maximal value v such that

$$\frac{\underline{N}(t) + \bar{N}(t)}{2N} \geq \alpha \tag{5.71}$$

This value is an estimation of $\bar{\beta}$. Then the procedure simulating the critical value $\bar{f}$ of $\text{Appr}\{\lambda | Pr\{f(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}\} \geq \beta\} \geq \alpha$ can be summarized as follows:

Procedure `Ra-Ro simulation for DCM`

Input: The decision vector $\boldsymbol{x}$
Output: α-chance $Ch\{f(\boldsymbol{x}, \boldsymbol{\xi}) \leq f\}(\alpha)$
Step 1. Generate $\underline{\lambda}_1, \underline{\lambda}_2, \ldots, \underline{\lambda}_N$ from the lower approximation $\underline{X}$ according to the approximation function Appr;
Step 2. Generate $\bar{\lambda}_1, \bar{\lambda}_2, \ldots, \bar{\lambda}_N$ from the upper approximation $\overline{X}$ according to the approximation function Appr;
Step 3. Find the maximal value t such that $\frac{\underline{N}(t)+\bar{N}(t)}{2N} \geq \alpha$ holds;
Step 4. Return t.

Example 5.14. Let $\tilde{\tilde{\xi}}_1$, $\tilde{\tilde{\xi}}_2$ and $\tilde{\tilde{\xi}}_3$ are three Ra-Ro variables as follows,

$$\begin{aligned}
&\tilde{\tilde{\xi}}_1 \sim \mathscr{U}(\tilde{\rho}_1, \tilde{\rho}_1 + 2), \text{ with } \tilde{\rho}_1 = ([1, 2], [1, 3]),\\
&\tilde{\tilde{\xi}}_2 \sim \mathscr{N}(\tilde{\rho}_2, 1), \qquad \text{with } \tilde{\rho}_2 = ([0, 1], [0, 3]),\\
&\tilde{\tilde{\xi}}_3 \sim exp(\tilde{\rho}_3), \qquad \text{with } \tilde{\rho}_3 = ([1, 2], [0, 3]),
\end{aligned}$$

A run of Ra-Ro simulation with 5000 cycles shows that

$$Ch\left\{\sqrt{\tilde{\xi}_1^2+\tilde{\xi}_2^2+\tilde{\xi}_3^2}\leq 2.6\right\}(0.9)=0.5.$$

5.5.3.2 Reactive Tabu Search Algorithm

The reactive tabu search (abbr. RTS) is an improved version of TS. It was first proposed by Battiti and Tecchiolli [18]. The tabu list length of RTS can be self-adapted to balance intensification and diversification. Furthermore, an escape mechanism is introduced to avoid recycling. The RTS method has been successfully applied in many fields [28, 38, 290].

The reactive tabu search (RTS) algorithm, goes further in the direction of robustness by proposing a simple mechanism for adapting the list size to the properties of the optimization problem. The configurations visited during the search and the corresponding iteration numbers are stored in memory so that, after the last movement is chosen, one can check for the repetition of configurations and calculate the interval between two visits. The basic fast reaction mechanism increases the list size when configurations are repeated. This is accompanied by a slower reduction mechanism so that the size is reduced in regions of the search space that do not need large sizes.

The reactive tabu search algorithm uses the reactive mechanism to adjust the length of the tabu list as well as balance the centralized strengthening search strategy and decentralized diversification search strategy. The RTS involves in the increasing adjustment coefficient ($N_{IN} > 1$) and decreasing adjustment coefficient ($0 < N_{DE} < 1$). In the searching process, all the solutions visited are stored. As operating a step of move, current solution is checked if it has been visited. If it has been visited, which shows that entering a cycle, then the length of tabu list become N_{IN} times of the original length. If there are not repeated solutions after several iterations, then the length of tabu list become N_{DE} times of the original length.

In order to cycle, RST presents the escape mechanism. In the searching process, when repeated times of a large number of repeated solutions exceed the given times R_{EP}, the escape mechanism is activated. The escape strategy is based on the execution of a series of random exchanges. Their number is random. To avoid an immediate return into the old region of the search space, all random steps executed are made tabu.

Tabu search algorithm uses historical memory optimization, search optimization with tabu list that, combined with the level of desire, the system achieved through an intensive search and distributed search for the balance of diversification. The RTS use active feedback strategies and use the escape mechanism to strengthen the balance. Thus, in theory, tabu search algorithm active than the general tabu search algorithm better, search for higher quality.

The key idea of RTS is feedback strategies and escape mechanism. In the real application, there are many ways to achieve feedback strategies and escape

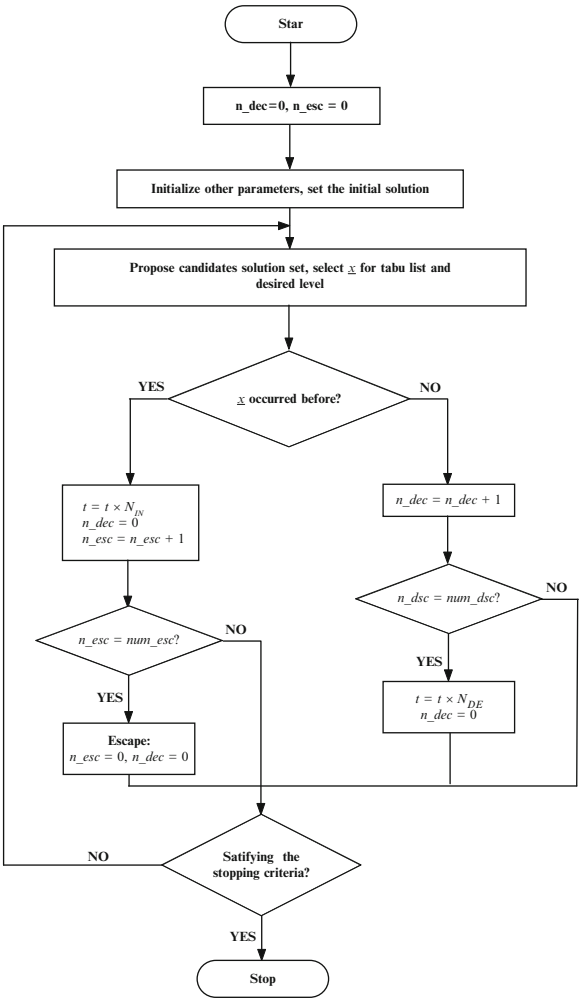


Fig. 5.18 Flowchart of RTS algorithm

mechanism. For example, then length of tabu list become the original list num_dec times if there is no repeated solutions in num_dec iteration. Escape mechanism is implemented if the times of repetition achieve num_esc.

The basic procedures of RTS are summarized as follows and the flowchart is shown by Fig. 5.18.

Step 1. Initialize two counters: $num\_dec = num\_esc = 0$.

Step 2. Initialize other parameters, present the initial solution.

Step 3. Propose candidate solutions set according to current solution.

Step 4. According to the tabu list situation and desired level, select a solution as the initial solution for next iteration update record list (including the tabu list, and all the normal solutions visited).

Step 5. If the selected solution occurred before, then the length of the tabu list $t = tN_{IN}, n\_esc = n\_esc + 1, n\_dec = 0$; otherwise $n\_dec = n\_dec + 1$.

Step 6. If $n\_dec = num\_dec$, then $t = tN_{DE}, n\_dec = 0$.

Step 7. if $n\_esc = num\_esc$, then implement escape mechanism, n_esc=0, $n\_dec = 0$.

Step 8. If the termination criterion is satisfied, stop; otherwise turn to **Step 3**.

5.5.4 Numerical Examples

Example 5.15. Consider the following multiobjective programming problem

$$\begin{cases} \max f_1(\boldsymbol{x}, \boldsymbol{\xi}) = Ch\{\tilde{\tilde{\xi}}_1 x_1 + \tilde{\tilde{\xi}}_2 x_2 + \tilde{\tilde{\xi}}_3 x_3 \geq f_1\}(\alpha) \\ \max f_2(\boldsymbol{x}, \boldsymbol{\xi}) = Ch\{c_1\tilde{\tilde{\xi}}_4 x_1 + c_2\tilde{\tilde{\xi}}_5 x_2 + c_3\tilde{\tilde{\xi}}_6 x_3 \geq f_2\}(\beta) \\ \text{s.t.} \begin{cases} x_1 + x_2 + x_3 \leq 15 \\ x_1 + x_2 + x_3 \geq 10 \\ x_1 + 4x_2 + 2x_3 \leq 30 \\ 2 \leq x_1, x_2, x_3 \leq 6 \end{cases} \end{cases} \tag{5.72}$$

where $c = (c_1, c_2, c_3) = (1.2, 0.8, 1.5)$,

$\tilde{\tilde{\xi}}_1 \sim \mathcal{N}(\rho_1, 1)$, with $\rho_1 \vdash ([1,2],[0,3])$, $\tilde{\tilde{\xi}}_2 \sim \mathcal{N}(\rho_2, 4)$, with $\rho_2 \vdash ([2,3],[1,4])$,
$\tilde{\tilde{\xi}}_3 \sim \mathcal{N}(\rho_3, 1)$, with $\rho_3 \vdash ([3,4],[2,5])$, $\tilde{\tilde{\xi}}_4 \sim \mathcal{N}(\rho_4, 2)$, with $\rho_4 \vdash ([0,1],[0,3])$,
$\tilde{\tilde{\xi}}_5 \sim \mathcal{N}(\rho_5, 1)$, with $\rho_5 \vdash ([1,2],[0,3])$, $\tilde{\tilde{\xi}}_6 \sim \mathcal{N}(\rho_6, 1)$, with $\rho_6 \vdash ([2,3],[0,3])$,

and $\rho_i (i = 1, 2, \ldots, 6)$ are rough variables. We set $\alpha = \beta = 0.9$, $f_1 = 15$, and $f_2 = 13$. Then we have the following equivalent model,

$$\begin{cases} \max[\alpha_0, \beta_0] \\ \text{s.t.} \begin{cases} \Phi\left(\dfrac{15 - (5.4x_1 + 6.4x_2 + 7.4x_3)}{\sqrt{x_1^2 + 4x_2^2 + x_3^2}}\right) \geq \alpha_0 \\ \Phi\left(\dfrac{13 - 1.8(3.6x_1 + 2.4x_2 + 4.5x_3)}{\sqrt{2x_1^2 + x_2^2 + x_3^2}}\right) \geq \beta_0 \\ x_1 + x_2 + x_3 \leq 15 \\ x_1 + x_2 + x_3 \geq 10 \\ x_1 + 4x_2 + 2x_3 \leq 30 \\ 2 \leq x_1, x_2, x_3 \leq 6 \end{cases} \end{cases} \tag{5.73}$$

In fact, since the function $\Phi(x)$ is monotonously increasing, we can get the optimal solution of problem (5.73) by solving the equivalent problem as follows,

$$\begin{cases} \max H_1(\boldsymbol{x}) = \dfrac{15-(5.4x_1+6.4x_2+7.4x_3)}{\sqrt{x_1^2+4x_2^2+x_3^2}} \\ \max H_2(\boldsymbol{x}) = \dfrac{13-1.8(3.6x_1+2.4x_2+4.5x_3)}{\sqrt{2x_1^2+x_2^2+x_3^2}} \\ \text{s.t.} \begin{cases} x_1+x_2+x_3 \le 15 \\ x_1+x_2+x_3 \ge 10 \\ x_1+4x_2+2x_3 \le 30 \\ 2 \le x_1, x_2, x_3 \le 6 \end{cases} \end{cases} \tag{5.74}$$

Then we use the fuzzy goal programming method to solve the above problem. Firstly, construct the fuzzy membership function by the following equation,

$$\mu_k(\boldsymbol{x}) = \frac{H_k(\boldsymbol{x}) - H_k^0}{H_k^1 - H_k^0},\ k = 1, 2,$$

where H_k^1 and H_k^0 can respectively obtained by $H_k^1 = \max_{x \in X} H_k(\boldsymbol{x})$ and $H_k^0 = \min_{x \in X} H_k(\boldsymbol{x})$ as follows,

$$H_1^1 = -3.974,\ H_1^0 = -7.952,\ H_2^1 = -6.074,\ H_2^0 = -8.636.$$

Secondly, let $\bar{\mu}_1 = \bar{\mu} = 0.9$, then the fuzzy goal programming problem can be got as follows,

$$\begin{cases} \max(d_1^- + d_2^-) \\ \text{s.t.} \begin{cases} \dfrac{\dfrac{15-(5.4x_1+6.4x_2+7.4x_3)}{\sqrt{x_1^2+4x_2^2+x_3^2}} + 7.952}{7.952-3.974} + d_1^- - d_1^+ = 0.9 \\ \dfrac{\dfrac{13-1.8(3.6x_1+2.4x_2+4.5x_3)}{\sqrt{2x_1^2+x_2^2+x_3^2}} + 8.636}{8.636-6.074} + d_2^- - d_2^+ = 0.9 \\ x_1+x_2+x_3 \le 15 \\ x_1+x_2+x_3 \ge 10 \\ x_1+4x_2+2x_3 \le 30 \\ 2 \le x_1, x_2, x_3 \le 6 \\ d_1^+, d_1^-, d_2^+, d_2^- \ge 0 \end{cases} \end{cases} \tag{5.75}$$

We obtain the optimal solution $x^* = (2.00, 6.00, 2.00)^T$. Thus, the optimal objective value $(\alpha_0^*, \beta_0^*) = (0.9673, 0.7692)$.

Example 5.16. Consider the following problem,

$$\begin{cases} \max f_1(x) = Ch\{\tilde{\tilde{\xi}}_1 x_1 + \tilde{\tilde{\xi}}_2 x_2 + \tilde{\tilde{\xi}}_3 x_3 \geq f_1\}(\alpha) \\ \max f_2(x) = Ch\{c_1\tilde{\tilde{\xi}}_4 x_1 + c_2\tilde{\tilde{\xi}}_5 x_2 + c_3\tilde{\tilde{\xi}}_6 x_3 \geq f_2\}(\beta) \\ \text{s.t.} \begin{cases} x_1 + x_2 + x_3 \leq 250, \\ x_1 + x_2 + x_3 \geq 200, \\ x_1 + 4x_2 + 2x_3 \leq 600, \\ x_1 \geq 20, x_2 \geq 20, x_3 \geq 20, \end{cases} \end{cases} \tag{5.76}$$

where $c = (c_1, c_2, c_3) = (1.2, 0.8, 1.5)$,

$\tilde{\tilde{\xi}}_1 \sim \mathcal{N}(\rho_1, 1)$, with $\rho_1 \vdash ([1,2],[0,3])$, $\tilde{\tilde{\xi}}_2 \sim \mathcal{N}(\rho_2, 4)$, with $\rho_2 \vdash ([2,3],[1,4])$,
$\tilde{\tilde{\xi}}_3 \sim \mathcal{N}(\rho_3, 1)$, with $\rho_3 \vdash ([3,4],[2,5])$, $\tilde{\tilde{\xi}}_4 \sim \mathcal{N}(\rho_4, 2)$, with $\rho_4 \vdash ([0,1],[0,3])$,
$\tilde{\tilde{\xi}}_5 \sim \mathcal{N}(\rho_5, 1)$, with $\rho_5 \vdash ([1,2],[0,3])$, $\tilde{\tilde{\xi}}_6 \sim \mathcal{N}(\rho_6, 1)$, with $\rho_6 \vdash ([2,3],[0,3])$,

and $\rho_i (i = 1, 2, \ldots, 6)$ are rough variables. We set $\alpha = \beta = 0.9$, $f_1 = 1500$, and $f_2 = 1300$.

Next, we apply the tabu search algorithm based on the Ra-Ro simulation to solve the nonlinear programming problem (5.41) with the Ra-Ro parameters.

Step 1. Set the move step $h = 0.5$ and the h neighbor $N(\boldsymbol{x}, h)$ for the present point $\boldsymbol{x}$ is defined as follows,

$$N(\boldsymbol{x}, h) = \left\{ \boldsymbol{y} | \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2} \leq h \right\}.$$

The random move of point $\boldsymbol{x}$ to point $\boldsymbol{y}$ in its h neighbor along direction s is given by

$$\boldsymbol{y}_s = \boldsymbol{x}_s + rh,$$

where r is a random number that belongs to [0,1], $s = 1, 2, 3$.

Step 2. Give the step set $H = \{h_1, h_2, \ldots, h_r\}$ and randomly generate a feasible point $\boldsymbol{x}_0 \in X$. One should empty the Tabu list T (the list of inactive steps) at the beginning.

Step 3. For each active neighbor $N(\boldsymbol{x}, h)$ of the present point $\boldsymbol{x}$, where $h \in H - T$, a feasible random move that satisfies all the constraints in problem (5.41) is to be generated.

Step 4. Construct the single objective function as follows,

$$\begin{aligned} f(\boldsymbol{x}, \boldsymbol{\xi}) = {} & w_1 Ch\{\tilde{\tilde{\xi}}_1 x_1 + \tilde{\tilde{\xi}}_2 x_2 + \tilde{\tilde{\xi}}_3 x_3 \geq f_1\}(\alpha) \\ & + w_2 Ch\{c_1\tilde{\tilde{\xi}}_4 x_1 + c_2\tilde{\tilde{\xi}}_5 x_2 + c_3\tilde{\tilde{\xi}}_6 x_3 \geq f_2\}(\beta) \end{aligned}$$

Table 5.6 The result computed by RTS algorithm

ω_1	ω_2	x_1	x_2	x_3	$f_1(x)$	$f_2(x)$
0.1	0.9	20.00	45.36	184.64	0.632	0.714
0.2	0.8	20.00	39.72	190.28	0.668	0.702
0.3	0.7	20.00	34.39	195.71	0.691	0.688
0.4	0.6	20.00	31.06	198.94	0.711	0.672
0.5	0.5	20.00	26.14	203.86	0.736	0.669

where $w_1 + w_2 = 1$. Compare the $f(\boldsymbol{x}, \xi)$ of the feasible moves with that of the current solution by the Ra-Ro simulation. If an augmenter in new objective function of the feasible moves exists, one should save this feasible move as the updated current one by adding the corresponding step to the Tabu list T and go to the next step; otherwise, go to the next step directly.

Step 5. Stop if the termination criteria are satisfied; other wise, empty T if it is full; then go to Step 3. Here, we set the computation is determined if the better solution doesn't change again (Table 5.16).

5.6 The Inventory System of Wal–Mart Supermarket

Classical inventory models generally deal with a single-item. But in real word situations, a single-item inventory seldom occurs. It is a common experience that the presence of a second item in an inventory favors the demand of the first and vice-versa; the effect may be different in the two cases. This is why companies and retailers deal with several items and stock them in their showrooms/warehouses. This leads to the idea of a multi-item inventory.

Many well-known books [123, 230] that introduce multi-item classical inventory models under resource constraints are available. In a multi-item inventory system, companies and retailers are required to maximize/minimize two or more objectives simultaneously over a given set of decision variables. This leads to the idea of a multi-objective mathematical programming problem. Toroslu and Arslanoglu [319] research a genetic algorithm for the personnel assignment problem with multiple objectives. The inventory system of supermarkets is a typical example of the multi-item inventory system. A supermarket contains many different goods and needs to determine how many of each should be ordered.

Generally, in classical inventory models, many parameters are assumed to be crisp, such as holding and set-up costs, demand, rate of replenishment/production and shortage costs and so on. However, in real-life situations, there are many uncertain factors leading to some imprecise parameters in inventory problems. Recently demand has been considered a fuzzy variable by some scholars [153, 198]. Ishii and Konno [145] considered shortage cost as a fuzzy number and demand as a random

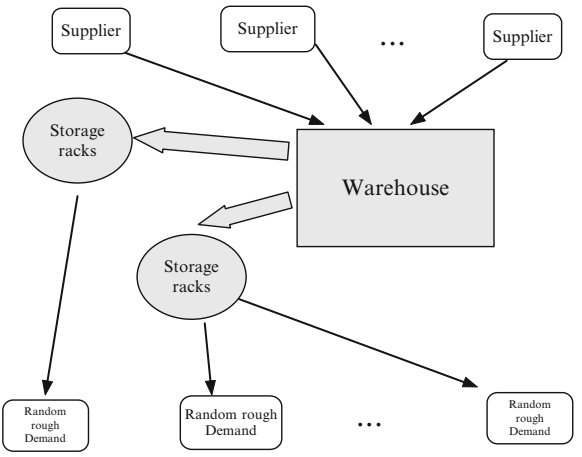


Fig. 5.19 Wal–Mart inventory system

variable in the classical newsbody problem. Then a single-period inventory problem with fuzziness and randomness simultaneously has been researched by Dutta, Chakraborty and Roy [92]. However, there has been no attempt to research another mixed environment, where randomness and roughness both appear simultaneously. For some seasonal items (Ice cream, Christmas trees, woolen materials), the demand may vary year to year. According to historical data or abundance of information, we can know demand in one year is subject to stochastic distribution. However, the expected value of stochastic distribution is vague and varies from year to year. Thus, it is difficult for decision makers to achieve a better decision. Hence, we have to consider it an uncertain variable. A rough variable can be applied to depict it well if the average sold amount is clear by the statistical data of each year. Thus, the demand of some seasonal items can be described as a Ra-Ro variable to help decision makers develop better strategies.

5.6.1 Background

In this section, an example from Wal-Mart supermarket inventory system is introduced. Sam's Gamble is a global Company with more than 1.8 million associates worldwide and nearly 6,500 stores and wholesale clubs across 15 countries. By August 31, 2006, the company had 1,135 Wal–Mart stores 2,121 Super-centers, 567 SAM'S CLUBS and 108 Neighborhood Markets in the United States. Internationally, the Company operated units in Argentina (12), Brazil (294), Canada (278), China (63), Costa Rica (133), Germany (85), Guatemala (122), Honduras (37), Japan (391), Mexico (828),Nicaragua (36),Puerto Rico (54), El Salvador (59), South Korea(16) and the United Kingdom (323). The most admired retailer according to FORTUNE magazine has just completed one of the best years in its history:

Wal–Mart generated more than 312.4 billion in global revenue in the fiscal year ended January 31, 2006.

To maintain and increase the percentage of sales, Wal–Mart stores sell products with their private-label to balance the pressure from global competitors. Now, the company plans to earn a handsome profit along with sufficiently large sales proceeds. Here, we suppose that the managers of the company are interested in eight new commodities, *reading lamp*, *chocolate*, *drink*, *hair drier*, *suitcase*, *cosmetic* and *washer*, and want these new products to be on sale in their supermarkets with the channel-firm label manufactured by other producers. Even though the eight new products are produced by eight different manufacturing companies and the purchasing (cost) price of each product is fixed. But the managers still need to determine order quantities and how to set a selling (retail) price for the eight new products, because they are all new products and have never been marketed. Due to many uncertain factors, the total average profit (PF), the total average cost (TC), the total wastage cost (WC), the selling or purchasing price and holding or set-up cost per product (not yet been finalized but) are expected to be Ra-Ro variables. For example, the demand for the *reading lamp* is a random variable following the normal distribution denoted by $\mathscr{N}(165, 10^2)$. We assume that the demand for the new *reading lamp* should also be a normally distributed variable, denoted by $\mathscr{N}(\tilde{\mu}, 5^2)$ based on the demand of the other congeneric commodities, where $\tilde{\mu} = ([100, 146], [80, 165])$ is a rough variable. Thus the demand for the new *reading lamp* is a Ra-Ro variable. That is the same for the other new productions. Here, based on the statistical data of similar commodities, we can know all the distribution of every commodity. Then we can deal with the eight new commodity inventory problem.

5.6.2 Assumptions

The complexity arises in modelling a realistic decision-making inventory situation is mainly due to the presence of some non-deterministic information, in the sense that they are not just capable of being encoded with the precision and certainty of classical mathematical reasoning. Actually, a realistic situation is no longer realistic when imprecise and uncertain information is neglected for the sake of mathematical model building. During the last three decades, considerable developments in the field of operations research has enabled theories of probabilistic and fuzzy sets to open ways for constructing metaphors that represents many aspects of uncertainty and impreciseness. These theories have been extensively applied to model decision-making situations in their respective environments. Generally, 'imprecision' and 'uncertainty' are modelled by fuzzy and stochastic approaches, respectively. Now, the existence of a mixed environment or the coexistence of imprecision and uncertainty in an inventory model is again a realistic phenomenon and the mathematical realization of this fact is an interesting field. Here, the extension of chance constrained programming to a random rough environment has been

investigated through an inventory model. We have developed an inventory model in a mixed environment where some constraints are imprecisely defined, and others probabilistically.

Realistically, a retailer/owner of a factory starts the business/production with a fixed amount of cash in hand to purchase the items/materials and a wherehouse of finite area to store the items/products. But, in the course of business/production, the retailer augments the said capital by some amount in the interest of the business, if the situation demands. Similarly, to avail of a certain transport facility/concession or to capitalize on some production situation, the items may be replenished/produced more than the capacity of the available warehouse and in that case, the owner manages to find some additional storage space, if it is required and situation is so called for. Hence, for real-life inventory problems, both budgetary amounts and storage space are not defined with certainty i.e. may be uncertain in a stochastic or non-stochastic (imprecise) sense.

