Springer Optimization and Its Applications 123

Panos M. Pardalos
Antanas Žilinskas
Julius Žilinskas

# Non-Convex Multi-Objective Optimization

Springer

# Springer Optimization and Its Applications

Volume 123



## Aims and Scope

Optimization has been expanding in all directions at an astonishing rate during the last few decades. New algorithmic and theoretical techniques have been developed, the diffusion into other disciplines has proceeded at a rapid pace, and our knowledge of all aspects of the field has grown even more profound. At the same time, one of the most striking trends in optimization is the constantly increasing emphasis on the interdisciplinary nature of the field. Optimization has been a basic tool in all areas of applied mathematics, engineering, medicine, economics, and other sciences.

The series *Springer Optimization and Its Applications* publishes undergraduate and graduate textbooks, monographs and state-of-the-art expository work that focus on algorithms for solving optimization problems and also study applications involving such problems. Some of the topics covered include nonlinear optimization (convex and nonconvex), network flow problems, stochastic optimization, optimal control, discrete optimization, multi-objective programming, description of software packages, approximation techniques and heuristic approaches.

More information about this series at http://www.springer.com/series/7393

Panos M. Pardalos • Antanas Žilinskas
Julius Žilinskas

# Non-Convex Multi-Objective Optimization

Springer

Panos M. Pardalos
Department of Industrial
and Systems Engineering
University of Florida
Gainesville, FL, USA

Research University Higher School
of Economics, Russia

Julius Žilinskas
Institute of Mathematics & Informatics
Vilnius University
Vilnius, Lithuania

Antanas Žilinskas
Institute of Mathematics & Informatics
Vilnius University
Vilnius, Lithuania

ISSN 1931-6828 ISSN 1931-6836 (electronic)
Springer Optimization and Its Applications
ISBN 978-3-319-61005-4 ISBN 978-3-319-61007-8 (eBook)
DOI 10.1007/978-3-319-61007-8

Library of Congress Control Number: 2017946557



Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

# Preface

Optimization is a very broad field of research with a wide spectrum of important applications. Until the 1950s, optimization was understood as a single-objective optimization, i.e., as the specification and computation of minimum/maximum of a function of interest taking into account some constraints for the solution. Such optimization problems were the focus of mathematicians from ancient times. The earliest methods of calculus were applied for the analysis and solution of optimization problems immediately following their development. Moreover, these applications gave way to important results in the basics of natural sciences. The importance of optimization for the understanding of nature is well formulated by Leonard Euler:

> Since the fabric of the universe is most perfect and the work of a most wise Creator, nothing at all takes place in the universe in which some rule of maximum or minimum does not appear.[1]

Further developments of optimization theory and computational methods were successfully applied not only in natural sciences but also in planning and design. Let us mention that several Nobel Memorial Prizes in Economics were awarded for the application of optimization methods to the problems of economics. Nevertheless, in the middle of the last century, it was understood that the model of single-objective optimization is not universal. In many problems of planning and design, a decisionmaker aims to minimize/maximize not a single but several objective functions. As an example, in industry, when producing metal sheets, the objectives are to minimize energy consumption, maximize process speed, and maximize the strength of the product at the same time. The purpose of multi-objective

[1] Cum enim Mundi universi fabrica sit perfectissima atque a Creatore sapientissimo absoluta, nihil omnino in mundo contingit, in quo non maximi minimive ratio quaepiam eluceat; quamobrem dubium prorsus est nullum, quin omnes Mundi effectus ex causis finalibus ope Methodi maximorum et minimorum aeque feliciter determinari queant, atque ex ipsis causis efficientibus (L. Euler, Methodus inveniendi lineas curvas maximi minimive proprietate gaudentes, sive solutio problematis isoperimetrici lattissimo sensu accepti, Lausanne and Geneva, 1744).

optimization is to give an understanding of how these objectives are conflicting and to provide the user the possibility to choose an appropriate trade-off between the objectives.

As another example, consider multi-objective path finding problems which have received a lot of attention in recent years. Routing problems are part of everyday activity. We move material through transportation networks, and we move huge amounts of data through telecommunication networks or the Internet. Many times, we are looking for the shortest path, but in real life, other objectives can be considered. Suppose we consider the problem to route hazardous materials in a transportation network. In addition to the minimum distance, we need to have objectives on minimizing environmental risks and risks to human populations.

When multiple objectives are present, the concept of an optimal solution as in the single-objective problems does not apply. Naturally, first of all, the well-known classical single-objective optimization methods were generalized for the multi-objective case, e.g., methods of multi-objective linear programming and of multi-objective convex optimization were developed. The single-objective non-convex optimization problems are known as the most difficult. The difficulties certainly increase in case of several objectives. The generalization of mathematical methods of single-objective global optimization to multi-objective case and the development of new methods present a real challenge for researchers. Heuristic methods are more prone to various modifications, and the generalization of heuristic methods of global optimization for the multi-objective case has been booming. Meanwhile, many multi-objective optimization problems of engineering can be solved by software implementing the heuristic methods. Nevertheless, the mathematical analysis of non-convex multi-objective optimization problems is urgent from the point of applications as well as of general global optimization theory. A subclass of those problems waiting for a more active attention of researchers is multi-objective optimization of non-convex expensive black box problems; this book is focused on the theoretically substantiated methods for problems of such a type.

**Acknowledgements** We would like to thank Springer for their help and the opportunity to publish this book. Work of the authors was supported by RSF grant 14-41-00039 (National Research University Higher School of Economics) and a grant (No. MIP-063/2012) from the Research Council of Lithuania.

Gainesville, FL, USA — Panos M. Pardalos
Vilnius, Lithuania — Antanas Žilinskas
2017 — Julius Žilinskas

# Contents

## Part II Theory and Algorithms

# Notation and Symbols

| | |
|---|---|
| $\mathbb{R}^d$ | $d$-Dimensional Euclidean space |
| $\mathbf{A} \in \mathbb{R}^d$ | Feasible region |
| $\mathbf{x} \in \mathbb{R}^d$ | A vector in $d$-dimensional space |
| $f_i(\cdot)$ | Objective function |
| $m$ | Number of objective functions |
| $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))^T$ | Vector-valued objective function |
| $\mathbf{P}(\mathbf{f})_O$ | The set of Pareto optimal solutions |
| $\mathbf{P}(\mathbf{f})_D$ | The set of Pareto optimal decisions |
| $L_i$ | Lipschitz constant of $f_i(\cdot)$ |
| $\xi_{\mathbf{x}}$ | Random variable used to model an unknown value of an objective function at point $\mathbf{x}$ |
| $\xi(\mathbf{x})$ | Random field |
| $\Xi(\mathbf{x})$ | Vector random field |

# Part I
# Basic Concepts

# Chapter 1
# Definitions and Examples

## 1.1 Definitions

The problem of multi-objective optimization is considered:

$$\min_{\mathbf{x}\in\mathbf{A}} \mathbf{f}(\mathbf{x}),\ \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}))^T. \tag{1.1}$$

Here $\mathbf{f}(\mathbf{x}) : \mathbf{A} \to \mathbb{R}^m$ is vector-valued objective function and $\mathbb{R}^m$ is the space of objectives.

The feasible region $\mathbf{A} \subset \mathbb{R}^d$ is usually expressed by a number of inequality constraints: $\mathbf{A} = \{\mathbf{x} \in \mathbb{R}^d | \, g_j(\mathbf{x}) \leq 0,\ j = 1, 2, \ldots, l\}$. If all the objective functions $f_i(\cdot)$ and the constraint functions $g_j(\cdot)$ are linear, then (1.1) is called a multi-objective linear programming problem. If at least one of the objective or constraint functions is nonlinear, then the problem is called a nonlinear multi-objective optimization problem. If the feasible region and all the objective functions are convex, then (1.1) becomes a convex multi-objective optimization problem. When at least one of the objective functions or the feasible region is non-convex, then we have a non-convex multi-objective optimization problem. This book is focused on such problems.

Theoretically the solution to problem (1.1) consists of two sets [135]:

- $\mathbf{P}(\mathbf{f})_O$—Pareto optimal solutions in the space of objectives (Pareto front)
- $\mathbf{P}(\mathbf{f})_D$—Pareto optimal decisions in $\mathbf{A}$ (Pareto set).

Analytically these sets can be found only in very specific cases. We are interested in the approximation of $\mathbf{P}(\mathbf{f})_O$, although most frequently it suffices to find several desirable members of the mentioned sets.

We aim at minimizing all the objective functions at the same time, however, objectives are usually conflicting to each other: if one is decreased, some others may be increased. Thus, we need to use an ordering concept in $\mathbb{R}^m$ in order to solve the problem comparing the solutions (objective function values).



P.M. Pardalos et al., *Non-Convex Multi-Objective Optimization*, Springer
Optimization and Its Applications 123, DOI 10.1007/978-3-319-61007-8_1

**Definition 1.1 (Dominance of Objective Vectors)** An objective vector $\mathbf{z} \in \mathbb{R}^m$ dominates an objective vector $\mathbf{u} \in \mathbb{R}^m$ if $z_i \leq u_i$ for all $i = 1, 2, \ldots, m$ and $z_j < u_j$ for at least one $j \in \{1, 2, \ldots, m\}$.

**Definition 1.2 (Dominance of Decisions)** A decision $\mathbf{x} \in \mathbf{A}$ with $\mathbf{f}(\mathbf{x})$ dominates a decision $\mathbf{y} \in \mathbf{A}$ with $\mathbf{f}(\mathbf{y})$ if $f_i(\mathbf{x}) \leq f_i(\mathbf{y})$ for all $i = 1, 2, \ldots, m$ and $f_j(\mathbf{x}) < f_j(\mathbf{y})$ for at least one $j \in \{1, 2, \ldots, m\}$.

Pareto optimality [155], named after the Italian economist Vilfredo Pareto, is a measure of efficiency in multi-objective optimization. The concept has a wide range of applications in economics, game theory, engineering, multi-criteria analysis, and social sciences. According to this concept, we look for objective vectors such that none of the components of each of those vectors can be improved without deterioration of at least one of the other components of the vector. Therefore, the Pareto optimality can be defined mathematically as follows:

**Definition 1.3 (Pareto Optimality)** A decision $\mathbf{x}^* \in \mathbf{A}$ with $\mathbf{f}(\mathbf{x}^*)$ is called (globally) Pareto optimal (efficient, non-dominated, non-inferior), if and only if there exists no decision $\mathbf{x} \in \mathbf{A}$ such that $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*)$ for all $i = 1, 2, \ldots, m$ and $f_j(\mathbf{x}) < f_j(\mathbf{x}^*)$ for at least one $j \in \{1, 2, \ldots, m\}$.

Local Pareto optimality can also be defined similarly to the local optimality in single-objective optimization:

**Definition 1.4 (Local Pareto Optimality)** A decision $\mathbf{x}^* \in \mathbf{A}$ with $\mathbf{f}(\mathbf{x}^*)$ is called locally Pareto optimal, if and only if there exists $\delta > 0$ such that $\mathbf{x}^*$ is Pareto optimal in $\mathbf{A} \cap \mathbf{B}(\mathbf{x}^*, \delta)$; where, $\mathbf{B}(\mathbf{x}^*, \delta)$ is an open ball of radius $\delta$ centered at $\mathbf{x}^* \in \mathbf{A}$, that is, $\mathbf{B}(\mathbf{x}^*, \delta) = \{\mathbf{x} \in \mathbb{R}^d | \, ||\mathbf{x} - \mathbf{x}^*|| < \delta\}$.

Note that every globally Pareto optimal decision is a locally Pareto optimal decision. However, the converse is not always true unless there are some assumptions in the problem.

Pareto optimal decisions compose the Pareto set $\mathbf{P}(\mathbf{f})_D$ and corresponding Pareto optimal solutions compose the Pareto front $\mathbf{P}(\mathbf{f})_O$. The Pareto front is bounded by the nadir and ideal objective vectors defined below.

**Definition 1.5 (Nadir Objective Vector)** The nadir objective vector is defined as $\mathbf{r} = (r_1, r_2, \ldots, r_m)^T$ with elements

$$r_i = \sup_{\mathbf{x} \in \mathbf{P}(\mathbf{f})_D} f_i(\mathbf{x}), \; i = 1, 2, \ldots, m.$$

**Definition 1.6 (Ideal Objective Vector)** The ideal objective vector is defined as $\mathbf{l} = (l_1, l_2, \ldots, l_m)^T$ with elements

$$l_i = \inf_{\mathbf{x} \in \mathbf{P}(\mathbf{f})_D} f_i(\mathbf{x}), \; i = 1, 2, \ldots, m.$$

**Definition 1.7 (Utopian Objective Vector)** The utopian objective vector is defined as $\mathbf{u} = (u_1, u_2, \ldots, u_m)^T$ with elements

$$u_i = l_i - \epsilon,\ \epsilon > 0,\ i = 1, 2, \ldots, m.$$

**Theorem 1.1 ([33])** *If the multi-objective optimization problem is convex, then every locally Pareto optimal decision is also globally Pareto optimal.*

*Proof* Suppose that $\mathbf{x}^*$ is a locally Pareto optimal decision which is not a globally Pareto optimal. Then there exists $\mathbf{y} \in \mathbf{A}$ such that $\mathbf{f}(\mathbf{y}) < \mathbf{f}(\mathbf{x}^*)$ and $\mathbf{f}(\mathbf{y}) \neq \mathbf{f}(\mathbf{x}^*)$. Consider a point $\mathbf{z} = \beta\mathbf{x}^* + (1-\beta)\mathbf{y}$ such that $\mathbf{z} \in \mathbf{A} \cap \mathbf{B}(\mathbf{x}^*, \delta)$ and $0 < \beta < 1$. By the convexity of the objective functions, we have

$$\mathbf{f}(\mathbf{z}) \leq \beta\mathbf{f}(\mathbf{x}^*) + (1-\beta)\mathbf{f}(\mathbf{y}) \leq \mathbf{f}(\mathbf{x}^*)$$

and

$$\mathbf{f}(\mathbf{x}^*) + (1-\beta)\mathbf{f}(\mathbf{y}) \neq \mathbf{f}(\mathbf{x}^*),$$

which shows that $\mathbf{x}^*$ would not be a locally Pareto optimal decision, a contradiction.

The following theorem extends the above result. For the proof, see [179].

**Theorem 1.2** *Consider the multi-objective optimization problem with a convex feasible region and quasiconvex objective functions. If at least one of the objective functions is strictly quasiconvex, then every locally Pareto optimal decision is also globally Pareto optimal.*

Let us introduce the weak Pareto optimality.

**Definition 1.8 (Weak Pareto Optimality)** A decision $\mathbf{x}^* \in \mathbf{A}$ with $\mathbf{f}(\mathbf{x}^*)$ is called weakly Pareto optimal, if and only if there exists no decision $\mathbf{x} \in \mathbf{A}$ such that $f_i(\mathbf{x}) < f_i(\mathbf{x}^*)$ for all $i = 1, 2, \ldots, m$.

Every Pareto optimal decision is weakly Pareto optimal.

For the theoretical details and methodological discussion we refer to [23, 135, 153, 269].

## 1.2 Optimality Conditions

Next we discuss optimality conditions for the multi-objective optimization problem

$$\begin{aligned} \min\ & \mathbf{f}(\mathbf{x}) \\ \text{s.t.}\ & \mathbf{g}(\mathbf{x}) \leq \mathbf{0}, \\ & \mathbf{x} \in \mathbf{B}, \end{aligned} \tag{1.2}$$

where $\mathbf{B}$ is an open set in $\mathbb{R}^d$. First we present necessary optimality conditions which can be derived without any convexity assumptions. Then we introduce sufficient optimality conditions under suitable convexity assumptions.

In order to define the optimality conditions we need the following notation. Let

$$\mathbf{I}(\mathbf{x}) = \{j \in 1, 2, \ldots, l |\ g_j(\mathbf{x}) = 0\}$$

be the index set of the active inequality constraints at $\mathbf{x}$, and let

$$\mathbf{A} = \{\mathbf{x} \in \mathbb{R}^d | \mathbf{g}(\mathbf{x}) \leq \mathbf{0}, \mathbf{x} \in \mathbf{B}\}.$$

**Theorem 1.3 (FrizJohn Necessary Condition for Pareto Optimality [135])** *Let $\mathbf{f}(\cdot), g_j(\cdot), j = 1, 2, \ldots, l$, be continuously differentiable in an open set containing the feasible set of (1.2), and let $\mathbf{x}^*$ be a locally Pareto optimal decision. Then the following optimality condition holds:*

*There exist vectors $\boldsymbol{\alpha} \in \mathbb{R}^m$ and $\boldsymbol{\lambda} \in \mathbb{R}^l$ such that*

$$\sum_{i=1}^{m} \alpha_i \nabla f_i(\mathbf{x}^*) + \sum_{j=1}^{l} \lambda_j \nabla g_j(\mathbf{x}^*) = 0,$$

$$(\boldsymbol{\alpha}, \boldsymbol{\lambda}) \neq (\mathbf{0}, \mathbf{0}),$$

$$\lambda_j g_j(\mathbf{x}^*) = 0,\ \lambda_j \geq 0,\ j = 1, 2, \ldots, l,$$

$$\alpha_i \geq 0,\ i = 1, 2, \ldots, m.$$

**Theorem 1.4 (Karush–Kuhn–Tucker [33])** *Let $\mathbf{f}(\cdot), g_j(\cdot), j = 1, 2, \ldots, l$, be continuously differentiable in an open set containing the feasible set of (1.2), and let $\mathbf{x}^*$ be a locally Pareto optimal decision. Further, assume that the vectors $\nabla g_j(\mathbf{x}^*),\ j \in \mathbf{I}(\mathbf{x}^*)$ are linearly independent. Then the following optimality condition holds:*

*There exist vectors $\boldsymbol{\alpha} \in \mathbb{R}^m$ and $\boldsymbol{\lambda} \in \mathbb{R}^l$ such that*

$$\sum_{i=1}^{m} \alpha_i \nabla f_i(\mathbf{x}^*) + \sum_{j=1}^{l} \lambda_j \nabla g_j(\mathbf{x}^*) = 0,$$

$$\lambda_j g_j(\mathbf{x}^*) = 0,\ \lambda_j \geq 0,\ j = 1, 2, \ldots, l,$$

$$\sum_{i=1}^{m} \alpha_i = 1,\ \alpha_i \geq 0,\ i = 1, 2, \ldots, m.$$

Let us introduce the scalar-valued function

$$F(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i f_i(\mathbf{x}).$$

The above conditions are equivalent to the claim that $\mathbf{x}^*$ is a Karush–Kuhn–Tucker point of the corresponding optimization problem with scalar-valued function $F(\mathbf{x})$ and the same constraints. Furthermore, if Problem (1.2) is convex, then $\mathbf{x}^*$ is a

Pareto optimal decision in Problem (1.2) if and only if $\mathbf{x}^*$ is the global minimizer of the corresponding scalar-valued function over the same constraint set as those in (1.2) [68]. This means that the above optimality conditions are sufficient for $\mathbf{x}^*$ to be a (globally) Pareto optimal decision for convex problems. In general, the optimality conditions do not provide the complete Pareto optimal set. We also consider second-order optimality conditions which are necessary for a decision $\mathbf{x}^*$ to be locally Pareto optimal.