5.6.3 Mathematical Model

To develop the proposed model, we adopt the following notations:

n	Number of items
A	Available floor/storage space
B	Available total budgetary cost
Q_i	Order level (decision variable)
s_i	Selling price of each product
p_i	Purchase price of each product
h_i	Holding cost per unit item per unit time
u_i	Set up cost per cycle
a_i	Constant rate of deterioration,$0 < a_i < 1$
A_i	Required storage area per unit quantity
T_i	Time period for each cycle
$q_i(t)$	Inventory level at time t
$Z_i(Q_i)$	Average profit of the ith item
$D_i(q_i)$	Quantity of demand at time$t$, $D_i(q_i) = b_i + c_i q_i(t)$(where b_i and c_i being constant, $0 < c_i < 1$)
$TC_i(Q_i)$	Total average cost of the ith item
$PF(Q_i)$	Total average profit$PF(Q_i) = \sum_{i=1}^{n} Z_i(Q_i)$
$WC(Q_i)$	Total cost of the ith item

Here, $D = (D_1, D_2, \ldots, D_n)^T$, $Q = (Q_1, Q_2, \ldots, Q_n)^T$. If $q_i(t)$ is the inventory level at time t of the ith, then $\frac{dq_i}{dt} = -D_i - a_i q_i$. Thus, the length of the cycle of the ith item is

$$T_i = \int_0^{Q_i} \frac{dq_i}{D_i + a_i q_i} = \int_0^{Q_i} \frac{dq_i}{b_i + (a_i + c_i)q_i} = \frac{1}{a_i + c_i} \ln\left(\frac{b_i + (a_i + c_i)Q_i}{b_i}\right).$$

The holding cost in each cycle for the ith item is $h_i g_i(Q_i)$, where

$$g_i(Q_i) = q_i T_i = \int_0^{Q_i} \frac{q_i dq_i}{b_i + (c_i + a_i)q_i} = \frac{Q_i}{a_i + c_i} - \frac{D_i}{(a_i + c_i)^2} \ln\left(\frac{b_i + (a_i + c_i)Q_i}{b_i}\right).$$

When the demand is $D_i(q_i)$ for the ith item, the real amount of sold items is

$$\min\{D_i, Q_i\} = \begin{cases} D_i, \text{ if } & D_i < Q_i \\ Q_i, \text{ if } & D_i \geq Q_i \end{cases}$$

The total number of deteriorating units of ith item is $\theta_i(Q_i) = a_i g_i(Q_i)$. The net revenue of the ith item is $N(Q_i) = (s_i - p_i)Q_i - s_i\theta_i(Q_i)$. Hence, total average profit of the ith item is

$$PF(Q_i) = \sum_{i=1}^{n} [N(Q_i) - h_i g_i(Q_i) - u_i]/T_i.$$

The total cost $WC(Q_i)(i = 1, 2, \ldots, n)$ of every item can be calculated by the following formula,

$$WC(Q_i) = \sum_{i=1}^{n} \theta_i(Q_i) p_i / T_i$$

Total average cost of the ith item is

$$TC_i(Q_i) = [p_i Q_i + h_i g_i(Q_i) + u_i]/T_i.$$

Usually, the demand is not a certain number, and it will vary according to the season or the country policy. When they are random rough variables, we assume that they are Ra-Ro variables, they have a log-normal distribution and its probability density function (see Fig. 5.20) is

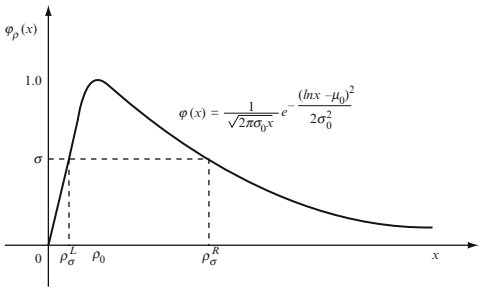


Fig. 5.20 The probability density function of the random variable ρ

$$\phi(x) = \frac{1}{\sqrt{2\pi}\sigma_0 x} e^{-\frac{(\ln x - \tilde{\mu}_0)^2}{2\sigma_0^2}}$$

where $\tilde{\mu}_0$ is a rough variable. If the decision maker wants to maximize the total average profit and minimize the total average cost, the problem can be formulated by the following model:

$$\begin{cases} \max PF(Q_i) = \sum_{i=1}^{n} [\tilde{\tilde{N}}(Q_i) - \tilde{\tilde{h}}_i g_i(Q_i) - \tilde{\tilde{u}}_i]/T_i \\ \min WC(Q_i) = \sum_{i=1}^{n} \theta_i(Q_i) p_i / T_i \\ \text{s.t.} \begin{cases} \sum_{i=1}^{n} T\tilde{\tilde{C}}_i(Q_i) \le B \\ \sum_{i=1}^{n} A_i Q_i \le A \\ Q_i \ge 0 \end{cases} \end{cases} \tag{5.77}$$

where $\tilde{\tilde{N}}(Q_i) = (\tilde{\tilde{s}}_i - \tilde{\tilde{p}}_i)Q_i - \tilde{\tilde{s}}_i \theta_i(Q_i)$, $T\tilde{\tilde{C}}_i(Q_i) = [\tilde{\tilde{p}}_i Q_i + \tilde{\tilde{h}}_i g_i(Q_i) + \tilde{\tilde{u}}_i]/T_i$.

Next, we consider the inventory model of Wal–Mart. Assume the wholesaler provides five kinds of ice cream wholesale and needs to develop an ordering strategy. According to last decade's statistics, the demand for every kind of ice cream is subject to exponential distribution, but the average demand every year is different. We can apply the rough variable to describe the expected value of demand. Take the maximum and minimum average demand in the last decade respectively as the upper and lower bounds of the rough variable. The two middle demand amounts can be taken as values of the rough variable. The total budgetary cost of order and available storage space are $B = 30{,}000$, $K = 300$. The other parameters can be seen in Table 5.8. How should the wholesaler make the order plan to get maximum probability that the total budgetary cost of the ith ice cream is less than the predetermined fee f_i? (Table 5.7).

Table 5.7 The different parameters in the inventory system

Group	Item	Parametrics							
		$\tilde{\tilde{s}}_i$	$\tilde{\tilde{p}}_i$	$\tilde{\tilde{h}}_i$	$\tilde{\tilde{u}}_i$	a_i	b_i	c_i	A_i
Pickle in bottle	Xin Fan Jun	$N(\bar{\mu}_{s1}, 0.5)$	$N(\bar{\mu}_{p1}, 1)$	$N(\bar{\mu}_{h1}, 0.05)$	$N(\bar{\mu}_{u1}, 0.5)$	0.1	60	0.3	0.5
	Chu Da Ge	$N(\bar{\mu}_{s2}, 0.3)$	$N(\bar{\mu}_{p2}, 0.5)$	$N(\bar{\mu}_{h2}, 0.02)$	$N(\bar{\mu}_{u2}, 0.3)$	0.1	65	0.3	0.5
	Baijia Pickles	$N(\bar{\mu}_{s3}, 0.1)$	$N(\bar{\mu}_{p3}, 0.5)$	$N(\bar{\mu}_{h3}, 0.05)$	$N(\bar{\mu}_{u3}, 0.1)$	0.15	78	0.5	0.5
SOYA OIL	Salad Oil Bottle	$N(\bar{\mu}_{s4}, 1)$	$N(\bar{\mu}_{p4}, 2)$	$N(\bar{\mu}_{h4}, 0.5)$	$N(\bar{\mu}_{u4}, 1)$	0.05	110	0.35	1
	Hong Qingting Bean Oil	$N(\bar{\mu}_{s5}, 1.2)$	$N(\bar{\mu}_{p5}, 2)$	$N(\bar{\mu}_{h5}, 0.5)$	$N(\bar{\mu}_{u5}, 0.3)$	0.05	135	0.35	1
	Wuhu soya bean oil	$N(\bar{\mu}_{s6}, 1)$	$N(\bar{\mu}_{p6}, 1)$	$N(\bar{\mu}_{h6}, 0.5)$	$N(\bar{\mu}_{u6}, 0.3)$	0.05	120	0.35	1
CHILLI KETCHUP	Tan Yutou sauce	$N(\bar{\mu}_{s7}, 0.5)$	$N(\bar{\mu}_{p7}, 0.4)$	$N(\bar{\mu}_{h7}, 0.05)$	$N(\bar{\mu}_{u7}, 0.8)$	0.3	230	0.55	0.5
	Mother Beef Sauce	$N(\bar{\mu}_{s8}, 0.8)$	$N(\bar{\mu}_{p8}, 1)$	$N(\bar{\mu}_{h8}, 0.2)$	$N(\bar{\mu}_{u8}, 0.1)$	0.3	235	0.55	0.5
	Baohua Bottle	$N(\bar{\mu}_{s9}, 1)$	$N(\bar{\mu}_{p9}, 0.8)$	$N(\bar{\mu}_{h9}, 0.2)$	$N(\bar{\mu}_{u9}, 0.3)$	0.3	180	0.55	0.5
GLUTAMATE	Guo Tai	$N(\bar{\mu}_{s10}, 0.1)$	$N(\bar{\mu}_{p10}, 0.2)$	$N(\bar{\mu}_{h10}, 0.05)$	$N(\bar{\mu}_{u10}, 0.1)$	0.05	150	0.35	0.1
	Hao Ji	$N(\bar{\mu}_{s11}, 0.2)$	$N(\bar{\mu}_{p11}, 0.3)$	$N(\bar{\mu}_{h11}, 0.02)$	$N(\bar{\mu}_{u11}, 0.1)$	0.05	135	0.35	0.1
	Tai Tai	$N(\bar{\mu}_{s12}, 0.5)$	$N(\bar{\mu}_{p12}, 0.2)$	$N(\bar{\mu}_{h12}, 0.02)$	$N(\bar{\mu}_{u12}, 0.1)$	0.05	180	0.35	0.1
MEAT IN CAN	Pork Luncheon Meat Square	$N(\bar{\mu}_{s13}, 1)$	$N(\bar{\mu}_{p13}, 1.2)$	$N(\bar{\mu}_{h13}, 1)$	$N(\bar{\mu}_{u13}, 2)$	0.4	85	0.35	1
	Qinhuangdao lunch meat	$N(\bar{\mu}_{s14}, 1)$	$N(\bar{\mu}_{p14}, 1)$	$N(\bar{\mu}_{h14}, 1)$	$N(\bar{\mu}_{u14}, 2)$	0.4	100	0.35	1
	Yuehua lunch meat	$N(\bar{\mu}_{s15}, 1.3)$	$N(\bar{\mu}_{p15}, 1.2)$	$N(\bar{\mu}_{h15}, 1)$	$N(\bar{\mu}_{u12}, 2)$	0.4	76	0.35	1
INSTANT ND CP	Kangshifu noodles	$N(\bar{\mu}_{s16}, 0.6)$	$N(\bar{\mu}_{p16}, 0.6)$	$N(\bar{\mu}_{h16}, 0.2)$	$N(\bar{\mu}_{u16}, 0.5)$	0.1	350	0.35	0.5
	President noodles	$N(\bar{\mu}_{s17}, 0.5)$	$N(\bar{\mu}_{p17}, 0.5)$	$N(\bar{\mu}_{h17}, 0.2)$	$N(\bar{\mu}_{u17}, 0.5)$	0.1	360	0.35	0.5
	HuaLong noodles	$N(\bar{\mu}_{s18}, 0.3)$	$N(\bar{\mu}_{p18}, 0.3)$	$N(\bar{\mu}_{h18}, 0.2)$	$N(\bar{\mu}_{u18}, 0.5)$	0.1	340	0.35	0.5
FOOD FOR CATS	Naughty Cat	$N(\bar{\mu}_{s19}, 0.3)$	$N(\bar{\mu}_{p19}, 0.3)$	$N(\bar{\mu}_{h19}, 0.1)$	$N(\bar{\mu}_{u19}, 0.2)$	0.05	78	0.35	1
	Friskies Fish for cat	$N(\bar{\mu}_{s20}, 0.5)$	$N(\bar{\mu}_{p20}, 0.5)$	$N(\bar{\mu}_{h20}, 0.2)$	$N(\bar{\mu}_{u20}, 0.2)$	0.05	60	0.35	1
	Whiskas Salmon	$N(\bar{\mu}_{s21}, 0.3)$	$N(\bar{\mu}_{p21}, 0.3)$	$N(\bar{\mu}_{h21}, 0.1)$	$N(\bar{\mu}_{u21}, 0.2)$	0.05	85	0.35	1

Table 5.8 The rough parameters

Rough variable	$\bar{\mu}_{si}$	$\bar{\mu}_{pi}$	$\bar{\mu}_{hi}$	$\bar{\mu}_{ui}$
1	([5.40,5.95],[5.30,6.00])	([5.00,5.35],[4.90,5.40])	([0.10,0.18],[0.08,0.20])	([0.60,0.80],[0.45,0.90])
2	([6.60,6.80],[6.50,7.00])	([5.90,6.25],[5.80,6.40])	([0.15,0.18],[0.12,0.20])	([0.60,0.80],[0.45,0.90])
3	([5.40,5.80],[5.35,6.00])	([4.95,5.15],[4.80,5.20])	([0.14,0.15],[0.12,0.18])	([0.60,0.80],[0.45,0.90])
4	([23.60,24.10],[23.00,24.30])	([21.60,21.95],[20.50,22.00])	([0.55,0.65],[0.50,0.70])	([0.80,0.90],[0.65,1.00])
5	([18.60,19.60],[18.50,20.30])	([16.80,17.85],[16.50,18.00])	([0.60,0.65],[0.50,0.70])	([0.80,0.90],[0.65,1.00])
6	([37.20,37.80],[37.00,38.00])	([33.80,34.50],[33.70,34.60])	([0.45,0.75],[0.40,0.80])	([0.80,0.90],[0.65,1.00])
7	([8.60,9.00],[8.50,9.10])	([8.00,8.15],[7.90,8.20])	([0.15,0.20],[0.10,0.25])	([0.70,0.90],[0.65,1.20])
8	([8.40,8.70],[8.35,8.80])	([7.60,7.95],[7.50,8.00])	([0.10,0.15],[0.10,0.20])	([0.70,0.90],[0.65,1.20])
9	([6.70,6.85],[6.65,6.85])	([6.10,6.17],[6.05,6.20])	([0.10,0.15],[0.05,0.20])	([0.70,0.90],[0.65,1.20])
10	([8.65,9.10],[8.50,9.15])	([7.80,8.00],[7.80,8.10])	([0.10,0.15],[0.10,0.20])	([0.50,0.70],[0.45,0.75])
11	([6.10,6.40],[6.00,6.40])	([5.60,5.80],[5.50,5.85])	([0.10,0.20],[0.05,0.20])	([0.50,0.70],[0.45,0.75])
12	([6.40,6.80],[6.40,6.85])	([5.80,5.98],[5.75,6.10])	([0.10,0.15],[0.05,0.20])	([0.50,0.70],[0.45,0.75])
13	([11.20,11.40],[11.15,11.50])	([10.35,10.50],[10.20,10.55])	([0.25,0.35],[0.20,0.40])	([0.30,0.50],[0.25,0.65])
14	([11.40,11.60],[11.35,11.65])	([10.40,10.55],[10.35,11.00])	([0.25,0.35],[0.10,0.35])	([0.30,0.50],[0.25,0.65])
15	([10.63,10.80],[10.50,11.00])	([9.50,9.80],[9.40,9.90])	([0.20,0.30],[0.15,0.40])	([0.30,0.50],[0.25,0.65])
16	([2.80,3.00],[2.75,3.10])	([2.60,2.69],[2.55,2.70])	([0.08,0.10],[0.05,0.15])	([0.10,0.20],[0.10,0.25])
17	([3.00,3.50],[3.00,3.55])	([2.70,2.85],[2.65,2.90])	([0.08,0.10],[0.05,0.15])	([0.10,0.20],[0.10,0.25])
18	([3.05,3.10],[3.00,3.15])	([2.75,2.86],[2.65,2.90])	([0.08,0.10],[0.05,0.15])	([0.10,0.20],[0.10,0.25])
19	([5.90,6.20],[5.85,6.20])	([5.58,5.60],[5.50,5.80])	([0.17,0.20],[0.15,0.25])	([0.50,0.60],[0.55,0.65])
20	([9.30,9.45],[9.30,9.50])	([8.50,8.53],[7.50,8.60])	([0.25,0.26],[0.20,0.30])	([0.50,0.60],[0.55,0.65])
21	([7.55,7.80],[7.40,7.80])	([7.00,7.10],[6.80,7.30])	([0.20,0.22],[0.20,0.25])	([0.50,0.60],[0.55,0.65])

According to the theory proposed before, we have the following crisp model,

$$
\begin{cases}
\max f_1(x) = \Phi(705.75 - 65.25x_1) \\
\max f_2(x) = \Phi(990.80 - 66.22x_2) \\
\max f_3(x) = \Phi(1680.69 - 44.12x_3) \\
\max f_4(x) = \Phi(914.55 - 64.97x_4) \\
\max f_5(x) = \Phi(336.25 - 66.77x_5) \\
\max f_6(x) = \Phi(1191.11 - 27.18x_6) \\
\max f_7(x) = \Phi(494.33 - 66.00x_7) \\
\max f_8(x) = \Phi(1780.50 - 32.48x_8) \\
\max f_9(x) = \Phi(3201.91 - 21.38x_9) \\
\max f_{10}(x) = \Phi(1983.30 - 26.70x_{10}) \\
\max f_{11}(x) = \Phi(1982.76 - 27.73x_{11}) \\
\max f_{12}(x) = \Phi(1635.50 - 46.28x_{12}) \\
\max f_{10}(x) = \Phi(2308.21 - 21.33x_{13}) \\
\max f_{14}(x) = \Phi(999.69 - 32.19x_{14}) \\
\max f_{15}(x) = \Phi(210.70 - 48.09x_{15}) \\
\max f_{16}(x) = \Phi(235.42 - 20.97x_{16}) \\
\max f_{17}(x) = \Phi(1528.53 - 23.85x_{17}) \\
\max f_{18}(x) = \Phi(690.25 - 34.75x_{18}) \\
\max f_{19}(x) = \Phi(315.75 - 48.62x_{19}) \\
\max f_{20}(x) = \Phi(615.46 - 32.31x_{20}) \\
\max f_{21}(x) = \Phi(497.45 - 31.36x_{21}) \\
s.t. \begin{cases}
0.5x_1 + 0.5x_2 + 0.5x_3 \le 29 \\
0.3x_4 + 0.3x_5 + 0.3x_6 \le 25.5 \\
0.1x_7 + 0.1x_8 + 0.1x_9 \le 19 \\
0.1x_{10} + 0.1x_{11} + 0.1x_{12} \le 17.5 \\
0.5x_{13} + 0.5x_{14} + 0.5x_{15} \le 67 \\
0.2x_{16} + 0.2x_{17} + 0.2x_{18} \le 36 \\
0.5x_{19} + 0.5x_{20} + 0.5x_{21} \le 20 \\
\sum_{i=1}^{21} h_i x_i \le 1200 \\
\sum_{i=1}^{21} x_i \ge 2500 \\
x_i \ge 0, i = 1, 2, \ldots, 21
\end{cases}
\end{cases}
\tag{5.78}
$$

We apply the TS algorithm to solve the above problem. The detail is as follows. After running 2000 cycles by TS, the corresponding satisfactory solutions are listed in Table 5.9. In real life, DM can increase or reduce the weight of different items. If DM hopes that the probability that the cost of item i is less than the budget is more than others, he or she can increase the corresponding weight. Then the ordering amount will reduce. DM can accurately compute the weight coefficients according to historical data. In this problem, we can be clear on the distribution of the demand and describe it as a crisp Ra-Ro variable according to historical data.

Table 5.9 The satisfactory solutions by TS

	$w_1 = 0.4$	$w_2 = 0.4$	$w_3 = 0.4$	$w_4 = 0.4$	$w_5 = 0.4$	$w_6 = 0.4$	$w_7 = 0.4$
x_1	6.79	6.54	6.23	6.40	6.66	6.71	6.72
f_1	0.94	0.86	0.84	0.87	0.84	0.86	0.88
x_2	15.17	16.63	14.20	16.11	15.12	15.64	14.86
f_2	0.93	0.83	0.82	0.86	0.80	0.82	0.84
x_3	38.03	38.45	39.20	37.89	38.21	38.15	38.07
f_3	0.96	0.91	0.90	0.90	0.91	0.92	0.89
x_4	14.05	13.54	14.23	16.40	14.66	13.71	15.72
f_4	0.74	0.85	0.74	0.72	0.76	0.73	0.74
x_5	5.02	5.54	6.23	5.40	4.66	4.71	5.72
f_5	0.82	0.90	0.74	0.72	0.76	0.73	0.74
x_6	43.76	6.54	6.23	6.40	6.66	6.71	6.72
f_6	0.88	0.90	0.74	0.72	0.76	0.73	0.74
x_7	7.47	6.54	6.23	6.40	6.66	6.71	6.72
f_7	0.81	0.90	0.74	0.72	0.76	0.73	0.74
x_8	54.74	6.54	6.23	6.40	6.66	6.71	6.72
f_8	0.96	0.90	0.74	0.72	0.76	0.73	0.74
x_9	149.63	6.54	6.23	6.40	6.66	6.71	6.72
f_9	0.94	0.90	0.74	0.72	0.76	0.73	0.74
x_{10}	74.19	6.54	6.23	6.40	6.66	6.71	6.72
f_{10}	0.83	0.90	0.74	0.72	0.76	0.73	0.74
x_{11}	71.42	6.54	6.23	6.40	6.66	6.71	6.72
f_{11}	0.81	0.90	0.74	0.72	0.76	0.73	0.74
x_{12}	35.29	6.54	6.23	6.40	6.66	6.71	6.72
f_{12}	0.90	0.90	0.74	0.72	0.76	0.73	0.74
x_{13}	108.08	6.54	6.23	6.40	6.66	6.71	6.72
f_{13}	0.76	0.90	0.74	0.72	0.76	0.73	0.74
x_{14}	31.00	6.54	6.23	6.40	6.66	6.71	6.72
f_{14}	0.92	0.90	0.74	0.72	0.76	0.73	0.74
x_{15}	4.36	6.54	6.23	6.40	6.66	6.71	6.72
f_{15}	0.67	0.90	0.74	0.72	0.76	0.73	0.74
x_{16}	11.19	6.54	6.23	6.40	6.66	6.71	6.72
f_{16}	0.74	0.90	0.74	0.72	0.76	0.73	0.74
x_{17}	64.01	6.54	6.23	6.40	6.66	6.71	6.72
f_{17}	0.88	0.90	0.74	0.72	0.76	0.73	0.74
x_{18}	19.84	6.54	6.23	6.40	6.66	6.71	6.72
f_{18}	0.82	0.90	0.74	0.72	0.76	0.73	0.74
x_{19}	6.48	6.54	6.23	6.40	6.66	6.71	6.72
f_{19}	0.72	0.90	0.74	0.72	0.76	0.73	0.74
x_{20}	19.02	6.54	6.23	6.40	6.66	6.71	6.72
f_{20}	0.84	0.90	0.74	0.72	0.76	0.73	0.74
x_{21}	15.81	6.54	6.23	6.40	6.66	6.71	6.72
f_{21}	0.86	0.90	0.74	0.72	0.76	0.73	0.74

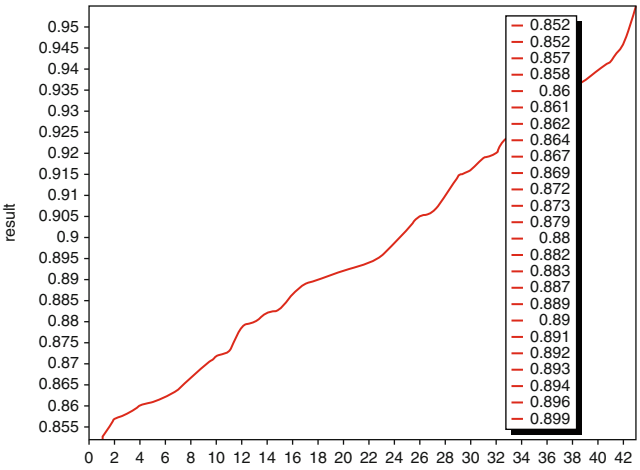


Fig. 5.21 The search process when $w_1 = 0.4$ and $w_i = 0.1, (i \neq 1)$

However, it is difficult to determine the distribution and we have to apply a Ra-Ro simulation to convert it into the crisp model and solve it by the fuzzy programming technique or TS.

Figure 5.21 shows the changes of the maximal chance measure when $w_1 = 0.4$ and $w_i = 0.1, (i \neq 1)$. It shows that the chance measure is gradually increasing for each generation.

5.6.4 Sensitivity Analysis

To illustrate the above problem, we give the results of the eight products corresponding discrete σ, α_1 values. For example, Table 5.10 shows the influence due to different σ, α_1 values of the first commodity when the σ, α_1 of the other commodities are all 0.5. For the first commodity, Table 5.10 shows the most superior total profits (PF), total costs (TC), total wastage costs (WC) and order quantities (Q_i) for the selling price $\tilde{\tilde{s}}_i$ corresponding discrete σ, α_1 values from Table 5.10 when $\sigma_{\tilde{\tilde{p}}_i} = \alpha_{1\tilde{\tilde{p}}_i} = 0.5, \sigma_{\tilde{\tilde{h}}_i} = \alpha_{1\tilde{\tilde{h}}_i} = 0.5, \sigma_{\tilde{\tilde{u}}_i} = \alpha_{1\tilde{\tilde{u}}_i} = 0.5$. Here, $0 \leq \alpha_1 \leq 0.6616$. In Table 5.10, if possibility value α_1 is 0.4 and probability value σ is 0.6, we obtain the most superior total profits (PF) is 7,822.42$, total costs (TC) are 32,219.71$, total wastage costs (WC) is 315.40$ and order quantities (Q_i) are 429 (428.71), 37 (37.28), 70 (69.90), 29 (28.64), 97 (96.71), 26 (26.19), 49 (49.30), 98 (98.35) units respectively. Thus, managers can devise stocking plans based on the most superior results. From Table 5.10, we can find that the influence of the results by the purchasing and selling prices is greater than that by the holding and set-up costs. The other parameters can be analyzed similarly.