**Theorem 1.5 (Second-Order Necessary Conditions [33])** *Let the objective and the constraint functions of Problem (1.2) be twice continuously differentiable at a feasible decision* $\mathbf{x}^*$. *Assume that the vectors* $\nabla g_j(\mathbf{x}^*),\ j \in \mathbf{I}(\mathbf{x}^*)$ *are linearly independent. The necessary conditions for* $\mathbf{x}^*$ *to be Pareto optimal are the following:*

*(i) There exist vectors* $\boldsymbol{\alpha} \in \mathbb{R}^m$ *and* $\boldsymbol{\lambda} \in \mathbb{R}^l$ *such that*

$$\sum_{i=1}^{m} \alpha_i \nabla f_i(\mathbf{x}^*) + \sum_{j=1}^{l} \lambda_j \nabla g_j(\mathbf{x}^*) = 0,$$

$$\lambda_j g_j(\mathbf{x}^*) = 0,\ \lambda_j \geq 0,\ j = 1, 2, \ldots, l,$$

$$\sum_{i=1}^{m} \alpha_i = 1,\ \alpha_i \geq 0,\ i = 1, 2, \ldots, m.$$

*(ii)* $\mathbf{d}^T \left( \sum_{i=1}^{m} \alpha_i \nabla^2 f_i(\mathbf{x}^*) + \sum_{j=1}^{l} \lambda_j \nabla^2 g_j(\mathbf{x}^*) \right) \mathbf{d} \geq 0$ *for all* $\mathbf{d} \in \{\mathbf{d} \in \mathbb{R}^d \backslash \{0\} |\ \nabla f_i(\mathbf{x}^*)^T \mathbf{d} \leq 0,\ i = 1, 2, \ldots, m,\ \nabla g_j(\mathbf{x}^*)^T \mathbf{d} = 0,\ j \in \mathbf{I}(\mathbf{x}^*)\}$.

For the proof, see [225]. Further optimality conditions for multi-objective programming can be found in [34, 120, 135, 225]. In the case the objective and constraint functions are convex, the conditions of Theorem 1.5 are sufficient.

## 1.3 Illustrative Examples

Let us consider some examples.

*Example 1.1* The first example problem has two quadratic objective functions:

$$\begin{aligned} f_1(\mathbf{x}) &= 5(x_1 - 0.1)^2 + (x_2 - 0.1)^2, \\ f_2(\mathbf{x}) &= (x_1 - 0.9)^2 + 5(x_2 - 0.9)^2. \end{aligned} \tag{1.3}$$

The feasible region is $\mathbf{A} : 0 \leq x_1, x_2 \leq 1$.

The objective functions are convex as well as the feasible region.

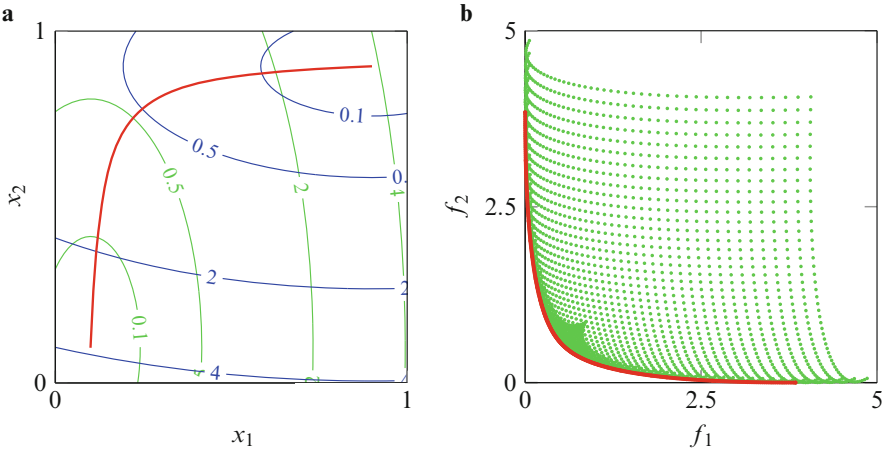


**Fig. 1.1** (**a**) Contour lines of the functions $f_1$ (*green*) and $f_2$ (*blue*) along with the set of Pareto optimal decisions (*red*). (**b**) Objective feasible region (represented by *green dots*) and the Pareto front (*red*) for Example 1.1

The set of Pareto optimal decisions is easily computable to be

$$x_2 = \frac{0.1(-224x_1 + 21.6)}{-24x_1 + 1.6}, \ 0.1 \le x_1 \le 0.9. \tag{1.4}$$

This set is illustrated in Figure 1.1a on the background of contour lines of functions $f_1(x)$ and $f_2(x)$. The Pareto front is depicted in Figure 1.1b. In this figure, we also show the objective feasible region; that is, the set of values of $\{(f_1(\mathbf{x}), f_2(\mathbf{x})), \mathbf{x} \in \mathbf{A}\}$.

*Example 1.2* The second example, proposed in [60], is referred to as the Fonseca problem; see also [42, pp. 339–340]. The objective functions are

$$\begin{aligned} f_1(\mathbf{x}) &= 1 - e^{-\sum_{i=1}^{d}(x_i - 1/\sqrt{d})^2}, \\ f_2(\mathbf{x}) &= 1 - e^{-\sum_{i=1}^{d}(x_i + 1/\sqrt{d})^2}, \ d = 2. \end{aligned} \tag{1.5}$$

The feasible region is $\mathbf{A} : -4 \le x_1, x_2 \le 4$.

The Pareto front is depicted in Figure 1.2. We see that in this example the Pareto front is not convex.

*Example 1.3* The third example is composed of two Shekel functions which are frequently used for testing of single-objective global optimization algorithms, see, e.g., [239]:

$$f_1(\mathbf{x}) = -\frac{0.1}{0.1 + (x_1 - 0.1)^2 + 2(x_2 - 0.1)^2} -$$

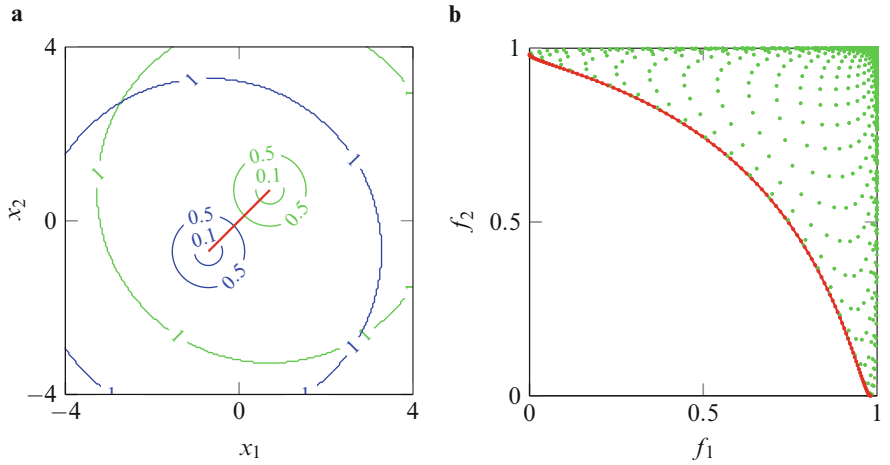


**Fig. 1.2** (**a**) Contour lines of the functions $f_1$ (*green*) and $f_2$ (*blue*) along with the set of Pareto optimal decisions (*red*). (**b**) Objective feasible region (represented by *green dots*) and the Pareto front (*red*) for Example 1.2

$$f_2(\mathbf{x}) = -\frac{0.1}{0.14 + 20((x_1 - 0.45)^2 + (x_2 - 0.55)^2)}, \\ -\frac{0.1}{0.15 + 40((x_1 - 0.55)^2 + (x_2 - 0.45)^2)} - \\ -\frac{0.1}{0.1 + (x_1 - 0.3)^2 + (x_2 - 0.95)^2}. \tag{1.6}$$

The feasible region is $\mathbf{A} : 0 \leq x_1, x_2 \leq 1$.

The set of Pareto optimal decisions is illustrated in Figure 1.3a on the background of contour lines of functions $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$. The Pareto front is depicted in Figure 1.3b. Here we see that the Pareto front consists of three non-contiguous parts.

*Example 1.4* As an example of combinatorial multi-objective optimization [220] problem we consider a bi-criterion cell formation [71]. The goal of the problem is to find groupings of machines simultaneously optimizing two conflicting objectives. Let us consider the problem described in [223] and also presented in [46], where two objectives are:

- Minimization of the total intercell moves: The number of cells processing each part is minimized.
- Minimization of within-cell load variation: The differences between workload induced by a part on a specific machine and the average workload induced by this part on the cell are minimized.

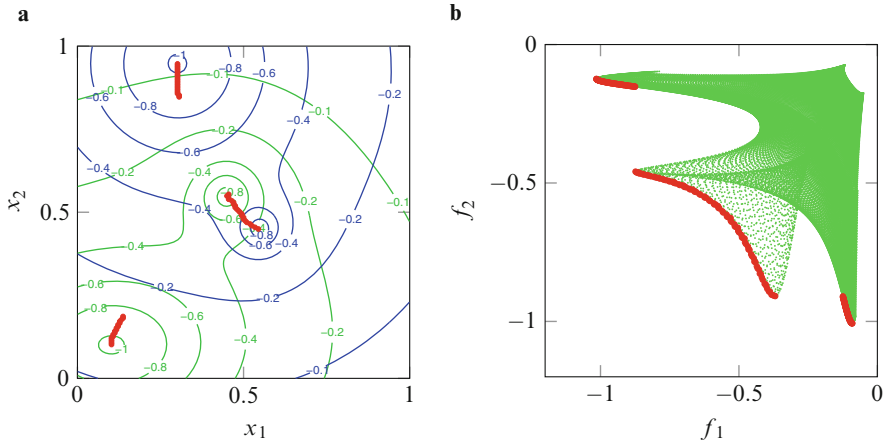


**Fig. 1.3** (**a**) Contour lines of the functions $f_1$ (*green*) and $f_2$ (*blue*) along with the set of Pareto optimal decisions (*red*). (**b**) Objective feasible region (represented by *green dots*) and the Pareto front (*red*) for Example 1.3

Let us denote the number of machines by $m$, the number of parts by $n$, and the number of cells by $k$. $\mathbf{X}$ is a Boolean $m \times k$ cell membership matrix where 1 in the $i$-th row and the $j$-th column means that the $i$-th machine is assigned to the $j$-th cell. Each machine can only be assigned to one cell: $\mathbf{X}\mathbf{e}_k = \mathbf{e}_m$, where $\mathbf{e}_t$ is a $t$-element column vector of ones. The input data are an $m \times n$ machine-part incidence matrix $\mathbf{W}$ specifying workload on each machine induced by each part and an $n$-element vector $\mathbf{p}$ of production requirements of parts.

We compute the number of intercell moves as follows:

$$f_1(\mathbf{X}) = \mathbf{p}^T(\Phi(\mathbf{W}^T\mathbf{X})\mathbf{e}_k - \mathbf{e}_n), \tag{1.7}$$

where the function $\Phi$ changes the nonzero elements of matrix to ones. The within-cell load variation is computed as

$$f_2(\mathbf{X}) = \langle \mathbf{W} - \mathbf{M}(\mathbf{X}), \mathbf{W} - \mathbf{M}(\mathbf{X}) \rangle, \tag{1.8}$$

where the matrix $\mathbf{M}(\mathbf{X})$ is an $m \times n$ matrix with the average cell loads: the element in the $i$-th row and the $j$-th column specifies the average load of the $j$-th part in the cell where the $i$-th machine is assigned.

The data of a problem instance presented in [223] is shown in Table 1.1, where zero elements are shown as spaces. There are $m = 15$ machines and $n = 30$ parts. This problem instance is evaluated as a large size in [46].

The Pareto optimal decision with objective vector (3974, 5.576) is illustrated in Figure 1.4. For this decision five cells are needed. The problem provides grouping of the machines to cells, but not the assignment of parts to cells. However, one

**Table 1.1** Data of a cell formation problem from [223]

| **W** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | 0.3 | | | 0.6 | 0.6 | 0.2 | 0.2 | 0.5 | 0.7 | | | | | | 0.4 | | 0.6 | | | | | | | | | | | |
| 2 | 0.4 | | 0.5 | | | 0.7 | 0.3 | 0.4 | 0.3 | 0.6 | 0.8 | | | | | | 0.9 | | 0.2 | | | | | | | | | | | |
| 3 | 0.6 | | 0.7 | 0.3 | | 0.2 | 0.4 | 0.9 | 0.6 | 0.2 | 0.2 | | | | | 0.4 | 0.3 | | 0.5 | | | | | | | | | | | |
| 4 | | 0.2 | | | 0.3 | | | | | | | | 0.4 | 0.7 | 0.5 | 0.6 | | 0.2 | | 0.4 | | 0.4 | | | | 0.5 | 0.6 | | | |
| 5 | | 0.2 | | | 0.3 | | | | | | | 0.4 | | 0.5 | | 0.7 | | 0.8 | | 0.9 | | 0.6 | | | | 0.8 | 0.2 | | | |
| 6 | | 0.8 | | | 0.9 | | | | | | | 1.0 | | 0.7 | | 0.2 | | 0.3 | | 0.4 | | 0.5 | | | | 0.6 | 0.8 | | | |
| 7 | 0.8 | | 0.9 | | | 0.3 | 0.5 | 0.5 | 0.7 | 0.3 | 0.5 | | | | | | 0.6 | | 0.9 | | | | | | | | | | | |
| 8 | | 1.1 | | | 1.2 | | | | | | | 0.3 | | 0.8 | | 0.3 | | 0.9 | | 0.2 | | 0.3 | | | | 0.4 | 0.5 | | | |
| 9 | | 0.4 | | | 0.5 | | | | | | | 0.6 | | 0.9 | | 0.5 | | 0.6 | | 0.7 | | 0.8 | | | | 0.9 | 1.0 | | | |
| 10 | 0.6 | | 0.2 | | | 0.3 | 0.9 | 0.2 | 0.3 | 0.4 | 0.5 | | | | | | 0.6 | | 0.8 | | | | | | | | | | | |
| 11 | 0.3 | | 0.3 | 0.2 | | | | | | | | | 0.3 | | 0.4 | | | | | | 0.5 | | 0.9 | 0.2 | 0.5 | | | 0.6 | 0.7 | 0.8 |
| 12 | | | | 0.6 | | | | | | | | | 0.7 | | 0.8 | | | | | | 0.9 | | 0.9 | 0.3 | 0.5 | | | 0.5 | 0.6 | 0.7 |
| 13 | | | | 0.7 | | | | | | | | | 0.5 | | 0.6 | | | | | | 0.8 | | 0.5 | 0.3 | 0.4 | | | 0.5 | 0.7 | 0.8 |
| 14 | | | | 0.2 | | | | | | | | | 0.6 | | 0.8 | | | | | | 1.0 | | 0.5 | 0.4 | 0.6 | | | 0.8 | 0.2 | 0.8 |
| 15 | | | | 0.5 | | | | | | | | | 0.7 | | 0.9 | | | | | | | | 0.3 | | 0.7 | | | 0.9 | 0.3 | 0.4 |
| **p** | 155 | 150 | 148 | 160 | 144 | 158 | 152 | 155 | 164 | 148 | 140 | 144 | 145 | 162 | 170 | 140 | 156 | 132 | 172 | 164 | 144 | 158 | 155 | 152 | 140 | 166 | 148 | 145 | 144 | 170 |

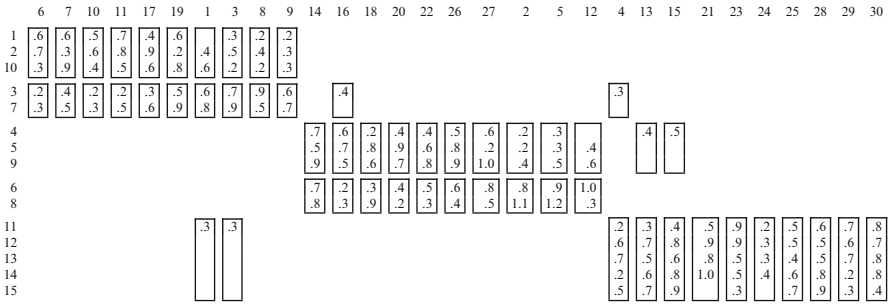


**Fig. 1.4** Illustration of non-dominated decision of example cell formation problem with objective vector (3974, 5.576)

can interpret that the part is supposed to be assigned to one of the cells where it needs processing if needed. We box each part in the cells where it needs processing. Intercell moves are required if the part needs processing in several cells. The numbers outside boxes are zeros and do not vary, therefore, only numbers in the boxes must be taken into account.

# Chapter 2
# Scalarization

## 2.1 General Idea

There were attempts to reduce multi-objective optimization problems to single-objective ones from the very beginning of their investigation [65, 99]. The reduction of a problem of multi-objective optimization to a single-objective optimization one normally is called scalarization. To find a discrete representation of the set of Pareto optimal solutions, a sequence of single-objective optimization problems should be solved, and they should hold the following theoretical properties:

- a solution of the single-objective optimization problem should provide an element of the set of Pareto optimal solutions,
- every element of the set of Pareto optimal solutions should be obtainable as a solution of a single-objective optimization problem constructed by the considered scalarization method.

In the case of nonlinear non-convex objective functions the single-objective problem, constructed by means of the scalarization, is non-convex as well, and therefore it can be solved exactly only in exceptional cases. Consequently, a found solution normally is not a true Pareto solution but a vector in some neighborhood of the Pareto set. To get a discrete representation (approximation) of the whole Pareto set, a sequence of auxiliary single-objective optimization problems should be solved. The quality of the approximation depends not only on the precision of a particular result of the single-objective optimization but also, and even stronger, on the sequence of parameters used in the construction of the auxiliary single-objective optimization problems. The methods of approximation of the whole Pareto set are discussed in the next chapter, while in this chapter we focus on the single-objective optimization problems which satisfy both theoretical requirements to the scalarization formulated above.



P.M. Pardalos et al., *Non-Convex Multi-Objective Optimization*, Springer
Optimization and Its Applications 123, DOI 10.1007/978-3-319-61007-8_2

A classical scalarization method is the weighted sum method where the auxiliary single-objective function is defined by the formula

$$F(\mathbf{x}) = \sum_{i=1}^{m} w_i f_i(\mathbf{x}),\ w_i \geq 0,\ \sum_{i=1}^{m} w_i = 1, \tag{2.1}$$

where the weights $w_i$ are parameters, and the selection of a vector $\mathbf{w} = (w_1, \ldots, w_m)^T$ gives the element of the Pareto front $\mathbf{f}(\mathbf{x}(\mathbf{w}))$

$$\mathbf{x}(\mathbf{w}) = \arg\min_{\mathbf{x}\in\mathbf{A}} F(\mathbf{x}). \tag{2.2}$$

The simplicity of the weighted sum scalarization is indeed an advantage of this method. However, its crucial disadvantage, in the case of non-convex objectives, is not guaranteed existence of $\mathbf{w}$ yielding an arbitrary element of the Pareto set. The latter fact is explained and illustrated in many textbooks, e.g., [135]. In the subsequent brief review of the scalarization methods, only the methods suitable for the computation of every Pareto optimal solution of non-convex problems are discussed.