As is shown in Fig. 5.22 and Table 5.10, when we fix one variable confidence level (α, β), the value of F_2 becomes bigger as other variables increase. Since DM

Table 5.10 The satisfactory solutions at different confidence levels

	PF	WC	TC	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	β_8
σ					$\alpha_1 = 0.2$						
0.20	7833.55	313.29	32080.60	449.14	38.67	64.81	29.26	89.56	27.15	51.22	93.26
0.40	7835.28	312.94	32059.03	451.82	38.88	64.11	29.35	88.56	27.30	51.53	92.44
0.60	7836.64	312.67	32042.04	453.88	39.05	63.58	29.43	87.79	27.42	51.77	91.79
0.80	7838.01	312.40	32024.89	455.90	39.23	63.05	29.50	87.04	27.54	52.02	91.12
1.00	7840.72	311.86	31990.97	459.81	39.57	62.03	29.64	85.58	27.78	52.51	89.78
σ					$\alpha_1 = 0.4$						
0.20	7816.31	309.16	32296.12	371.68	36.05	71.12	27.89	102.29	25.38	48.76	104.70
0.40	7819.70	312.48	32253.74	402.15	36.72	70.45	28.30	99.15	25.83	49.07	101.28
0.60	7822.42	315.40	32219.71	428.71	37.28	69.90	28.64	96.71	26.19	49.30	98.35
0.80	7825.20	314.96	32185.06	434.51	37.63	68.60	28.81	94.83	26.43	49.76	97.09
1.00	7830.67	313.87	32116.59	443.88	38.31	66.14	29.12	91.36	26.90	50.69	94.63
σ					$\alpha_1 = 0.6$						
0.20	7800.20	297.86	32497.54	261.65	33.19	74.32	26.06	118.68	23.49	46.97	117.50
0.40	7804.94	300.29	32438.30	287.46	33.96	73.39	26.56	113.70	24.01	47.52	114.40
0.60	7808.73	302.68	32390.89	311.25	34.62	72.64	26.98	109.83	24.44	47.95	111.57
0.80	7812.62	305.56	32342.21	338.91	35.32	71.88	27.43	105.99	24.91	48.38	108.32
1.00	7820.57	312.89	32242.87	406.61	36.87	70.31	28.39	98.55	25.92	49.15	100.62
σ					$\alpha_1 = 0.6616$						
0.20	7795.42	295.66	32550.49	238.21	32.51	75.20	25.61	123.63	23.03	46.45	120.13
0.40	7800.58	297.95	32492.79	262.95	33.25	74.25	26.09	118.33	23.53	47.01	117.31
0.60	7804.70	300.06	32441.30	285.33	33.92	73.44	26.53	114.00	23.98	47.49	114.61
0.80	7808.92	302.69	32388.51	311.67	34.64	72.62	27.00	109.70	24.46	47.97	111.48
1.00	7817.53	309.70	32280.93	377.69	36.25	70.92	28.01	101.39	25.52	48.87	103.82

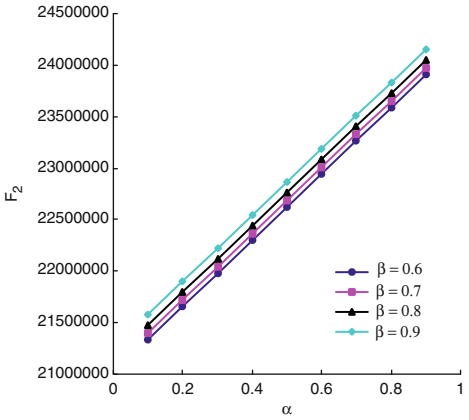


Fig. 5.22 Objectives F_2 with different confidence levels (α, β)

Table 5.11 The highest results corresponding discrete γ, δ

δ	F_1	F_2	F_1	F_2	F_1	F_2	F_1	F_2
	$\delta = 0.1$		$\delta = 0.2$		$\delta = 0.3$		$\delta = 0.4$	
0.1	4.21	24069500	4.21	24069500	4.21	24069500	4.21	24069500
0.15	4.25	24078650	4.25	24078650	4.25	24078650	4.25	24078650
0.2	4.21	24099670	4.21	24099670	4.21	24099670	4.21	24099670
0.25	4.75	24155900	4.75	24155900	4.75	24155900	4.75	24155900

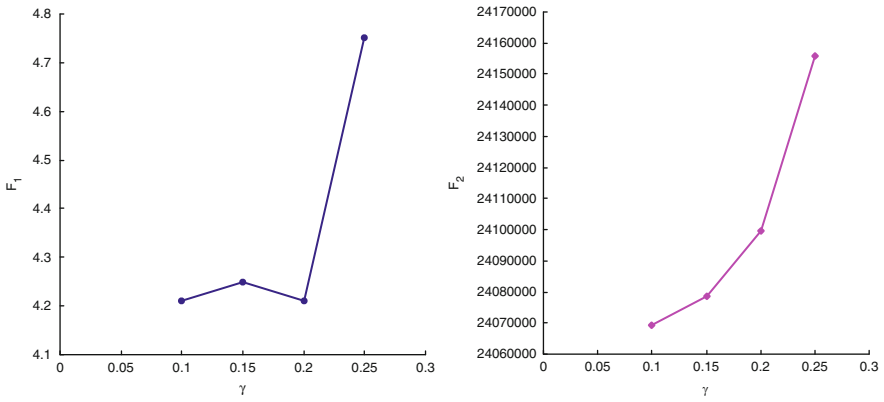


Fig. 5.23 Objectives with different confidence levels (γ, δ)

wants to set the confidence level so that the total cost doesn't exceed the predetermined value, the objective of minimizing total cost becomes hard to achieve. It satisfies the real-life situation. However, if the optimal solution doesn't vary, then the satisfying level of DM also doesn't.

When $\alpha = \beta = 0.9$, the highest results corresponding discrete γ, δ values are from the Table 5.11. Here, other parameters do not alter. The results listed in Fig. 5.23 show that the solutions can be affected by the different γ, since δ just makes small changes that only marginally affect the objectives. The total cost increases when γ becomes larger the same as the situation when α becomes larger.

Chapter 6
Methodological System for RLMODM

Random-like multiple objective decision making (RLMODM) considers multiple objective decision making with random-like phenomena. In this book, we focus on random-like uncertainty, and develop a series of multiple objective decision making:

- Multiple objective decision making with random phenomena
- Multiple objective decision making with bi-random phenomena
- Multiple objective decision making with random fuzzy phenomena
- Multiple objective decision making with random rough phenomena

In RLMODM, we use the following random-like variables to describe the coefficients, and consider the random-like phenomena:

- Random variable
- Ra-Ra variable
- Ra-Fu variable
- Ra-Ro variable

For the general MODM models with random-like coefficients, the meaning is not clear, so we have to adopt some philosophy to deal with them, and the following six kinds of random-like models are proposed:

- Expected value model (EVM): $\mathscr{M}(\boldsymbol{x}|E(\vartheta), E(\vartheta))$
- Chance constrained model (CCM): $\mathscr{M}(\boldsymbol{x}|C(\vartheta), C(\vartheta))$
- Dependent chance model (DCM): $\mathscr{M}(\boldsymbol{x}|D(\vartheta), C(\vartheta))$
- Expectation model with chance constraint (ECM): $\mathscr{M}(\boldsymbol{x}|E(\vartheta), C(\vartheta))$
- Chance constrained model with expectation constraint (CEM): $\mathscr{M}(\boldsymbol{x}|C(\vartheta), E(\vartheta))$
- Dependent chance model with expectation constraint

(DEM): $\mathscr{M}(\boldsymbol{x}|D(\vartheta), E(\vartheta))$, where $\vartheta \in \mathbf{U} = \{Ra, Ra-Ra, Ra-Fu, Ra-Ro\}$ expresses random-like uncertain coefficients. For the above random-like models, some of them can be directly converted into crisp equivalent models. However, many cannot be solved only by the mathematical transformation, so then we have to design a hybrid intelligent algorithm to get the approximate solutions.

J. Xu and L. Yao, *Random-Like Multiple Objective Decision Making*, Lecture Notes in Economics and Mathematical Systems 647, DOI 10.1007/978-3-642-18000-2_6,

RLMODM has also been applied to optimization problems with random-like parameters, for example, vendor selection problems [338], supply chain management problems [339,341], inventory problems [344,346], facility location-allocation problems [208], transportation problems [342, 343] and so on.

6.1 Motivation of Researching RLMODM

Why should we research RLMODM? Let us recall the two foundations: multiple objective decision making (MODM) and probability theory, see Fig. 6.1.

Optimization is a procedure for finding and comparing feasible solutions until no better solution can be found. Solutions are termed good or bad in terms of an objective, which is often the cost of fabrication, amount of harmful gases, efficiency of a process, product reliability or other factors. A significant portion of research and application in the field of optimization considers a single objective, although most real-world problems involve more than one objective. The presence of multiple conflicting objectives (such as simultaneously minimizing the cost of fabrication and maximizing product reliability) is natural in many problems and makes the optimization problem interesting to solve. Since no one solution can be termed as an optimum solution to multiple conflicting objectives, the resulting multiobjective optimization problems resorts to a number of trade-off optimal solutions. Classical optimization methods can at best find one solution in one simulation run, thereby making those methods inconvenient for solving multiobjective optimization problems.

Multiple objective decision making (abbr. MODM) is a part of mathematical programming dealing with decision problems characterized by multiple and conflicting objective functions that are to be optimized over a feasible set of decisions. Such

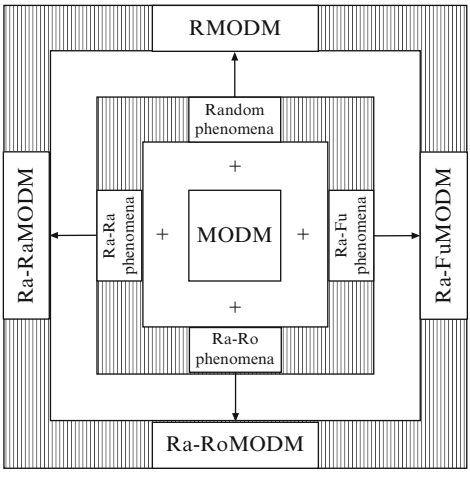


Fig. 6.1 Development of RLMODM

problems, referred to as multiobjective programming, are commonly encountered in many areas of human activity including engineering, management, and others. The research of the certain MODM can be traced back in the eighteenth century. Franklin introduced how to coordinate multiple objective problems in 1772. Then Cournot proposed the multiple objective model from an economic point of view in 1836. Pareto [246] firstly introduced multiple objective decision making models from the mathematical point of view in 1896. The seeds of what is a strong branch of operations research can be traced to the early work of Kunh and Tucker [64] and Koopmans [170] in 1951. Later, Arrow [10] proposed the concept of efficient points in 1953. MODM has gradually been widespread concerned and developed. MODM was not really considered a separate speciality, however, until a 1972 conference in South Carolina [61]. From then, it has become an area that attracted an enormous amount of attention because it is so useful for real-world decision making [360]. The monographs of Chankong and Hamies [44], Cohon [63], Hwang and Masud [138], Osyczka [242], Sawaragi et al. [278], Steuer [303] provide an extensive overview of the area of multiobjective optimization. Theory and methods for multiobjective optimization have been developed chiefly during the last century. Here we do not go into the history as the orgin and the achievements are widely treated in [300]. A brief summary of the development is also given in [104]. A great deal of theoretical, methodological and applied studies and related algorithms [6, 79, 103, 115, 124, 197, 199, 270, 271, 273–275, 282, 366] have been undertaken in the area of multiobjective programming.

Generally speaking, there are five elements in a MODM problem:

1. Decision variable: $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T \in \mathbf{R}^n$.
2. Objective function: $f(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})), m \geq 2$.
3. Feasible solution set: $X = \{\boldsymbol{x} \in \mathbf{R}^n | g_i(\boldsymbol{x}) \leq 0, h_r(\boldsymbol{x}) = 0, i = 1, 2, \ldots, p, r = 1, 2, \ldots, q\}$.
4. Preference relation: In the image set $f(X) = \{f(\boldsymbol{x}) | \boldsymbol{x} \in X\}$, there is a certain binary relation which could reflect the preference of the decision maker.
5. Definition of the solution. Define the optimal solution of f in X based on the known preference relation.

Thus, an MODM problem can be described as follows:

$$\begin{cases} \min \ f(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})] \\ \text{s.t. } \boldsymbol{x} \in X \end{cases}$$

The start of probability theory is tracked to a gambling problem regarding how to derive exact probabilities. People were broadly attracted to pay close attention to uncertain events (especially random events) around them and further studied random events. Afterwards, probability theory has been widely applied to many social problems and technology problems, such as, vital statistics, premium theory, astro observation, theory of errors, quality control and so on. From the seventeenth to nineteenth century. Many distinguished scholars such as, Bernoulli, De-Moivre, Laplace, Gauss, Poisson, Tchebychev, Markov have made contributions to the

development of probability theory. During this time, the development of probability theory fascinated everyone. However, as probability theory is applied to more and more real-life problems in many fields, the basic definition has been proved to be limiting, even proved that it cannot be used to deal with usual random events. Great progress was achieved when Von Mises [327] initialized the concept of sample space, filling the gaps between probability theory and measure theory in 1931. The strict theoretical principle didn't exist until 1933, when the outstanding mathematician Kolmogorov [168] from former Soviet Union published the famous paper 'The basic concept of probability theory', in which he put forward the axiomatization structure which is considered as the milestone, and the foundation of the development of probability theory. A general random MODM model can be written as follows:

$$\begin{cases} \min \ f(\boldsymbol{x}, \tilde{\boldsymbol{a}}) = [f_1(\boldsymbol{x}, \tilde{\boldsymbol{a}}), f_2(\boldsymbol{x}, \tilde{\boldsymbol{a}}), \ldots, f_m(\boldsymbol{x}, \tilde{\boldsymbol{a}})] \\ \text{s.t.} \begin{cases} g_i(\boldsymbol{x}, \tilde{\boldsymbol{b}}) \leq 0, \ i = 1, 2, \ldots, p \\ h_r(\boldsymbol{x}, \tilde{\boldsymbol{c}}) = 0, r = 1, 2, \ldots, q \\ \boldsymbol{x} \geq 0 \end{cases} \end{cases}$$

where $\tilde{\boldsymbol{a}}$, $\tilde{\boldsymbol{b}}$, $\tilde{\boldsymbol{c}}$ are the vectors of random coefficients. It should be noted that "$\leq$" denotes "basically less than or equal to", "=" denotes "basically equal to", and "min" denote "minimize the value of the objective functions as much as possible".

Actually, in order to make a satisfactory decision in practice, an important problem is to determine the type and accuracy of information. If complete information is required in the decision making process, it will mean the expenditure of extra time and money. If incomplete information is used to make a decision quickly, then it is possible to take non-optimal action. In fact, we cannot have complete accuracy in both information and decision because the total cost is the sum of the cost of running the target system and the cost of getting decision information. Since we have to balance the advantage of making better decisions against the disadvantages of getting more accurate information, incomplete information will almost surely be used in a real-life decision process, and uncertain programming is an important tool in dealing with the decision making with imperfect information. Among all of the uncertain programming, random programming approach [186, 203, 220] is useful and efficient in handling a programming problem with uncertainty.

As we know, the fuzzy random variable was proposed by Kwakernaak [180] who regarded it as "random variables whose values are not real, but fuzzy numbers". Kruse and Meyer [176] applied the fuzzy random variable to asymptotic statistics with vague data. From another view, Puri and Ralescu [256] and Klement et al. [162] regarded a fuzzy random variable as a random fuzzy set. The two views regarded it as a random variable and a fuzzy set in essence, respectively. They described two different kinds of uncertain environments and solved two different problems. Haldar and Reddy [126] considered both the randomness in some of the design parameters and the fuzzy imprecision in some other parameters representing the in-place condition of aged structures to estimate the reliability of existing structures. They used a hybrid approach in the random-fuzzy domain to evaluate reliability using an α-level

concept. Körner [171] inherited Kwakernaak's view and went down to rename the first as a random fuzzy variable in order to distinguish them, and he regarded it as a random variable with fuzzy parameters. From then on, several research works [65, 98, 172, 173, 206, 215, 236, 285, 332] about the random fuzzy variable have been published in recent years. We usually use the expected value operator or the probability operator to deal with programming problems with random coefficients and use the possibility or credibility operator to deal with problems of fuzzy coefficients. Similarly, the expected value operator and chance operator can be used to convert the programming problem with random fuzzy parameters into a crisp one.

If the value of a random variable ξ is also a random variable, then ξ is called the Ra-Ra variable. Similarly, if the value of a random variable ξ is a fuzzy/rough variable, then ξ is called the Ra-Fu/Ra-Ro variable. In realistic problems, the information may be described as a Ra-Ra variable, a Ra-Fu variable or a Ra-Ro variable. For example, we already know information is random variables, fuzzy variables and rough variables, and when we want to integrate the experiences and the knowledge of human beings, a good method is to add a tolerance interval to the former random variable, fuzzy variable or rough variable, and thus a Ra-Ra variable, a Ra-Fu variable or a Ra-Ro variable will exist. So how to deal with MODM with those random-like coefficients? It is very necessary and important for us to research random-like multiple objective decision making.

6.2 Physics-Based Model System

Random-like multiple objective decision making deals with multiple objective decision making problems with random-like phenomena. In other words, when some parameters or coefficients for a multiple objective decision making problem are some random-like coefficients, then this multiple objective decision making problem is called a random-like multiple objective decision making problem, which includes those problems with Ra-Ra, Ra-Fu, and Ra-Ro phenomena. In this book, we use three typical problems-transportation problems, network design problems and inventory problems to illustrate the Ra-Ra, Ra-Fu and Ra-Ro multiple objective decision making, respectively. Among all kinds of typical problems, we choose the flow shop scheduling problem, supply chain network design problem, and single-period inventory problem to clarify corresponding random-like multiple objective decision making in detail, see Fig. 6.2.

In the flow shop scheduling problem, we notice that the objective of flow shop scheduling problems mostly focus on minimizing total completion time, makespan. Additionally, objectives such as total flow time, tardiness, and idle time are also considered. But there are few which considered earliness time. In fact, the decision maker (DM) often wants to minimize the completion time and earliness, but the objectives conflict with one another. Each of these objectives is valid from a general point of view. Sometimes people have also used hybrid uncertain variables to be more precise, such as the bi-random variable. For example, according to historical

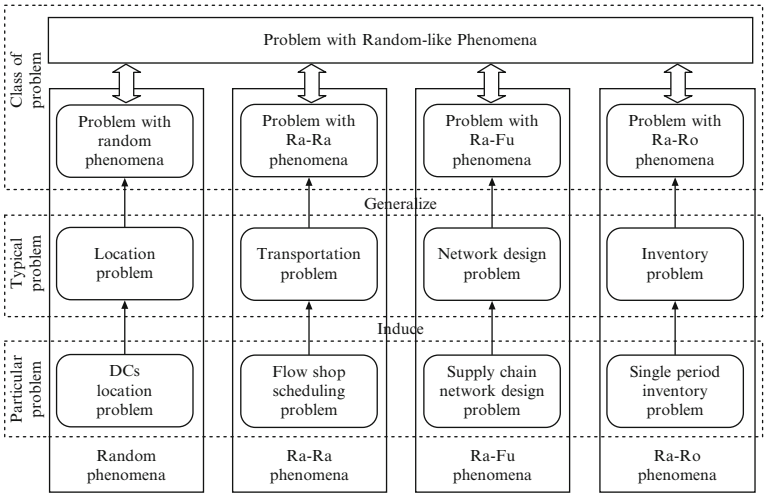


Fig. 6.2 Problem with random-like phenomena

data or abundance of information, we know the demand in one year is subject to stochastic distribution. However, the expected value of stochastic distribution is vague and varies from year to year. This results in the decision maker being unable to achieve a better decision. Sometimes, it could be considered as a random variable. Hence, we have to make a multiobjective decision in this situation.

The supply chain network (SCN) design problem has been gaining importance due to increasing competitiveness introduced by market globalization. A supply chain, beginning with the production of raw material by a supplier and ending with the consumption of a product by the customer, is a set of suppliers, facilities, products, customers, and inventory control, purchasing, and distribution. Traditionally, marketing, distribution, planning, manufacturing, and purchasing organizations along the supply chain are operated independently. Unfortunately, the SCN design problem is subject to many sources of uncertainty besides random uncertainty and fuzzy uncertainty [137]. In a practical decision-making process, we often face a hybrid uncertain environment. We consider the amount of demand on the products as normally distributed variable $\mathcal{N}(\mu, \sigma^2)$ from the view point of probability theory, and the values of μ as a triangular fuzzy variable (a, b, c) because of the lack of analytical data. Therefore, probability SCN with fuzzy parameters appears. In this case, the Ra-Fu variable can be used to deal with this kind of combined uncertainty of randomness and fuzziness. Hence, it is necessary to consider random fuzzy multiobjective decision making in this situation.

The inventory problem, known as a classical and complex problem, has been paid considerable attention. Nevertheless, most quantitative analysis on the inventory problem is mainly concerned with a single item and the deterministic parameters, such as crisp yield, crisp demand, crisp cost and so on. The uncertain inventory problem is also difficult and deserves to be researched. Some scholars have well

researched some inventory problems with crisp and vague parameters. Order, or demand, or planning horizons have been considered as fuzzy or random variables in some literatures [153, 213]. However, there is no attempt to research other mixed environments, where randomness and roughness both appear simultaneously. For some seasonal items (Ice cream, Christmas trees, woolen materials), the demand may vary from year to year. According to historical data or abundance of information, we know the demand in one year is subject to stochastic distribution. However, the expected value of the stochastic distribution is vague and varies year to year. Hence, we have to consider it as an uncertain variable. Rough variables can be applied to depict it well if the average sold amount is clear by the statistical data of each year. Thus, the demand of some seasonal items can be described as a random rough variable to help decision makers develop better strategies. Thus the Ra-Ro multiple objective model should be built for the inventory decision making problem under a Ra-Ro environment.

It is noted that the problems introduced in this book are just some example problems, and readers can obviously extend the application areas.

6.3 Mathematical Model System

The initial random-like multiple objective decision making model is as follows:

$$\begin{cases} \max\ [f_1(\boldsymbol{x},\boldsymbol{\xi}), f_2(\boldsymbol{x},\boldsymbol{\xi}), \ldots, f_m(\boldsymbol{x},\boldsymbol{\xi})] \\ \text{s.t.} \begin{cases} g_r(\boldsymbol{x},\boldsymbol{\xi}) \le 0,\ r = 1,2,\ldots,p \\ \boldsymbol{x} \in X \end{cases} \end{cases}$$

or

$$\begin{cases} \min\ [f_1(\boldsymbol{x},\boldsymbol{\xi}), f_2(\boldsymbol{x},\boldsymbol{\xi}), \ldots, f_m(\boldsymbol{x},\boldsymbol{\xi})] \\ \text{s.t.} \begin{cases} g_r(\boldsymbol{x},\boldsymbol{\xi}) \le 0,\ r = 1,2,\ldots,p \\ \boldsymbol{x} \in X \end{cases} \end{cases}$$

where $\boldsymbol{\xi}$ is a random-like vector, that is, the Ra-Ra vector, Ra-Fu vector and Ra-Ro vector and $\boldsymbol{x} \in X \subset \mathbf{R}^n$ is the decision vector.

It is necessary for us to know that the above models are conceptual models rather than mathematical models, because we cannot maximize an uncertain quantity. There does not exist a natural order relation in an uncertain world. Since there exists random-like variables, the above models have an ambiguous explanation. The meaning of maximizing/minimizing $f_1(\boldsymbol{x},\boldsymbol{\xi}), f_2(\boldsymbol{x},\boldsymbol{\xi}), \ldots, f_m(\boldsymbol{x},\boldsymbol{\xi})$ is unclear, and the constraints $g_r(\boldsymbol{x},\boldsymbol{\xi}) \le 0\ (r = 1,2,\ldots,p)$ do not define a deterministic feasible set.

Because of the existence of random-like uncertainty, we have to adopt some philosophies to deal with and make the above model solvable. In this book, three techniques including the expected value operator, the chance operator, the independent chance operator are introduced to deal with objective functions and constraints. Hence, philosophy 1–5 will be used to deal with decision making models with

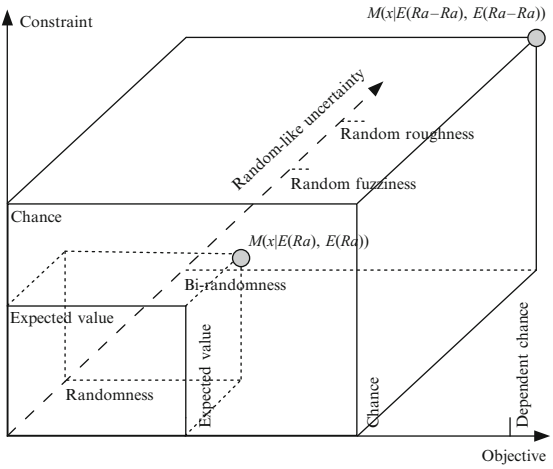


Fig. 6.3 Total space structure of model system

random-like phenomena. Generally, there are 3 techniques including philosophy 1–3 to deal with objective functions and 2 techniques including philosophy 4–5 to deal with the constraints. Meanwhile, random-like uncertainty includes randomness, bi-randomness, random fuzziness, and random roughness.

Total Space Structure. A space structure of all RLMODM models can be summarized and their relationship can be found in Fig. 6.3.

$$\mathscr{M}(\boldsymbol{x}|\alpha(\vartheta), \beta(\vartheta))$$

where $\boldsymbol{x}$ is the decision variable, $\vartheta \in \mathbf{U} = \{Ra, Ra - Ra, Ra - Fu, Ra - Ro\}$ expresses random-like uncertain coefficients, $\alpha \in \nabla = \{E, C, D\}$ expresses the technique dealing with the objective function, and $\beta \in \triangle = \{E, C\}$ expresses the technique dealing with the constraints. This structure can express all the RLMODM model. For example, $\mathscr{M}(\boldsymbol{x}|E(Ra - Ra), E(Ra - Ra))$ expresses a multiobjective programming model with Ra-Ra coefficients dealt by the expected objective and expected constraint. This is a typical model which is called EVM in Chap. 3. For the four random-like uncertainty and three techniques dealing with the objectives and constraints, there are total 24 basic models in total as follows:

$$\begin{gathered}
\mathscr{M}(\boldsymbol{x}|E(Ra), E(Ra)),\ \mathscr{M}(\boldsymbol{x}|E(Ra), C(Ra)),\\
\mathscr{M}(\boldsymbol{x}|C(Ra), E(Ra)),\ \mathscr{M}(\boldsymbol{x}|C(Ra), C(Ra)),\\
\mathscr{M}(\boldsymbol{x}|D(Ra), E(Ra)),\ \mathscr{M}(\boldsymbol{x}|E(Ra), C(Ra)),\\
\mathscr{M}(\boldsymbol{x}|E(Ra - Ra), E(Ra - Ra)),\ \mathscr{M}(\boldsymbol{x}|E(Ra - Ra), C(Ra - Ra)),\\
\mathscr{M}(\boldsymbol{x}|C(Ra - Ra), E(Ra - Ra)),\ \mathscr{M}(\boldsymbol{x}|C(Ra - Ra), C(Ra - Ra)),\\
\mathscr{M}(\boldsymbol{x}|D(Ra - Ra), E(Ra - Ra)),\ \mathscr{M}(\boldsymbol{x}|E(Ra - Ra), C(Ra - Ra)),\\
\mathscr{M}(\boldsymbol{x}|E(Ra - Fu), E(Ra - Fu)),\ \mathscr{M}(\boldsymbol{x}|E(Ra - Fu), C(Ra - Fu)),
\end{gathered}$$

$$\mathscr{M}(\boldsymbol{x}|C(Ra-Fu), E(Ra-Fu)),\ \mathscr{M}(\boldsymbol{x}|C(Ra-Fu), C(Ra-Fu)),$$
$$\mathscr{M}(\boldsymbol{x}|D(Ra-Fu), E(Ra-Fu)),\ \mathscr{M}(\boldsymbol{x}|E(Ra-Fu), C(Ra-Fu)),$$
$$\mathscr{M}(\boldsymbol{x}|E(Ra-Ro), E(Ra-Ro)),\ \mathscr{M}(\boldsymbol{x}|E(Ra-Ro), C(Ra-Ro)),$$
$$\mathscr{M}(\boldsymbol{x}|C(Ra-Ro), E(Ra-Ro)),\ \mathscr{M}(\boldsymbol{x}|C(Ra-Ro), C(Ra-Ro)),$$
$$\mathscr{M}(\boldsymbol{x}|D(Ra-Ro), E(Ra-Ro)),\ \mathscr{M}(\boldsymbol{x}|E(Ra-Ro), C(Ra-Ro)).$$

For the detail about these 5 philosophies, we will introduce them as follows. Firstly, let us consider the objective functions

$$\max\ [f_1(\boldsymbol{x}, \boldsymbol{\xi}), f_2(\boldsymbol{x}, \boldsymbol{\xi}), \ldots, f_m(\boldsymbol{x}, \boldsymbol{\xi})],$$

where $\boldsymbol{\xi}$ is the random-like variables.

There are three types of philosophy to handle objectives.