An equivalent definition of proper efficiency was proposed in [234]. With the new definition of properness, a multi-objective problem is transformed to a more convenient one. Under determined conditions the original and the transformed problems have the same Pareto and properly efficient solutions. Therefore, this transformation may be employed for convexification and simplification in order to improve the computational efficiency of the solution of a given problem. Some existing results about the weighted sum method in the multi-objective optimization literature are generalized using the special case of the transformation scheme [234].

## 2.2 Tchebycheff Method

Different version of parametric scalar problems were proposed involving a reference point besides of weights of objectives. The weighted Tchebycheff method is one of the well-known methods of such a type, see, e.g., [42, 135]. Let $\mathbf{u} \in \mathbb{R}^m$ be a utopian point, i.e., $u_i < \min_{\mathbf{x}\in\mathbf{A}} f_i(\mathbf{x})$, and $\mathbf{w}$ be a vector of weights. The weighted Tchebycheff problem is of the form

$$\min_{x\in\mathbf{A}} \max_{i=1,\ldots,m} w_i\,(f_i(\mathbf{x}) - u_i). \tag{2.3}$$

Any Pareto optimal solution can be found by solving the weighted Tchebycheff problem with appropriate parameters, and with any parameters its solution corresponds to a weakly Pareto optimal solution of the original problem. Let us emphasize that the stated property of the weighted Tchebycheff problem holds for the non-convex problems.

The objective function of minimization problem (2.3) is nondifferentiable even in the case of smooth objectives of the original multi-objective problem. It is well known that the numerical solution of nondifferentiable problems is more complicated than the solution of smooth ones. However, the problem (2.3) with smooth $f_i(\cdot)$ can be reduced to the following equivalent differentiable form:

$$\begin{aligned} &\min_{t\geq 0} t, \\ &w_i\left(f_i(\mathbf{x}) - u_i\right) \leq t,\ i = 1, \ldots, m \\ &\mathbf{x} \in \mathbf{A}. \end{aligned} \tag{2.4}$$

The disadvantage of the Tchebycheff method is the presence of weakly Pareto optimal solutions among the found ones. The other disadvantage is the difficulty to solve the single-objective optimization problem (2.3) in case of non-convex objective functions. The latter problem, however, is inherent to the non-convex problems, whereas the weakly Pareto optimal solutions can be avoided by an adjustment of the formulation of the single-objective problem. For example, in [203, 204] it is suggested to augment the objective function in (2.3) by a small term proportional to the Manhattan distance between the decision and the utopian points

$$\min_{\mathbf{x}\in\mathbf{A}} \left( \left[ \max_{i=1,\ldots,m} w_i \left(f_i(\mathbf{x}) - u_i\right) \right] + \rho \sum_{i=1}^{m} \left(f_i(\mathbf{x}) - u_i\right) \right), \tag{2.5}$$

where $\rho$ is a small positive scalar.

A similar modification of the weighted Tchebycheff problem is proposed in [97]

$$\min_{\mathbf{x}\in\mathbf{A}} \left( \max_{i=1,\ldots,m} w_i \left[ \left(f_i(\mathbf{x}) - u_i\right) + \rho \sum_{i=1}^{m} \left(f_i(\mathbf{x}) - u_i\right) \right] \right). \tag{2.6}$$

For both modified weighted Tchebycheff problems (2.5) and (2.6) it is true that they generate only strongly Pareto optimal solutions, and any such solution can be found by selecting the appropriate parameters [135]. The metrics used in (2.5) and (2.6) are indeed very similar and differ in the slightly different slopes of the hyper-planes which form the sphere in the considered metrics. Such a sphere can be imagined as a merger of spheres in $L_1$ and $L_\infty$ metrics; we refer to [97] for details and further references.

The weakly Pareto optimal solutions can be excluded in the Tchebycheff metric-based approach also by the application of the lexicographic minimization of two objectives where the second objective is $L_1$ distance between the utopian and searched solutions:

$$\operatorname{lex}\min_{\mathbf{x}\in\mathbf{A}} \left[ \left( \max_{i=1,\ldots,m} w_i \left(f_i(\mathbf{x}) - u_i\right) \right), \sum_{i=1}^{m} \left(f_i(\mathbf{x}) - u_i\right) \right]. \tag{2.7}$$

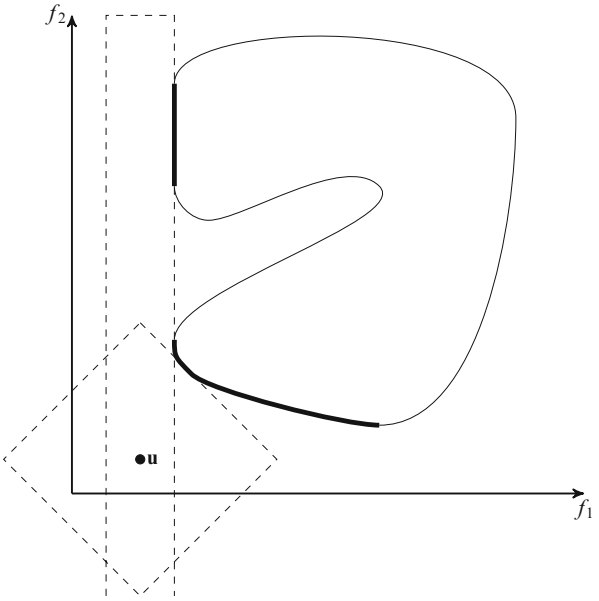


**Fig. 2.1** Illustration of the lexicographic weighted Tchebycheff method

For illustration we refer to Figure 2.1.

As shown in [204] the solution of the lexicographic weighted Tchebycheff problem (2.7) is Pareto optimal. On the other hand, for a Pareto optimal decision $\tilde{\mathbf{x}}$ there exists a vector of weights $\mathbf{w}$, $w_i > 0, i = 1, \ldots, m$, $\sum_{i=1}^{m} w_i = 1$ such that $\tilde{\mathbf{x}}$ is a unique optimal decision for the problem (2.7).

The most important advantage of the Tchebycheff metric-based approach is the very weak assumptions ensuring the potential possibility to find every Pareto optimal solution. Various versions of the Tchebycheff methods are widely used for solving practical problems.

## 2.3 Achievement Scalarization Function

A method which is formally similar to the weighted Tchebycheff method is defined by the formula (2.3) with the only difference, namely instead of the utopian vector $\mathbf{u}$ any reference point $\mathbf{z}$ can be used

$$\min_{x \in \mathbf{A}} \max_{i=1,\ldots,m} w_i \, (f_i(\mathbf{x}) - z_i). \tag{2.8}$$

The solution $\mathbf{f}(\mathbf{x}_{\mathbf{w},\mathbf{z}})$, where $\mathbf{x}_{\mathbf{w},\mathbf{z}}$ is the minimizer in (2.8), is a weakly Pareto optimal solution independently of the feasibility or infeasibility of $\mathbf{z}$. The formula (2.8) defines a special case of the Achievement Scalarizing Function method proposed in [228].

## 2.4 $k$th-Objective Weighted-Constraint Problem

A scalarization method proposed in [26] is applicable not only to the problems with a disconnected Pareto front but also to the problems with a disconnected feasible set under the mild assumptions that the objective functions are continuous and bounded from below with a known lower bound; the latter assumption is reformulated as $f_i(\mathbf{x}) > 0$. This scalarization technique is named by its authors the "$k$th-objective weighted-constraint problem," since for each fixed $k$, the $k$th-objective is minimized, while the other weighted objective functions are incorporated as constraints:

$$\begin{aligned} &\min_{\mathbf{x}\in\mathbf{A}} w_k f_k(\mathbf{x}), \\ &w_i f_i(\mathbf{x}) \le w_k f_k(\mathbf{x}),\ i = 1,\dots,m,\ i \ne k, \\ &w_i > 0,\ \sum_{i=1}^{m} w_i = 1. \end{aligned} \tag{2.9}$$

As shown in [26], $\tilde{\mathbf{x}}$ is a weakly Pareto optimal decision of the original multi-objective optimization problem if and only if there exists some $\mathbf{w}$ such that $\tilde{\mathbf{x}}$ is an optimal decision of (2.9) for all $k = 1,\dots,m$.

Despite different formulas in the problem statement (2.3) and (2.9), the weakly Pareto optimal solution defined by the $m$ times repeated solution of (2.9) with different $k$ can be obtained by the solution of (2.3) with the same weights. Since the objective functions in (2.9) are assumed positive, the utopian vector in (2.3) is assumed equal to zero.

Let $\tilde{\mathbf{x}}$ be an optimal decision for $m$ problems (2.9) with different $k$. Then the following inclusion is true

$$\begin{aligned} &\tilde{\mathbf{x}} \in \bigcap_{i=1}^{m} \mathbf{A_i} \\ &\mathbf{A_i} = \{\mathbf{x} :\ \mathbf{x} \in \mathbf{A},\ w_i f_i(\mathbf{x}) \ge w_j f_j(\mathbf{x}),\ j = 1,\dots,m,\ j \ne i\}. \end{aligned} \tag{2.10}$$

Because of the equality $w_i f_i(\tilde{\mathbf{x}}) = w_j f_j(\tilde{\mathbf{x}})$, $j \ne i$, which follows from (2.10), and since

$$\tilde{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathbf{A_i}} f_i(\mathbf{x}),\ i = 1,\dots,m, \tag{2.11}$$

$\tilde{\mathbf{x}}$ is a minimizer of the minimization problem

$$\min_{\mathbf{x}\in\mathbf{A}} \max_{1\le i\le m} w_i f_i(\mathbf{x}), \tag{2.12}$$

which for $\mathbf{f}(\mathbf{x}) > \mathbf{0}$, $\mathbf{u} = \mathbf{0}$, is coincident with the optimal decision for (2.3).

## 2.5 Pascoletti and Serafini Scalarization

Pascoletti and Serafini scalarization [156] involves two vectors of parameters $\mathbf{u} \in \mathbb{R}^m$ and $\mathbf{v} \in \mathbb{R}^m$, and is defined by the following minimization problem:

$$\begin{aligned} &\min_{t,\mathbf{x}} t, \\ &\mathbf{u} + t\mathbf{v} - \mathbf{f}(\mathbf{x}) \in \mathbb{R}^m_+, \\ &t \in \mathbb{R},\ \mathbf{x} \in \mathbf{A}, \end{aligned} \tag{2.13}$$

where $\mathbb{R}^m_+$ denotes the non-negative hyper-orthant. The formulation (2.13) is a special case of more general definition in [50, 51] where a nonempty closed pointed convex cone is used instead of $\mathbb{R}^m_+$. Assuming that the functions $f_i(\cdot)$ are continuously differentiable, and the feasible region $\mathbf{A}$ is compact, it can be proved that a weakly Pareto optimal solution of the multi-objective minimization problem

$$\min_{\mathbf{x}\in\mathbf{A}} \mathbf{f}(\mathbf{x}), \tag{2.14}$$

can be reduced to the solution of (2.13), i.e., the following statements are true:

- If $\tilde{\mathbf{x}}$ is a Pareto optimal decision vector of (2.14), then $(0, \tilde{\mathbf{x}})^T$ is a global minimizer of (2.13).
- Let $(\tilde{t}, \tilde{\mathbf{x}})^T$ be a global minimizer of (2.13), then $\tilde{\mathbf{x}}$ is a weakly Pareto optimal decision of (2.14).

# Chapter 3
# Approximation and Complexity

## 3.1 Some Results on the Complexity of Multi-Objective Optimization

In computer science, the theory of computational complexity was developed to evaluate the hardness of problems, regarding the number of steps to obtain a solution; for the basic results of that theory we refer to [147]. The model of computations, used in that theory, is either the Turing machine or any other equivalent model. The input and output of the Turing machine should be encoded as finite strings of bits. Such an encoding can be used to represent the objects of discrete nature, and the theory of computational complexity is favorable, e.g., to the investigation of problems of single-objective combinatorial optimization [150, 151]. However, the Turing machine model is not suitable for the algorithms using real numbers. Therefore alternative complexity theories have been developed for the investigation of problems of continuous nature. For the fundamentals of the complexity of real number algorithms we refer to [19, 218], and for the complexity of problems of mathematical programming to [143, 150, 151].

Many important combinatorial optimization problems are NP-complete in the single-objective version, and obviously their multi-objective versions are also NP-complete. To answer the question about the complexity of multi-objective versions of the problems, the single-objective versions of which are tractable, is not so simple. Similarly to the case of single-objective optimization problems, the existence of the polynomial complexity approximation of the NP-complete multi-objective problems is of great interest. To illustrate a typical relation between complexity of single- and multi-objective versions of a problem in combinatorial optimization, a well-known example of the shortest path problem is considered.

Let us start with a statement of combinatorial optimization problems. A finite set of items $\mathbf{E} = \{e_1, \ldots, e_n\}$ is given, and a set of combinations of interest $\mathbf{S}$ is a subset of $2^{\mathbf{E}}$. A subset of $\mathbf{S}$ should be found which consists of non-dominated



P.M. Pardalos et al., *Non-Convex Multi-Objective Optimization*, Springer
Optimization and Its Applications 123, DOI 10.1007/978-3-319-61007-8_3

elements with respect to a vector objective function $\mathbf{f}(\mathbf{e}) = (f_1(\mathbf{e}), \ldots, f_m(\mathbf{e}))^T$, $\mathbf{e} \in \mathbf{S}$. The objective functions of many combinatorial problems are defined as the sums of weights of elements in $\mathbf{e}$

$$f_i(\mathbf{e}) = \sum_{j=1}^{n} w_i(e_j) x_j,$$

where $w_j(\cdot)$ is the weight function representing the $j$ objective, and $x_j \in \{0, 1\}$. Now the problem is formulated as a 0-1 multi-objective optimization problem where the feasible region for $\mathbf{x} = (x_1, \ldots, x_n)^T$ is obtained as the image of $\mathbf{S}$ by the mapping

$$2^{\mathbf{E}} \to \{\mathbf{x} : x_i \in \{0, 1\}\}.$$

As a special case the bi-objective shortest path problem is considered below. Let $(\mathbf{V}, \mathbf{E})$ be a graph with the set of vertices $\mathbf{V} = \{v_1, \ldots, v_K\}$ and the set of edges $\mathbf{E} = \{e_1, \ldots, e_N\}$. The shortest path should be found from the source vertex $s \in \mathbf{V}$ to the target vertex $t \in \mathbf{V}$ taking into account two weight functions $w_1(e_i)$, $w_2(e_i)$, $i = 1, \ldots, N$. This problem is a bi-objective combinatorial minimization problem

$$\min_{\mathbf{x} \in \mathbf{A}} \mathbf{f}(\mathbf{x}),$$

where $\mathbf{x} = (x_1, \ldots, x_N)^T$, $x_i \in \{0, 1\}$, $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}))^T$,

$$f_j(\mathbf{x}) = \sum_{i=1}^{N} w_j(e_i) x_i,\ j = 1, 2,$$

and $\mathbf{A}$ is a set of $\mathbf{x}$ corresponding to paths from $s$ to $t$ in $(\mathbf{V}, \mathbf{E})$.

As it is well known, several efficient algorithms are available for the single-objective shortest path problem; we refer to the textbooks on graph theory, e.g., the Dikstra's algorithm for graphs with edges of positive weights, and the Bellman–Ford algorithm for graphs with edges of arbitrary sign weights. The efficiency here means that the solution time can be bounded by a low degree polynomial in the problem size. Let us analyze the complexity of the bi-objective shortest path problem. The associated decision problem is the following. Given constants $c_1 > 0$, $c_2 > 0$ does there exist a feasible solution $\mathbf{x} \in \mathbf{A}$ such that $f_j(\mathbf{x}) \leq c_j$, $j = 1, 2$? The complexity of a decision problem respective to a multi-objective optimization problem is determined in the same way as in the single-objective case. If there exists a deterministic polynomial time Turing machine which solves the considered decision problem, then the problem belongs to the class of polynomial complexity. The term "polynomial time" means that the needed number of steps of the considered Turing machine programme is bounded by a polynomial in the size of the input. If there exists a nondeterministic polynomial time Turing machine which solves the considered decision problem, then the problem belongs to the class of

NP-complete problems. However, the well-developed theory of complexity enables the analysis of the complexity of a concrete problem without tedious involvement of Turing machines. The supposed NP-completeness can be proved by means of polynomial reducibility from a known NP-complete problem.

**Theorem 3.1** *The bi-objective shortest path problem is NP-complete.*

*Proof* The statement is proved by the polynomial reduction of the 0-1-Knapsack problem to the bi-objective shortest path problem [188]. The 0-1-Knapsack problem is formulated as the maximization problem

$$\max_{\mathbf{x}\in\mathbf{A}} \sum_{i=1}^{N} v_i x_i, \tag{3.1}$$

where

$$\mathbf{A} = \{\mathbf{x} : x_i \in \{0, 1\}, \sum_{i=1}^{N} u_i x_i \leq U\},$$

and the input data is interpreted as follows: $N$ is the number of items supposed for potential selection, $v_i$ is the value of $i$-th item, $u_i$ is the weight of $i$-th item, and $U$ is the maximum allowed weight of selected items. The maximum value collection of items should be selected not exceeding, however, the allowed weight. The values and weights are assumed positive integer numbers. The decision problem version of the 0-1-Knapsack problem is the following: does there exist $\mathbf{x} \in \mathbf{A}$ such that

$$\sum_{i=1}^{N} v_i x_i \geq C, \tag{3.2}$$

for any positive integer constant $C$.

Let us consider the bi-objective shortest path problem for the directed graph $(\mathbf{V}, \mathbf{E})$ where

$$\mathbf{V} = \{r_i,\ i = 0, \ldots, N+1\},\ \mathbf{E} = \{e_i,\ i = 0, \ldots, N,\ d_i,\ i = 1, \ldots, N,\ \},$$

and the edges $e_i$ and $d_i$ connect the vertices $r_i$ and $r_{i+1}$, $e_o = (r_o, r_1)$. The bi-objective valued weights of the edges are defined as follows:

$$\mathbf{w}(e_o) = (2C, 0)^T,\ \mathbf{w}(e_i) = (-v_i, u_i)^T,\ i = 1, \ldots, N, \tag{3.3}$$

$$\mathbf{w}(d_i) = (0, 0)^T,\ i = 1, \ldots, N. \tag{3.4}$$

The bi-objective shortest path problem is to find non-dominated paths from $r_o$ to $r_{N+1}$ taking into account two objectives, i.e., the following optimization problem should be solved:

$$\min_{\mathbf{x}\in\mathbf{A}} \mathbf{f}(\mathbf{x}), \tag{3.5}$$

$$\mathbf{f}(\mathbf{x}) = \left(2C - \sum_{i=1}^{N} v_i x_i, \quad \sum_{i=1}^{N} u_i x_i\right)^T,$$

where $\mathbf{A} = \{\mathbf{x} : x_i \in \{0, 1\}\}$; while constructing a path the edge $e_i$ is selected for $x_i = 1$, and $d_i$ is selected for $x_i = 0$.

To show that the bi-objective shortest path problem is polynomial time reducible from the knapsack problem, let us assume that the decision problem version of the shortest path problem is solvable for the constants $C$, $U$, i.e., we assume that there exists such $\mathbf{x}$ that

$$2C - \sum_{i=1}^{N} v_i x_i \le C, \quad \sum_{i=1}^{N} u_i x_i \le U. \tag{3.6}$$

The first inequality of (3.6) can be rewritten in the form

$$\sum_{i=1}^{N} v_i x_i \ge C \tag{3.7}$$

meaning that the solution of (3.6) is also a solution of the decision version of the knapsack problem (3.2). The vice versa is also obvious: the solution of (3.2) is also the solution of (3.6).

The NP-completeness of the bi-objective shortest path problem is implied by its polynomial reducibility from the NP-complete 0-1-Knapsack problem. □

Similar conclusions are true, e.g., concerning the so-called unconstrained combinatorial optimization problem and assignment problem: their single-objective versions are solvable in polynomial time but their bi-objective versions are NP-complete; see [49].

Let us note, as the complement to the theorem, that an example of the bi-objective shortest path problem is known where the number of non-dominated paths is exponential in size of the problem [77].

Since important multi-objective optimization problems are NP-hard their approximate solution is of great interest. The construction of special approximation algorithms is urged by various applications, however, frequently challenged by the complexity of not only exact but also approximate solution. For example, the complexity of computing an approximate solution of multi-objective shortest path problem is considered in [148] in relation with the optimal access to the Web resources. Some theoretical results below represent the state of the art in the complexity of the approximate solution of the multi-objective NP-complete problems.

The approaches to approximation in computer science and in numerical analysis are focused on different assumptions, e.g., in computer science algorithms operate on the finite binary strings while in numerical analysis operations with real numbers are considered. Accordingly to the basic assumptions the results of both approaches are formulated.

To formulate the properties of the approximating problems in the terms of computer science the multi-objective optimization problems should be stated, respectively; we follow here the original presentation in [148]. The problem has a set of valid instances $\mathbf{I} = \{I_k,\, k = 1, \ldots, N\}$ represented by strings over a fixed finite alphabet. An instance $I \in \mathbf{I}$ is related to the set of feasible decisions $\mathbf{D}(I)$ which are polynomially bounded and polynomially recognizable in the size of the instance; e.g., a decision $\mathbf{x} \in \mathbf{D}(I)$ corresponds to a finite dimensional vector with the components either 0 or 1. The objective function $\mathbf{f}(I, \mathbf{x})$ is defined by a polynomial time algorithm which, for given $I, \mathbf{x}$, computes an $m$-dimensional vector, the components of which are the rational numbers. We use term "decision" for $\mathbf{x}$ instead of usual for computer science "solution" to agree with the terminology of the theory of multi-objective optimization accepted in the introductory chapter of this book where "solution" is reserved for function values $\mathbf{f}(\cdot)$. The notation $\mathbf{f}(\mathbf{x})$ will be used where the omission of the first argument will not cause a misunderstanding.

Assume the values of objective functions be positive: $f_i(I, \mathbf{x}) > 0$, $i = 1, \ldots, m$; this assumption, together with the assumptions on polynomially bounded size of $\mathbf{x}$, and on the polynomial time algorithm defining $\mathbf{f}(I, \mathbf{x})$, implies the following bounds for $\mathbf{f}(I, \mathbf{x})$:

$$2^{-p(|I|)} \le f_i(I, \mathbf{x}) \le 2^{p(|I|)},$$

for some polynomial $p(\cdot)$.

Let $\mathbf{P}(I)$ denotes the set of Pareto optimal solutions of an instance of the multi-objective optimization problem $I$. In some publications, e.g., [43, 148, 213] $\mathbf{P}(I)$ is called "Pareto curve" also in the case $m > 2$. Since $\mathbf{P}(I)$ is exponentially large for many problems of interest, investigating the size of its approximation is an urgent problem, investigation of which was initiated in [148].

**Definition 3.1** The set of decisions $\mathbf{x}$ of a multi-objective optimization problem $I$ is called an $\varepsilon$-approximate Pareto set and denoted $\mathbf{P}_\varepsilon(I)$, if

$$\mathbf{f}(I, \mathbf{x})/(1 + \varepsilon),\ \mathbf{x} \in \mathbf{P}_\varepsilon(I),$$

is not dominated by any $\mathbf{f}(I, \mathbf{x})$, $\mathbf{x} \in \mathbf{D}(I)$.

The answer to the question on the existence of a polynomial size approximation of the Pareto set is given by the following theorem [148]:

**Theorem 3.2** *For any multi-objective optimization problem and any $\varepsilon > 0$ there exists a $\mathbf{P}_\varepsilon(I)$ consisting of a number of solutions that is polynomial in $|I|$ and $\frac{1}{\varepsilon}$ (but may be exponential in the number of objectives).*

Although this result is valid for any multi-objective optimization problem satisfying rather weak assumptions, it remains to answer the question whether $\mathbf{P}_\varepsilon(I)$ can be constructed in polynomial time or not. A general result, generalizing the characterization of the approximability in the single-objective case, is presented in [148]:

**Theorem 3.3** *There is an algorithm for constructing* $\mathbf{P}_\varepsilon(I)$ *polynomial in* $|I|$ *and* $\frac{1}{\varepsilon}$ *if and only if the following problem can be solved: given* $I$ *and* $\mathbf{b} = (b_1, \ldots, b_m)^T$ *either return* $\mathbf{x}$ *with* $\mathbf{f}(I, \mathbf{x}) \leq \mathbf{b}$ *or answer that there is no decision* $\grave{\mathbf{x}}$ *with* $\mathbf{f}(I, \grave{\mathbf{x}}) \leq \mathbf{b}/(1+\varepsilon)$.

In some special cases (e.g., linear objective functions, discrete feasible regions) Theorem 3.3 is applied to derive the conditions for the existence of fully polynomial time approximation schema for constructing $\mathbf{P}_\varepsilon$ [148]. These conditions can also be verified relatively easily, e.g., multi-objective versions of the shortest path, minimum spanning tree, and matching problems. Such results show that for some classical graph problems the approximate solution of their multi-objective versions is not much harder than the solution of their single-objective versions. For further investigation of complexity of the approximate solution of similar multi-objective problems (e.g., linear or convex) we refer to [43, 44, 49, 213].

In the present book we focus mainly on the non-convex continuous multi-objective optimization where the complexity problem is researched not so well as in discrete optimization. Further in the book, we will touch upon the question on complexity of the single-objective and multi-objective versions of an optimization problem (in Chapter 6) where the approximate worst-case solutions of multi-objective optimization problems with Lipschitzian objectives are considered.

The complexity of single-objective optimization problems is measured by the growth rate of solution time and of memory requirements in response to growth of the problem size, e.g., the dimensionality of domain of the objective function in question. Similar complexity measures are used also with regard to the multi-objective problems. However, in the case of multi-objective problems a new complexity aspect emerges, namely the growth of required computational resources in response to the growth of the number of objectives. The latter growth is drastically fast, and a multi-objective optimization problem yet with four objectives can be intractable in case special favorable properties are not available. To emphasize the difficulties in solution of multi-objective problems with four and larger number of objectives a new term is suggested in [39] to define such problems: "many-objective optimization". The extension of known methods, well applicable for problems with up to three objectives, for problems of larger number of objectives is challenging and slow.

## 3.2 Approximate Representation of Pareto Sets

The publications on the approximation of Pareto sets by the methods referable to numerical analysis are indeed numerous. A Pareto set can be approximated by means of the repeated application of well-developed single-objective optimization methods where one vector of the Pareto set approximation is obtained as a solution of an appropriately formulated single-objective optimization problem. This widely used approach is called a parametric scalarization since the mentioned auxiliary single-objective optimization problem is normally selected from a parametric class of problems. To ensure the existence of the solution of a considered auxiliary single-objective optimization problems we assume that the feasible regions are compact, and the objective functions are continuous if explicitly is not stated otherwise.

To obtain a good representation of the set of Pareto optimal solutions, a scalarized problem should be solved repeatedly selecting various values of parameters, e.g., in the case of (2.13) appropriate values of $\mathbf{u}$ and $\mathbf{v}$ should be selected. The selection of an appropriate set of parameters is a challenge for all scalarization methods. Many various methods for selecting parameters of scalarization problems have been proposed, see, e.g., [154]. From the point of view of applications, a good approximation method should guarantee uniform distribution of representative points in (or close to) the true Pareto set and exclude the non-Pareto and local Pareto points.

Below we discuss several popular methods for the Pareto set approximation which are applicable in the case of non-convex objectives and discontinuous Pareto sets. The appropriate result should be found at reasonable computational cost. Therefore, scalarization methods can be not appropriate for the problems with expensive objective functions where the repeated solution of the auxiliary single-objective optimization problems can be challenging in view of the computational burden.

### *3.2.1 The ε-Constraint Method*

The $\varepsilon$-constraint method is a reformulation of an approach well known in engineering optimization where all but one objectives of interest are moved to the constraints, and the most important objective is optimized with respect to the imposed constraints. The $\varepsilon$-constraint method was introduced in [74] as follows:

$$\min_{\mathbf{x}\in\mathbf{A}(\boldsymbol{\varepsilon})} f_l(\mathbf{x}), \tag{3.8}$$

$$\mathbf{A}(\boldsymbol{\varepsilon}) = \{\mathbf{x} : \mathbf{x} \in \mathbf{A},\ f_j(\mathbf{x}) \le \varepsilon_j,\ \ j = 1,\ldots,m, j \ne l\},$$

where $\mathbf{A}$ is a feasible region of the original optimization problem, and $\boldsymbol{\varepsilon} = (\varepsilon_1,\ldots,\varepsilon_m)^T$.

The minimum point of minimization problem (3.8) $\mathbf{x}^*$ defines a weakly Pareto optimal solution $\mathbf{f}(\mathbf{x}^*)$. The proof of the respective theorem can be found, e.g., in [135] as well as the following theorem and its proof.

**Theorem 3.4** *A decision vector* $\mathbf{x}^* \in \mathbf{A}$ *is Pareto optimal if and only if it is an optimal decision for the* $\varepsilon$*-constraint problem (3.8) for every* $l = 1, \ldots, m$*, where* $\varepsilon_j = f_j(\mathbf{x}^*)$ *for* $j = 1, \ldots, m$*,* $j \neq l$*.*

The statements above show the possibility to approximate the set of Pareto optimal solutions for non-convex problems, without any further assumptions, by solving a series of $\varepsilon$-constraint problem. However, to ensure that the found solution is Pareto optimal, either $m$ different problems (3.8) should be solved or the uniqueness of the solution should be proved; for the theoretical substantiation of the latter statement we refer to [135].

It is relatively easy to distribute parameters $\boldsymbol{\varepsilon}$ evenly in a relevant region, however, such a distribution usually does not imply the uniform distribution of solutions closely to the true Pareto set. The relation between the distributions of the parameters of the auxiliary single-objective problems and found candidate solutions of the original multi-objective problem is too complicated to be decoded as illustrated in Figure 3.1. Therefore, the selection of a sequence of $\boldsymbol{\varepsilon}$, implying the desirable distribution of the obtained candidate solutions, is challenging. In the case of a simple structure of $\mathbf{A}$ the addition of a nonlinear non-convex constraint in the definition of the feasible region in (3.8) can make the problem more challenging than the minimization of $f_l(\mathbf{x})$ over $\mathbf{A}$.

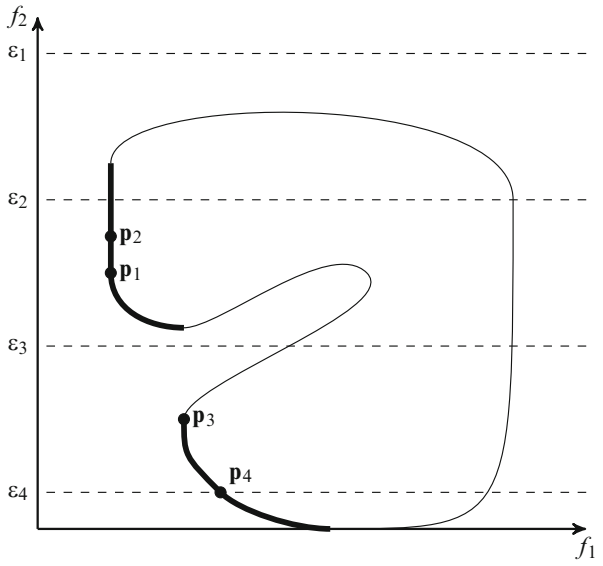


**Fig. 3.1** Illustration of the $\varepsilon$-constraint method

### 3.2.2 The Normalized Normal Constraint Method

The normalized normal constraint (NNC) method, proposed in [134], is based on a similar idea to the $\varepsilon$-constraint method: a Pareto optimal solution is found by means of minimization of one of the objective functions in the reduced feasible region where the reduction is implied by the constraints defined for the objective functions. The step of normalization includes $m$ single-objective minimizations

$$\bar{\mathbf{x}}_i = \arg\min_{\mathbf{x}\in\mathbf{A}} f_i(\mathbf{x}),\ \bar{\mathbf{y}}_i = f_i(\bar{\mathbf{x}}_i), \tag{3.9}$$

and subsequent re-scaling of the objective space so that $\bar{\mathbf{y}} = (0,\ldots,0)^T$, and the nadir point is equal to $\mathbf{r} = (1,\ldots,1)^T$. In the re-scaled objective space $m$ anchors are selected. Let the anchors be coincident with the vertices of the $m-1$-dimensional simplex $\mathbf{S}$

$$\mathbf{S} = \{\mathbf{y} :\ \sum_{i=1}^{m} y_i = 1,\ 0 \le y_i \le 1\},$$

and the points $\mathbf{v}_i \in \mathbf{S}$, $i = 1,\ldots,K$, are uniformly distributed in $\mathbf{S}$. The following $K$ single-objective minimization problems are solved yielding Pareto optimal solutions $\mathbf{p}_i$, $i = 1,\ldots,K$:

$$\mathbf{p}_i = \mathbf{f}(\mathbf{x}_i^*),\ \mathbf{x}_i^* = \arg\min_{\mathbf{x}\in\mathbf{A}_i} \mathbf{f}_m(\mathbf{x})$$

$$\mathbf{A}_i = \mathbf{A}\bigcap\mathbf{C}_i,\ i = 1,\ldots,K,$$

where $\mathbf{C}_i$ is the subset in the decision space defined by the hyper-planes passing through $\mathbf{v}_i$, and orthogonal to the edges of $\mathbf{S}$ which connect the vertex $(0,\ldots,0,1)^T$ with other vertices of $\mathbf{S}$:

$$\mathbf{C}_i = \{\mathbf{x} : f_j(\mathbf{x}) - f_m(\mathbf{x}) \le v_{ij} - v_{im},\ j = 1,\ldots,m-1\},$$
$$\mathbf{v}_i = (v_{i1},\ldots,v_{im})^T.$$

The bi-objective examples in [134] demonstrate the performance of the algorithm. Some further improvements are proposed in [127].

The challenges in the implementation and applications of the NNC method are similar to that of the $\varepsilon$-constraint method though the number of additional non-convex constraints in the definition of $\mathbf{C}_i$ is equal to $m-1$.

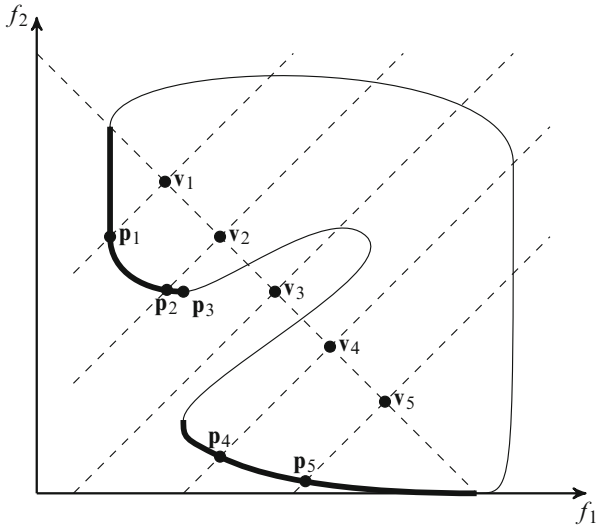


**Fig. 3.2** Illustration of the normalized normal constraint method

### *3.2.3 The Method of Normal Boundary Intersection*

The method of normal boundary intersection (NBI) [40, 41] has some similarity with NNC with respect to the construction. The idea of NBI is to find the Pareto optimal solutions as the points of intersection of lines orthogonal to the hyperplane extending $\mathbf{S}$ and passing the points $\mathbf{v}_i$. A shorthand CHIM is introduced by the authors of [41] to refer to the simplex defined by the individual minima of objectives $\mathbf{S}$. We use here the same notation as in the description of the NNC, however, it should be noted that for the formulas below, the re-scaling used in the description of NBC is not needed. The lines in question are illustrated in Figure 3.2. The method is defined by the formulas below

$$\max_{t,\,\mathbf{x}\in\mathbf{A}(\varepsilon)} \mathbf{t}, \tag{3.10}$$
$$\boldsymbol{\Phi}\mathbf{w} + t\mathbf{N} = \mathbf{f}(\mathbf{x}),$$

where $\mathbf{w}$ is the vector of weights

$$w_i \geq 0, \ \sum_{i=1}^{m} w_i = 1,$$

$t \in \mathbb{R}$, $\mathbf{N}$ is the unit vector orthogonal to CHIM simplex pointing to the origin, $\boldsymbol{\Phi}$ is the $m \times m$ matrix constituted of columns $\mathbf{f}(\bar{\mathbf{x}}_i) - \bar{\mathbf{y}}$, $\bar{\mathbf{x}}_i$ are defined by (3.9), and $\bar{\mathbf{y}}$ is the ideal point. The points $\mathbf{v}_i$ mentioned in the informal description of the NBI

method appear in (3.10) as the term $\boldsymbol{\Phi}\mathbf{w}$ with the appropriate vector of weights $\mathbf{w}$; a favorable sequence of $\mathbf{v}_i$ is selected via an appropriate selection of the sequence of $\mathbf{w}_i$.

In [41] examples are provided which show that the found solutions are well distributed in the Pareto front. However, in the case of non-convex objectives a solution of (3.10) does not necessarily define a Pareto optimal solution; an example is shown in Figure 3.2 where the closer to the origin intersection of the border of the feasible objective region with the line passing the point $\mathbf{v}_3$ is not a Pareto optimal solution. The other disadvantage of NBI is the existence of such cases ($m \geq 3$) where not all Pareto optimal solutions can be found by the solution of (3.10).