Philosophy 1: Making the decision by optimizing the expected value of the objectives. That is, maximizing the expected values of the objective functions for the Max problem, or minimizing the expected values of the objective functions for the Min problem.

$$\max\ [E[f_1(\boldsymbol{x}, \boldsymbol{\xi})], E[f_2(\boldsymbol{x}, \boldsymbol{\xi})], \ldots, E[f_m(\boldsymbol{x}, \boldsymbol{\xi})]],$$

or

$$\min\ [E[f_1(\boldsymbol{x}, \boldsymbol{\xi})], E[f_2(\boldsymbol{x}, \boldsymbol{\xi})], \ldots, E[f_m(\boldsymbol{x}, \boldsymbol{\xi})]].$$

Philosophy 2: Making a decision which provides the best optimal objective values with a given confidence level. That is, maximizing the referenced objective values $\bar{f}_i$ subjects to $f_i(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}_i$ with a confidence level α_i, or minimizing the referenced objective values $\bar{f}_i$ subjects to $f_i(\boldsymbol{x}, \boldsymbol{\xi}) \leq \bar{f}_i$ with a confidence level α_i.

$$\max\ [\bar{f}_1, \bar{f}_2, \ldots, \bar{f}_m]$$
$$\text{s.t. } Ch\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}_i\} \geq \alpha_i,\ i = 1, 2, \ldots, m,$$

or

$$\min\ [\bar{f}_1, \bar{f}_2, \ldots, \bar{f}_m]$$
$$\text{s.t. } Ch\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) \leq \bar{f}_i\} \geq \alpha_i,\ i = 1, 2, \ldots, m,$$

where α_i should be predetermined, $\bar{f}_1, \bar{f}_2, \ldots, \bar{f}_n$ are called critical values.

Philosophy 3: Making a decision by maximizing the chance of the events. That is, maximizing the chance of the events $f_i(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}_i$ or $f_i(\boldsymbol{x}, \boldsymbol{\xi}) \leq \bar{f}_i$.

$$\max \begin{bmatrix} Ch\{f_1(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}_1\}, \\ Ch\{f_2(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}_2\}, \\ \cdots \\ Ch\{f_m(\boldsymbol{x}, \boldsymbol{\xi}) \geq \bar{f}_m\}, \end{bmatrix}$$

or

$$\max \begin{bmatrix} Ch\{f_1(\boldsymbol{x},\boldsymbol{\xi}) \le \bar{f}_1\}, \\ Ch\{f_2(\boldsymbol{x},\boldsymbol{\xi}) \le \bar{f}_2\}, \\ \cdots \\ Ch\{f_m(\boldsymbol{x},\boldsymbol{\xi}) \le \bar{f}_m\}, \end{bmatrix}$$

where $\bar{f}_i$ should be predetermined.

Secondly, let us consider the constraints

$$\text{s.t.} \begin{cases} g_r(\boldsymbol{x},\boldsymbol{\xi}) \le 0,\ r = 1,2,\ldots,p \\ \boldsymbol{x} \in X \end{cases}$$

where $\boldsymbol{\xi}$ is the random-like variables.

There are two types of philosophy to handle the constraints.

Philosophy 4: Making the optimal decision subject to the expected constraints. That is,

$$E[g_r(\boldsymbol{x},\boldsymbol{\xi}) \le 0],\ r = 1,2,\ldots,p,$$

Philosophy 5: Making the optimal decision with chance constraints.

$$Ch\{g_r(\boldsymbol{x},\boldsymbol{\xi}) \le 0\} \ge \beta_r,\ r = 1,2,\ldots,p.$$

where β_r is predetermined.

By combining the 3 philosophies for the objective functions and 2 philosophies for the constraints, we can get six types of models which can deal with the initial random-like multiple objective decision making models: $\mathscr{M}(\boldsymbol{x}|E(\boldsymbol{\xi}),E(\boldsymbol{\xi}))$, $\mathscr{M}(\boldsymbol{x}|C(\boldsymbol{\xi}),C(\boldsymbol{\xi}))$, $\mathscr{M}(\boldsymbol{x}|D(\boldsymbol{\xi}),C(\boldsymbol{\xi}))$, $\mathscr{M}(\boldsymbol{x}|E(\boldsymbol{\xi}),C(\boldsymbol{\xi}))$, $\mathscr{M}(\boldsymbol{x}|C(\boldsymbol{\xi}),E(\boldsymbol{\xi}))$ and $\mathscr{M}(\boldsymbol{x}|D(\boldsymbol{\xi}),E(\boldsymbol{\xi}))$, see Fig. 6.4.

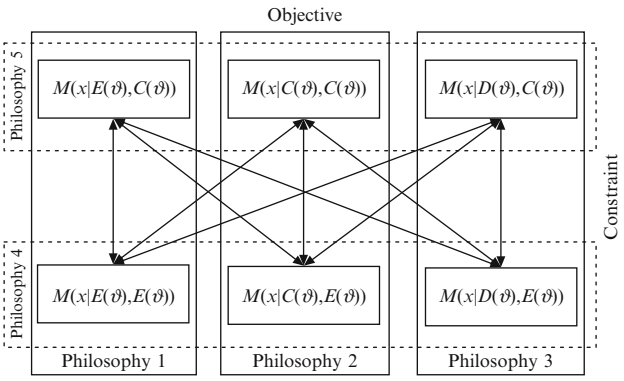


Fig. 6.4 Random-like model system

$$(\mathscr{M}(\boldsymbol{x}|E(\boldsymbol{\xi}),E(\boldsymbol{\xi})))\qquad \begin{cases}\max\ E[f_1(\boldsymbol{x},\boldsymbol{\xi}),f_2(\boldsymbol{x},\boldsymbol{\xi}),\ldots,f_m(\boldsymbol{x},\boldsymbol{\xi})]\\ \text{s.t.}\begin{cases}E[g_r(\boldsymbol{x},\boldsymbol{\xi})]\le 0,\ r=1,2,\ldots,p\\ \boldsymbol{x}\in X\end{cases}\end{cases}$$

$$(\mathscr{M}(\boldsymbol{x}|C(\boldsymbol{\xi}),C(\boldsymbol{\xi})))\qquad \begin{cases}\max\ \left[\bar{f}_1,\bar{f}_2,\ldots,\bar{f}_m\right]\\ \text{s.t.}\begin{cases}Ch\{f_i(\boldsymbol{x},\boldsymbol{\xi})\ge\bar{f}_i\}\ge\alpha_i,\ i=1,2,\ldots,m\\ Ch\{g_r(\boldsymbol{x},\boldsymbol{\xi})\le 0\}\ge\beta_r,\ r=1,2,\ldots,p\\ \boldsymbol{x}\in X\end{cases}\end{cases}$$

where $\alpha_i(i=1,2,\ldots,m),\beta_r(r=1,2,\ldots,p)$ are the predetermined confidence levels.

$$(\mathscr{M}(\boldsymbol{x}|D(\boldsymbol{\xi}),C(\boldsymbol{\xi})))\qquad \begin{cases}\max\ \begin{bmatrix}Ch\{f_1(\boldsymbol{x},\boldsymbol{\xi})\ge\bar{f}_1\},\\ Ch\{f_2(\boldsymbol{x},\boldsymbol{\xi})\ge\bar{f}_2\},\\ \cdots\\ Ch\{f_m(\boldsymbol{x},\boldsymbol{\xi})\ge\bar{f}_m\},\end{bmatrix}\\ \text{s.t.}\begin{cases}Ch\{g_r(\boldsymbol{x},\boldsymbol{\xi})\le 0\}\ge\beta_r,\ r=1,2,\ldots,p\\ \boldsymbol{x}\in X\end{cases}\end{cases}$$

where $\bar{f}_i(i=1,2,\ldots,m),\beta_r(r=1,2,\ldots,p)$ are the predetermined referenced objective values and confidence levels.

$$(\mathscr{M}(\boldsymbol{x}|E(\boldsymbol{\xi}),C(\boldsymbol{\xi})))\qquad \begin{cases}\max\ E[f_1(\boldsymbol{x},\boldsymbol{\xi}),f_2(\boldsymbol{x},\boldsymbol{\xi}),\ldots,f_m(\boldsymbol{x},\boldsymbol{\xi})]\\ \text{s.t.}\begin{cases}Ch\{g_r(\boldsymbol{x},\boldsymbol{\xi})\le 0\}\ge\beta_r,\ r=1,2,\ldots,p\\ \boldsymbol{x}\in X\end{cases}\end{cases}$$

where $\beta_r(r=1,2,\ldots,p)$ are the predetermined confidence levels.

$$(\mathscr{M}(\boldsymbol{x}|C(\boldsymbol{\xi}),E(\boldsymbol{\xi})))\qquad \begin{cases}\max\ \left[\bar{f}_1,\bar{f}_2,\ldots,\bar{f}_n\right]\\ \text{s.t.}\begin{cases}Ch\{f_i(\boldsymbol{x},\boldsymbol{\xi})\ge\bar{f}_i\}\ge\alpha_i,\ i=1,2,\ldots,m\\ E[g_r(\boldsymbol{x},\boldsymbol{\xi})]\le 0,\ r=1,2,\ldots,p\\ \boldsymbol{x}\in X\end{cases}\end{cases}$$

where $\alpha_i(i=1,2,\ldots,m)$ are the predetermined confidence levels.

$$(\mathscr{M}(\boldsymbol{x}|D(\boldsymbol{\xi}),E(\boldsymbol{\xi})))\qquad \begin{cases}\max\ \begin{bmatrix}Ch\{f_1(\boldsymbol{x},\boldsymbol{\xi})\ge\bar{f}_1\},\\ Ch\{f_2(\boldsymbol{x},\boldsymbol{\xi})\ge\bar{f}_2\},\\ \cdots\\ Ch\{f_m(\boldsymbol{x},\boldsymbol{\xi})\ge\bar{f}_m\},\end{bmatrix}\\ \text{s.t.}\begin{cases}E[g_r(\boldsymbol{x},\boldsymbol{\xi})]\le 0,\ r=1,2,\ldots,p\\ \boldsymbol{x}\in X\end{cases}\end{cases}$$

where $\bar{f}_i (i = 1, 2, \ldots, m)$ are the predetermined referenced objective values and confidence levels.

In this book, we mainly discuss the first three models, the techniques are all incorporated when we deal with EVM, CCM and DCM, that is, $\mathscr{M}(\boldsymbol{x}|E(\boldsymbol{\xi}), E(\boldsymbol{\xi}))$, $\mathscr{M}(\boldsymbol{x}|C(\boldsymbol{\xi}), C(\boldsymbol{\xi}))$ and $\mathscr{M}(\boldsymbol{x}|D(\boldsymbol{\xi}), C(\boldsymbol{\xi}))$. And the rest of the models ECM, CEM and DEM ($\mathscr{M}(\boldsymbol{x}|E(\boldsymbol{\xi}), C(\boldsymbol{\xi}))$, $\mathscr{M}(\boldsymbol{x}|C(\boldsymbol{\xi}), E(\boldsymbol{\xi}))$ and $\mathscr{M}(\boldsymbol{x}|D(\boldsymbol{\xi}), E(\boldsymbol{\xi}))$) can be handled in the same way. The reader can use the model when they use different philosophies, and it is possible to use every model.

6.4 Model Analysis System

For the linear random-like multiple objective decision making model,

$$\begin{cases} \max \left[\tilde{\tilde{c}}_1 \boldsymbol{x}, \ldots, \tilde{\tilde{c}}_m \boldsymbol{x}\right] \\ \text{s.t.} \begin{cases} \tilde{\tilde{e}}_r \boldsymbol{x} \leq \tilde{\tilde{b}}_r, \ r = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases}$$

where $\tilde{\tilde{c}}_i, \tilde{\tilde{e}}_r, \tilde{\tilde{b}}_r, (i = 1, 2, \ldots, m; r = 1, 2, \ldots, p)$ are special random-like coefficients, that is, the Ra-Ra variables, Ra-Fu variables and Ra-Ro variables. We introduced how to transform the 3 types of objective functions and the 2 types of constraints into their crisp equivalent formulas in detail. In this book, we introduced the equivalent models for EVM, CCM and DCM in detail, we simplify them as EEVM, ECCM and EDCM. See Fig. 6.5.

For the Ra-Ra linear multi-objective models, there are 4 basic theorems for handling the objective functions and the constraints: Theorems 2.2, 2.3, 2.6, and 2.9. And according to these 4 theorems, we can get the crisp equivalent models for random EVM, CCM and DCM.

The EVM with random coefficients can be converted into

$$(\mathscr{M}(\boldsymbol{x}|E(Ra), E(Ra))_1) \qquad \begin{cases} \max \left[\mu_1^{cT} \boldsymbol{x}, \mu_2^{cT} \boldsymbol{x}, \ldots, \mu_m^{cT} \boldsymbol{x}\right] \\ \text{s.t.} \begin{cases} \mu_r^{eT} \boldsymbol{x} \leq \mu_r^b, \ r = 1, 2, \ldots, p \\ \boldsymbol{x} \geq 0 \end{cases} \end{cases}$$

where the random variable follows normal distribution. When it follows exponential distribution, the EVM with random coefficients can be converted into

$$(\mathscr{M}(\boldsymbol{x}|E(Ra), E(Ra))_2) \qquad \begin{cases} \max \left[\lambda_1^{cT} \boldsymbol{x}, \lambda_2^{cT} \boldsymbol{x}, \ldots, \lambda_m^{cT} \boldsymbol{x}\right] \\ \text{s.t.} \begin{cases} \lambda_r^{eT} \boldsymbol{x} \leq \dfrac{1}{\lambda_r^b}, r = 1, 2, \ldots, p \\ \boldsymbol{x} \geq 0 \end{cases} \end{cases}$$

where $\lambda_i^c = \left(\frac{1}{\lambda_{i1}^c}, \frac{1}{\lambda_{i2}^c}, \ldots, \frac{1}{\lambda_{in}^c}\right)^T$ and $\lambda_i^e = \left(\frac{1}{\lambda_{r1}^e}, \frac{1}{\lambda_{r2}^e}, \ldots, \frac{1}{\lambda_{rn}^e}\right)^T$.

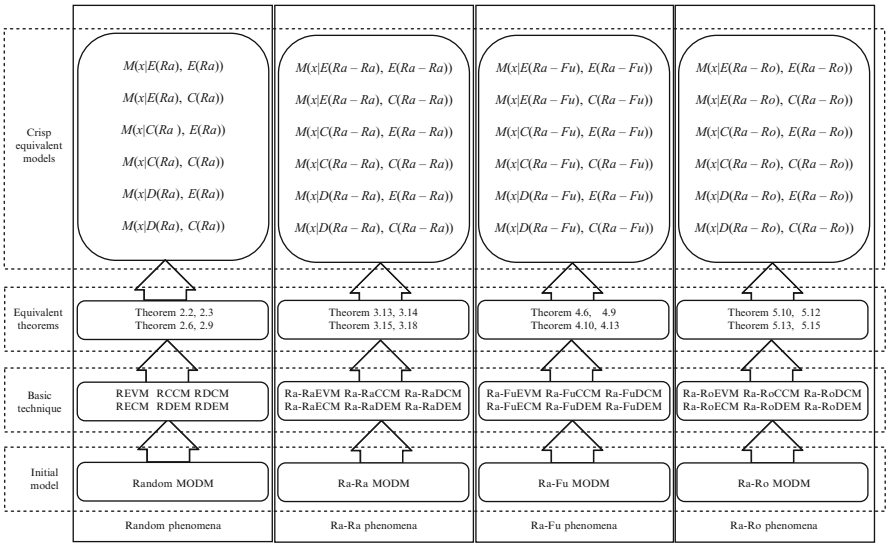


Fig. 6.5 Transformation to crisp equivalent models

The CCM with random coefficients can be converted into

$$(\mathscr{M}(\boldsymbol{x}|C(Ra), C(Ra))) \quad \begin{cases} \max\ [H_1(\boldsymbol{x}), H_2(\boldsymbol{x}), \ldots, H_m(\boldsymbol{x})] \\ \text{s.t.} \begin{cases} g_r(\boldsymbol{x}) \le 0, r = 1, 2, \ldots, p \\ \boldsymbol{x} \ge 0, 0 \le \alpha_r, \beta_i \le 1 \end{cases} \end{cases}$$

where $H_i(\boldsymbol{x}) = \Phi^{-1}(1 - \beta_i)\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}} + \mu_i^{cT}\boldsymbol{x}$, $g_r(\boldsymbol{x}) = \Phi^{-1}(\alpha_r)$ $\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2} + \mu_r^{eT}\boldsymbol{x} - \mu_r^b$ and Φ is the standardized normal distribution function.

The DCM with random coefficients can be converted into

$$(\mathscr{M}(\boldsymbol{x}|D(Ra), C(Ra))) \quad \begin{cases} \max \left[1 - \Phi\left(\dfrac{f_i - \mu_i^{cT}\boldsymbol{x}}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}} \right), i = 1, 2, \ldots, m \right] \\ \text{s.t.} \begin{cases} \Phi^{-1}(\beta_r)\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2} + \mu_r^{eT}\boldsymbol{x} - \mu_r^b \le 0 \\ \boldsymbol{x} \ge 0, r = 1, 2, \ldots, p \end{cases} \end{cases}$$

For the Ra-Ra linear multi-objective models, there are 4 basic theorems for handling the objective functions and the constraints: Theorems 3.13, 3.14, 3.15, and 3.18. According to these 4 theorems, we can get the crisp equivalent models for Ra-Ra EVM, CCM and DCM.

The EVM with Ra-Ra coefficients can be converted into

$$(\mathscr{M}(\boldsymbol{x}|E(Ra-Ra),E(Ra-Ra))) \qquad \begin{cases} \max\left[H_1(\boldsymbol{x}),H_2(\boldsymbol{x}),\ldots,H_m(\boldsymbol{x})\right] \\ \text{s.t.} \begin{cases} G_r(\boldsymbol{x}) \le K_r, r=1,2,\ldots,p \\ \boldsymbol{x} \ge 0 \end{cases} \end{cases}$$

where

$$H_i(\boldsymbol{x}) = -\sqrt{\boldsymbol{x}^T \Xi_i^c \boldsymbol{x}}\left(F\left(\frac{\mu_i^{cT}\boldsymbol{x}}{\sqrt{\boldsymbol{x}^T\Xi_i^c\boldsymbol{x}}}\right)+F\left(\frac{\mu_i^{cT}\boldsymbol{x}}{\sqrt{\boldsymbol{x}^T\Xi_i^c\boldsymbol{x}}}\right)-2F(-\infty)\right),$$

$$G_r(\boldsymbol{x}) = -\sqrt{\boldsymbol{x}^T \Xi_r^e \boldsymbol{x}}\left(F\left(\frac{\mu_r^{eT}\boldsymbol{x}}{\sqrt{\boldsymbol{x}^T\Xi_r^e\boldsymbol{x}}}\right)+F\left(-\frac{\mu_r^{eT}x}{\sqrt{x^T\Xi_r^e x}}\right)-2F(-\infty)\right),$$

$$K_r = -\sqrt{\Xi_r^b}\left(F\left(\frac{\mu_r^b}{\Xi_r^b}\right)+F\left(-\frac{\mu_r^b}{\Xi_r^b}\right)-2F(-\infty)\right).$$

The CCM with Ra-Ra coefficients can be converted into

$$(\mathscr{M}(\boldsymbol{x}|C(Ra-Ra),C(Ra-Ra))) \qquad \begin{cases} \max\left[H_1(\boldsymbol{x}),H_2(\boldsymbol{x}),\ldots,H_m(\boldsymbol{x})\right] \\ \text{s.t. } \boldsymbol{x} \in X \end{cases}$$

where $R = \Phi^{-1}(1-\beta_i)\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}$, $H_i(\boldsymbol{x}) = R + \mu_i + \sigma_i \Phi^{-1}(1-\alpha_i)$, $i = 1,2,\ldots,m$, and $X := \{\boldsymbol{x} \in \mathbf{R}^n | Pr\{\omega | Pr\{\tilde{\tilde{e}}_r^T \boldsymbol{x} \le \tilde{\tilde{b}}_r\} \ge \theta_r\} \ge \eta_r, r = 1,2,\ldots,p; \boldsymbol{x} \ge 0\}$.

The DCM with Ra-Ra coefficients can be converted into

$$(\mathscr{M}(\boldsymbol{x}|D(Ra-Ra),C(Ra-Ra)))$$
$$\begin{cases} \max\left[\Phi\left(\dfrac{\Phi^{-1}(1-\alpha_i)\sqrt{\sum\limits_{j=1}^{n} x_{ij}^2\sigma_{ij}^2} + d_i^{cT}\boldsymbol{x} - \bar{f}_i}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right), \right] \\ i = 1,2,\ldots,m \\ \text{s.t. } \boldsymbol{x} \in X, \end{cases}$$

where $X = \{\boldsymbol{x} | \Phi^{-1}(\theta_r)\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2} - \Phi^{-1}(\eta_r)\sqrt{(\delta_r^b)^2 + \sum\limits_{j=1}^{n} x_{ij}^2 (\delta_r^e)^2} - (d_r^b - d_r^e \boldsymbol{x}) \le 0, \boldsymbol{x} \ge 0\}$.

For the Ra-Fu linear multi-objective models, there are 4 important theorems for handling the objective functions and the constraints: Theorems 4.6, 4.9, 4.10 and 4.13. According to these 4 theorems, we get the crisp equivalent models for Ra-Fu EVM, CCM and DCM.

The EVM with Ra-Fu coefficients can be converted into

$$(\mathscr{M}(\boldsymbol{x}|E(Ra-Fu),E(Ra-Fu))) \qquad \begin{cases} \max\ [H_1(\boldsymbol{x}),H_2(\boldsymbol{x}),\ldots,H_m(\boldsymbol{x})] \\ \text{s.t.} \begin{cases} K_r(\boldsymbol{x}) \le B,\ r=1,2,\ldots,p \\ \boldsymbol{x} \ge 0 \end{cases} \end{cases}$$

where

$$H_i(\boldsymbol{x}) = \frac{1}{2}\delta_i^c \boldsymbol{x}\left[F(\mu_i^c \boldsymbol{x}) - F(\mu_i^c \boldsymbol{x} - \delta_i^c \boldsymbol{x})\right] + \frac{1}{2}\gamma_i^c \boldsymbol{x}\left[G(\mu_i^c \boldsymbol{x} + \gamma_i^c \boldsymbol{x}) - G(\mu_i^c \boldsymbol{x})\right],$$

$$K_r(\boldsymbol{x}) = \frac{1}{2}\delta_r^e \boldsymbol{x}\left[F(\mu_r^e \boldsymbol{x}) - F(\mu_r^e \boldsymbol{x} - \delta_r^e \boldsymbol{x})\right] + \frac{1}{2}\gamma_r^e \boldsymbol{x}\left[G(\mu_r^e \boldsymbol{x} + \gamma_r^e \boldsymbol{x}) - G(\mu_r^e \boldsymbol{x})\right],$$

$$B = \frac{1}{2}\delta_r^b \left[F(\mu_r^b) - F(\mu_r^b - \delta_r^b)\right] + \frac{1}{2}\gamma_r^b \left[G(\mu_r^b + \gamma_r^b) - G(\mu_r^b)\right].$$

The CCM with Ra-Fu coefficients can be converted into

$$(\mathscr{M}(\boldsymbol{x}|C(Ra-Fu),C(Ra-Fu))) \qquad \begin{cases} \max\ [H_1(\boldsymbol{x}),H_2(\boldsymbol{x}),\ldots,H_m(\boldsymbol{x})] \\ \text{s.t. } \boldsymbol{x} \in X \end{cases}$$

where $H_i(\boldsymbol{x}) = \Phi^{-1}(1-\beta_i)\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}} + R^{-1}(\alpha_i)\gamma_{ij}^c \boldsymbol{x} + d_i^{cT}\boldsymbol{x}, i = 1,2,\ldots,m.$

The DCM with Ra-Fu coefficients can be converted into

$$(\mathscr{M}(\boldsymbol{x}|D(Ra-Fu),C(Ra-Fu))) \qquad \begin{cases} \max \left[\Phi\left(\dfrac{R^{-1}(\alpha_i)\gamma_i^c \boldsymbol{x} + d_i^{cT}\boldsymbol{x} - \bar{f}_i}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}} \right), i = 1,2,\ldots,m \right] \\ \text{s.t. } \boldsymbol{x} \in X \end{cases}$$

where $X = \{\boldsymbol{x} \in R^n | \Phi^{-1}(\theta_r)\sqrt{\boldsymbol{x}^T V_r^e \boldsymbol{x} + (\sigma_r^b)^2} - (d_r^b - d_r^e \boldsymbol{x}) - R^{-1}(\eta_r)(\gamma_r^b + \delta_r^e \boldsymbol{x}) \le 0, \boldsymbol{x} \ge 0\}$.

For the Ra-Ro linear multi-objective models, there are 4 important theorems for handling the objective functions and the constraints: Theorems 5.10, 5.12, 5.13 and 5.15. And according to these 4 theorems, we can get the crisp equivalent models for Ra-Ro EVM, CCM and DCM.

The EVM with Ra-Ro coefficients can be converted into

$$(\mathscr{M}(\boldsymbol{x}|E(Ra-Ro),E(Ra-Ro))) \qquad \begin{cases} \max\ [\Psi_1^c \boldsymbol{x}, \Psi_2^c \boldsymbol{x}, \ldots, \Psi_m^c \boldsymbol{x}] \\ \text{s.t.} \begin{cases} \Psi_r^e \boldsymbol{x} \le \Psi_r^b, r = 1,2,\ldots,p \\ \boldsymbol{x} \ge 0 \end{cases} \end{cases}$$

where $\Psi_i^c = \frac{1-\eta}{2}(a_{ij}^c + b_{ij}^c) + \frac{\eta}{2}(c_{ij}^c + d_{ij}^c)$, $\Psi_i^e = \frac{1-\eta}{2}(a_i^e + b_i^e) + \frac{\eta}{2}(c_i^e + d_i^e)$, $\Psi_i^b = \frac{1-\eta}{2}(a_i^b + b_i^b) + \frac{\eta}{2}(c_i^b + d_i^b)$, $\chi_i^{\Xi}(\chi = a, b, c, d, \Xi = e, b)$ respectively denote the vectors.

The CCM with Ra-Ro coefficients can be converted into

$$(\mathscr{M}(\boldsymbol{x}|C(Ra-Ro), C(Ra-Ro))) \qquad \begin{cases} \max\ [H_1(\boldsymbol{x}), H_2(\boldsymbol{x}), \ldots, H_m(\boldsymbol{x})] \\ \text{s.t. } \boldsymbol{x} \in X' \end{cases}$$

where $H_i(\boldsymbol{x}) = d - 2\gamma_i(d-c) + R$, $M = \bar{f_i} - \Phi^{-1}(1-\delta_i)\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}$, $R = \Phi^{-1}(1-\delta_i)\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}$ and $X' = \{\boldsymbol{x} \in X | M \le d\}$.

The DCM with Ra-Ro coefficients can be converted into

$$(\mathscr{M}(\boldsymbol{x}|D(Ra-Ro), C(Ra-Ro))_1)$$

$$\begin{cases} \max\ [\beta_1, \beta_2, \ldots, \beta_m] \\ \text{s.t.} \begin{cases} \Phi\left(\dfrac{f_i - a}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right) \le \beta_i, & i = 1, 2, \ldots, m \\ \Phi\left(\dfrac{f_i - c - 2\alpha_i(d-c)}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right) \ge \beta_i, & i = 1, 2, \ldots, m \\ e_r^T \boldsymbol{x} \le b_r, & r = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases}$$

$$(\mathscr{M}(\boldsymbol{x}|D(Ra-Ro), C(Ra-Ro))_2)$$

$$\begin{cases} \max\ [\beta_1, \beta_2, \ldots, \beta_m] \\ \text{s.t.} \begin{cases} \Phi\left(\dfrac{f_i - b}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right) \le \beta_i, i = 1, 2, \ldots, m \\ \Phi\left(\dfrac{f_i - l}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right) \ge \beta_i, i = 1, 2, \ldots, m \\ e_r^T \boldsymbol{x} \le b_r, r = 1, 2, \ldots, p \\ \boldsymbol{x} \in X \end{cases} \end{cases}$$

where $l = \max\left\{a, \dfrac{c(b-a) + a(d-c) - 2\alpha_i(d-c)(b-a)}{b-a+d-c}\right\}$.

$$
(\mathscr{M}(\boldsymbol{x}|D(Ra-Ro),C(Ra-Ro))_3)
$$

$$
\begin{cases}
\max\ [\beta_1,\beta_2,\ldots,\beta_m] \\
\text{s.t.} \begin{cases}
\Phi\left(\dfrac{f_i-d}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right) \le \beta_i,\ i=1,2,\ldots,m \\
\Phi\left(\dfrac{f_i-t}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right) \ge \beta_i,\ i=1,2,\ldots,m \\
e_r^T \boldsymbol{x} \le b_r, \qquad r=1,2,\ldots,p \\
\boldsymbol{x} \in X
\end{cases}
\end{cases}
$$

where $t = \max\{b, (2\alpha_i - 1)(d-c) + c\}$.

$$
(\mathscr{M}(\boldsymbol{x}|D(Ra-Ro),C(Ra-Ro))_4)
$$

$$
\begin{cases}
\max\ [\beta_1,\beta_2,\ldots,\beta_m] \\
\text{s.t.} \begin{cases}
\Phi\left(\dfrac{f_i-d}{\sqrt{\boldsymbol{x}^T V_i^c \boldsymbol{x}}}\right) \ge \beta_i,\ i=1,2,\ldots,m \\
e_r^T \boldsymbol{x} \le b_r, \qquad r=1,2,\ldots,p \\
\boldsymbol{x} \in X
\end{cases}
\end{cases}
$$

6.5 Algorithm System

For the RLMODM, two kinds of algorithms are discussed. One aims at those decision making problem which are directly converted into crisp equivalent models. 12 tradition solution methods solving multiobjective programming problem are introduced. The other aims at those decision making problems which can not be converted into crisp equivalent models. A hybrid intelligent algorithm is introduced to obtain an approximate solution. The total flow chart of the two algorithms can found in Fig. 6.6.

After we get the crisp equivalent models, we can employ basic solution methods to solve those multiobjective programming problems. There are 12 solution methods detailed in the book, which includes:

- Two-stage method
- Goal programming method
- Ideal point method
- Fuzzy satisfied method
- Surrogate worth trade-off method
- Satisfying trade-off method
- Step method

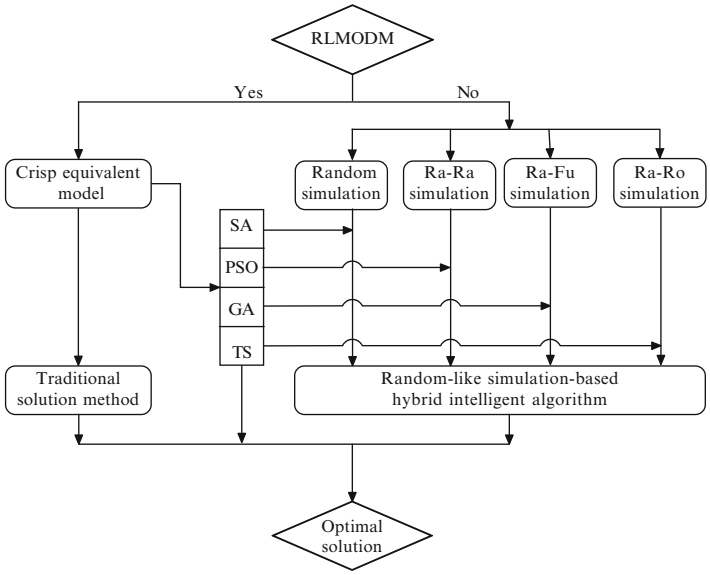


Fig. 6.6 Total flow chart of algorithm

- Lexicographic method
- Weight sum method
- Maximin point method
- Fuzzy goal method
- ε-constraint method

The above 12 solution methods are the most popular methods for multiple objective decision making. The decision maker can choose different method when they face different requests or under different conditions.

For the nonlinear random-like multiple objective decision making models, it is very difficult to transform the 3 types of objective functions and 2 types of constraints into their crisp equivalences. So we proposed several random-like simulations to simulate the objective functions and constraints. There are three kinds of simulations for each kind of random-like uncertainty. In Sects. 3.3.3.1, 3.4.3.1 and 3.5.3.1, we propose Ra-Ra simulation for EVM, Ra-Ra simulation for CCM and Ra-Ra simulation for DCM, respectively. In Sects. 4.3.3.1, 4.4.3.1, and 4.5.3.1, we propose Ra-Fu simulation for EVM, Ra-Fu simulation for CCM, and Ra-Fu simulation for DCM, respectively. In Sects. 5.3.3.1, 5.4.3.1, and 5.5.3.1, we propose Ra-Ro simulation for EVM, Ra-Ro simulation for CCM, and Ra-Ro simulation for DCM. By combining the random-like simulations and intelligent algorithms, we obtain some hybrid algorithms. Then for the six kinds of models: EVM, CCM, DCM, ECM, CEM, DEM, we can obtain several kinds of random-like hybrid algorithms to deal with, see Fig. 6.7.

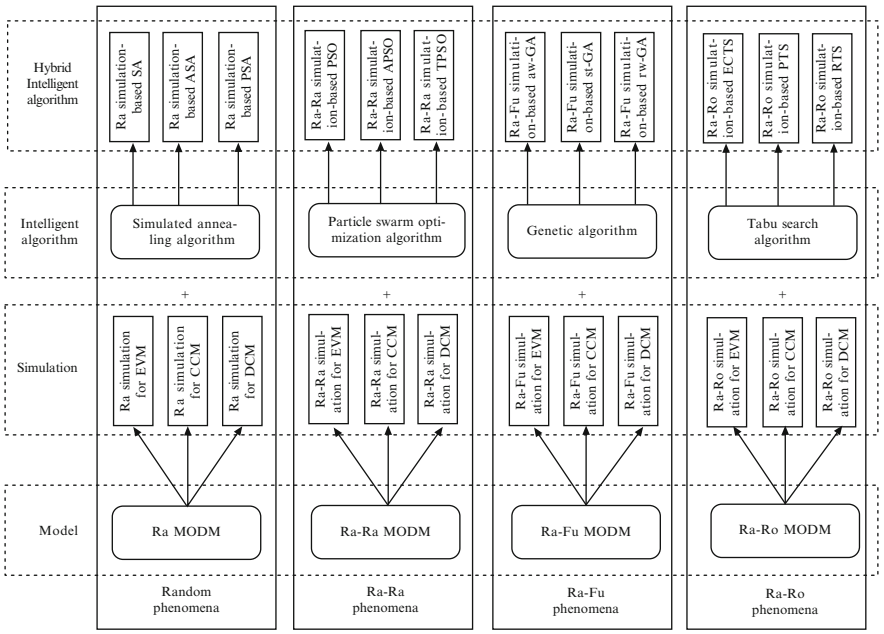


Fig. 6.7 Random-like hybrid algorithm system

These random-like simulations will embed into 4 types of basic intelligent algorithms, which includes

– Particle swarm optimization algorithm (PSO)
– Genetic algorithm (GA)
– Simulated annealing algorithm (SA)
– Tabu search algorithm (TS)

So for the general random-like MODM, we present the following ideas for designing the algorithm. For the linear random-like multiple decision making model with some particular random-like variables, we can transform them into some crisp equivalent models and use the above 12 traditional solution methods to solve them directly. For the normal random-like multiple decision making model, especially the nonlinear model, we embed the corresponding random-like simulations into the intelligent algorithm to find the solutions.

Application domains for each intelligent algorithm is as follows [333].
Some example areas for the application of PSO are:

- Machine Learning
- Function Optimization
- Geometry and Physics
- Operations Research

- Chemistry, Chemical Engineering
- Electrical Engineering and Circuit Design

Some example areas for the application of GA are:

- Scheduling
- Chemistry, Chemical Engineering
- Medicine
- Data Mining and Data Analysis
- Geometry and Physics
- Economics and Finance
- Networking and Communication
- Electrical Engineering and Circuit Design
- Image Processing
- Combinatorial Optimization

Some example areas for the application of Simulated Annealing are:

- Combinatorial Optimization
- Function Optimization
- Chemistry, Chemical Engineering
- Image Processing
- Economics and Finance
- Electrical Engineering and Circuit Design
- Machine Learning
- Geometry and Physics
- Networking and Communication.

Some example areas for the application of TS are:

- Combinatorial Optimization
- Machine Learning
- Biochemistry
- Operations Research
- Networking and Communication

Although we used these 4 algorithms in the book, there are some other excellent intelligent algorithms, such as the ant colony optimization algorithm (ACO), artificial neural network (ANN), immune algorithms (IA) and so on. We expect more advanced intelligent algorithms and we are willing to use them if it is appropriate in future research.

6.6 Research Ideas and Paradigm: 5MRP

RLMODM solves a class of real-life problems which seems to be uncertain (especially with random-like phenomena), for example, random demand in DCs location problem, bi-random transportation cost in the flow shop scheduling, random fuzzy

demand in the supply chain problem, random rough yield in the inventory problem, and so on. Why should we apply RLMODM to solve these problems? How should we construct a physical model to express these real-life problems? How should we deal with uncertainty in RLMODM? How can we describe this problem through scientific language? How can we design an efficient algorithm to solve a practiced problem? Finally how can be apply this integrated method to the engineering fields? All these questions must be answered under a new paradigm following a certain methodology. This new paradigm will enable researchers to draw scientific results and conclusions under the guidance of science, and will play a significant guiding role in conducting scientific research.

The research ideal of 5MRP expresses the initial relationship among Research, Model and Problem. R stands for a research system that includes research specifics, research background, research base, research reality, research framework, and applied research; M refers to a model system that includes concept models, physical models, physical and mathematical models, mathematical and physical models, designed models for algorithms, and describing the specific models. P represents a problem system that includes a particular problem, a class of problems, abstract problems, problem restoration, problem solutions, and problem settlements. Next, we take RLMODM as an example to present how to use 5MPR to start research.

Total ideal route. Let us summarize the research ideas and the framework of the research work, see Fig. 6.8. When research is started, we usually proceed to study a particular problem, which has research value and can be described as a concept model. This is the introduction of the research. After studying a particular problem and a problem with the same essence of the particular problem, then we can obtain the typical problem which has universality and can be abstracted to a physical model. This is the background to research. Then we generalize the typical problem ulteriorly to a class of problems which can be abstracted to common mathematical problems, then we can propose the mathematical model. This is the foundation of

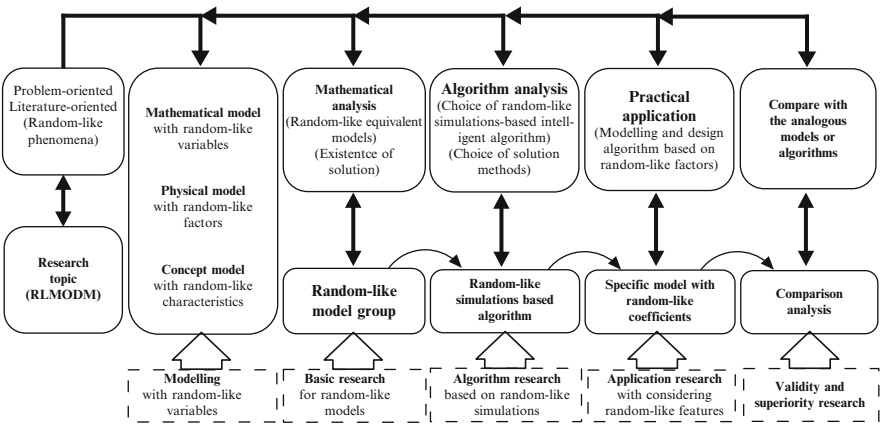


Fig. 6.8 Ideal route

the research. Then we design the algorithm and obtain the model for the procedure of the algorithm. This is the framework of the research. Finally we should apply the above models to a practical problem and establish a numerical model for the specific problem, and employ an algorithm to get the solution to illustrate the efficiency and validity. This is the application of the research.

Then we use Figs. 6.9–6.11 to describe the relationship between problem system, model system and research system.

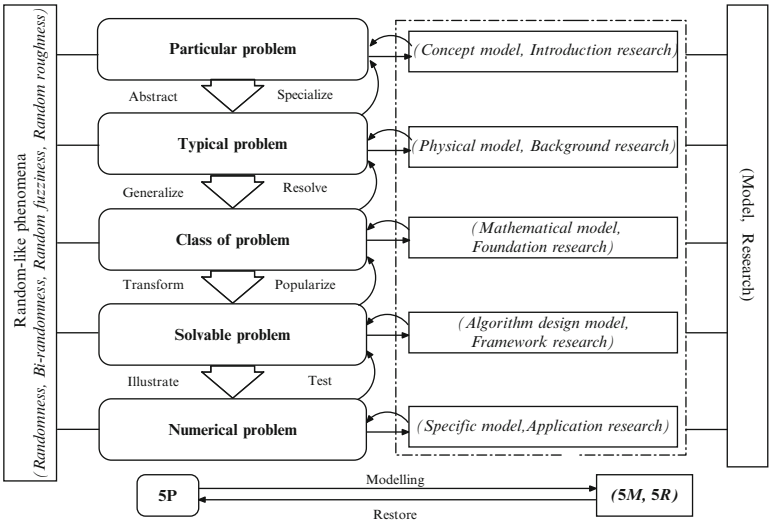


Fig. 6.9 Problem system

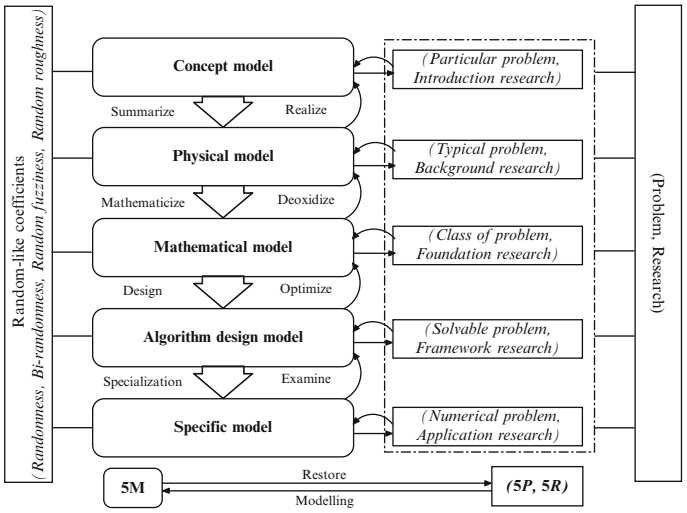


Fig. 6.10 Model system

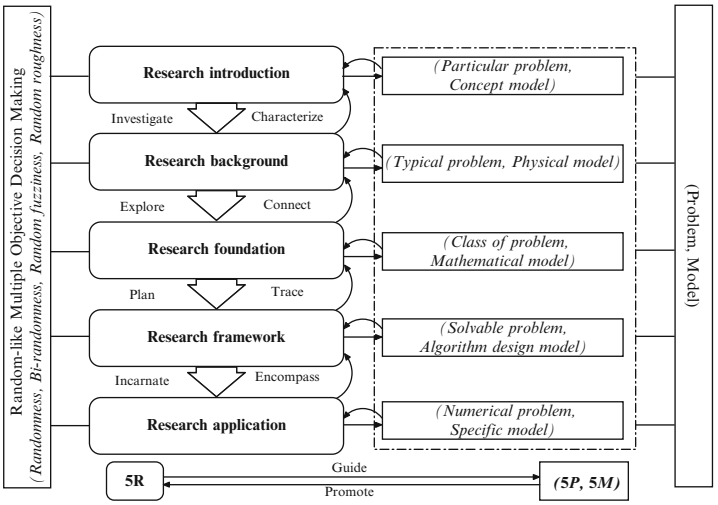


Fig. 6.11 Research system

Problem-driven. Figure 6.9 emphasizes the problem system, and presents the train of thought of dealing with the problem. In the real-life world, we usually face many problems which make can confuse us as we don't know how to deal with them, for example, the random demand in DCs location problem, the bi-random transportation cost in the flow shop scheduling, the random fuzzy demand in the supply chain problem, the random rough yield in the inventory problem, and so on. We don't know how to make a decision to control total cost and maximize total profit. This confusion drives us to start new research. Then a typical problem with random-like phenomena is formed in the brain, and we generalize it to be a class-problem with random-like phenomena, that is, it represents all the features of a class of problems with random-like phenomena. How can we solve it? This question reminds us that we should convert it to a solvable problem with random-like parameters. Finally, a numerical problem can be presented to confirm that these problems can be solved. Conversely, a numerical problem just checks if the solvable problem can be solved using the proposed model and algorithm. If so, it can be popularized to a class of problems and used to solve some typical problems. Finally, the particular problem we at first faced can be easily dealt with using the proposed technique.

Model system. Figure 6.10 emphasizes the model system, and presents a series of models which are used to deal with the corresponding problems. When we face those particular problems which confuse us, a concept with random-like coefficients is formed in our brain. Aiming at the typical problem, a physical model with random-like coefficients can be constructed to present the real-life structure of those problems. To help decision makers, a mathematical model with random coefficients can be constructed to quantifiably analyze those problems. Naturally, after constructing a mathematical model, we can design an algorithm model to solve it. Hence, this is called the algorithm design model with random-like coefficients.

Finally, a specific model with random-like coefficients according to a particular problem can be constructed to show the rationality of the proposed model and the efficiency of the proposed algorithm design model. Conversely, a specific model can be used to examine the efficiency and convergence of the designed algorithm model. If so, the algorithm can be applied to optimize the mathematical model. We can further deoxidize the mathematical model to a physical model to describe a realistic problem and solve a particular problem.

Research method. Figure 6.11 emphasizes the research system, and presents the technological process when we face a problem and conduct our research. The research process is also rooted in the problem and model. Firstly, a basic research introduction can be done when we face a particular problem with random-like phenomena. It means that we can give a basic description and construct a concept model about the particular problem. Secondly, the research background can be clarified when it is abstracted as a typical problem with random-like phenomena. In this process, the essence that the problem occurs can be investigated and then a physical model is given. Thirdly, we can prepare the basic research foundation when the physical model with random-like coefficients is abstracted to a mathematical model with random-like coefficients. The research foundation basically includes the following parts: (a) Which class of problems should these typical problems divide into? (b) What research has been done on this class of problems? (c) What literature is useful for our research about problems, models, algorithms? Fourthly, a research framework should be given to assist us in doing research. Finally, a research application is given to assist us in doing research. show the whole research process. Conversely, the application should include the whole research framework and reflect the research method. The research framework can also trace the research foundation and further connect the research background. In the end, the research goes back to the research introduction of the particular problem and obtains the optimal strategy.

Figure 6.12 presents how we use the 5MPR research ideal to do our research: random, bi-random, random fuzzy and random rough multiple objective decision making. Above all, 5MRP is an effective paradigm that can be widely used in various fields of scientific research and can contribute to research in all areas in a standardized and efficient manner. In the area of decision making, 5MRP is well reflected because of its rigorous logical and effective applicability, and it will have significant guiding role in the practice side of research.

6.7 Future Research

In the past 50 years, RLMODM has been a growing subject, whose theory and research methods are being rapidly developed and whose application has also become enlarged. Therefore, the summative conclusion is obtained about the future research of RLMODM: (1) a great surmounter in theory and method; (2) an important connection with realistic problems. The importance of the development of theory and method can only be measured by applying them to realistic problems. On

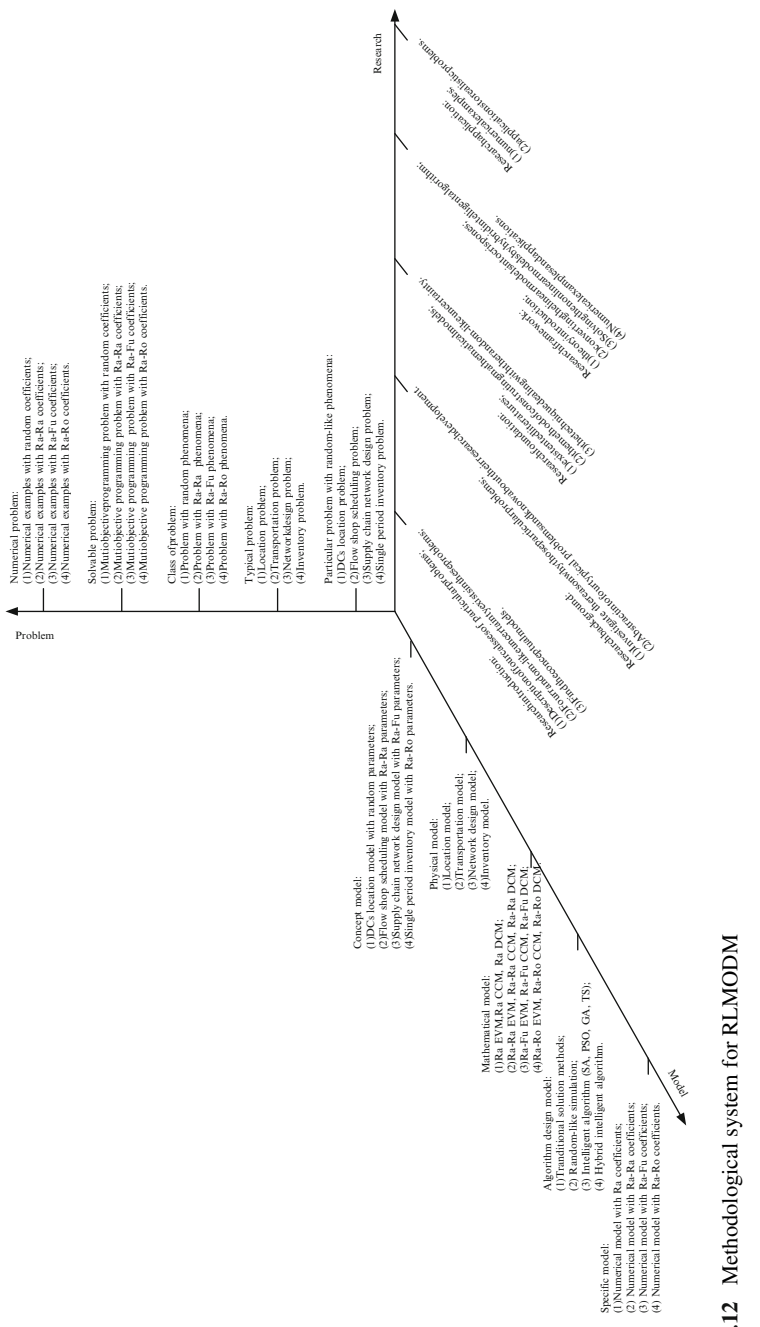


Fig. 6.12 Methodological system for RLMODM

the other hand, only a reliable theory can guarantee the correctness of those methods dealing with realistic problems. Here we provide some details about further research appearing in this area.

From the theoretical aspect, although it is basically perfect in the current system, readers can also use the new method to deal with RLMODM problems. In the current system, the authors only considered three techniques (EVM, CCM, DCM) to deal with uncertainty in the multiobjective programming model with random-like coefficients, then some other methods such as minimizing the variance can be hunted to deal with the uncertainty. In addition, some crisp equivalent conditions for some special distributions of the random-like variables in the linear RLMODM models are discussed, but the equivalency for other distributions could be further researched. For some special nonlinear models, their equivalent models maybe be obtained using the mathematical technique. This is also an interesting and worthy topic. For mathematical properties, we should consider sensitivity analysis, dual theorems, optimality conditions, and so on.

Form the viewpoint of the solutions method, some effective and powerful algorithms are designed in the current book. For example, twelve traditional methods are presented to solve those crisp multiobjective programming problems which are obtained by transforming the linear RLMODM model with special distributions. Four random-like simulations, and four intelligent algorithms including the genetic algorithm, the simulated annealing algorithm, the particle swarm optimization algorithm and the tabu search algorithm are proposed to solve those nonlinear RLMODM problems. Of course, there are many other solution methods for solving the multiobjective programming problems with crisp parameters. Hence, how to select an effective and appropriate method deserves more research. In addition, some new random-like simulation-based evolutionary algorithms can also be applied to the nonlinear RLMODM model. The convergence and the convergent speed are also future research focus.

In the perspective of the applications, RLMODM may be applied to four decision making problems with random-like phenomena in this book, for example, the DCs location problem, the flow shop scheduling problem, the supply chain network problem and the inventory problem. In fact, in the real-life world, we usually face many complicated problems. Hence, those applications in other fields such as finance, manufacturing, engineering management and so on should be also considered. In these realistic problems, mixed uncertainty may occur, for example, a supply chain system with random demand and random fuzzy cost. This is an important focus in our future research. In addition, how to select an approximate model is also a key factor when we face these realistic problems.

Above all, systematic research about RLMODM has been provided by this book and a scientific research area can be also found for future research from these aspects: theory, algorithm and applications.