### 3.2.4 The Method of kth-Objective Weighted-Constraint

The method named "$k$th-objective weighted-constraint" [26] can be applied for the construction of the discrete representation of the set of Pareto optimal solutions under the mild assumptions that the objective functions are continuous and bounded from below with a known lower bound; the latter assumptions without loss of generality can be reformulated as $f_i(\mathbf{x}) > 0$, $\mathbf{x} \in \mathbf{A}$, $i = 1, \ldots, m$. As discussed in the previous chapter the set of solutions of

$$\min_{\mathbf{x}\in\mathbf{A}} w_k f_k(\mathbf{x}), \tag{3.11}$$

$$w_i f_i(\mathbf{x}) \leq w_k f_k(\mathbf{x}),\ i = 1, \ldots, m,\ i \neq k,$$

$$w_i > 0,\ \sum_{i=1}^{m} w_i = 1, \tag{3.12}$$

is coincident with the set of decisions respective to the weakly Pareto optimal solutions of the original multi-objective optimization problem. We briefly present the description of the version of the algorithm for bi-objective problems [26]. An approximation of the set of Pareto optimal solutions is found solving the series of the single-objective optimization problems with varying $k$ and $\mathbf{w}$ where a utopian vector $\mathbf{u}$ can be chosen in view of known lower bounds for the objective functions values:

**Step 1** Initialization: utopian vector $\mathbf{u} = (u_1, u_2)^T$; the number of partition points $N$ for the interval of weights.

**Step 2** Objective functions are minimized separately, and the interval of weights is defined

$$\mathbf{z}_i = \arg\min f_i(\mathbf{x}),\ f_i^* = f_i(\mathbf{z}_i),\ i = 1, 2,$$

$$v_1 = f_1(\mathbf{z}_2),\ v_2 = f_2(\mathbf{z}_1),$$

$$w_0 = (f_2^* - u_2)/(v_1 - u_1 + f_2^* - u_2),$$
$$w_N = (v_2 - u_2)/(v_2 - u_2 + f_1^* - u_1),$$
$$\mathbf{F}_0 = (f_1(\mathbf{z}_1), f_2(\mathbf{z}_1))^T,\ \mathbf{F}' = (f_1(\mathbf{z}_2), f_2(\mathbf{z}_2))^T$$

**Step 3** Discretization of the interval of weights

$$w_t = w_0 + t(w_N - w_0)/N, t = 0, 1, \ldots, N.$$

**Step 4** Computation of the (weak) Pareto front of the solutions of scalar problems
$s = 1$;
**for** $t = 0 : N$ **do**

$$\mathbf{x}_1 = \arg\min w_t f_1(\mathbf{x}),\ w_t f_1(\mathbf{x}) \geq (1 - w_t) f_2(\mathbf{x});$$
$$\mathbf{x}_2 = \arg\min (1 - w_t) f_2(\mathbf{x}),\ (1 - w_t) f_2(\mathbf{x}) \geq w_t f_1(\mathbf{x});$$
**if** $\mathbf{x}_1 = \mathbf{x}_2$ **then** $\mathbf{F}_s = \mathbf{f}(\mathbf{x}_1),\ s = s + 1;$
**else**
**if** $f_2(\mathbf{x}_1) \leq f_2(\mathbf{x}_2)$ **then** $\mathbf{F}_s = \mathbf{f}(\mathbf{x}_1),\ s = s + 1;$ **end**
**if** $f_1(\mathbf{x}_2) = f_1(\mathbf{x}_1)$ **then** $\mathbf{F}_s = \mathbf{f}(\mathbf{x}_2),\ s = s + 1;$ **end**
**end**

**end**
$\mathbf{F}_s = \mathbf{F}'$;
**Step 5** Output the discrete representation of the (weak) Pareto front $\mathbf{F}_i$, $i = 0, \ldots, s$.

### *3.2.5 An Example of Adaptive Method*

The discrete representation of the set of Pareto optimal solutions is computed by means of the solution of a sequence of single-objective optimization problems which depend on some parameters. To obtain a favorable distribution of the representation vectors, an appropriate sequence of parameters should be selected. However, the a priori selection of the sequence of parameters ensuring a desirable distribution of the solutions is possible very rarely, only in exceptional cases. The adaptive selection of parameters during the solution process can be more advantageous than its a priori definition.

To illustrate the adaptive approach let us consider a bi-objective minimization problem

$$\min_{\mathbf{x}\in\mathbf{A}} \mathbf{f}(\mathbf{x}), \quad \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}))^T,$$

with convex objectives. In this case a weighted sum scalarization can be applied, i.e., a Pareto optimal solution $\mathbf{y}_w$ is found as follows:

$$\mathbf{y}_w = f(\mathbf{x}_w), \quad \mathbf{x}_w = \arg\min_{\mathbf{x}\in\mathbf{A}} \left(wf_1(\mathbf{x}) + (1-w)f_2(\mathbf{x})\right), \tag{3.13}$$

$$0 \le w \le 1.$$

A reasonable initial sequence of weights is uniformly distributed in the interval $[0, 1]$.

A simple adaptive algorithm [154] follows:

**Step 1** *Initialization.* The representation of the Pareto front is a two-dimensional array containing the found solutions which are increasingly ordered with respect to the first objective. Solve the problem (3.13) with the weights equal to 0 and 1 (i.e., find the anchor points of the Pareto front). Normalize the function values to fit in the range $[0, 1]$. The representation of the Pareto front is initialized by two found solutions corresponding to the weights equal to 0 and 1.

**Step 2** *Selection of a subinterval for bisection.* Find the longest distance between neighbor solutions representing the Pareto front, and select the corresponding subinterval of weights for bisection.

**Step 3** *Update the representation.* Solve the problem (3.13) with the weight respective to the middle of the selected subinterval of weights. Insert the newly computed solution into the ordered representation of the Pareto front, and update the array of weights correspondingly.

**Step 4** *Termination.* If the stopping criterion is satisfied, then terminate. Otherwise, go to **Step 2**.

A termination criterion, for example, can be the decrease of the longest interval between the neighboring solutions below the predefined level.

# Chapter 4
# A Brief Review of Non-convex Single-Objective Optimization

## 4.1 Introduction

Single-objective optimization methods are among the mathematical methods most widely used in various applications. The classical methods starting with linear programming have been proved very valuable tools for solving various economic and engineering problems. However, the growing complexity of the applied problems demanded the development of new ideas, methods, and algorithms. The classical mathematical optimization methods are based on the assumption that an objective function is convex. The convexity assumption, which is very fruitful for theoretical investigation, is hardly provable in many practical applications. Moreover, it is not truth very frequently. Therefore in the 1960s of the last century there begun active research in global optimization of non-convex problems. Naturally, single-objective problems foremost attracted attention of researchers. In this section we briefly review the approaches to single-objective global optimization, the multi-objective counterparts of which are considered in subsequent chapters. For the comprehensive presentation we refer to the representative monographs.

The fundamental difficulty of non-convex problems is the possibility of existence of many local optima. Classical methods are not aimed at finding the best of them, the global optimum, but an arbitrary local optimum corresponding to some initial search conditions. Difficulties in extending the classical optimization theory to non-convex problems motivated a vast development of heuristic methods. Many of these methods exploited randomization of search and ideas from nature. An early development of a heuristic approach was represented, e.g., in [81, 175, 186]. During the initial period, the development of theoretically substantiated methods was considerably slower than that of various heuristics. Nevertheless, several theoretic approaches have emerged, e.g., based on statistics of extreme values [138]. An example of a heuristic method with subsequently well-developed theory is simulated annealing [1, 221]. During the past years research in heuristic and theoretically substantiated global optimization expanded. Very important impact to intensify the



P.M. Pardalos et al., *Non-Convex Multi-Objective Optimization*, Springer
Optimization and Its Applications 123, DOI 10.1007/978-3-319-61007-8_4

theory of global optimization has made the Journal of Global Optimization, the publication of which started in 1990. The results of the last century are summarized in [84, 152]. Further in the present book we consider extensions of theoretically substantiated methods of single-objective global optimization to the multi-objective case. Therefore, in this brief review of single-objective global optimization we do not consider heuristic methods, and refer to [8, 69] for thorough presentation of that subject.

In the 1960s, the very general idea of a branch and bound approach was fruitfully hybridized with Lipschitz function model starting the development of Lipschitz global optimization. According to the definition, the rate of changing of Lipschitz functions is bounded, and this property can be relatively easily exploited to construct a lower bound for a considered objective function. Availability of a sufficiently tight lower bound is a prerequisite for the construction of an efficient branch and bound type optimization algorithm. For the details of single-objective Lipschitz optimization we refer to monographs [86, 87].

At the same time several groups of researchers started intensive investigation of statistical models-based global optimization algorithms. In this approach the uncertainty in the behavior of an objective function is modeled by a statistical model, e.g., it is assumed that the considered single variable objective function is a sample function of a random process. An algorithm is constructed as an optimal (in some sense) procedure with respect to the chosen statistical model. The theory, algorithms, and applications of the statistical models-based global optimization algorithms are presented in monographs [139, 206, 208, 216, 239, 244].

Besides of substantiation by various similarities with processes in nature, random search was shown to be a prospective approach also mathematically [184, 233]. However, we are not going to cover all directions in the theory of random search but will focus on the methods based on extreme value statistics. For the thorough presentation of these methods we refer to [236, 238, 239].

This brief review is limited by the mentioned above approaches since the multi-objective optimization methods considered in the subsequent chapters can be seen as generalized versions of the reviewed single-objective counterparts.

## 4.2 Lipschitz Optimization

Lipschitz optimization [78, 86, 87, 159, 168, 208] is based on the assumption that the real-valued objective function $f(\mathbf{x})$ is Lipschitz continuous, i.e.,

$$|f(\mathbf{x}) - f(\mathbf{y})| \leq L \|\mathbf{x} - \mathbf{y}\|, \ \forall \mathbf{x}, \mathbf{y} \in \mathbf{A}, \quad 0 < L < \infty, \tag{4.1}$$

where $L$ is the Lipschitz constant, $\mathbf{A} \subset \mathbb{R}^d$ is compact, and $\|\cdot\|$ denotes a norm. Sometimes it is also assumed that the first derivative of the objective is also Lipschitz continuous. In Lipschitz optimization the Euclidean norm is used most often, but other norms can also be considered [157, 158].

Lipschitz optimization algorithms can estimate how far the current approximation is from the optimal function value, and hence can use stopping criteria that are more meaningful than a simple iteration limit. The methods can guarantee to find an approximation of the solution to a specified accuracy within finite time. Lipschitz optimization may be used in situations when an analytical description of the objective function is not available.

An important question in Lipschitz optimization is how to obtain the Lipschitz constant of the objective function or at least its estimate. There are several approaches [195]:

1. The Lipschitz constant is assumed given a priori [10, 86, 136, 170]. This case is very important from the theoretical viewpoint although in practice it is often difficult to use.
2. Adaptive global estimate over the whole search region is used [86, 109, 167, 208].
3. Local Lipschitz constants are estimated adaptively [112, 119, 189, 190, 194, 197, 208];
4. A set of possible values for the Lipschitz constant is used [59, 96, 160, 193, 194].

Overestimation of the Lipschitz constant implies the overestimated bounds and loss of the optimization efficiency. In this book we will consider that the Lipschitz constant is given a priori.

A univariate Lipschitz optimization is a well-studied case for which several algorithms have been proposed, compared, and theoretically investigated. A comprehensive survey is contained in [78].

The lower bound for $f(\mathbf{x})$ over a subregion $\mathbf{D} \subseteq \mathbf{A}$ may be evaluated by exploiting the Lipschitz condition (4.1). For all $\mathbf{x}, \mathbf{y} \in \mathbf{D}$

$$f(\mathbf{x}) \geq f(\mathbf{y}) - L\|\mathbf{x} - \mathbf{y}\|.$$

If $\mathbf{y} \in \mathbf{D}$ is sampled, then the concave function

$$F_{\mathbf{y}}(\mathbf{x}) = f(\mathbf{y}) - L\|\mathbf{x} - \mathbf{y}\| \tag{4.2}$$

underestimates $f(\mathbf{x})$ over $\mathbf{D}$.

Thus an algorithm for Lipschitz optimization may be built. Start from an arbitrary point $\mathbf{x}_1 \in \mathbf{A}$, compute the objective function value at this point $f(\mathbf{x}_1)$, and define the first lower-bounding function by

$$F_1(\mathbf{x}) = f(\mathbf{x}_1) - L\|\mathbf{x} - \mathbf{x}_1\|. \tag{4.3}$$

Then choose a next point $\mathbf{x}_2 \in \mathbf{A}$ where the lower-bounding function $F_1(\mathbf{x})$ is minimal:

$$\mathbf{x}_2 = \arg\min_{\mathbf{x} \in \mathbf{A}} F_1(\mathbf{x}).$$

Again concave lower-bounding function $f(\mathbf{x}_2) - L\|\mathbf{x} - \mathbf{x}_2\|$ may be defined and the best under-estimator of $f(\mathbf{x})$ on $\mathbf{A}$, given the information obtained so far, is

$$F_2(\mathbf{x}) = \max_{i=1,2}\{f(\mathbf{x}_i) - L\|\mathbf{x} - \mathbf{x}_i\|\}.$$

In the step $k$ the underestimating function $F_k(\mathbf{x})$ is

$$F_k(\mathbf{x}) = \max_{i=1,\ldots,k}\{f(\mathbf{x}_i) - L\|\mathbf{x} - \mathbf{x}_i\|\}, \tag{4.4}$$

and the next point $\mathbf{x}_{k+1}$ is given by

$$\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}\in\mathbf{A}} F_k(\mathbf{x}). \tag{4.5}$$

In the univariate case the function $\max_{\mathbf{x}_i\in\mathbf{A}} F_{\mathbf{x}_i}(\mathbf{x})$ is piecewise linear, and its minimum can be determined in a simple straightforward way [78]. Therefore, many univariate algorithms use the described underestimating function. The well-known methods are due to Piyavskii [169, 170] and Shubert [199].

Extensions of Piyavskii's method [169, 170] to the multivariate case have been proposed [94, 130, 136, 137, 170, 229, 230, 235]. When the Euclidean norm is used in the multivariate case, the lower bounding functions are envelopes of circular cones with parallel symmetry axes. A problem of finding the minimum of such a bounding function becomes a difficult global optimization problem. Therefore, other ways to build multivariate Lipschitz optimization algorithms were investigated.

One way is to reduce multivariate Lipschitz optimization to the univariate case. A nested optimization scheme was proposed by Piyavskii [169, 170]. Numerical approximations of space-filling curves may be used to reduce the original Lipschitz multidimensional problem to a univariate one satisfying the Hölder condition [27, 197, 207].

Another way is to partition the feasible region into simplices or hyper-rectangles and consider a small number of sample points in each subregion. Most often such algorithms are built using a branch and bound technique and fit into the general framework proposed by Horst and Tuy [83, 85, 86]. The proposed algorithms differ in the selection, branching and bounding rules [62, 63, 72, 132, 133, 159, 164–166].

## 4.3 Statistical Models-Based Global Optimization

### *4.3.1 General Assumptions*

The development of single-objective global optimization algorithms for some subclasses of non-convex functions is facilitated by the exploitation of analytical properties of objectives [87]. However, in some applications there occur optimization

problems where objectives are available either as a complicated computational model or as a closed code software. Such types of problems usually are named black-box optimization problems. According to this concept, the assumption on the uncertainty in properties of $f(\mathbf{x})$ seems quite natural. Nevertheless, normally that assumption can be softened postulating some relevant properties of the problem in question. At least, the assumption that $f(\cdot)$ is a continuous function normally is acceptable as well as $\mathbf{A}$ is a compact set. Besides the continuity, other analytical properties of $f(\mathbf{x})$ can be difficult to substantiate. Such unfavorable, from the optimization point of view, properties of $f(\mathbf{x})$ as non-differentiability, non-convexity, and multimodality cannot be excluded. To justify the search strategy in the described situation it is important to choose a model of uncertainty corresponding to the relevant information on the problem. A "rational optimizer" can choose a statistical model of uncertainty as it is justified in the expected utility theory [58], although other models, such as fuzzy logic and rough sets, are also applicable.

The focus of statistical models-based global optimization is on the black-box problems where objective functions are expensive; expensiveness here means a long-lasting computation of a value of the objective function normally implied by the complexity of the underlying computational model of the considered problem. The black-box optimization of expensive functions in many respects is noticeably different from the optimization of objective functions defined by analytical formulae. The expensiveness of objective functions imply limitations in both exploration and exploitation during the search. Therefore, the rationality of the search strategy is strongly required, i.e., the choice of points where to compute the objective function values should be well substantiated. The algorithms, founded on the principles of rational decision theory, here are of special interest. To construct such algorithms in the single-objective optimization case, statistical models of multimodal functions have proved very helpful [139, 208, 216, 239].

### *4.3.2 Statistical Models for Global Optimization*

Black-box optimization is aimed at the optimization of objective functions the properties of which are rather uncertain. The models of functions under uncertainty developed in probability theory are stochastic functions: random processes in the case of functions of one variable, and random fields in the case of functions of many variables. Assuming that the objective function is a sample function of a random process/field it would be attractive to construct a method of the best average performance with respect to the chosen statistical model. Stochastic models of objective functions are also helpful for the theoretical research on average complexity of global optimization problems; see, e.g., [28, 29].

For the successful implementation of the algorithms, by the considered approach, the selection/construction of an appropriate statistical model is crucial. We discuss here the relevant properties of the statistical models in detail since they are equally important for the case of multi-objective optimization.

The first paper on statistical model-based global optimization was the paper by Kushner [108]. The choice of a statistical model for constructing global optimization algorithms should be justified by the usual arguments of adequacy and simplicity. This is not always an easy task. The theory of stochastic functions is well developed from the point of view of probabilistic analysis. However, the properties known from the theory of stochastic functions are not always helpful in constructing global optimization algorithms. On the other hand, important properties for global optimization are largely outside of the interests of probability theoreticians. For example, a numerically tractable stochastic function with prescribed multimodality characteristics would be of great interest for global optimization. Since the Wiener process was the first statistical model used for global optimization, many discussions on advantages and disadvantages of the approach were related to its properties, e.g., the Wiener process seems to be an acceptable model as long as its local properties (non-differentiability of sampling functions) can be ignored. The Wiener process is not only used in global optimization but it is also an important model in the average case analysis of various numerical problems, e.g., one-dimensional interpolation, integration, and zero finding [28, 178]. The Wiener process $\xi(x),\ x \geq 0$, is a Gaussian random process with zero mean and covariance function $cov(\xi(x_1), \xi(x_2)) = \sigma_0^2 \min(x_1, x_2)$, where $\sigma_0^2$ is a parameter. The increments of the Wiener process $\xi(x+\delta) - \xi(x)$ are Gaussian random variables $N(0, \sigma_0^2 \delta)$, where the mean value is zero, and the variance is equal to $\sigma_0^2 \delta$; the increments corresponding to disjoint time intervals are independent. If a target problem is expected to be very complicated, with many local minima, then the assumption of independence of increments of function values for sufficiently distant disjoint subintervals seems acceptable. This is the main argument in favor of the global adequacy of the Wiener process to model complicated multimodal functions. The Wiener process is favorable also from the computational point of view because it is Markovian.

An important advantage of the Wiener process is the availability of an analytical formula for the probability distribution of the minimum [198]. This formula is used to compute the probability of finding the global minimum with predefined tolerance which in turn defines a natural stopping condition of a global minimization algorithm: to stop if the mentioned probability exceeds a predefined threshold, e.g., equal to 0.99 [216, 240, 241].

However, the sampling functions of the Wiener process are not differentiable with probability one, almost everywhere. This feature draws criticism for the use of the Wiener process as a model because objective functions with such severe local behavior do not arise in applications. Summarizing the advantages and disadvantages, the Wiener process often seems an acceptable model for a global description of objective functions but not as a local model. The latter conclusion has motivated the introduction of a dual statistical/local model where the global behavior of an objective function is described by the Wiener process, and its behavior in small subintervals of the main local minimizers is described by a quadratic function; see [216, 239].