Appendix A
Some Mathematical Aids

A.1 Series

- *Geometric series:* From $(1-r)(1+r+r^2+\cdots+r^n) = 1-r^{n+1}$, we obtain

$$\sum_{k=0}^{n} r^k = \frac{1-r^{n+1}}{1-r}, \textit{ for } r \neq 1.$$

For $|r| < 1$, these sums converge to the geometric series $\sum_{k=0}^{\infty} r^k = 1/(1-r)$. Differentiation yields the following two useful series:

$$\sum_{k=1}^{\infty} k r^{k-1} = \frac{1}{(1-r)^2}, \textit{ for } |r| < 1$$

and

$$\sum_{k=2}^{\infty} k(k-1) r^{k-2} = \frac{2}{(1-r)^3}, \textit{ for } |r| < 1.$$

For the finite sum, differentiation and algebraic manipulation yields

$$\sum_{k=0}^{n} k r^{k-1} = \frac{1-r^n(1+n(1-r))}{(1-r)^2},$$

which converges to

$$\frac{1}{(1-r)^2}, \textit{ for } |r| < 1.$$

J. Xu and L. Yao, *Random-Like Multiple Objective Decision Making*, Lecture Notes in Economics and Mathematical Systems 647, DOI 10.1007/978-3-642-18000-2_7,

- *Exponential series:* $e^x = \sum_{i=0}^{\infty} \frac{x^k}{k!}$ and $e^{-x} = \sum_{i=0}^{\infty} (-1)^k \frac{x^k}{k!}$ for any x. Simple algebraic manipulation yields the following equalities useful for the Poisson distribution:

$$\sum_{k=n}^{\infty} k \frac{x^k}{k!} = x \sum_{k=n-1}^{\infty} \frac{x^k}{k!}$$

and

$$\sum_{k=n}^{\infty} k(k-1) \frac{x^k}{k!} = x^2 \sum_{k=n-2}^{\infty} \frac{x^k}{k!}$$

A.2 Some Useful Integrals

- *The gamma function:* $\Gamma(r) = \int_0^{\infty} t^{r-1} e^{-t} dt$ for $r > 0$.
  Integration by parts shows $\Gamma(r) = (r-1)\Gamma(r-1)$ for $r > 1$.
  By induction $\Gamma(r) = (r-1)(r-2)\cdots(r-k)\Gamma(r-k)$ for $r > k$.
  For a positive integer n, $\Gamma(n) = (n-1)!$ with $\Gamma(1) = 0! = 1$.
- By a change of variable in the gamma integral, we obtain

$$\int_0^{\infty} t^r e^{-\lambda t} dt = \frac{\Gamma(r+1)}{\lambda^{r+1}}, \quad r > -1, \lambda > 0.$$

- A well-known indefinite integral gives

$$\int_a^{\infty} t e^{-\lambda t} dt = \frac{1}{\lambda^2} e^{-\lambda a}(1 + \lambda a)$$

and

$$\int_a^{\infty} t^2 e^{-\lambda a t} dt = \frac{1}{\lambda^3} e^{-\lambda a}(1 + \lambda a + (\lambda a)^2/2).$$

For any positive integer m,

$$\int_a^{\infty} t^m e^{-\lambda t} dt = \frac{m!}{\lambda^{m+1}} e^{-\lambda a} \left(1 + \lambda a + \frac{(\lambda a)^2}{2!} + \cdots + \frac{(\lambda a)^m}{m!}\right).$$

- The following integrals are important for the Beta distribution

$$\int_0^1 u^r (1-u)^s du = \frac{\Gamma(r+1)\Gamma(s+1)}{\Gamma(r+s+2)}, \; r > -1, s > -1$$

For nonnegative integers m, n,

$$\int_0^1 u^m (1-u)^n \, du = \frac{m!n!}{(m+n+1)!}.$$

A.3 Chebyshev's Inequality

Let $(\Omega, \mathscr{A}, Pr)$ be a probability space and $\xi = \xi(\omega)$ a nonnegative random variable. Then

$$Pr\{|\xi - E[\xi]| \geq \varepsilon\} \leq D[\xi]/\varepsilon^2 \tag{A.1}$$

for all $\varepsilon > 0$.

Proof. Let $F(x)$ be the distribution function of ξ, then it is apparent that

$$\begin{aligned} Pr\{|\xi - E[\xi]| \geq \varepsilon\} &= \int_{|\xi - E[\xi]| \geq \varepsilon} dF(x) \\ &\leq \int_{|\xi - E[\xi]| \geq \varepsilon} \frac{(x - E[\xi])^2}{\varepsilon^2} dF(x) \\ &\leq \frac{1}{\varepsilon^2} \int_{-\infty}^{\infty} (x - E[\xi])^2 dF(x) \\ &= \frac{D[\xi]}{\varepsilon^2} \end{aligned}$$

This completes the proof. □

Sometimes, (A.1) can be rewritten as

$$Pr\{|\xi - E[\xi]| < \varepsilon\} \geq 1 - \frac{D[\xi]}{\varepsilon^2}$$

or equivalently,

$$Pr\left\{ \left| \frac{\xi - E[\xi]}{\sqrt{D[\xi]}} \right| \geq \delta \right\} \leq \frac{1}{\delta^2}$$

A.4 Cauchy's Equation

- Let f be a real-valued function defined on $(0, \infty)$, such that
  (a) $f(t+u) = f(t) + f(u)$ for $t, u > 0$.
  (b) There is an open interval I on which f is bounded above (or is bounded below).

  Then $f(t) = f(1)t$ for $t > 0$.

- Let f be a real-valued function defined on $(0, \infty)$ such that
  (a) $f(t+u) = f(t)f(u)$ for $t, u > 0$.
  (b) There is an interval on which f is bounded above.
  Then, either $f(t = 0)$ for $t > 0$, or there is a constant a such that $f(t) = e^{at}$ for $t > 0$.

A.5 Some Useful Lemmas

Lemma A.1. *$ln(1+z) = z + R(z)$, where $|R(z)| < |z|^2$ for $|z| < 1/2$.*

Proof.

$$ln(1+z) = z + \frac{z^2}{2} + \frac{z^3}{3} + \cdots = z + R(z), for\ |z| < 1$$

where

$$R(z) = \frac{z^2}{2}\left(1 + \frac{2}{3}z + \frac{2}{4}z^2 + \cdots\right).$$

For $|z| < 1/2$,

$$|R(z)| < \frac{|z|^2}{2}\sum_{k=0}^{\infty}\frac{1}{2^k} = |z|^2.$$

□

Lemma A.2. *Let z_i, w_i, $1 \le i \le m$, be complex numbers with $|z_i| \le 1$, $|w_i| \le 1$, $1 \le i \le m$. Then*

$$\left|\prod_{i=1}^{m} z_i - \prod_{i=1}^{m} w_i\right| \le \sum_{i=1}^{m} |z_i - w_i|.$$

Proof. Consider

$$\prod_{i=1}^{k+1} z_i - \prod_{i=1}^{k+1} w_i = (z_{k+1} - w_{k+1})\prod_{i=1}^{k} z_i + w_{k+1}\left[\prod_{i=1}^{k} z_i - \prod_{i=1}^{k} w_i\right].$$

By hypothesis,

$$\left|\prod_{i=1}^{k} z_i\right| = \prod_{i=1}^{k} |z_i| \le 1, \quad and\ |w_{k+1}| \le 1.$$

From elementary properties of absolute value, we have

$$\left|\prod_{i=1}^{k+1} z_i - \prod_{i=1}^{k+1} w_i\right| \leq |z_{k+1} - w_{k+1}| \leq |z_{k+1} - w_{k+1}| + \left|\prod_{i=1}^{k} z_i - \prod_{i=1}^{k} w_i\right|.$$

Since the theorem is trivially true for $m = 1$, the last inequality provides the basis for a proof by mathematical induction. This completes the proof. □

As a corollary, we have that, if $|z| \leq 1$ and $|w| \leq 1$, then $|z^n - w^n| \leq n|z - w|$.
The Little-o Notation:

- Suppose $g(x) \neq 0$ for all x in some deleted neighborhood of a. We say

$$f(x) = o(g(x)) \; as \; x \to a \;\; iff \lim_{x \to a} \frac{f(x)}{g(x)} = 0.$$

The expression $f(x) = o(g(x))$ is read "f is little- of g at a" or "f is of smaller order than g at a".

- A number of properties of the little-o symbol are worth nothing.

(a) If f_1 and f_2 are little-o of g at a, then so is $f_1 \pm f_2$.
(b) If $c \neq 0$ and f is little-o of g at a, then so is cf.
(c) If h is bounded and nonzero in deleted neighborhood of a and f is little-o of g at a, then so is hf.
(d) If h is little-o of f at a and f is little-o of g at a, then h is little-o of g at a.
(e) If $g(x) \to 0$ as $x \to a$, then $1/(1 + g(x)) = 1 - g(x) + o(g(x))$.

Appendix B
Some Procedures

B.1 The Procedure of Random Simulation-Based SA

```
clc;
clear all;
clf;
figure(1);
initial_temperature=500;
finish_temperature=0;
cooling_tem=2;
B=1;
N_ycss=10;
h=2.0;
Max=0;
Max_1=0;
Max_2=0;
tic j=0;
w1=0.5;
w2=0.5;
while(j<1)
    x0=unifrnd(0,5);
    y0=unifrnd(0,5);
    t=constraint_check(x0,y0);
    if(t==1)
        j=j+1;
    end
end
z0=w1*Ob1(x0,y0)+w2*Ob2(x0,y0);
T=initial_temperature;
x_best=[];

while T>=finish_temperature
    n=1;
```

J. Xu and L. Yao, *Random-Like Multiple Objective Decision Making*, Lecture Notes in Economics and Mathematical Systems 647, DOI 10.1007/978-3-642-18000-2_8,

```
while n<N_ycss
    x_best=[];
    x_best=[x_best,[x0,y0]'];
    r=unifrnd(-1,1);
    x1=x0+r*h;
    y1=y0+r*h;
    if(constraint_check(x1,y1)==0)
        continue;
    end
    t1=Ob1(x1,y1);
    t2=Ob2(x1,y1);
    z1=w1*t1+w2*t2;
    diff=z0-z1;
    if diff<0
        z0=z1;
        if Max<z0
            Max=z0;
            Max_1=t1;
            Max_2=t2;
        end
        continue;
    else
        if (diff==0)
            x_best=[x_best,[x1,y1]'];
        end
        if exp(-diff/T)>unifrnd(0,1)
            z0=z1;
        end
    end
    n=n+1;
    plot(T,z0,'--b.');
    axis([0 500 0 6]);
    title('Cooling process');
    legend('Start value at current temperature',
         'Best value so far');
    hold on;
    pause(0.005);
    grid on;
end
plot(T,Max,'-r.');
axis([0 500 0 6]);
title('Cooling process');
legend('Start value at current temperature',
  'Best value so far');
hold on;
```

```
    pause(0.005);
    grid on;
    plot(T,Max_1,'-r.');
    plot(T,Max_2,'--b.')
    axis([0 500 0 8]);
    title('Cooling process');
    legend('f2','f1');
    hold on;
    pause(0.005);
    grid on;
    n=n+1;
    T=T-1;
end
z0
Max
Max_1
Max_2
x_best
toc

function [t] = constraint_check(x1,x2)
t=0;
rrs=0;
if((x1>=0)&(x2>=0))
    if(x1+x2<=5)
            t=1;
    end
end

function Ob1=Ob1(x1,x2)
N=20;
EXP=0;
for i=1:N
    t1=normrnd(2,0.5);
    t2=unifrnd(1,2);
    EXP=EXP+sqrt((t1-x1)^2+(t2-x2)^2);
end Ob1=EXP/N;

function Ob2=Ob2(x1,x2)
N=50;
EXP=0;
for i=1:N
    t1=normrnd(2,0.5);
    t2=unifrnd(1,2);
```

```
    EXP=EXP+sqrt((t1+x1)^2+(t2+x2)^2);
end Ob2=EXP/N;
```

B.2 The Procedure of Ra-Ra Simulation-Based PSO

```
clc;
clear all;
clf;
figure(1);
c1=1.4962;
c2=1.4962;
w=0.7298;
MaxDT=100;
N=10;
P=[];
v=[];
pbest=[];
Gmax=0;
p=[];
con=0;

tic
j=1;
while(j<=N)
    x0=unifrnd(0,5);
    y0=unifrnd(0,5);
    t=constraint_check(x0,y0);
    if(t==1)
        P=[P,[x0,y0]'];
        u=unifrnd(0,5);
        v=[v,u];
        j=j+1;
    end
end
for Gen=1:MaxDT
    pbest=P;
    for i=1:N
        g=fitness(P(1,i),P(2,i));
        p=[p,g];
        if(p(i)>=Gmax)
            Gmax=p(i);
            gbest_1=P(1,i);
            gbest_2=P(2,i);
```

```
        end
        v(i)=w*v(i)+c1*unifrnd(0,1)*sqrt((pbest
        (1,i)-P(1,i))^2+(pbest(2,i)-P(2,i))^2)+c2*
        unifrnd(0,1)*sqrt((gbest_1-P(1,i))^2+(gbest_2
        -P(2,i))^2);
        t1=P(1,i)+v(i);
        t2=P(2,i)+v(i);
        con=constraint_check(t1,t2);
        if(con==0)
            continue;
        end
        P(1,i)=t1;
        P(2,i)=t2;
        t=fitness(P(1,i),P(2,i));
        if (t>p(i))
            pbest(1,i)=P(1,i);
            pbest(2,i)=P(2,i);
            p(i)=t;
        end
        if (t>Gmax)
            Gmax=t;
            gbest_1=P(1,i);
            gbest_2=P(2,i);
            T_1=Ob1(gbest_1,gbest_2);
            T_2=Ob2(gbest_1,gbest_2);
        end
    plot(Gen,t,'--b.');
    axis([0 MaxDT 0 8]);
    title('Searching process');
    legend('Best so far');
    hold on;
    pause(0.005);
    grid on;
    end
    plot(Gen,Gmax,'--r.');
    axis([0 MaxDT 0 8]);
    title('Searching process');
    legend('Fitness value','Best so far');
    hold on;
    pause(0.005);
    grid on;
    plot(Gen,T_1,'-b.');
    plot(Gen,T_2,'--r.');
    axis([0 MaxDT 0 10]);
    title('Searching process');
```

```
    legend('H2','H1');
    hold on;
    pause(0.005);
    grid on;
end
gbest_1
gbest_2
T_1
T_2
Gmax
toc

function [t] = constraint_check(x1,x2)
t=0;
rrs=0;
if((x1>=0) &
(x2>=0))
    if(x1+x2<=5)
            t=1;
    end
end


function fitness=fitness(x1,x2)
w1=0.5;
w2=0.5;
fitness=w1*Ob1(x1,x2)+w2*Ob2(x1,x2);

function Ob1=Ob1(x1,x2)
N=50;
M=50;
EXP=0;
for i=1:N
    T=0;
    t1=normrnd(3,1);
    t2=normrnd(2,0.5);
    for i=1:M
       T=T+sqrt((normrnd(t1,1)-x1)^2
       +(normrnd(t2,1)-x2)^2);
   end
   EXP=EXP+T/M;
end
Ob1=EXP/N;

function Ob2=Ob2(x1,x2)
```

```
N=20;
M=20;
EXP=0;
for i=1:N
    T=0;
    t1=normrnd(3,1);
    t2=normrnd(2,0.5);
    for i=1:M
       T=T+sqrt((normrnd(t1,1)+x1)^2
       +(normrnd(t2,1)+x2)^2);
   end
   EXP=EXP+T/M;
end
Ob2=EXP/N;
```

B.3 The Procedure of Ra-Fu Simulation-Based GA

```
clc;
clear all;
clf;
figure(1);
N=3;
GEN=100;
POP_SIZE=10;
P_MUTATION=0.2;
P_CROSSOVER=0.3;
Chr=[];
Obj=[];
Ob1=[];
Ob2=[];
w_1=0.5;
w_2=0.5;

tic
j=1;
while(j<=POP_SIZE)
    x0=unifrnd(0,10);
    y0=unifrnd(0,10);
    z0=unifrnd(0,10);
    t=constraint_check(x0,y0,z0);
    if(t==1)
        Chr=[Chr,[x0,y0,z0]'];
        j=j+1;
```

```
    end
end
for gen=1:GEN
    INDX=[];
    z1Max=0;
    z2Max=0;
    rTemp=[];
    eval=[];
    for j=1:POP_SIZE
        t1=Obfunction1(Chr(1,j),Chr(2,j),Chr(3,j));
        if(t1>z1Max)
            z1Max=t1;
        end
        t2=Obfunction2(Chr(1,j),Chr(2,j),Chr(3,j));
        if(t2>z2Max)
            z2Max=t2;
        end
        u=w_1*t1+w_2*t2;
        Obj=[Obj,u];
        Ob1=[Ob1,t1];
        Ob2=[Ob2,t2];
    end
    [Obj,INDX]=sort(Obj);
    for i=1:POP_SIZE
        t=sqrt(w_1^2*(Ob1(j)-z1Max)^2+w_2^2
        *(Ob2(j)-z2Max)^2);
        rTemp=[rTemp,t];
    end
    rMax=max(rTemp);
    rMin=min(rTemp);
    rr=unifrnd(0,1);
    for i=1:POP_SIZE
        r_x(i)=sqrt(w_1^2*(Obfunction1(Chr(1,j),
        Chr(2,j),Chr(3,j))
        -z1Max)^2+w_2^2*(Obfunction2(Chr(1,j),
        Chr(2,j),Chr(3,j))-z2Max)^2);
        t=(rMax-r_x(i)+rr)/(rMax-rMin+rr);
        eval=[eval,t];
    end
    temp=[];
    qTemp=[];
    qTemp=[qTemp,0];
    q(1)=eval(1);
    qTemp=[qTemp,q(1)];
    for i=2:POP_SIZE
```

```
        q(i)=q(i-1)+eval(i);
        qTemp=[qTemp,q(i)];
    end
    for i=1:POP_SIZE
        r=unifrnd(0,q(POP_SIZE));
        for j=1:POP_SIZE
            if(r>=qTemp(j)&& r<qTemp(j+1))
                   temp=[temp,[Chr(1,j),Chr(2,j),
                   Chr(3,j)]'];
               break;
            end
        end
    end

    for i=1:POP_SIZE
        for k=1:N
            Chr(k,i)=temp(k,i)
        end
    end
    pop=POP_SIZE/2;
    for i=1:pop
        if (unifrnd(0,1)>P_CROSSOVER) continue;
        end
        j=floor(unifrnd(1,POP_SIZE));
        jj=floor(unifrnd(1,POP_SIZE));
        r=unifrnd(0,1);
        for k=1:N
            x(k)=r*Chr(k,j)+(1-r)*Chr(k,jj);
            y(k)=r*Chr(k,jj)+(1-r)*Chr(k,j);
        end
        if(constraint_check(x(1),x(2),x(3))==1)
            for k=1:N
                Chr(k,j)=x(k)
            end
        end
        if(constraint_check(y(1),y(2),y(3))==1)
            for k=1:N
                Chr(k,jj)=y(k)
            end
        end
    end
    INFTY=10;
    precision=0.0001;
    for i=1:POP_SIZE
        if (unifrnd(0,1)>P_MUTATION) continue;
```

```
        end
        for k=1:N
            x(k)=Chr(k,i);
        end
        for k=1:N
            if(unifrnd(0,1)<0.5)
            direction(k)=unifrnd(-1,1);
            else direction(k)=0;
            end
        end
        infty=unifrnd(0,INFTY);
        while(infty>precision)
            for j=1:N
                y(j)=x(j)+infty*direction(j)
            end
            if(constraint_check(y(1),y(2),y(3))==1)
                for k=1:N
                    Chr(k,i)=y(k);
                end
                break;
            end
            infty=unifrnd(0,infty);
        end
    end
scatter(Obj(POP_SIZE),gen,8,'r*');
axis([0 GEN 0 50]);
title('Search
process');
legend('Best so far');
hold on;
pause(0.005);
hold off;
toc
end

function [t] = constraint_check(x1,x2,x3)
t=0;
rfs=0;
if((x1>=0) &
(x2>=0) & (x3>=0))
    if((x1+x2+x3<=10) & (3*x1+5*x2+3*x3>=4))
        t=1;
    end
end
```

```
function [Obfunction1] = Obfunction1(X,Y,Z)
N=20;
M=20;
Obfunction1=0;
v=[];
E=[];
for i=1:N
    T=0;
     x1=unifrnd(5,7);
    if 5<x1&&x1<6
        mu1=x1-5;
    end
    if 6<=x1&&x1<7
        mu1=7-x1;
    end

    x2=unifrnd(6.5,10);
    if 6.5<=x2&&x2<8
        mu2=2*(x2-6.5)/3;
    end
    if 8<=x2&&x2<=10
        mu2=0.5*(10-x2);
    end
    for j=1:M
        k1=normrnd(x1,2);
        k2=normrnd(x2,1);
        T=T+(3*k1^2*X-2*k1*k2*Y+1.3*k2^2*Z);
    end
    E=[E,T/M];
    v=[v,min(mu1,mu2)];
end MIN=min(E); MAX=max(E); for k=1:N
    r=unifrnd(MIN,MAX);
    b1=0;
    b2=0;
    if r>=0
        for i=1:N
            if E(i)>=r&&b1<=v(i)
                b1=v(i);
            end
            if E(i)<r&&b2<=v(i)
                b2=v(i);
            end
        end
         Obfunction1=Obfunction1+(b1+1-b2)/2;
    else
```

```
        for i=1:N
            if E(i)<=r&&b1<v(i)
                b1=v(i);
            end
            if E(i)>r&&b2<v(i)
                b2=v(i);
            end
        end
        Obfunction1= Obfunction1-(b1+1-b2)/2;
    end
end if MIN<=0
    a=0;
else
    a=MIN;
end if MAX>=0
    b=0;
else
    b=MAX;
end Obfunction1=Obfunction1*(MAX-MIN)/N+a+b;


function [Obfunction2] = Obfunction2(t1,t2,t3)
N=20;
M=20;
Obfunction2=0;
v=[];
E=[];
for i=1:N
    T=0;
    x3=unifrnd(4,6);
    if 4<=x3&&x3<5
        mu3=x3-4;
    end
    if 5<=x3&&x3<=6
        mu3=6-x3;
    end

    x4=unifrnd(5,8);
    if 5<=x4&&x4<7
        mu4=0.5*(x4-5);
    end
    if 7<=x4&&x4<=8
        mu4=8-x4;
    end
    for j=1:M
```

```
            T=T+2.5*(normrnd(x3,1.5))^2*t1+
            3*normrnd(x3,1.5)*normrnd(x4,2)*t2
            +5*(normrnd(x4,2))^2*t3;
        end
        E=[E,T/M];
        v=[v,min(mu3,mu4)];
    end
    MIN=min(E);
    MAX=max(E);
    for k=1:N
        r=unifrnd(MIN,MAX);
        b1=0;
        b2=0;
        if r>=0
            for i=1:N
                if E(i)>=r&&b1<=v(i)
                    b1=v(i);
                end
                if E(i)<r&&b2<=v(i)
                    b2=v(i);
                end
            end
             Obfunction2=Obfunction2+(b1+1-b2)/2;
        else
            for i=1:N
                if E(i)<=r&&b1<v(i)
                    b1=v(i);
                end
                if E(i)>r&&b2<v(i)
                    b2=v(i);
                end
            end
            Obfunction2= Obfunction2-(b1+1-b2)/2;
        end
    end if MIN<=0
        a=0;
    else
        a=MIN;
    end if MAX>=0
        b=0;
    else
        b=MAX;
    end
    Obfunction2=Obfunction2*(MAX-MIN)/N+a+b;
```

B.4 The Procedure of Ra-Ro Simulation-Based TS

```
clc;
clear all;
clf;
figure(1);
N_jinji=1000;
N_ycss=1000;
j=0;
while(j<1)
    x0=unifrnd(0,5);
    y0=unifrnd(0,5);
    t=constraint_check(x0,y0);
    if(t==1)
        j=j+1;
    end
end

Tlist_x=[x0];
Tlist_y=[y0];
tic
for i=1:N_jinji
    j=0;
    h=5;
    z0=0;
    z1=0;
    z_ciyou=0;
    x_ciyou=0;
    y_ciyou=0;


    z0=x0^2+y0^2;
    for i=1:N_ycss
        r=unifrnd(-1,1);
        R=unifrnd(-1,1);
        x1=x0+r*h;
        if(x1<0)
            continue
        end
        y1=y0+R*h;
        if(y1<0)
            continue
        end
        if(x1+y1>5)
            continue
```

```
        end
        if(x1==find(Tlist_x) & y1==find(Tlist_y))
        %judge if it is in the tabu list
            continue;
        end
        z1=x1^2+y1^2;
        if(z1<z0)
            if(z1>z_ciyou)
                z_ciyou=z1;
                x_ciyou=x1;
               y_ciyou=y1;
            end
            continue
        elseif(z1==z0)
            Tlist_x=[Tlist_x,x1];
            Tlist_y=[Tlist_x,y1];
            continue
        else
            j=j+1;
            break;
        end
    end
    if(j==0)
        t=constraint_check(x_ciyou,y_ciyou);
        if(t==1)
            x0=x_ciyou;
            y0=y_ciyou;
        end
    end
    if(j==1)
        t=constraint_check(x1,y1);
        if(t==1)
            x0=x1;
            y0=y1;
            z0=z1;
            Tlist_x=[x0];
            Tlist_y=[y0];
        end
    end
    scatter(x_ciyou,y_ciyou,8,'bo')
    scatter(Tlist_x,Tlist_y,8,'r*');
    axis([0 5 0 5]);
    title('Search process');
    legend('Best so far', 'Sub-optimal so far');
    hold on;
```

```
    pause(0.005);
end hold off; toc Tlist_x Tlist_y z=x0^2+y0^2


function [t] = constraint_check(x1,x2)
t=0; rrs=0; if((x1>=0) &
(x2>=0))
    if(x1+x2<=5)
        rrs=rrsimulation(x1,x2);
        if(rrs<=7)
            t=1;
        end
    end
end

function [rrsimulation]=rrsimulation(x1,x2)
Mcount=100;
Racount=100;
E_1=0;
E_2=0;
lambda_1=0;
lambda_2=0;
rrsimulation=0;

for i=1:Mcount
    a=0;
    b=0;
    lambda_1=unifrnd(1,2);
    for i=1:Racount
        a=a+((x1-normrnd(lambda_1,1))^2
        +(x2-normrnd(lambda_1,1))^2)^0.5;
    end
    E_1=E_1+a/Racount;

    lambda_2=unifrnd(0,3);
    for i=1:Racount
        b=b+((x1-normrnd(lambda_2,1))^2
        +(x2-normrnd(lambda_2,1))^2)^0.5;
    end
    E_2=E_2+b/Racount;
end rrsimulation=(E_1+E_2)/(2*Mcount);
```

References

1. Aarts, E., Korst, J.: Simulated annealing and Boltzmann machines (1989)
2. Ahuj, R., Magnanti, T., Orlin, J.: Network Flows. New Jersey (1993)
3. Alefeld, G., Herzberger, J.: Introduction to interval computations (1983)
4. Alonso-Ayuso, A., Escudero, L., Garín, A., Ortuno, M., Pérez, G.: An approach for strategic supply chain planning under uncertainty based on stochastic 0-1 programming. Journal of Global Optimization **26**(1), 97–124 (2003)
5. Altiparmak, F., Gen, M., Lin, L., Paksoy, T.: A genetic algorithm approach for multi-objective optimization of supply chain networks. Computers & Industrial Engineering **51**(1), 196–215 (2006)
6. Amid, A., Ghodsypour, S., O'Brien, C.: Fuzzy multiobjective linear model for supplier selection in a supply chain. International Journal of Production Economics **104**(2), 394–407 (2006)
7. Angeline, P.: Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences. In: Evolutionary Programming VII, pp. 601–610. Springer (1998)
8. Araujo, A., Giné, E.: The central limit theorem for real and Banach valued random variables. Wiley New York (1980)
9. Ardagna, D., Francalanci, C., Trubian, M.: A multi-model algorithm for the cost-oriented design of Internet-based systems. Information Sciences **176**(21), 3105–3131 (2006)
10. Arrow, K., Barankin, E., Blackwell, D.: Admissible points of convex sets. Contributions to the Theory of Games **2**, 87–91 (1953)
11. Ash, R.: Real analysis and probability. Probability and Mathematical Statistics **11** (1972)
12. Azizi, N., Zolfaghari, S.: Adaptive temperature control for simulated annealing: a comparative study. Computers & Operations Research **31**(14), 2439–2451 (2004)
13. Badeau François, P., Potvin Eric, J.: A parallel tabu search heuristic for the vehicle routing problem with time windows. Transportation Research Part C: Emerging Technologies **5**(2), 109–122 (1997)
14. Baker, R., Urban, T.: A deterministic inventory system with an inventory-level-dependent demand rate. Journal of the Operational Research Society **39**(9), 823–831 (1988)
15. Baker Timothy, L., et al.: Deterministic fixed order-level inventory models: An application for replenishment of radioactive source material for irradiation sterilizers. European Journal of Operational Research **50**(3), 249–256 (1991)
16. Banerjee, P., Jones, M., Sargent, J.: Parallel simulated annealing algorithms for cell placement on hypercube multiprocessors. Parallel and Distributed Systems, IEEE Transactions on **1**(1), 91–106 (2002)
17. Barahona, F., Jensen, D.: Plant location with minimum inventory. Mathematical Programming **83**(1), 101–111 (1998)
18. Battiti, R., Tecchiolli, G., et al.: The reactive tabu search. ORSA journal on computing **6**, 126–126 (1994)
19. Beamon, B.: Supply chain design and analysis: Models and methods. International journal of production economics **55**(3), 281–294 (1998)