An alternative choice of a statistical model is a stationary random process. The choice of Gaussian processes is dictated by implementation considerations. A process in this class is defined by the mean, variance, and correlation function. The desired model of smoothness can be chosen simply by choosing a random process with an appropriate correlation function. For example, a version of the P-algorithm for a smooth function model was constructed the implementation of which is of similar complexity to that of the P-algorithm based on the Wiener model [30].

Natural candidates for the statistical models of multimodal objective functions of many variables are random fields. There are many close analogies between properties of random processes and properties of random fields. For example, homogeneous isotropic random fields are generalizations of stationary processes. A Gaussian homogeneous isotropic random field is defined by its mean, variance, and correlation function $\varrho(\mathbf{x}, \mathbf{y}) = r(||\mathbf{x} - \mathbf{y}||)$, $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$; the functions $r(t) = \exp(-c|t|)$, $r(t) = \exp(-ct^2)$, $c > 0$, provide two examples of correlation functions used in different applications [140, 202]. However, not all properties of random processes can be easily generalized to random fields.

A significant simplification of computations implied by the Markov property in the one-dimensional case cannot be maintained in the multidimensional case. The inherent computational complexity of random fields restricts their application to the optimization of expensive (consuming much computing time) functions. For examples of the application of Gaussian homogeneous isotropic fields as statistical models for global optimization we refer to [139, 216].

Computational difficulties related to the use of random fields models motivated search for simpler statistical models. An axiomatic approach to the construction of statistical models is considered in [242] where the family of random variables $\xi_{\mathbf{x}}, \mathbf{x} \in \mathbf{A}$, is shown an acceptable statistical model of a class of objective functions. The distribution of $\xi_{\mathbf{x}}$ is normally accepted to be Gaussian because of computational reasons. The parameters of the family of random variables can be defined in a methodologically similar way, axiomatically defining a predictor (extrapolator/interpolator) of unknown value of the objective function [216]. For the simplified models used in the Bayesian approach we refer to [139].

The use of statistical models in optimization by some authors is called "kriging"; see, e.g., [95]. This term is borrowed from geo-statistics where random field models are used to predict values of functions of several variables. The main subject of kriging is linear predictors [202].

Several algorithms for global optimization of expensive objectives have been constructed using radial basis functions to model the behavior of the objective function considered. The radial basis function method is one of the most promising methods for multivariate interpolation using scattered data points [25]. Since we do not know a dedicated monograph on radial basis methods for global optimization, we refer to the original paper [73] and to the recent papers [176, 177]. Surprisingly, there exist deep analogies between the algorithms based on statistical and radial basis functions models; for details we refer to [239, 247].

### 4.3.3 Algorithms

It seems attractive to construct a global optimization algorithm that would search for the minimum where expected function values are small. Such an idea is appropriate in local minimization. However, some generalization is needed to apply this natural idea in global optimization. Of course, a local model of the function would be changed by a global model, e.g., by a global response surface. Nevertheless, searching in the neighborhood of the global minimizer of a response surface does not guarantee finding a global minimum of the objective function which can be located in a not yet researched subregion. The approximation of the objective function by a response surface in such a not well-researched subregion can be inaccurate, and therefore this subregion qualified as not prospective for search. For the rational search, the prediction of function values is needed as well as assessment of the uncertainty measure of the prediction. The rational planning of the current optimization step should balance the predicted value and its uncertainty. Such a balancing is facilitated by using a statistical model discussed in previous subsections.

Let us consider the choice of a point for the current computation of the objective function value. Such a choice in the black-box situation is a decision under uncertainty, and the rational decision theory [58] can be applied to make the choice rationally. The theory suggests to make a decision by maximizing the average utility. To compute the latter a statistical model of uncertainty is needed as well as a utility function. The axioms in [242] substantiate the acceptance of a random variable as a model of uncertainty for an unknown value of the objective function. Accordingly, a family of random variables $\xi_{\mathbf{x}}$ is acceptable as a statistical model of the objective function. A utility function corresponding to the conception of global optimization is proposed in [243]. These results substantiate the P-algorithm, i.e., to chose the point of current computation of the objective function value, where the probability of improvement is maximal. Let $n$ function values are computed at the previous minimization steps, and the values $y_i = f(\mathbf{x}_i)$, $i = 1, \ldots, n$, are obtained. The unknown value $f(\mathbf{x})$, $\mathbf{x} \neq \mathbf{x}_i$ is modeled by the random variable $\xi_{\mathbf{x}}$ whose conditional distribution with respect to the known function values is defined by the chosen statistical model. The site of $n + 1$ computation of the function value is defined by the P-algorithm according to the following formula:

$$\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}\in\mathbf{A}} \mathbf{P}\{\xi_{\mathbf{x}} \leq y^{*n} \mid \xi_{\mathbf{x}_1} = y_1, \ldots, \xi_{\mathbf{x}_n} = y_n\}, \tag{4.6}$$

where $\mathbf{P}(\cdot|\cdot)$ denotes the conditional probability and $y^{*n} < \min_{1\leq i\leq n} y_i$.

The one-variable version of the algorithm (4.6) based on the Wiener process model was proposed in [108]. For the axiomatic substantiation, theoretical analysis, and applications of the P-algorithm we refer to [216, 239].

The Bayesian method, where the idea of minimization of the average error is implemented, is presented in detail in [139]. In the later publications this algorithm was started to be called efficient global optimization (EGO) and kriging [88]. Another well-established direction in the development of statistical model-based global optimization algorithms is rooted in the information theory [208].

## 4.4 Branch and Probability Bound Methods

The deterministic branch and bound methods are briefly reviewed in Section 4.2. However, the construction of deterministic bounds is not always possible because of uncertainty in properties of the objective function considered. Such a situation is typical in the case of black-box optimization.

Let us consider a version of the branch and bound technique, which has been entitled by the branch and probability bound technique; see [236, 238] for a detailed description. Let us consider the search situation after $n$ objective function values $y_i = f(\mathbf{x}_i)$, $i = 1, \ldots, n$, were computed. According to this technique, the feasible region is subdivided into, say $K$ subsets (e.g., hyper-rectangles) $\mathbf{A}_k$, $k = 1, \ldots, K$, and a sample of uniformly distributed independent random points is generated in every subset. Similarly to the deterministic case, the subsets $\mathbf{A}_k$ are evaluated with respect to the prospectiveness of the continuation of further search. In the case a deterministic lower bound $y_k^-$ for function values $f(\mathbf{x})$, $\mathbf{x} \in \mathbf{A}_k$, is available, the subset $\mathbf{A}_k$ with $y_k^- < y_n^*$ is considered suitable for further search, where $y_n^* = \min_{i=1,\ldots,n} y_i$. The subsets with $y_k^- \geq y_n^*$ are discarded since guaranteed not to contain the global minimizer. Statistical methods cannot give guaranteed assessment on the prospectiveness of search in a considered subset but can be applied to evaluate the probability to find a better than $y_n^*$ objective function value during the subsequent search by means of generation of uniformly distributed independent random points in the considered subset. The subsets $\mathbf{A}_k$, for which such probabilities are small, are discarded. One of the subsets with sufficiently large probability is subdivided. The concrete strategies of subdivision and of selection of a subset for subdivision are defined taking into account specifics of a considered problem.

The probabilistic evaluation of the prospectiveness of $\mathbf{A}_k$ with respect to the continuation of the search is based on the results in statistics of extremal values known also as the record statistics [145]. Let $\xi$ denotes the objective function value at the random point with uniform distribution in $\mathbf{A}_k$. The cumulative distribution function of $\xi$ is denoted by $G_k(\cdot)$, and $\mathsf{m}_k = \inf_{\mathbf{x} \in \mathbf{A}_k} f(\mathbf{x})$. Let $\xi_i$, $i = 1, \ldots, K_k$, be the values of the objective function at the points $\mathbf{z}_i$ which are generated at random independently with uniform distribution in $\mathbf{A}_k$. The parameter $\mathsf{m}_k$ can be estimated using a functional defined on the sample $\xi_i$, $i = 1, \ldots, K_k$:

$$\tilde{\mathsf{m}}_k = \nu(\xi_1, \xi_2, \ldots, \xi_{K_k}).$$

In the theory of records [145], various estimators for $\mathsf{m}_k$ are proposed and investigated. Their relevance to the global optimization is considered in detail in [236, 238]. A most suitable estimator for a concrete situation is selected taking into account properties of the considered optimization problem, but the general schema of the algorithm remains the same as outlined next.

For each $\mathbf{A}_k$ the probability $\mathbf{P}(\tilde{\mathsf{m}}_k < y_n^* | G_k(\cdot))$ is estimated where dependence on $G_k(\cdot)$ is explicitly shown. If the estimated probability is smaller than a predefined threshold, then $\mathbf{A}_k$ is discarded. One of those $\mathbf{A}_k$, the estimated probabilities of which are larger than the threshold, is subdivided. For example, the subset with the largest probability can be selected for subdivision. In the case, where $\mathbf{A}_k$ are hyper-rectangular, subdivision frequently is performed as bisection by a hyper-plane orthogonal to the longest edge of $\mathbf{A}_k$. In the new subsets, obtained that way, the samples of $\mathbf{z}_i$ are supplemented by some number of random points generated with the uniform distribution. In case a smaller than yet known function value is found, $y_n^*$ is updated; in such a case it can be reasonable to perform a local search aiming at further improvement of $y_n^*$. The search is terminated when all subsets are discarded.

# Part II
# Theory and Algorithms

# Chapter 5
# Multi-Objective Branch and Bound

Branch and bound approaches for optimization problems were developed in the 1960s [114, 121]. The main concept of a branch and bound algorithm is to detect and discard sets of feasible decisions which cannot contain optimal decisions. The search process can be illustrated as a search tree with the root corresponding to the search space and branches corresponding to its subsets. An iteration of the algorithm processes a node in the search tree that represents an unexplored subset of feasible decisions. The iteration has three main components: selection of the subset to be processed, branching corresponding to subdivision of the subset, and bound calculation.

The rules of selection, branching and bounding may differ from algorithm to algorithm, but the general scheme of branch and bound remains the same. The algorithms start from initialization of the list of candidate sets covering the set of feasible decisions by one or more candidates. The branch and bound scheme aims to reduce the list of candidates and make it converge to the list of globally optimal decisions.

Three main strategies of selection are: best first—select a candidate with the minimal lower bound, depth first—select the youngest candidate, breadth first—select the oldest candidate. The total number of evaluated candidate sets for the depth first strategy may be a bit larger, but it saves memory and requires considerably less time for maintaining data structures [162]. This search strategy allows avoidance of storing of candidate sets [254], the number of the candidates may be very large and storing may require large amount of memory when other search strategies are used.

Evaluation of the bounds for the objective functions is a very important part of a branch and bound algorithm. If the bounds are not tight, the search may lead to a complete enumeration of all feasible decisions. Construction of bounds depends on the objective functions and the type of subsets of feasible decisions over which the bounds are evaluated.



P.M. Pardalos et al., *Non-Convex Multi-Objective Optimization*, Springer
Optimization and Its Applications 123, DOI 10.1007/978-3-319-61007-8_5

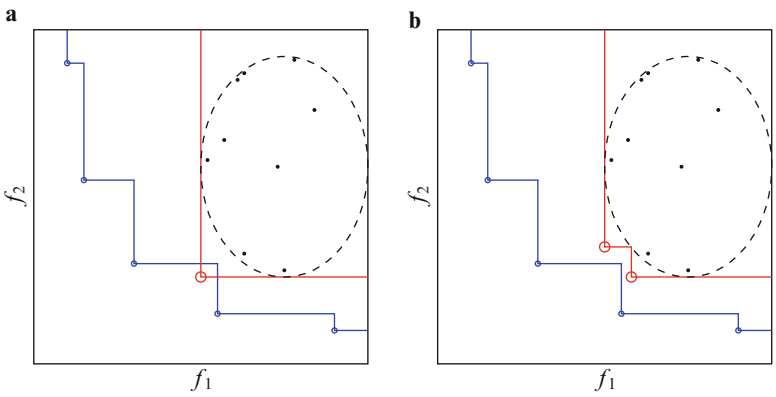


**Fig. 5.1** Bounding front: (**a**) single bounding vector, (**b**) multiple bounding vectors

In single-objective optimization, the subset cannot contain optimal decisions, and the branch of the search tree corresponding to the subset can be pruned, if the bound for the objective function over the subset is worse than a known function value.

In multi-objective optimization the subset cannot contain Pareto optimal decisions if the bounds for the objective functions over the subset are worse than a known objective vector. A bounding front may also be used—a set of bounding vectors in the objective space dominating all possible objective vectors corresponding to the subset of feasible decisions. The subset cannot contain efficient (Pareto optimal) decisions if each bounding vector $\mathbf{b} \in \mathbf{B}$ in bounding front $\mathbf{B}$ is dominated by at least one already known solution $\mathbf{a}$ in the current approximation $\mathbf{S}$ of the Pareto front:

$$\forall \mathbf{b} \in \mathbf{B}\ \exists \mathbf{a} \in \mathbf{S} :\ \forall i \in \{1, 2, \ldots, d\} :\ a_i \leq b_i\ \&\ \exists j \in \{1, 2, \ldots, d\} :\ a_j < b_j.$$

The simplest bounding front consists of a single ideal vector composed of lower bounds for each objective function.

Figure 5.1 illustrates a bounding front. The *dashed ellipse* circumscribes objective vectors (some are shown by *dots*) of a subset of the feasible decisions. Bounding vectors are shown with the largest *solid circles* shown in *red* and their dominated space is shown by the *red solid line*. They bound the objectives over the subset of the feasible decisions: all objective vectors (*dots*) are dominated by (are left and higher than) bounding front (*red solid lines*). *Small blue circles* represent currently known non-dominated objective vectors—they represent the current approximation of the Pareto front shown by *blue lines*. These currently known non-dominated objective vectors do not dominate the single bounding vector shown in Figure 5.1a, but both bounding vectors shown in Figure 5.1b are dominated—they are higher than the *blue lines*. Therefore, in such a situation branch and bound algorithm could discard the subset of the feasible decisions if bounding front with multiple bounding vectors is used, but cannot do so if the single bounding vector is used.

## 5.1 Branch and Bound for Continuous Optimization Problems

The rules of covering and branching depend on the type of partitions used. In interval methods for global optimization [79] hyper-rectangular partitions are used. In Lipschitz optimization [78, 86, 87, 168, 208] apart from such partitions, simplices may also be used [159].

Simplicial partitions are preferable when the objective function is evaluated at the vertices of subregions since a simplex has the minimal number of vertices. Initial triangulation is needed to cover hyper-rectangular feasible region by simplices [246]. However, feasible regions of different shapes may also be covered by simplices, for example, in optimization problems with linear constraints [161]. This can also be exploited when solving optimization problems where the objective functions have symmetries, for example, nonlinear least squares regression [257] and center-based clustering of data [160]. Subdivision of simplices is usually performed by bisecting the longest edge.

A bisection may be used to subdivide hyper-rectangles as well. In such a case a hyper-rectangle is subdivided perpendicular to the direction of the widest width into two. However, if the objective function is evaluated at all vertices of a hyper-rectangle, $2^{n-1}$ new samples are needed to subdivide a hyper-rectangle.

A central sampling strategy may be used in Lipschitz optimization where the value of the objective function at the center of a hyper-rectangle is used to evaluate bounds. If the bisection is used together with the central sampling, the known function value at the center is not used to evaluate descendant hyper-rectangles and two new samples are needed for subdivision. In order to reduce the number of function evaluations, a trisection may be used subdividing a hyper-rectangle into three, the center remains the same for the middle one and again two new samples are needed for subdivision. Although the objective function is evaluated at the same number of points, the resulting hyper-rectangles are smaller in the case of trisection. This is illustrated in Figure 5.2 where two new samples (*empty circles*) are used in subdivision in both cases.

Another popular approach in Lipschitz optimization is a diagonal approach where the objective function is evaluated at the endpoints of a diagonal [191, 192, 194] and the bounds are computed taking into account two samples per each hyper-rectangle. In the case of a bisection two function evaluations are needed during subdivision. The same number of evaluations is needed in the case of the trisection if the direction of the diagonal in the central descendant hyper-rectangle is changed. This is illustrated in Figure 5.3 where we see that two new endpoints of the diagonals (*empty circles*) in addition to previously evaluated ones (*filled circles*) are needed in both cases. The objective function is evaluated at the same number of points, but the resulting rectangles are smaller in the case of trisection.

The same number of function evaluations is needed during subdivision in both sampling strategies. However, in the case of the diagonal approach each hyper-rectangle contains two samples and only one sample in the case of the central

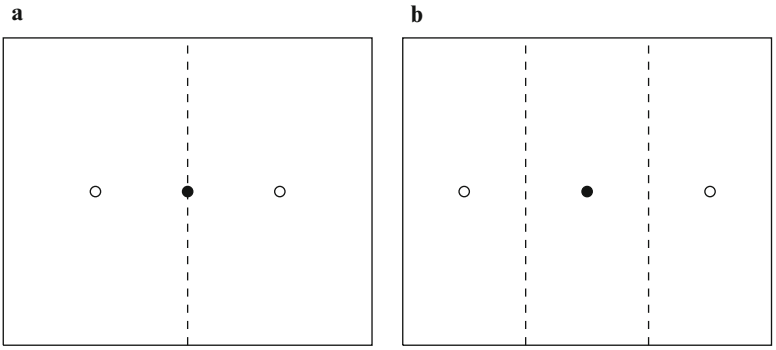


**Fig. 5.2** Central sampling: (**a**) Bisection of a rectangle, (**b**) Trisection of a rectangle

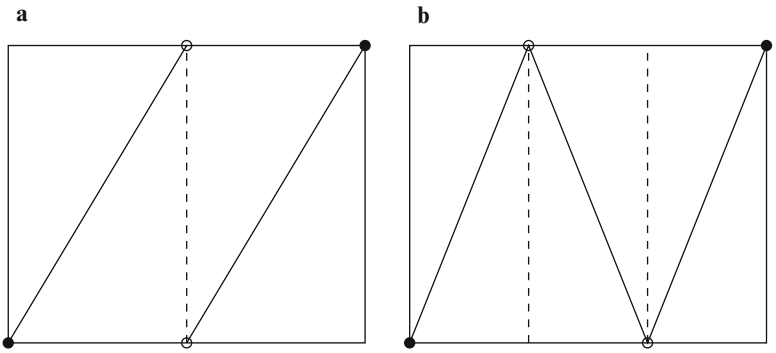


**Fig. 5.3** Diagonal partitioning: (**a**) Bisection of a rectangle, (**b**) Trisection of a rectangle

sampling. Moreover, a vertex database may be used to avoid reevaluation of the objective function at repeating sample points in neighboring subregions.