J. Xu and L. Yao, *Random-Like Multiple Objective Decision Making*, Lecture Notes in Economics and Mathematical Systems 647, DOI 10.1007/978-3-642-18000-2,

20. Benayoun, R., De Montgolfier, J., Tergny, J., Laritchev, O.: Linear programming with multiple objective functions: Step method (STEM). Mathematical programming **1**(1), 366–375 (1971)
21. Berberian, S.: Measure and integration (1965)
22. Berman, O., Larson, R.: Optimal 2-facility network districting in the presence of queuing. Transportation Science **19**(3), 261 (1985)
23. Bevilacqua, A.: A methodological approach to parallel simulated annealing on an SMP system. Journal of Parallel and Distributed Computing **62**(10), 1548–1570 (2002)
24. Bhatia, G.: Optimality and mixed saddle point criteria in multiobjective optimization. Journal of Mathematical Analysis and Applications **342**(1), 135–145 (2008)
25. Billingsley, P.: The Lindeberg-Levy theorem for martingales. Proceedings of the American Mathematical Society **12**(5), 788–792 (1961)
26. Billingsley, P.: Probability and measure. John Wiley & Sons (1965)
27. Blackwell, D.: On a class of probability spaces. In: Proceedings of the third Berkeley Symposium on mathematical statistics and probability: contributions to econometrics industrial research and psychometry, p. 1. University of California Press (1956)
28. Blöchliger, I., Zufferey, N.: A graph coloring heuristic using partial solutions and a reactive tabu scheme. Computers & Operations Research **35**(3), 960–975 (2008)
29. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems (Santa Fe Institute Studies in the Sciences of Complexity Proceedings) (1999)
30. Bratley, P., Fox, B., Schrage, L.: A guide to simulation. Springer (1987)
31. Brown, B.: Martingale central limit theorems. The Annals of Mathematical Statistics **42**(1), 59–66 (1971)
32. Buckley, J.: Fuzzy probability and statistics. Springer Verlag (2006)
33. Bulfin, R., M'hallah, R.: Minimizing the weighted number of tardy jobs on a two-machine flow shop. Computers & Operations Research **30**(12), 1887–1900 (2003)
34. Campos, L., Verdegay, J.: Linear programming problems and ranking of fuzzy numbers. Fuzzy Sets and Systems **32**(1), 1–11 (1989)
35. Canós, M., Ivorra, C., Liern, V.: The fuzzy p-median problem: A global analysis of the solutions. European Journal of Operational Research **130**(2), 430–436 (2001)
36. Carlier, J.: Ordonnancements a contraintes disjonctives. RAIRO Recherche operationelle/Operations Research **12**(4), 333–351 (1978)
37. Casavant, T., Kuhl, J.: A taxonomy of scheduling in general-purpose distributed computing systems. IEEE Transactions on Software Engineering, pp. 141–154 (1988)
38. Castellani, U., Fusiello, A., Gherardi, R., Murino, V.: Automatic selection of MRF control parameters by reactive tabu search. Image and Vision Computing **25**(11), 1824–1832 (2007)
39. Černỳ, V.: Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. Journal of optimization theory and applications **45**(1), 41–51 (1985)
40. Chakrapani, J., Skorin-Kapov, J.: Massively parallel tabu search for the quadratic assignment problem. Annals of Operations Research **41**(4), 327–341 (1993)
41. Chamberlain, R., Edelman, M., Franklin, M., Witte, E.: Simulated annealing on a multiprocessor. In: Computer Design: VLSI in Computers and Processors, 1988. ICCD'88., Proceedings of the 1988 IEEE International Conference on, pp. 540–544. IEEE (2002)
42. Chan, F., Chung, S.: A multi-criterion genetic algorithm for order distribution in a demand driven supply chain. International Journal of Computer Integrated Manufacturing **17**(4), 339–351 (2004)
43. Chan, F., Chung, S., Wadhwa, S.: A hybrid genetic algorithm for production and distribution. Omega **33**(4), 345–355 (2005)
44. Chankong, V., Haimes, Y.: Multiobjective decision making: theory and methodology. North-Holland New York (1983)
45. Charnes, A., Cooper, W.: Chance-constrained programming. Management Science **6**(1), 73–79 (1959)
46. Charnes, A., Cooper, W.: Management Models and Industrial Applications of Linear Programming, Vols. 1 and 2 (1961)
47. Chelouah, R., Siarry, P.: Tabu search applied to global optimization. European Journal of Operational Research **123**(2), 256–270 (2000)

48. Chen, C.: A fuzzy approach to select the location of the distribution center. Fuzzy Sets and Systems **118**(1), 65–73 (2001)
49. Chen, C., Lee, W.: Multi-objective optimization of multi-echelon supply chain networks with uncertain product demands and prices. Computers & chemical engineering **28**(6–7), 1131–1144 (2004)
50. Chen, H., Flann, N., Watson, D.: Parallel genetic simulated annealing: a massively parallel SIMD algorithm. IEEE Transactions on Parallel and Distributed Systems, pp. 126–136 (1998)
51. Chen, K., Li, T., Cao, T.: Tribe-PSO: A novel global optimization algorithm and its application in molecular docking. Chemometrics and Intelligent Laboratory Systems **82**(1–2), 248–259 (2006)
52. Chen, S., Luk, B., Liu, Y.: Application of adaptive simulated annealing to blind channel identification with HOC fitting. Electronics Letters **34**(3), 234–235 (2002)
53. Cheung, R., Powell, W.: Models and algorithms for distribution problems with uncertain demands. Transportation Science **30**(1), 43 (1996)
54. Choi, B., Yoon, S., Chung, S.: Minimizing maximum completion time in a proportionate flow shop with one machine of different speed. European Journal of Operational Research **176**(2), 964–974 (2007)
55. Choo E.V., A.D.: An interactive algorithm for multiobjective programming. Computers and Operations Reasearch **7**, 81–88 (1980)
56. Choudhary, A., Ashutosh Singh, K., Tiwari, M.: A statistical tolerancing approach for design of synchronized supply chains. Robotics and Computer-Integrated Manufacturing **22**(4), 315–321 (2006)
57. Chouinard, M., D'Amours, S., Aït-Kadi, D.: A stochastic programming approach for designing supply loops. International Journal of Production Economics **113**(2), 657–677 (2008)
58. Clerc, M.: 'TRIBES-un exemple d' optimisation par essaim particulaire sans parametres de contrôle. Optimisation par Essaim Particulaire (OEP 2003), Paris, France (2003)
59. Clerc, M.: Stagnation analysis in particle swarm optimisation or what happens when nothing happens. alea **5**, 0 (2006)
60. Clerc, M., Kennedy, J.: The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE transactions on Evolutionary Computation **6**(1), 58–73 (2002)
61. Cochrane, J., Zeleny, M.: Multiple criteria decision making. Univ of South Carolina Pr (1973)
62. Cohn, D.: Measure theory. Birkhäuser (1980)
63. Cohon, J.: Multicriteria programming: Brief review and application. Design optimization, pp. 163–191 (1985)
64. Cohon, J.: Multiobjective programming and planning. Dover Pubns (2004)
65. Colubi, A., Fernandez-Garcia, C., Gil, M.: Simulation of random fuzzy variables: an empirical approach to statistical/probabilistic studies with fuzzy experimental data. Fuzzy Systems, IEEE Transactions on **10**(3), 384–390 (2002)
66. Crainic, T., Gendreau, M., Toulouse, M., Centre for Research on Transportation (Montréal, Q.: Towards a taxonomy of parallel tabu search algorithms. Centre for Research on Transportation= Centre de recherche sur les transports (1993)
67. Crainic, T., Toulouse, M., Gendreau, M.: Toward a taxonomy of parallel tabu search heuristics. INFORMS Journal on Computing **9**(1), 61–72 (1997)
68. Cvijovic, D., Klinowski, J.: Taboo search: an approach to the multiple minima problem. Science (Washington) **267**(5198), 664–664 (1995)
69. Czech, Z., Czarnas, P.: Parallel simulated annealing for the vehicle routing problem with time windows. In: Parallel, Distributed and Network-based Processing, 2002. Proceedings. 10th Euromicro Workshop on, pp. 376–383. IEEE (2002)
70. Czyzak, P., Hapke, M., Jaszkiewicz, A.: Application of the Pareto-simulated annealing to the multiple criteria shortest path problem. Tech. rep., Technical Report, Politechnika Poznanska Instytut Informatyki, Poland (1994)
71. Czyż, P., et al.: Pareto simulated annealing-a metaheuristic technique for multiple-objective combinatorial optimization. Journal of Multi-Criteria Decision Analysis **7**(1), 34–47 (1998)

72. Dagpunar, J.: Principles of random variate generation. Oxford University Press, USA (1988)
73. De, P., Ghosh, J., Wells, C.: Heuristic estimation of the efficient frontier for a bi-criteria scheduling problem. Decision Sciences **23**(3), 596–609 (1992)
74. De Jong, K.: Analysis of the behavior of a class of genetic adaptive systems (1975)
75. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE transactions on evolutionary computation **6**(2), 182–197 (2002)
76. Dempster, M.: Stochastic programming: based on proceedings. Academic Press (1980)
77. Devroye, L., Devroye, L.: Non-uniform random variate generation. Citeseer (1986)
78. Diekmann, R., Luling, R., Simon, J.: A general purpose distributed implementation of simulated annealing. In: Parallel and Distributed Processing, 1992. Proceedings of the Fourth IEEE Symposium on, pp. 94–101. IEEE (2002)
79. Dietz, A., Aguilar-Lasserre, A., Azzaro-Pantel, C., Pibouleau, L., Domenech, S.: A fuzzy multiobjective algorithm for multiproduct batch plant: Application to protein production. Computers & Chemical Engineering **32**(1–2), 292–306 (2008)
80. Dobrushin, R., Major, P.: Non-central limit theorems for non-linear functional of Gaussian fields. Probability Theory and Related Fields **50**(1), 27–52 (1979)
81. Dogan, K., Goetschalckx, M.: A primal decomposition method for the integrated design of multi-period production–distribution systems. IIE Transactions **31**(11), 1027–1036 (1999)
82. Dos Santos Coelho, L., Alotto, P.: Tribes Optimization Algorithm Applied to the Loney's Solenoid. Magnetics, IEEE Transactions on **45**(3), 1526–1529 (2009)
83. Dowsland, K.: Some experiments with simulated annealing techniques for packing problems. European Journal of Operational Research **68**, 389–399 (1993)
84. Dubois, D., Prade, H.: Operations on fuzzy numbers. International Journal of Systems Science **9**(6), 613–626 (1978)
85. DuBois, D., Prade, H.: Fuzzy sets and systems: theory and applications. Academic Press (1980)
86. Dubois, D., Prade, H.: Fuzzy numbers: an overview. Analysis of fuzzy information, Mathematics and Logics **1**, 3–39 (1987)
87. Dubois, D., Prade, H.: Possibility theory: an approach to computerized processing of uncertainty. Plenum Press, New York (1988)
88. Dubois, D., Prade, H.: Rough Fuzzy Sets and Fuzzy Rough Sets. International Journal of general systems **17**(2), 191–209 (1990)
89. Dudley, R.: Central limit theorems for empirical measures. The Annals of Probability **6**(6), 899–929 (1978)
90. Dudley, R.: Uniform central limit theorems. Cambridge Univ Press (1999)
91. Durrett, R., Durrett, R.: Probability: theory and examples. Cambridge Univ Pr (2010)
92. Dutta, P., Chakraborty, D., Roy, A.: A single-period inventory model with fuzzy random variable demand. Mathematical and Computer Modelling **41**(8–9), 915–922 (2005)
93. Eberhart, R., Shi, Y., Kennedy, J.: Swarm intelligence. Morgan Kaufmanns (2001)
94. Erenguc, S., Simpson, N., Vakharia, A.: Integrated production/distribution planning in supply chains: An invited review. European journal of operational research **115**(2), 219–236 (1999)
95. Evans, L., Gariepy, R.: Measure theory and fine properties of functions. CRC (1992)
96. Fei, W.: Existence and uniqueness of solution for fuzzy random differential equations with non-Lipschitz coefficients. Information Sciences **177**(20), 4329–4337 (2007)
97. Feller, W.: An introduction to probability theory and its application (1971)
98. Ferrero, A., Salicone, S.: The random-fuzzy variables: A new approach to the expression of uncertainty in measurement. Instrumentation and Measurement, IEEE Transactions on **53**(5), 1370–1377 (2004)
99. Fiechter, C.: A parallel tabu search algorithm for large traveling salesman problems. Discrete Applied Mathematics **51**(3), 243–267 (1994)
100. Fieldsend, J., Singh, S.: A Multi-Objective Algorithm based upon Particle Swarm Optimisation, an Efficient Data Structure and Turbulence. (2002)
101. Fishman, G.: Principles of discrete event simulation. John Wiley & Sons, Inc. New York, NY, USA (1978)

102. Fishman, G.: Monte Carlo: concepts, algorithms, and applications. Springer (1996)
103. Fonseca, C., Fleming, P.: An overview of evolutionary algorithms in multiobjective optimization. Evolutionary computation **3**(1), 1–16 (1995)
104. Gál, T., Stewart, T., Hanne, T.: Multicriteria decision making: advances in MCDM models, algorithms, theory, and applications. Springer Netherlands (1999)
105. Gangadharan, R., Rajendran, C.: A simulated annealing heuristic for scheduling in a flowshop with bicriteria. Computers & Industrial Engineering **27**(1–4), 473–476 (1994)
106. Gelfand, S., Mitter, S.: Recursive stochastic algorithms for global optimization in Rd. SIAM Journal on Control and Optimization **29**(5), 999–1018 (1991)
107. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. Journal of Applied Statistics **20**(5), 25–62 (1993)
108. Gen, M., Cheng, R.: Genetic algorithms and engineering design. Wiley-Interscience (1997)
109. Gen, M., Cheng, R.: Genetic algorithms and engineering optimization. Wiley-interscience (2000)
110. Gendreau M., et al.: Parallel tabu search for real-time vehicle routing and dispatching. Transportation Science **33**(4), 381–390 (1999)
111. Gendreau, M., Laporte, G., Semet, F.: A dynamic model and parallel tabu search heuristic for real-time ambulance relocation. Parallel Computing **27**(12), 1641–1653 (2001)
112. Geoffrion A.M., G.G.: Multicommodity Distribution System Design by Benders Decomposition. A Long View of Research and Practice in Operations Research and Management Science, pp. 35–61 (2010)
113. Glauber, R.: Time-dependent statistics of the Ising model. Journal of Mathematical Physics (New York)(US) **4** (1963)
114. Glover, F.: Future paths for integer programming and links to artificial intelligence. Computers & Operations Research **13**(5), 533–549 (1986)
115. Goicoechea, A., Hansen, D., Duckstein, L.: Multiobjective decision analysis with engineering and business applications. Wiley, New York (1982)
116. Goldberg, D.: Genetic algorithms in search, optimization, and machine learning. Addison-wesley (1989)
117. González, A.: A study of the ranking function approach through mean values. Fuzzy Sets and Systems **35**(1), 29–41 (1990)
118. Grabowski, J., Pempera, J.: Some local search algorithms for no-wait flow-shop problem with makespan criterion. Computers & Operations Research **32**(8), 2197–2212 (2005)
119. Greco, S., Matarazzo, B., Slowinski, R.: Rough sets theory for multicriteria decision analysis. European journal of operational research **129**(1), 1–47 (2001)
120. Greco, S., Matarazzo, B., Slowinski, R.: Rough sets methodology for sorting problems in presence of multiple attributes and criteria. European Journal of Operational Research **138**(2), 247–259 (2002)
121. Greening, D.: Parallel simulated annealing techniques. Physica D: Nonlinear Phenomena **42**(1–3), 293–306 (1990)
122. Gupta, J., Stafford, E., et al.: Flowshop scheduling research after five decades. European Journal of Operational Research **169**(3), 699–711 (2006)
123. Hadley, G., Whitin, T.: Analysis of inventory systems. Prentice Hall (1963)
124. Haimes, Y., Hall, W.: Multiobjectives in water resource systems analysis: The surrogate worth trade off method. Water Resources Research **10**(4), 615–624 (1974)
125. Haimes, Y., Lasdon, L., Wismer, D.: On a bicriterion formulation of the problems of integrated system identification and system optimization. IEEE Transactions on Systems, Man, and Cybernetics **1**(3), 296–297 (1971)
126. Haldar, A., Reddy, R.: A random-fuzzy analysis of existing structures. Fuzzy Sets and Systems **48**(2), 201–210 (1992)
127. Halmos, P.: Measure theory. Van Nostrad Reinhold Company (1950)
128. Halmos, P.: Naive set theory. Springer (1974)
129. Hansen, E.: Global optimization using interval analysis: the one-dimensional case. Journal of Optimization Theory and Applications **29**(3), 331–344 (1979)

130. Hansen, E.: Global optimization using interval analysis – the multi-dimensional case. Numerische Mathematik **34**(3), 247–270 (1980)
131. Hansen, P.: The steepest ascent mildest descent heuristic for combinatorial programming. In: Congress on Numerical Methods in Combinatorial Optimization, Capri, Italy, pp. 70–145 (1986)
132. Harkness, J., ReVelle, C.: Facility location with increasing production costs. European Journal of Operational Research **145**(1), 1–13 (2003)
133. Hasuike, T., Katagiri, H., Ishii, H.: Probability maximization model of 0–1 knapsack problem with random fuzzy variables. In: Fuzzy Systems, 2008. FUZZ-IEEE 2008.(IEEE World Congress on Computational Intelligence). IEEE International Conference on, pp. 548–554. IEEE (2008)
134. Ho S.L., e.a.: A particle swarm optimization method with enhanced global search ability for design optimizations of electromagnetic devices. Magnetics, IEEE Transactions on **42**(4), 1107–1110 (2006)
135. Holland, J.: Adaptation in natural and artificial systems. MIT Press Cambridge, MA, USA (1992)
136. Hu J., e.a.: Adaptive hierarchical fair competition (ahfc) model for parallel evolutionary algorithms. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 772–779. Morgan Kaufmann Publishers Inc. (2002)
137. Huang, X.: Optimal project selection with random fuzzy parameters. International journal of production economics **106**(2), 513–522 (2007)
138. Hwang, C., Masud, A.: Md.,(1979). Multiple Objective Decision Making Methods and Applications
139. Ignizio, J.: Goal programming and extensions. Lexington Books Lexington, MA (1976)
140. Ijiri, Y.: Management goals and accounting for control. North-Holland Amsterdam (1965)
141. Ingber, L.: Very fast simulated re-annealing. Mathematical and Computer Modelling **12**(8), 967–973 (1989)
142. Ingber, L.: Simulated annealing: Practice versus theory. Mathematical and computer modelling **18**(11), 29–57 (1993)
143. Ingber, L.: Adaptive simulated annealing (ASA): Lessons learned. Arxiv preprint cs/0001018 (2000)
144. Ishibuchi, H., Murata, T.: A multi-objective genetic local search algorithm and its application to flowshop scheduling. IEEE Transactions on Systems, Man, and CyberneticsPart C: Applications and Reviews **28**(3) (1998)
145. Ishii, H., Konno, T.: A stochastic inventory problem with fuzzy shortage cost. European Journal of Operational Research **106**(1), 90–94 (1998)
146. Ishii, H., Nishida, T.: Stochastic linear knapsack problem: probability maximization model. Mathematica Japonica **29**, 273–281 (1984)
147. Jacobs, K., Kurzweil, J.: Measure and integral. Academic Press (1978)
148. Jacod, J., Shiryaev, A.: Limit theorems for stochastic processes. Springer Berlin (1987)
149. Johnson, S.: Optimal two-and three-stage production schedules with setup times included. Naval research logistics quarterly **1**(1), 61–68 (1954)
150. Kabe, D.: On some coverage probability maximization problems. Communications in Statistics-Simulation and Computation **9**(1), 73–79 (1980)
151. Kall, P., Mayer, J.: Stochastic linear programming: models, theory, and computation. Springer (2005)
152. Kall, P., Wallace, S., Kall, P.: Stochastic programming. Citeseer (1994)
153. Kao, C., Hsu, W.: A single-period inventory model with fuzzy demand. Computers & Mathematics with Applications **43**(6–7), 841–848 (2002)
154. Kaufmann, A.: Introduction to the Theory of Fuzzy Subsets. Academic Press, New York (1975)
155. Kelley, J., Srinivasan, T.: Measure and integral. Springer (1988)
156. Kendall, K., Lee, S.: Formulating blood rotation policies with multiple objectives. Management Science **26**(11), 1145–1157 (1980)

157. Kennedy, J., Eberhart, R.: A discrete binary version of the particle swarm algorithm. In: Systems, Man, and Cybernetics, 1997.'Computational Cybernetics and Simulation'., 1997 IEEE International Conference on, vol. 5, pp. 4104–4108. IEEE (2002)
158. Kennedy, J., Eberhart, R., et al.: Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks, vol. 4, pp. 1942–1948. Perth, Australia (1995)
159. Kennedy, J., Mendes, R.: Population structure and particle swarm performance. Proceedings of the Evolutionary Computation on pp. 1671–1676 (2002)
160. Kirkpatrick, S.: Optimization by simulated annealing: Quantitative studies. Journal of Statistical Physics **34**(5), 975–986 (1984)
161. Kirkpatrick, S., Gelatt Jr, C., Vecchi, M.: Optimization by simulated annealing. Science **220**(4598), 671 (1983)
162. Klement, E., Puri, M., Ralescu, D.: Limit theorems for fuzzy random variables. Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences **407**(1832), 171–182 (1986)
163. Kliewer, G., Tschoke, S.: A general parallel simulated annealing library and its application in airline industry. In: Parallel and Distributed Processing Symposium, 2000. IPDPS 2000. Proceedings. 14th International, pp. 55–61. IEEE (2002)
164. Klose, A., Drexl, A.: Facility location models for distribution system design. European Journal of Operational Research **162**(1), 4–29 (2005)
165. Knuth, D.: The art of computer programming. Reading, Ma (1973)
166. Ko, H., Ko, C., Kim, T.: A hybrid optimization/simulation approach for a distribution network design of 3PLS. Computers & Industrial Engineering **50**(4), 440–449 (2006)
167. Kolmogorov, A.: Foundations of the theory of probability. (1950)
168. Kolmogorov, A.: Foundations of the Theory of Probability. Chelsea Pub. Co. (1956)
169. Kondakci, S., Azizoglu, M., Köksalan, M.: Note: Bicriteria scheduling for minimizing flow time and maximum tardiness. Naval Research Logistics **43**(6), 929–936 (1996)
170. Koopmans, T.: Activity analysis of production and allocation. New York (1951)
171. Körner, R.: Linear models with random fuzzy variables. Disseration, Technische Universität Bergakademie Freiberg, Freiberg (1997)
172. Körner, R., Näther, W.: Linear regression with random fuzzy variables: extended classical estimates, best linear estimates, least squares estimates. Information Sciences **109**(1–4), 95–118 (1998)
173. Korner, R., Nather, W.: On the variance of random fuzzy variables. Studies in Fuzziness and Soft Computing **87**, 25–42 (2002)
174. Koumousis, V., Katsaras, C.: A saw-tooth genetic algorithm combining the effects of variable population size and reinitialization to enhance performance. Evolutionary Computation, IEEE Transactions on **10**(1), 19–28 (2006)
175. Krickeberg, K.: Probability theory. Addison-Wesley Pub. Co. (1965)
176. Kruse, R., Meyer, K.: Statistics with vague data. Springer (1987)
177. Krusinacute, E., et al.: Discriminant versus rough sets approach to vague data analysis. Applied Stochastic Models and Data Analysis **8**(1), 43–56 (1992)
178. Kulak, O., Kahraman, C.: Fuzzy multi-attribute selection among transportation companies using axiomatic design and analytic hierarchy process. Information Sciences **170**(2–4), 191–210 (2005)
179. Kushner, H.: Asymptotic global behavior for stochastic approximation and diffusions with slowly decreasing noise effects: Global minimization via Monte Carlo. SIAM Journal on Applied Mathematics **47**(1), 169–185 (1987)
180. Kwakernaak, H.: Fuzzy random variables–I. Definitions and theorems. Information Sciences **15**(1), 1–29 (1978)
181. Kwakernaak, H.: Fuzzy random variables–II. Algorithms and examples for the discrete case. Information Sciences **17**(3), 253–278 (1979)
182. Laarhoven, P.: Theoretical and computational aspects of simulated annealing. Centrum voor Wiskunde en Informatica (1988)
183. Laarhoven, P., van Laarhoven, P., Aarts, E.: Simulated annealing: theory and applications. Springer (1987)