Lipschitz condition (4.1) may be used to build bounds for the objective functions over a subregion $\mathbf{D}$. The values of the objective functions are computed at sample points $\mathbf{x}_j \in \mathbf{D}$ in the subregion to be evaluated and lower-bounding functions are built

$$F_i(\mathbf{x}) = \max_{j=1,\ldots,k} \left\{ f_i\left(\mathbf{x}_j\right) - L\|\mathbf{x} - \mathbf{x}_j\| \right\}.$$

A single bounding vector may be computed from the minima of bounding functions:

$$\mathbf{b} = \left( \min_{\mathbf{x}\in\mathbf{D}} F_1(\mathbf{x}), \min_{\mathbf{x}\in\mathbf{D}} F_2(\mathbf{x}), \ldots, \min_{\mathbf{x}\in\mathbf{D}} F_m(\mathbf{x}) \right)^T.$$

Bounding fronts for multidimensional bi-objective Lipschitz optimization will be described in Section 6.3.2.

Interval optimization methods are based on interval arithmetic proposed in [141]. The interval arithmetic operates with real intervals whose ends are real numbers. Every real arithmetic operation has the corresponding interval arithmetic operation whose result contains every possible number which can be formed by the real arithmetic operation with any real numbers from the intervals.

The lower and upper bounds for the function values over a subregion defined by the intervals of variables can be estimated applying interval operations with intervals instead of real operations in the algorithm of calculation the function values. The bounds may be used to detect subregions not containing a global minimizer. Such subregions may be discarded from the further search. The first version of the interval global optimization algorithm was oriented to minimization of a rational function by bisection of subregions [200].

Interval methods for global optimization were further developed in [75, 76], e.g., the interval Newton method and the test of strict monotonicity were introduced. A thorough description including theoretical as well as practical aspects can be found in [79] where a very efficient interval global optimization method involving monotonicity and non-convexity tests and special interval Newton method is proposed. If the objective function is differentiable it is possible to compute the intervals of the derivatives and discard the subregions where the objective function is monotone. If the objective function is twice continuously differentiable it is also possible to compute the intervals of the second derivatives and discard the subregions where the objective function is concave. If the objective function is twice differentiable the special interval Newton method can be applied to reduce the subregions, and discard the subregions where there are no stationary points.

Interval branch and bound algorithm for global optimization was generalized to multi-objective optimization problems [185]. Multi-objective discarding tests based on conditions for Pareto optimality and interval bounds for the derivatives may be used. For example, if there is a variable for which the partial derivatives of the objective functions are either all positive or all negative over subregion, then the subregion cannot contain any Pareto optimal decision.

An interesting approach to solve multi-objective optimization problems is using a branch and bound search in the outcome space of the problem instead of the decision space [16, 17]. Since the dimension of the outcome space is usually smaller than the dimension of the decision space such an approach can be expected to shorten the search. The algorithm can be also modified to obtain an approximate global optimal weakly efficient solution after a finite number of iterations.

Although the rules of bounding, selection, covering, and branching may differ from algorithm to algorithm, the general scheme of branch and bound remains the same. Therefore the general branch and bound algorithm for continuous problems may be outlined in the following steps:

**Step 1** Initialization: Cover the feasible region $\mathbf{A}$: $\mathbf{L} \leftarrow \{\mathbf{L}_j | \mathbf{A} \subseteq \bigcup \mathbf{L}_j\}$ using *covering rule*.

**Step 2** Repeat remaining **Steps 3–7** while there are candidate subregions ($\mathbf{L} \neq \emptyset$):

**Step 3** Choose subregion $\mathbf{D} \in \mathbf{L}$ using *selection rule*, $\mathbf{L} \leftarrow \mathbf{L} \setminus \mathbf{D}$.
**Step 4** Subdivide $\mathbf{D}$ into $p$ subregions $\mathbf{D}_j$ using *branching rule*.
**Step 5** Compute bounds over $\mathbf{D}_j$ using *bounding rules*.
**Step 6** Update the approximation of the Pareto set taking into account sample points where the objective functions were evaluated when computing bounds.
**Step 7** If the bounding front or bounding vector over $\mathbf{D}_j$ is not dominated by solutions from the current approximation of the Pareto set, add the subregion into the candidate list $\mathbf{L} \leftarrow \{\mathbf{L}, \mathbf{D}_j\}$.

## 5.2 Branch and Bound for Combinatorial Optimization Problems

In this book we build branch and bound algorithms for combinatorial optimization problems following a scheme similar to presented in [24]. The main components of such algorithms are *branching*, *bounding*, *pruning*, and *retracting*.

The *branching* corresponds to the creation of one or more candidate subsets from existing one. For combinatorial problems we use the depth first search strategy to avoid storing of the candidate subsets and save memory. Instead of storing candidate subsets the *branching* and *retracting* is used to change from one candidate to another.

In combinatorial optimization a set of decisions (subset of feasible decisions) may be represented by a partial (incomplete) decision where some objects are not yet assigned. This is in contrast to a complete decision where all objects are assigned and for which the values of the objective functions may be evaluated. During the process of branching, decisions are gradually constructed by assigning the objects and eventually completing the decision.

The *branching* is implemented as assigning a new object (branch forward) or changing the assignment of the last object to not yet explored alternative (branch right). The *retracting* corresponds to unassigning of objects for which no alternative unexplored assignment is possible.

If it is possible to determine that a partial decision cannot lead to an optimal decision, it can be eliminated from the further search together with all partial and complete decisions which could be composed from it by assigning some other objects during the search. This elimination corresponds to the *pruning* of the branch of the search tree.

The *bounding* is used to determine if *pruning* is possible. While the objective functions are evaluated for complete decisions, bounds for the objective functions are needed to determine if the set of decisions represented by a partial decision may be eliminated. The bounds for the objective functions include the direct contribution from the partial decision itself to the values of the objective functions and the most favorable contribution to them from completing the partial decision by assigning further objects. The direct contribution can be imagined as evaluation of

the objective function taking into account only already assigned objects. Probably the simplest bounding vector may be composed of such direct contributions for all objective functions. Better bounding vectors may be composed from bounds for every objective function composed from the direct contribution from already assigned objects as well as the favorable contribution from not yet assigned objects.

In multi-objective optimization a favorable contribution to one of the objectives may impose a less favorable contribution to other objectives. By taking this into account, it is possible to build better bounding fronts than a bounding vector composed of independent bounds for objectives. An example of such a bounding front will be presented in Section 10.3.

The general branch and bound algorithm for combinatorial optimization problems may be outlined in the following steps:

**Step 1** *Initialization*: Start with the first possible complete decision. Set the number of assigned objects $n'$ to the total number of objects $n$: $n' \leftarrow n$.

**Step 2** Repeat the remaining steps while $n' > 0$:

**Step 3** If the current decision is complete ($n' = n$)

- *Evaluation of complete decision:*
  - Compute objective functions.
  - If no solutions in the current approximation of the Pareto set dominate the current solution, add it to the approximation.
  - If there are solutions in the current approximation of the Pareto set dominated by the current solution, remove them from the approximation.
- *Retract* if needed and *branch right*:
  - *Retract* if needed: while there are no new possible assignment for $n'$th object, $n' \leftarrow n' - 1$.
  - *Branch right*: assign $n'$th object by the next possible assignment.

**Step 4** Otherwise

- *Bounding*: Compute bounds.
- If bounding front or bounding vector is dominated by solutions from the current approximation of the Pareto set, *prune*:
  - *Retract* if needed: while there are no new possible assignment for $n'$th object, $n' \leftarrow n' - 1$.
  - *Branch right*: assign $n'$th object by the next possible assignment.
- Otherwise *branch forward*:
  - Increase $n'$: $n' \leftarrow n' + 1$.
  - Assign $n'$th object by the first possible assignment.

As a particular algorithm let us discuss a bi-objective branch and bound algorithm for cell formation problem [264]. The problem is described in Section 1.3. We will represent a subset of feasible decisions of the bi-criterion cell formation problem as

a partial decision where only some ($m' < m$) first machines are considered. In this case, a partial decision is represented by the $m' \times k$ cell membership matrix $\mathbf{X}'$.

Instead of operating with zero-one matrix $\mathbf{X}$ we will operate with the integer $m$-element vector $\mathbf{c}$ defining labels of cells to which machines are assigned. The vector $(1, 1, 2, 3, 3, 4, \dots)$ means that the first and the second machines are assigned to the first cell, the third machine is assigned to the second cell, and the fourth machine is assigned to the third cell. The matrix $\mathbf{X}$ can be easily built from $\mathbf{c}$:

$$x_{ij} = \begin{cases} 1, \ c_i = j, \\ 0, \ \text{otherwise}, \end{cases} \quad i = 1, 2, \dots, m, \ j = 1, 2, \dots, k.$$

In this case, $\mathbf{X}\mathbf{e}_k = \mathbf{e}_m$ and $\mathbf{X} \in \{0, 1\}^{m \times k}$ hold for any $\mathbf{c} \in \{1, \dots, k\}^m$.

In order to avoid equivalent decisions [245] some restrictions are set:

$$\min_{\mathbf{c}_i = j} i < \min_{\mathbf{c}_i = j+1} i.$$

Such restrictions correspond to the arrangement of $\mathbf{X}$ so that

$$\min_{x_{ij}=1} i < \min_{x_{il}=1} i \ \leftrightarrow \ j < l.$$

Taking into account such restrictions a search tree of the problem with $m = 4$ is shown in Figure 5.4. Only numerals are shown to save space. At each level of the tree the number of nodes corresponds to the number of partitions of a set (Bell number) of the corresponding number of machines. The minimal ($L$) and maximal ($U$) number of machines in each cell or the maximal number of cells ($K$) may be restricted.

The bounds for objective functions (1.7)–(1.8) may be computed as

$$b_1(\mathbf{X}') = \mathbf{p}^T(\Phi(\mathbf{W}'^T\mathbf{X}')\mathbf{e}_k - \Phi(\Phi(\mathbf{W}'^T\mathbf{X}')\mathbf{e}_k)), \tag{5.1}$$

$$b_2(\mathbf{X}') = \langle \mathbf{W}' - \mathbf{M}(\mathbf{X}'), \mathbf{W}' - \mathbf{M}(\mathbf{X}') \rangle, \tag{5.2}$$

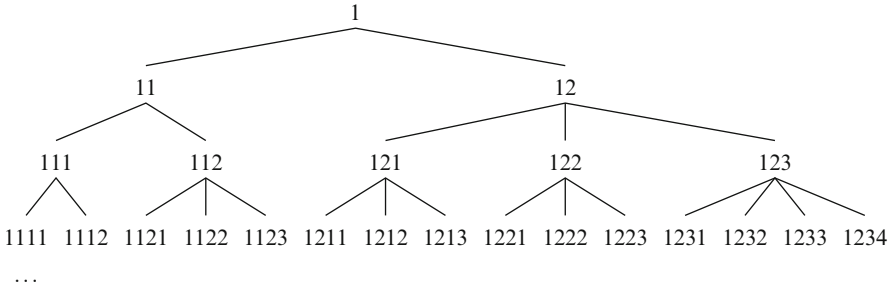


**Fig. 5.4** An example of the search tree for the cell formation problem

where $\mathbf{W}'$ denotes the matrix composed of $m'$ first rows of matrix $\mathbf{W}$. Assignment of other machines to cells later on during the search process cannot reduce intercell moves since already assigned machines will be not changed and the newly assigned machines can only introduce new intercell moves. The cell load variation will remain the same if the other machines are assigned to new separate cells and cannot be decreased after assignment of other machines.

The branch and bound algorithm for the bi-criterion cell formation problem can be outlined in the following steps:

**Step 1** *Initialization*: Start with the first possible assignment in $\mathbf{c}$:

- If there are no restrictions on the maximal number of machines in the cell $U$, $\mathbf{c} = (1, 1, \ldots, 1)$,
- otherwise $c_i = \left\lceil \frac{i}{U} \right\rceil$, $i = 1, 2, \ldots, m$, where $\lceil \cdot \rceil$ defines rounding upwards.

$m' \leftarrow m$

**Step 2** Repeat the remaining steps:

**Step 3** If the current decision is complete ($m' = m$)

*Evaluation of complete decision*: Compute objective functions and update the decision list.

- Find $\mathbf{X}$ corresponding to $\mathbf{c}$:

$$x_{ij} = \begin{cases} 1, \ c_i = j, \\ 0, \ \text{otherwise.} \end{cases} \quad i = 1, 2, \ldots, m, \ j = 1, 2, \ldots, k.$$

- Compute $f_1(\mathbf{X})$ and $f_2(\mathbf{X})$ according to (1.7)–(1.8)
- If no decisions in the decision list dominate the current decision, then add it to the decision list:

$$\text{If } \nexists \mathbf{a} \in S : \mathbf{a} \prec \mathbf{X}, \ \text{then } S \leftarrow S \cup \{\mathbf{X}\}$$

- If there are decisions in the decision list dominated by the current decision, then remove them from the decision list:

$$S \leftarrow S \setminus \{\mathbf{a} \in S : \mathbf{X} \prec \mathbf{a}\}.$$

- Do retracting in **Step 5**.

**Step 4** Otherwise

*Bounding*: Compute bounds

- Find $\mathbf{X}'$ corresponding to $\mathbf{c}$:

$$x'_{ij} = \begin{cases} 1, \ c_i = j, \\ 0, \ \text{otherwise,} \end{cases} \quad i = 1, 2, \ldots, m', \ j = 1, 2, \ldots, k.$$

- Compute bounding vector **b** from $b_1(\mathbf{X}')$ and $b_2(\mathbf{X}')$ computed according to (5.1)–(5.2).
- If no decisions in the decision list dominate **b** of the current set of decisions represented by the current partial decision, then do branching forward in **Step 5**.
- Otherwise do retracting in **Step 5**.

**Step 5** *Branching*:

- If branching forward, then append 1 to the tail of **c** ($c_{m'} = 1$), increase $m'$.
- If retracting, then remove from the tail of **c** all numbers appearing only once in **c** or not smaller than the maximal number of cells ($K$). If **c** is empty, finish; otherwise set $m'$ accordingly and increase the last remaining number $c_{m'}$.
- Check restrictions on **c**. If the number of machines in a cell violates the minimal ($L$) or maximal ($U$) numbers, then repeat retracting.

To illustrate the branch and bound algorithm, let us investigate the results of it on example problem with data given in Table 1.1. In the worst-case situation a branch and bound algorithm could lead to a complete enumeration of all feasible decisions. In a complete enumeration all possible labels in **c** satisfying the described restrictions are generated and non-dominated decisions are retained as the Pareto set. Let us look how far the practical case is from the worst one. It is also interesting how the solution time depends on the number of cells.

The computational time and the number of functions evaluations for the branch and bound algorithm (B&B) and complete enumeration (CE) with various maximal number of cells are presented in Table 5.1. In the case of the branch and bound

**Table 5.1** Experimental comparison of the branch and bound algorithm and complete enumeration

| | B&B | | CE | | Speed-up | |
|---|---|---|---|---|---|---|
| $K$ | $t$, s | NFE | $t$, s | NFE | $t$ | NFE |
| 2 | 0 | 3197 | 0.05 | 16,384 | | 5 |
| 3 | 0.02 | 4077 | 7.41 | 2,391,485 | 371 | 587 |
| 4 | 0.02 | 12,108 | 148.63 | 44,747,435 | 7432 | 3696 |
| 5 | 0.08 | 29,738 | 894.31 | 255,514,355 | 11,179 | 8592 |
| 6 | 0.21 | 57,544 | 2497 | 676,207,628 | 11,890 | 11,751 |
| 7 | 0.32 | 90,280 | 4643 | 1,084,948,961 | 14,509 | 12,018 |
| 8 | 0.44 | 119,939 | 5082 | 1,301,576,801 | 11,550 | 10,852 |
| 9 | 0.53 | 140,514 | 5382 | 1,368,705,291 | 10,155 | 9741 |
| 10 | 0.61 | 150,819 | 5442 | 1,381,367,941 | 8921 | 9159 |
| 11 | 0.61 | 154,462 | 6066 | 1,382,847,419 | 9944 | 8953 |
| 12 | 0.59 | 155,485 | 5457 | 1,382,953,889 | 9249 | 8894 |
| 13 | 0.68 | 155,692 | 5459 | 1,382,958,439 | 8028 | 8883 |
| 14 | 0.64 | 155,717 | 6055 | 1,382,958,544 | 9461 | 8881 |
| | 0.67 | 155,718 | 6134 | 1,382,958,545 | 9155 | 8881 |

algorithm the number of functions evaluations includes evaluations of bounds which are computed similarly to the objective functions. Simultaneous evaluation of $f_1(\mathbf{X})$ and $f_2(\mathbf{X})$ counts as one evaluation.

The results show that the computational time and the number of function evaluations grow very fast for the complete enumeration. If more than six cells are allowed, the enumerations last more than 1 h. In the case of the branch and bound algorithm the computational time and NFE grow much slower and flatten after nine cells. We see that the branch and bound algorithm is faster than the complete enumeration by approximately four orders of magnitude for the problems with more than four cells. As expected, both the branch and bound algorithm and the complete enumeration find the same non-dominated decision vectors. The number of function evaluations for the complete enumeration of the whole problem is the number of partitions of a set (Bell number) of 15 machines. An important result seen in Table 5.1 is that the branch and bound algorithm solves the complete problem instance with up to $m$ cells in less than 1 s. This is an acceptable time for such problem instances.

The Pareto front of the problem instance is illustrated in Figure 5.5. In this figure we also show the solutions found by the multi-objective GP-SLCA Algorithm [46] (these solutions are shown as plus symbol) and the solution found by the genetic algorithm for the cell formation problem [223] (this solution is shown as cross symbol). Only the non-dominated solution with the objective vector (918, 8.596) has been identified in [223]. We see that the multi-objective GP-SLCA identified the extreme solutions: when all machines are assigned to a single cell and when each machine composes a separate cell, as well as Pareto optimal solutions with two and three cells. The latter solutions can also be identified by a complete enumeration in relatively short time (see Table 5.1). Unfortunately, other four solutions provided as

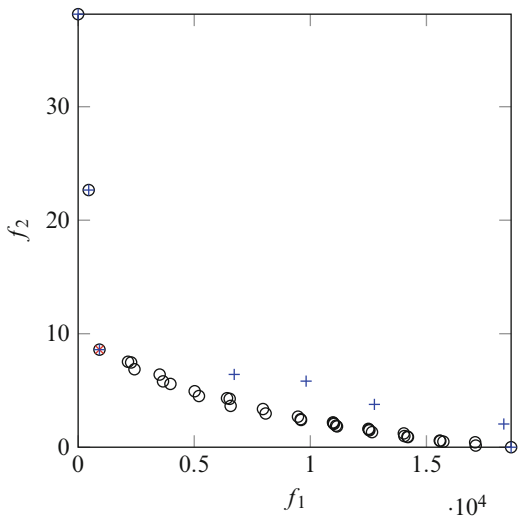


**Fig. 5.5** Pareto front for Problem 3 found by the branch and bound algorithm (*open circle*) and solutions given in literature: *plus symbol* [46] and *cross symbol* [223]

a result of the multi-objective GP-SLCA are not Pareto optimal as we can see in the plot (see Figure 5.5). It is not surprising since the multi-objective GP-SLCA is a heuristic approach. However, these solutions are quite far from the actual Pareto front. It can also be seen that the spread of the actual Pareto front is also better than that of approximation found by multi-objective GP-SLCA. Moreover, the approximation found by GP-SLCA may give a wrong impression that the Pareto front has a sharp turn at (918, 8.596) what can make the decision maker to choose the corresponding decision vector. We can see that the turn of the actual Pareto front is not that sharp and other Pareto optimal solutions can also be considered. The Pareto optimal decision with objective vector (3974, 5.576) is illustrated in Section 1.3 as Figure 1.4.