184. Laha, R., Rohatgi, V.: Probability Theory. Wiley, New York (1985)
185. Lang, S.: Algebraic numbers. Addison-Wesley Pub. Co. (1964)
186. Laporte, G., Louveaux, F., van Hamme, L.: Exact solution to a location problem with stochastic demands. Transportation Science **28**(2), 95–103 (1994)
187. Laskari, E., Parsopoulos, K., Vrahatis, M.: Particle swarm optimization for integer programming. In: Proceedings of the IEEE 2002 congress on evolutionary computation, pp. 1576–1581. Citeseer (2002)
188. Law, A., Kelton, W., Kelton, W.: Simulation modeling and analysis. McGraw-Hill New York (1991)
189. Lee, S.: Goal programming for decision analysis. Auerbach Publishers (1972)
190. Lee, S., Lee, K.: Synchronous and asynchronous parallel simulated annealing with multiple Markov chains. Parallel and Distributed Systems, IEEE Transactions on **7**(10), 993–1008 (2002)
191. Lei, W., Qi, K., Hui, X., Qidi, W.: A modified adaptive particle swarm optimization algorithm. In: Industrial Technology, 2005. ICIT 2005. IEEE International Conference on, pp. 209–214. IEEE (2006)
192. Lempel, R., Moran, S.: The stochastic approach for link-structure analysis (SALSA) and the TKC effect1. Computer Networks **33**(1–6), 387–401 (2000)
193. Lewis, C.: Scientific inventory control. Butterworths, London (1970)
194. Lewis, P., Shedler, G.: Simulation of nonhomogeneous Poisson processes by thinning. Naval Research Logistics Quarterly **26**(3), 403–413 (1979)
195. Li N., C.J.L.Y.: A parallel simulated annealing algorithm based on functional feature tree modeling for 3D engineering layout design. Applied Soft Computing **10**, 592–601 (2010)
196. Li, D., Gao, L., Zhang, J., Li, Y.: Power system reactive power optimization based on adaptive particle swarm optimization algorithm. In: Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on, vol. 2, pp. 7572–7576. IEEE (2006)
197. Li, J., Xu, J., Gen, M.: A class of multiobjective linear programming model with fuzzy random coefficients. Mathematical and Computer Modelling **44**(11–12), 1097–1113 (2006)
198. Li, L., Kabadi, S., Nair, K.: Fuzzy models for single-period inventory problem. Fuzzy sets and systems **132**(3), 273–289 (2002)
199. Li, R.: Multiple objective decision making in a fuzzy environment. Ph.D. thesis, Kansas State University (1990)
200. Li, X.: A non-dominated sorting particle swarm optimizer for multiobjective optimization. In: Genetic and Evolutionary ComputationłGECCO 2003, pp. 198–198. Springer (2003)
201. Lian, Z., Jiao, B., Gu, X.: A similar particle swarm optimization algorithm for job-shop scheduling to minimize makespan. Applied Mathematics and Computation **183**(2), 1008–1017 (2006)
202. Listes, O., Dekker, R.: Stochastic approaches for product recovery network design: a case study. Econometric Institute Report EI **8** (2001)
203. Listes, O., Dekker, R.: A stochastic approach to a case study for product recovery network design. European Journal of Operational Research **160**(1), 268 (2005)
204. Liu, B.: Dependent-chance programming: A class of stochastic programming. Computers & Mathematics with Applications **34**(12), 89–104 (1997)
205. Liu, B.: Random fuzzy dependent-chance programming and its hybrid intelligent algorithm. Information Sciences **141**(3–4), 259–271 (2002)
206. Liu, B.: Theory and practice of uncertain programming. Physica Verlag (2002)
207. Liu, B.: Uncertainty theory: an introduction to its axiomatic foundations. Springer Verlag (2004)
208. Liu, Q., Xu, J.: A study on facility location–allocation problem in mixed environment of randomness and fuzziness. Journal of Intelligent Manufacturing, pp. 1–10 (2009)
209. Liu, Y., Liu, B.: Expected value operator of random fuzzy variable and random fuzzy expected value models. International Journal of Uncertainty Fuzziness and Knowledge Based Systems **11**(2), 195–216 (2003)
210. López Díaz, M., Gil, M.: An extension of Fubini's theorem for fuzzy random variables. Information Sciences **115**(1–4), 29–41 (1999)

211. Loukil, T., Teghem, J., Tuyttens, D.: Solving multi-objective production scheduling problems using metaheuristics. European journal of operational research **161**(1), 42–61 (2005)
212. Luhandjula, M.: Multiple objective programming problems with possibilistic coefficients. Fuzzy Sets and Systems **21**(2), 135–145 (1987)
213. Maiti, A., Maiti, M., Maiti, M.: Two storage inventory model with random planning horizon. Applied Mathematics and Computation **183**(2), 1084–1097 (2006)
214. Malek, M., Guruswamy, M., Pandya, M., Owens, H.: Serial and parallel simulated annealing and tabu search algorithms for the traveling salesman problem. Annals of Operations Research **21**(1), 59–84 (1989)
215. Malinowski, M.: On random fuzzy differential equations. Fuzzy Sets and Systems **160**, 3152–3165 (2009)
216. McLeish, D.: Dependent central limit theorems and invariance principles. The Annals of Probability **2**(4), 620–628 (1974)
217. Meise, C.: On the convergence of parallel simulated annealing. Stochastic Processes and their Applications **76**(1), 99–115 (1998)
218. Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E., et al.: Equation of state calculations by fast computing machines. The journal of chemical physics **21**(6), 1087 (1953)
219. Michalewicz, Z.: Genetic Algorithms Plus Data Structures Equals Evolution Programs. Springer-Verlag New York, Inc. Secaucus, NJ, USA (1994)
220. Mirchandani, P., Oudjit, A., Wong, R.: Multidimensional extensions and a nested dual approach for the m-median problem. European Journal of Operational Research **21**(1), 121–137 (1985)
221. MirHassani, S.e.a.: Computational solution of capacity planning models under uncertainty. Parallel Computing **26**(5), 511–538 (2000)
222. Modares, H., Alfiand, M., et al.: Parameter estimation of bilinear systems based on an adaptive particle swarm optimization. Engineering Applications of Artificial Intelligence (2010)
223. Monk, J.: Introduction to set theory. McGraw-Hill (1969)
224. Moore, J., Chapman, R.: Application of particle swarm to multiobjective optimization. Department of Computer Science and Software Engineering, Auburn University (1999)
225. Moreno Pérez, J., Moreno Vega, J., Verdegay, J.: Fuzzy location problems on networks. Fuzzy sets and systems **142**(3), 393–405 (2004)
226. Morgan, B.: Elements of simulation. Chapman & Hall/CRC (1984)
227. Mori, H., Goto, Y.: A parallel tabu search based method for determining optimal allocation of FACTS in power systems. In: Power System Technology, 2000. Proceedings. PowerCon 2000. International Conference on, vol. 2, pp. 1077–1082. IEEE (2002)
228. Murata, T., Ishibuchi, H., Tanaka, H.: Multi-objective genetic algorithm and its applications to flowshop scheduling. Computers & Industrial Engineering **30**(4), 957–968 (1996)
229. Naddor, E.: Inventory systems. Wiley, New York (1966)
230. Nador, E.: Inventory systems (1996)
231. Nahmias, S.: Fuzzy variables. Fuzzy sets and systems **1**(2), 97–110 (1978)
232. Nakayama, H.: Use of random noise in problems of optimization and learning. Problemy Peredaci Informacii, (in Russian) **1**, 113–117 (1965)
233. Nakayama, H.: Proposal of satisfying trade-off method for multiobjective programming. Trans. Soc. Inst. Contorl Eng. (in Japanese) **20**, 29–53 (1984)
234. Nakayama, H.: Basic Handbook of Operation (1999)
235. Natarajan, K.: Graph-partitioning on shared-memory multiprocessor systems. In: Proceedings of the 1991 International Conference on Parallel Processing August 12–16, 1991: Algorithms & Applications. CRC (1991)
236. Näther, W.: On random fuzzy variables of second order and their application to linear statistical inference with fuzzy data. Metrika **51**(3), 201–221 (2000)
237. Nijkamp, P., Spronk, J.: Interactive multiple goal programming (1978)
238. Okada, S., Soper, T.: A shortest path problem on a network with fuzzy arc lengths. Fuzzy Sets and Systems **109**(1), 129–140 (2000)

239. Onwubolu, G.: Optimization of milling operations for the selection of cutting conditions using Tribes. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture **219**(10), 761–771 (2005)
240. Onwubolu, G., Babu, B.: TRIBES application to the flow shop scheduling problem. New Optimization Techniques in Engineering **141**, 517–536
241. Orosz, J., Jacobson, S.: Analysis of static simulated annealing algorithms. Journal of Optimization theory and Applications **115**(1), 165–182 (2002)
242. Osyczka, A.: Multicriterion optimization in engineering with FORTRAN programs. JOHN WILEY & SONS, INC., 605 THIRD AVE., NEW YORK, NY 10158, USA, 1984, 200 (1984)
243. Pan, J., Chen, J., Chao, C.: Minimizing tardiness in a two-machine flow-shop. Computers & Operations Research **29**(7), 869–885 (2002)
244. Pan, Q., Fatih Tasgetiren, M., Liang, Y.: A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem. Computers & Operations Research **35**(9), 2807–2839 (2008)
245. Panigrahi, B., Ravikumar Pandi, V., Das, S.: Adaptive particle swarm optimization approach for static and dynamic economic load dispatch. Energy conversion and management **49**(6), 1407–1415 (2008)
246. Pareto, V., di Economica Polittica, M.: Societa Editrice Libraia. Milan, Italy (1906)
247. Parsopoulos, K., Vrahatis, M.: Particle swarm optimization method in multiobjective problems. In: Proceedings of the 2002 ACM symposium on applied computing, pp. 603–607. ACM (2002)
248. Parsopoulos, K., Vrahatis, M.: On the computation of all global minimizers through particle swarm optimization. Evolutionary Computation, IEEE Transactions on **8**(3), 211–224 (2004)
249. Pawlak, Z.: Rough probability. Bulletin of the Polish academy of sciences. Mathematics **32**(9–10), 607–612 (1984)
250. Pawlak, Z.: Rough sets and fuzzy sets. Fuzzy sets and Systems **17**(1), 99–102 (1985)
251. Peng, J., Liu, B.: Birandom variables and birandom programming. Computers & Industrial Engineering **53**(3), 433–453 (2007)
252. Pfeiffer, P.: Probability for applications. Springer Verlag (1990)
253. Polkowski, L., Skowron, A.: Rough mereology. Methodologies for Intelligent Systems, pp. 85–94 (1994)
254. Porto, S., Ribeiro, C.: Parallel tabu search message-passing synchronous strategies for task scheduling under precedence constraints. Journal of heuristics **1**(2), 207–223 (1996)
255. Prem, P., et al.: Analysis of multi-item inventory systems under resource constraints: A nonlinear goal programming approach. Engineering Costs and Production Economics **20**(2), 121–127 (1990)
256. Puri, M., Ralescu, D.: Fuzzy random variables. Journal of Mathematical Analysis and Applications **114**(2), 409–422 (1986)
257. Reeves, C.: A genetic algorithm for flowshop sequencing. Computers & Operations Research **22**(1), 5–13 (1995)
258. Rego, C., Roucairol, C.: A parallel tabu search algorithm using ejection chains for the vehicle routing problem. Meta-heuristics: Theory and applications **1**, 661–675 (1996)
259. Reyes-Sierra, M., Coello, C.: Multi-objective particle swarm optimizers: A survey of the state-of-the-art. International Journal of Computational Intelligence Research **2**(3), 287–308 (2006)
260. Rezaei, J., Davoodi, M.: A deterministic, multi-item inventory model with supplier selection and imperfect quality. Applied Mathematical Modelling **32**(10), 2106–2116 (2008)
261. Riezebos, J., Gaalman, G.: A single-item inventory model for expected inventory order crossovers. International Journal of Production Economics **121**(2), 601–609 (2009)
262. Ripley, B.: Computer generation of random variables: a tutorial. International Statistical Review/Revue Internationale de Statistique pp. 301–319 (1983)
263. Ripley, B.: Stochastic simulation. Wiley New York et al. (1987)
264. Romeijn, H., Shu, J., Teo, C.: Designing two-echelon supply networks. European Journal of Operational Research **178**(2), 449–462 (2007)

265. Rommelfanger, H.: Fuzzy linear programming and applications. European Journal of Operational Research **92**(3), 512–527 (1996)
266. Rosenblatt, M.: A central limit theorem and a strong mixing condition. Proceedings of the National Academy of Sciences of the United States of America **42**(1), 43 (1956)
267. Ross, S.: Simulation. 3rd Edition, Posts & Telecom Press, Peking (2006)
268. Roussel-Ragot, P., Dreyfus, G.: A problem independent parallel implementation of simulated annealing: models and experiments. Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on **9**(8), 827–835 (2002)
269. Rubinstein, R., Kroese, D.: Simulation and the Monte Carlo method. Wiley-interscience (2008)
270. Sakawa, M.: Interactive fuzzy goal programming for multiobjective nonlinear problems and its application to water quality management. Control and Cybernetics **13**(2), 217–228 (1984)
271. Sakawa, M.: Fuzzy sets and interactive multiobjective optimization. Plenum Press New York (1993)
272. Sakawa, M., Kato, K., Katagiri, H.: An interactive fuzzy satisficing method for multiobjective linear programming problems with random variable coefficients through a probability maximization model. Fuzzy sets and systems **146**(2), 205–220 (2004)
273. Sakawa, M., Kato, K., Nishizaki, I.: An interactive fuzzy satisficing method for multiobjective stochastic linear programming problems through an expectation model. European journal of operational research **145**(3), 665–672 (2003)
274. Sakawa, M., Yano, H.: Interactive fuzzy decision making for multiobjective nonlinear programming using augmented minimax problems. Fuzzy Sets and Systems **20**(1), 31–43 (1986)
275. Sakawa, M., Yano, H., Yumine, T.: An interactive fuzzy satisficing method for multiobjective linear-programming problems and its application. IEEE Transactions on Systems, Man and Cybernetics **17**(4), 654–661 (1987)
276. Santoso, T., Ahmed, S., Goetschalckx, M., Shapiro, A.: A stochastic programming approach for supply chain network design under uncertainty. European Journal of Operational Research **167**(1), 96–115 (2005)
277. Sawaragi, Y., Nakayama, H., Tanino, T.: Theory of multiobjective optimization. Academic Press Orlando (1997)
278. Sawaragi, Y., Nakayama, H., Tanino, T., service), S.O.: Theory of multiobjective optimization. Academic Press Orlando (1985)
279. Schmeiser, B.: Random Variate Generation: A Survey. (1980)
280. Schneider, M.: Probability maximization in networks. In: Proc. Int. Conf. Transportation Research, pp. 748–755 (1973)
281. Sengupta, J.: Stochastic programming: Methods and applications. North-Holland Pub. Co. (1973)
282. Seo, F., Sakawa, M.: Multiple criteria decision analysis in regional planning: concepts, methods, and applications. D Reidel Pub Co (1988)
283. Sha, D., Che, Z.: Supply chain network design: partner selection and production/distribution planning using a systematic model. Journal of the Operational Research Society **57**(1), 52–62 (2005)
284. Shao, Z., Ji, X.: Fuzzy multi-product constraint newsboy problem. Applied Mathematics and Computation **180**(1), 7–15 (2006)
285. Shen, Q., Zhao, R., Tang, W.: Random fuzzy alternating renewal processes. Soft Computing-A Fusion of Foundations, Methodologies and Applications **13**(2), 139–147 (2009)
286. Shen, Z.: A profit-maximizing supply chain network design model with demand choice flexibility. Operations Research Letters **34**(6), 673–682 (2006)
287. Shimizu, K.: Theory of Multiobjective and Conflict. Kyoritsu Syuppan,(in Japanese) (1982)
288. Shiryaev, A.: Probability, vol. 95 of Graduate Texts in Mathematics (1996)
289. Siarry, P., Berthiau, G.: Fitting of tabu search to optimize functions of continuous variables. International Journal for Numerical Methods in Engineering **40**(13), 2449–2457 (1997)

290. da Silva, L., Fernandes Pereira, R., Abbad, J., Sanches Mantovani, J.: Optimised placement of control and protective devices in electric distribution systems through reactive tabu search algorithm. Electric Power Systems Research **78**(3), 372–381 (2008)
291. Silver, E., Peterson, R.: Decision systems for inventory management and production planning. Wiley New York (1979)
292. Skowron, A., Grzymalła-Busse, J.: From rough set theory to evidence theory. In: Advances in the Dempster-Shafer Theory of Evidence, p. 236. John Wiley & Sons, Inc. (1994)
293. Skowron, A., Rauszer, C.: The discernibility matrices and functions in information systems. Intelligent Decision Support **362** (1992)
294. Slowiński, K., Slnowiński, R., Stefanowski, J.: Rough sets approach to analysis of data from peritoneal lavage in acute pancreatitis. Informatics for Health and Social Care **13**(3), 143–159 (1988)
295. Slowinski, R.: Rough sets. International Journal of Information and Computer Sciences **11**(5), 341–356 (1982)
296. Slowinski, R.: Rough set processing of fuzzy information. Soft Computing: Rough Sets, Fuzzy Logic, Neural Networks, Uncertainty Management, Knowledge Discovery. Simulation Councils, San Diego, CA, pp. 142–145 (1995)
297. Slowinski, R., Vanderpooten, D., et al.: A generalized definition of rough approximations based on similarity. IEEE Transactions on Knowledge and Data Engineering **12**(2), 331–336 (2000)
298. Spronk, J., Telgen, J.: An ellipsoidal interactive multiple goal programming method. Lecture notes in economics and mathematical systems **190**, 380–7 (1981)
299. Srivastava, S.: A course on Borel sets. Springer Verlag (1998)
300. Stadler, W.: Sufficient conditions for preference optimality. Journal of Optimization Theory and Applications **18**(1), 119–140 (1976)
301. Stancu-Minasian, I., Giurgiutiu, V.: Stochastic programming with multiple objective functions. Editura Academiei (1984)
302. Steel, J.: A review of Laarhoven. J. Amer. Statist. Ass. **85**, 596–596 (1990)
303. Steuer, R.: Multiple criteria optimization: Theory, computation, and application. Wiley New York (1986)
304. Suman, B.: Multiobjective simulated annealing-a metaheuristic technique for multiobjective optimization of a constrained problem. Foundations of Computing and Decision Sciences **27**(3), 171–191 (2002)
305. Suman, B.: Simulated annealing-based multiobjective algorithms and their application for system reliability. Engineering Optimization **35**(4), 391–416 (2003)
306. Suman, B.: Study of simulated annealing based algorithms for multiobjective optimization of a constrained problem. Computers & chemical engineering **28**(9), 1849–1871 (2004)
307. Suppapitnarm, A., Parks, G.: Simulated annealing: an alternative approach to true multiobjective optimization (1999)
308. Syam, S.: A model and methodologies for the location problem with logistical components. Computers & Operations Research **29**(9), 1173–1193 (2002)
309. Syarif, A., Yun, Y., Gen, M.: Study on multi-stage logistic chain network: a spanning tree-based genetic algorithm approach. Computers & Industrial Engineering **43**(1–2), 299–314 (2002)
310. Syswerda, G.: A study of reproduction in generational and steady-state genetic algorithms. Foundations of genetic algorithms **94**, 101 (1991)
311. Szidarovszky, F., Gershon, M., Duckstein, L.: Techniques for multiobjective decision making in systems management. Elsevier New York, NY, USA: Distributors for the US and Canada, Elsevier Science Pub. Co., Amsterdam; Tokyo (1986)
312. Taillard, E.: Robust taboo search for the quadratic assignment problem. Parallel computing **17**(4–5), 443–455 (1991)
313. Taillard, É.: Parallel iterative search methods for vehicle routing problems. Networks **23**(8), 661–673 (1993)
314. Talbi, E., Hafidi, Z., Geib, J.: A parallel adaptive tabu search approach. Parallel Computing **24**(14), 2003–2019 (1998)

315. Tasgetiren, M.e.a.: A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem. European Journal of Operational Research **177**(3), 1930–1947 (2007)
316. Tavakkoli-Moghaddam, R., Rahimi-Vahed, A., Mirzaei, A.: A hybrid multi-objective immune algorithm for a flow shop scheduling problem with bi-objectives: weighted mean completion time and weighted mean tardiness. Information Sciences **177**(22), 5072–5090 (2007)
317. Teghem Jr, J.e.a.: STRANGE: an interactive method for multi-objective linear programming under uncertainty. European Journal of Operational Research **26**(1), 65–82 (1986)
318. Terán, P.: Probabilistic foundations for measurement modelling with fuzzy random variables. Fuzzy Sets and Systems **158**(9), 973–986 (2007)
319. Toroslu, I., Arslanoglu, Y.: Genetic algorithm for the personnel assignment problem with multiple objectives. Information Sciences **177**(3), 787–803 (2007)
320. Tragantalerngsak, S., Holt, J., Rönnqvist, M.: An exact method for the two-echelon, single-source, capacitated facility location problem. European Journal of Operational Research **123**(3), 473–489 (2000)
321. Triki, E., Collette, Y., Siarry, P.: A theoretical study on the behavior of simulated annealing leading to a new cooling schedule. European Journal of Operational Research **166**(1), 77–92 (2005)
322. Tsiakis, P., Shah, N., Pantelides, C.: Design of multi-echelon supply chain networks under demand uncertainty. Ind. Eng. Chem. Res **40**(16), 3585–3604 (2001)
323. Ulungu, E., Teghem, J., Fortemps, P., Tuyttens, D.: MOSA method: a tool for solving multi-objective combinatorial optimization problems. Journal of Multi-Criteria Decision Analysis **8**(4), 221–236 (1999)
324. Ulungu, L., Teghem, J., Ost, C.: Interactive simulated annealing in a multiobjective framework: application to an industrial problem. J. Oper. Res. Soc **49**, 1044–1050 (1998)
325. Van Hop, N.: Solving linear programming problems under fuzziness and randomness environment using attainment values. Information Sciences **177**(14), 2971–2984 (2007)
326. Vidal, C., Goetschalckx, M.: Strategic production-distribution models: A critical review with emphasis on global supply chain models. European Journal of Operational Research **98**(1), 1–18 (1997)
327. Von Mises, R.: Probability, statistics, and truth. Dover Pubns (1981)
328. Voss, S.: Tabu search: Applications and prospects. Inst. f. Betriebswirtschaftslehre (1992)
329. Wang Y., e.a.: Self-adaptive learning based particle swarm optimization. Information Sciences p. doi:10.1016/j.ins.2010.07.013 (2008)
330. Wang, J., Daniel Ng, C., Cheng, T., Liu, L.: Minimizing total completion time in a two-machine flow shop with deteriorating jobs. Applied Mathematics and Computation **180**(1), 185–193 (2006)
331. Wang, J., Shu, Y.: A possibilistic decision model for new product supply chain design. European Journal of Operational Research **177**(2), 1044–1061 (2007)
332. Wang, X., Tang, W., Zhao, R.: Random fuzzy EOQ model with imperfect quality items. Fuzzy Optimization and Decision Making **6**(2), 139–153 (2007)
333. Weise, T.: Global Optimization Algorithms–Theory and Application. URL: http://www. it-weise. de, Abrufdatum **1** (2008)
334. Weiss, E.: Algebraic number theory. Dover Pubns (1998)
335. Wetzel, R.: Freshwater ecosystems, modelling and simulation. Aquatic Botany **31**(3–4), 373–374 (1988)
336. Witte, E., Chamberlain, R., Franklin, M.: Parallel simulated annealing using speculative computation. Parallel and Distributed Systems, IEEE Transactions on **2**(4), 483–494 (2002)
337. Worrall, B., Hall, M.: The analysis of an inventory control model using posynomial geometric programming. International Journal of Production Research **20**(5), 657–667 (1982)
338. Xu, J., Ding, C.: A class of chance constrained multiobjective linear programming with birandom coefficients and its application to vendors selection. International Journal of Production Economics, Doi:10.1016/j.ijpe.2011.02.020
339. Xu, J., He, Y., Gen, M.: A class of random fuzzy programming and its application to supply chain design. Computers & Industrial Engineering **56**(3), 937–950 (2009)

340. Xu, J., Li, J.: Multiple Objective Decision Making Theory and Methods. Tsinghua University Press (in Chinese) (2005)
341. Xu, J., Liu, Q., Wang, R.: A class of multi-objective supply chain networks optimal model under random fuzzy environment and its application to the industry of Chinese liquor. Information Sciences **178**(8), 2022–2043 (2008)
342. Xu, J., Tao, Z.: A class of birandom multi-objective linear programming with equilibrium chances for constraints and objective functions. Sichuan University Research Report (2010)
343. Xu, J., Tao, Z.: A class of equilibrium chance constrained multiobjective linear programming with birandom coefficients. Sichuan University Research Report (2010)
344. Xu, J., Yao, L.: A class of expected value multi-objective programming problems with random rough coefficients. Mathematical and Computer Modelling **50**(1–2), 141–158 (2009)
345. Xu, J., Yao, L.: A class of multiobjective linear programming models with random rough coefficients. Mathematical and Computer Modelling **49**(1–2), 189–206 (2009)
346. Xu, J., Yao, L.: A class of bi-level expected value programming with random rough coefficients and its application to production-inventory problem. Sichuan University Research Report (2010)
347. Xu, J., Yao, L.: Random rough variable and random rough programming. Sichuan University Research Report (2010)
348. Xu, J., Zhou, X.: A class of multi-objective expected value decision-making model with birandom coefficients and its application to flow shop scheduling problem. Information Sciences **179**(17), 2997–3017 (2009)
349. Yager, R.: A procedure for ordering fuzzy subsets of the unit interval. Information Sciences **24**(2), 143–161 (1981)
350. Yager, R.: On the evaluation of uncertain courses of action. Fuzzy Optimization and Decision Making **1**(1), 13–41 (2002)
351. Yang, L., Ji, X., Gao, Z., Li, K.: Logistics distribution centers location problem and algorithm under fuzzy environment. Journal of Computational and Applied Mathematics **208**(2), 303–315 (2007)
352. Yeh, J., Fu, J.: Parallel adaptive simulated annealing for computer-aided measurement in functional MRI analysis. Expert Systems with Applications **33**(3), 706–715 (2007)
353. Yisu J., e.a.: The landscape adaptive particle swarm optimizer. Applied Soft Computing **8**(1), 295–304 (2008)
354. Youssef, K., Yousef, H., Sebakhy, O., Wahba, M.: Adaptive fuzzy APSO based inverse tracking-controller with an application to DC motors. Expert Systems with Applications **36**(2), 3454–3458 (2009)
355. Yu C.S., L.H.: A robust optimization model for stochastic logistic problems. International Journal of Production Economics **64**(1–3), 385–397 (2000)
356. Zadeh, L.: Fuzzy sets. Information and control **8**(3), 338–353 (1965)
357. Zadeh, L.: A fuzzy-set-theoretic interpretation of linguistic hedges. Cybernetics and Systems **2**(3), 4–34 (1972)
358. Zadeh, L.: Fuzzy sets as a basis for a theory of possibility. Fuzzy sets and systems **1**(1), 3–28 (1978)
359. Zadeh, L., et al.: Calculus of fuzzy restrictions. Fuzzy sets and their applications to cognitive and decision processes pp. 1–39 (1975)
360. Zeleny, M.: The attribute-dynamic attitude model (ADAM). Management Science **23**(1), 12–26 (1976)
361. Zhang, Z.: Constrained Multiobjective Optimization Immune Algorithm: Convergence and Application. Computers and Mathematics with Applications **52**, 791–808 (2006)
362. Zhou, G., Min, H., Gen, M.: The balanced allocation of customers to multiple distribution centers in the supply chain network: a genetic algorithm approach. Computers & Industrial Engineering **43**(1–2), 251–261 (2002)
363. Zhou, X., Xu, J.: A class of integrated logistics network model under random fuzzy environment and its application to Chinese beer company. Int. J. Unc. Fuzz. Knowl. Based Syst. **17**(06), 807–831 (2009)

364. Zimmermann, H.: Fuzzy programming and linear programming with several objective functions. Fuzzy sets and systems **1**(1), 45–55 (1978)
365. Zimmermann, H.: Fuzzy sets, decision making, and expert systems. Springer (1987)
366. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary computation **8**(2), 173–195 (2000)

Index