# Chapter 6
# Worst-Case Optimal Algorithms

## 6.1 Optimal Algorithms for Lipschitz Functions

### *6.1.1 Introduction*

The question of the possibility to construct an optimal (in some sense) algorithm is of importance to the theory of multi-objective optimization similarly to any other theory of algorithmically solvable problems. In this chapter, we aim at finding a worst-case optimal approximation of the Pareto optimal set for multi-objective optimization problems, where the convexity of objective functions is not assumed. The class of Lipschitz functions is chosen as a model of objective functions since that model is one of the simplest and best researched models of global optimization [87]. Worst-case optimal algorithms are constructed for the cases of passive (non-adaptive) and sequential (adaptive) search in [249]. These results are the generalization to the multi-objective case of the results by Sukharev who investigated the worst-case optimal single-objective optimization algorithms in [210, 211]. It is shown that the worst-case optimal passive algorithm, with the fixed in advance number of computations of values of objective functions $n$, can be reduced to the computation of centers of $n$ balls producing the optimal cover of a feasible region, where the balls are of equal minimum radius. It is also shown that in the worst-case, adaptivity does not improve the guaranteed accuracy achievable by the passive algorithm.

In the case when either the number of the computations of objectives is not fixed in advance or the estimates of the Lipschitz constants can be changing during the search, the concept of the one-step optimality can be attractive in the development of respective algorithms. Several single-objective optimization algorithms were developed in accordance with this concept. Below a bi-objective optimization problem with univariate non-convex objectives is considered, assuming the objective functions are Lipschitz continuous. An algorithm is presented which is based on the concept of one-step worst-case optimality. Some numerical examples are included



P.M. Pardalos et al., *Non-Convex Multi-Objective Optimization*, Springer
Optimization and Its Applications 123, DOI 10.1007/978-3-319-61007-8_6

to illustrate the performance of the algorithm and to compare its performance with that of one-step optimal algorithm based on the concept of rational decisions under uncertainty.

### 6.1.2 Mathematical Model

A class of Lipschitz objective functions is considered, i.e., $f_k(\mathbf{x}) \in F(L_k)$ iff

$$|f_k(\mathbf{x}) - f_k(\mathbf{z})| \le L_k \cdot \|\mathbf{x} - \mathbf{z}\|_D, \tag{6.1}$$

where $\mathbf{x} \in \mathbf{A}$, $\mathbf{z} \in \mathbf{A}$, $\mathbf{L} = (L_1, \ldots, L_m)^T$, $L_k > 0$, $k = 1, \ldots, m$, and $\|\cdot\|_D$ is the Euclidean norm in the decision space $\mathbb{R}^d$ where the subscript $D$ indicates the decision space (the alternative subscript $O$ indicates the space of objectives/solutions). The feasible region $\mathbf{A}$ is supposed to be compact. To simplify the notation below, let us change the scales of function values so that

$$L_k = 1/\sqrt{m},\ k = 1, \ldots, m. \tag{6.2}$$

An algorithm consists of two sub-algorithms. The first sub-algorithm selects the points $\mathbf{x}_i \in \mathbf{A}$, and computes $\mathbf{f}(\mathbf{x}_i)$, $i = 1, \ldots, n$. The second sub-algorithm computes an approximation of $\mathbf{P}(\mathbf{f})_O$ using the output of the first sub-algorithm. The number of points $n$ is supposed to be chosen in advance. Such an assumption seems reasonable in the case of expensive objectives where the lengthy computation of a single value of the vector of objectives essentially restricts the permissible value of $n$. To exploit rationally an essentially restricted resource it is important to know its limit in advance. For example, in some heuristic single-objective global optimization algorithms some percentage of the admissible number of computations $n$, which is defined in advance, is assigned for global search, and the remaining computations are supposed for a local improvement of the found candidate solution.

Two versions of the first sub-algorithm are considered. The passive (non-adaptive) algorithm selects $\mathbf{x}_i$, $i = 1, \ldots, n$, only taking into account information on the problem, i.e., on the class of objective functions and on $(n, \mathbf{A})$. The sequential (adaptive) algorithm consists of the algorithmic steps which compute $\mathbf{x}_i$ taking into account the points and function values computed at the previous steps:

$$\begin{aligned}
\mathbf{x}_1 &= \mathbf{x}_1(n, \mathbf{A}), \\
\mathbf{y}_i &= \mathbf{f}(\mathbf{x}_i), \\
\mathbf{x}_i &= \mathbf{x}_i(n, \mathbf{A}, \mathbf{x}_j, \mathbf{y}_j, j = 1, \ldots, i-1),\ i = 2, \ldots, n.
\end{aligned}$$

The output of the first sub-algorithm is $(\mathbf{X}_n, \mathbf{Y}_n)$, where $\mathbf{X}_n = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, and $\mathbf{Y}_n = (\mathbf{y}_1, \ldots, \mathbf{y}_n)$. The second sub-algorithm computes an approximation of $\mathbf{P}(\mathbf{f})_O$ using $(\mathbf{X}_n, \mathbf{Y}_n)$ as an input.

An approximation of $\mathbf{P}(\mathbf{f})_O$ is defined by its lower and upper bounds. A lower bound is defined as the subset of non-dominated vectors in the set of realizable solutions with respect to $(\mathbf{X}_n, \mathbf{Y}_n)$, i.e., of $\mathbf{y} \in \mathbb{R}^m$ such that there exist $\mathbf{x} \in \mathbf{A}$ and a function $\mathbf{f}(\cdot)$ (in the considered set of functions), which satisfy $\mathbf{y}_i = \mathbf{f}(\mathbf{x}_i)$, $i = 1, \ldots, n$ and $\mathbf{y} = \mathbf{f}(\mathbf{x})$. For the class of Lipschitz functions $F(L_k)$, (see (6.1), (6.2)), the lower bound $\check{\mathbf{P}}(\mathbf{f})_O$ is defined as

$$\check{\mathbf{P}}(\mathbf{f})_O = \bigcup_{i \in \mathbf{I}(\mathbf{Y}_n)} \left\{ \mathbf{z} : \ \|\mathbf{z} - \mathbf{y}_i\|_O = r(\mathbf{X}_n),\ z_j \le y_{ij},\ j = 1, \ldots, m \right\}, \tag{6.3}$$

where $\mathbf{I}(\mathbf{Y}_n)$ is the set of indices of non-dominated elements of $\{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$,

$$r(\mathbf{X}_n) = \max_{\mathbf{x}} \in \mathbf{A} \ \min_{1 \le i \le n} \ \|\mathbf{x} - \mathbf{x}_i\|_D. \tag{6.4}$$

The upper bound for $\mathbf{P}(\mathbf{f})_O$, denoted by $\hat{\mathbf{P}}(\mathbf{f})_O$ is the south-west boundary of the set

$$\mathbf{D}(\mathbf{Y}_n) = \bigcup_{i=1}^{n} \{\mathbf{y} : \ \mathbf{y} \in \mathbb{R}^m,\ \mathbf{y} \ge \mathbf{f}(\mathbf{x}_i)\}, \tag{6.5}$$

where $\mathbf{y} \ge \mathbf{z}$ is valid iff $y_i \ge z_i$, $i = 1, \ldots, m$; $\mathbf{D}(\mathbf{Y}_n)$ is the subset of $\mathbb{R}^m$, whose each point is dominated by at least one solution belonging to $\mathbf{Y}_n$.

The set

$$\tilde{\mathbf{Y}}_n = \{\mathbf{y}_i : \ \mathbf{y}_i \in \mathbf{Y}_n,\ i = 1, \ldots, n,\ \mathbf{S}(\mathbf{y}_i, r(\mathbf{X}_n)) \backslash \mathbf{D}(\mathbf{Y}_n) \ne \emptyset\}, \tag{6.6}$$

is called the support of bounds; here $\mathbf{S}(\mathbf{y}, r)$ denotes a ball with the center $\mathbf{y}$ and radius $r$.

The distance between $\check{\mathbf{P}}(\mathbf{f})_O$ and $\hat{\mathbf{P}}(\mathbf{f})_O$

$$\Delta(\mathbf{f}) = \max_{\mathbf{y} \in \check{\mathbf{P}}(\mathbf{f})_O} \ \min_{\mathbf{z} \in \hat{\mathbf{P}}(\mathbf{f})_O} \|\mathbf{y} - \mathbf{z}\|_O, \tag{6.7}$$

is an upper bound for the approximation error of $\mathbf{P}(\mathbf{f})_O$. The assessment of the approximation error by (6.7) is inspired by the following metric, frequently used in the testing of multi-objective optimization algorithms

$$\delta(\mathbf{f}) = \max_{\mathbf{y} \in \mathbf{P}(\mathbf{f})_O} \ \min_{\tilde{\mathbf{y}} \in \tilde{\mathbf{P}}(\mathbf{f})_O} \|\mathbf{y} - \tilde{\mathbf{y}}\|_O, \tag{6.8}$$

where $\tilde{\mathbf{P}}(\mathbf{f})_O$ denotes the approximation for $\mathbf{P}(\mathbf{f})_O$ found by any considered algorithm.

Various performance metrics are used in experimental testing of multi-objective optimization algorithms; see, e.g., [42, 60, 183, 267]. Their appropriateness depends

on the properties of the problems considered. We have chosen the metric (6.8) since it corresponds well to the quality of approximation of $\mathbf{P}(\mathbf{f})_O$ by $\tilde{\mathbf{P}}(\mathbf{f})_O$. A serious disadvantage of (6.8), in terms of applications, is caused by the involvement of $\mathbf{P}(\mathbf{f})_O$ into the computations of this metric. In an experimental investigation, $\mathbf{P}(\mathbf{f})_O$ usually is replaced by a high precision approximation which should be obtained by an exhaustive a priori experimentation, using various algorithms of multi-objective optimization. The approximation error (6.7) is a generalization of $\delta(\mathbf{f})$, and the upper bound of (6.7) can be assessed through the fundamental property of the considered class of objective functions defined by (6.1).

**Lemma 6.1** *The approximation defined by* $\check{\mathbf{P}}(\mathbf{f})_O$ *and* $\hat{\mathbf{P}}(\mathbf{f})_O$ *is worst-case optimal for the sub-class of Lipschitz functions with the Lipschitz constants defined by (6.2), and the values of which are computed at points* $\mathbf{x}_i$, $1 \le i \le n$. *The minimum worst-case error is equal to* $r(\mathbf{X}_n)$.

*Proof* According to (6.4) a solution $\mathbf{y} \in \mathbf{P}(\mathbf{f})_O$ is in the ball whose center is a solution $\mathbf{f}(\mathbf{x}_i) \in \check{\mathbf{Y}}_n$ closest to $\mathbf{y}$. The center of the ball either belongs to $\hat{\mathbf{P}}(\mathbf{f})_O$ or is an interior point of $\mathbf{D}(\mathbf{Y}_n)$. Therefore, the inequality

$$\min_{\mathbf{z} \in \hat{\mathbf{P}}(\mathbf{f})_O} \|\mathbf{y} - \mathbf{z}\|_O \le \Delta(\mathbf{f}) = r(\mathbf{X}_n), \tag{6.9}$$

is obviously valid.

Next we will show that in the most unfavorable case, inequality (6.9) becomes an equality. Indeed, let us denote by $j$ an index for which the equality

$$\begin{aligned} \|\tilde{\mathbf{x}} - \mathbf{x}_j\|_D &= r(\mathbf{X}_n) \\ \tilde{\mathbf{x}} = \arg \max_{\mathbf{x} \in \mathbf{A}} \min_{1 \le i \le n} &\|\mathbf{x} - \mathbf{x}_i\|_D, \end{aligned}$$

holds.

Let the function $\mathbf{g}(\mathbf{x})$ be defined by the formula

$$g_k(\mathbf{x}) = \max_{1 \le i \le n} \left( c_k - \frac{1}{\sqrt{m}} \|\mathbf{x} - \mathbf{x}_i\|_D \right), \; k = 1, \ldots, m. \tag{6.10}$$

In such a case the set $\mathbf{Y}_n$ consists of the identical vectors $\mathbf{c} = (c_1, \ldots, c_m)^T$, and $\check{\mathbf{P}}(\mathbf{f})_O$ is a part of the sphere defined by the following formula:

$$\check{\mathbf{P}}(\mathbf{f})_O = \{\mathbf{y} : \; \mathbf{y} \in \mathbb{R}^m, \; \|\mathbf{y} - \mathbf{c}\|_O = r(\mathbf{X}_n), \; y_i - c_i \le 0\}.$$

Since

$$\|\mathbf{g}(\tilde{\mathbf{x}}) - \mathbf{c}\|_O = \|\mathbf{x}_j - \tilde{\mathbf{x}}\|_D = r(\mathbf{X}_n),$$

then the inclusion $\mathbf{g}(\tilde{\mathbf{x}}) \in \check{\mathbf{P}}(\mathbf{f})_O$ is true. The functions $\mathbf{g}(\mathbf{x})$, defined by (6.10) with different vectors $\mathbf{c}$, constitute a subset of $F(L_k)$, and, assuming $\mathbf{X}_n$ given, they are the worst-case functions with respect to the Pareto set approximation.

### 6.1.3 Optimal Passive Algorithm

Assume that a passive algorithm has computed $\mathbf{X}_n = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ at the first stage. The worst-case error, in this case, is equal to $r(\mathbf{X}_n)$, as shown in Lemma 6.1. We are interested in the selection of points $\mathbf{x}_1, \ldots, \mathbf{x}_n$, minimizing $r(\mathbf{X}_n)$, and in the worst-case functions.

**Theorem 6.1** *The worst-case optimal multi-objective optimization algorithm selects the points $\mathbf{x}_1, \ldots, \mathbf{x}_n$, at the centers of n balls of minimal radius which cover the feasible region $\mathbf{A}$. The minimum worst-case error is equal to the radius r of the balls of the optimal cover. The worst-case objective functions $\boldsymbol{\varphi}(\cdot)$ and $\mathbf{g}(\cdot)$ are defined by the following formulae:*

$$\begin{aligned} \varphi_k(\mathbf{x}) &= c_k,\ \mathbf{x} \in \mathbf{A},\ k = 1, \ldots, m, \\ g_k(\mathbf{x}) &= \max_{1 \le i \le n} c_k - (\|\mathbf{x} - \mathbf{x}_i\|_D),\ \mathbf{x} \in \mathbf{A},\ k = 1, \ldots, m, \end{aligned} \tag{6.11}$$

*where $c_k$, $k = 1, \ldots, m$, are arbitrary constants.*

*Proof* The problem of minimization of $r(\mathbf{X}_n)$

$$r = \min_{\mathbf{x}_1, \ldots, \mathbf{x}_n,\ \mathbf{x}_i \in \mathbf{A}} r(\mathbf{X}_n) = \min_{\mathbf{x}_1, \ldots, \mathbf{x}_n,\ \mathbf{x}_i \in \mathbf{A}} \max_{\mathbf{x} \in \mathbf{A}} \min_{1 \le i \le n} \|\mathbf{x} - \mathbf{x}_i\|_D,$$

is equivalent to the problem of covering a feasible region with equal balls of the minimal radius. Therefore, the centers of balls of optimal covering define the points, where the values of the objective functions should be computed.

Repeating the arguments of Lemma 6.1 it is easy to check that

$$\max_{\mathbf{y} \in \mathbf{P}(\boldsymbol{\varphi})_O} \min_{\mathbf{z} \in \check{\mathbf{P}}(\boldsymbol{\varphi})_O} \|\mathbf{y} - \mathbf{z}\|_O = r, \tag{6.12}$$

$$\max_{\mathbf{y} \in \mathbf{P}(\mathbf{g})_O} \min_{\mathbf{z} \in \hat{\mathbf{P}}(\mathbf{g})_O} \|\mathbf{y} - \mathbf{z}\|_O = r. \tag{6.13}$$

### 6.1.4 Optimal Sequential Algorithm

A user of optimization algorithms is inclined, as a rule, to believe that a passive algorithm will always be outperformed by an adaptive algorithm. Although it is normally true in applications, the theory of optimal algorithms shows that generally

it is not true in the worst-case analysis [217]. As shown in [210], adaptation does not help in the worst-case Lipschitz single-objective optimization. A similar statement in the case of numerical integration is proved in [212]. Similar arguments are also valid in the case of multi-objective optimization. For example, in the case where $f_k(\mathbf{x}_i) = 0$, $i = 1, \ldots, n$, $k = 1, \ldots, m$, the selection of the points $\mathbf{x}_i \in \mathbf{A}$, $i = 1, \ldots, n$, different from the centers of balls of the optimal covering of $\mathbf{A}$, obviously implies the error larger than the radius of balls of the optimal cover $r$. Since the worst-case analysis is not always best suitable to real world problems, it seems reasonable to slightly enhance the concept of optimality. The term "the best algorithm" has been introduced in [211] to name the sequential algorithm which in worst-case generates the same set of $\mathbf{x}_i$, $i = 1, \ldots, n$, as the worst-case passive algorithm, however, performs better in other cases. If the available information on the considered problem at a current optimization step deviates from the worst-case information, that deviation is taken into account in an optimal way. The next step is performed optimally with respect to the less adversarial conditions where the worst-case conditions are relaxed by the acquired favorable information. In terms of the game theory we are interested in the strategy seeking the extra winning in the situation where an adversary deviates from his optimal strategy. The best sequential algorithms for Lipschitz optimization and integration are constructed in [211] and [212] correspondingly. However, the implementation of the best sequential algorithm of global Lipschitz optimization is very complicated, and, to our best knowledge, it has never been implemented. Nevertheless, theoretical investigation of the best sequential multi-objective optimization algorithm for the functional class (6.1) would be interesting, at least to acquire recommendations for the development of heuristics inspired by properties of the best algorithm.

### *6.1.5 Discussion*

An algorithm, optimal with respect to the error bound (6.7), is expected to spread the found non-dominated solutions not only close to the optimal Pareto set but also evenly. It is proven, however, that the worst-case optimal (with respect to the class of Lipschitz functions) passive algorithm distributes the decisions (the sites for computation the values of objective functions) uniformly in the feasible region. The set of optimal sites is coincident with the centers of the equal minimal radius balls covering the feasible region. It was unexpected since the uniformity of distribution of the non-dominated solutions normally corresponds to the non-uniform distribution of the decisions; see, e.g., [154]. However, the *worst-case* is very specific: here the Pareto optimal set consists of a single point, and the solutions are represented by a single point $\mathbf{f}(\mathbf{x}_i) = (c_1, \ldots, c_m)^T$ as well. In the *worst-case*, the adaptivity does not help in multi-objective optimization as well as in single-objective optimization, this result is in accordance with the general theory of optimal algorithms [217].

A surprising result is that the components of the *worst-case* vector objective functions (6.11) coincide up to affine scaling. Such a coincidence means that the *worst-case* model of multi-objective global optimization degrades to the single-objective model.

The Euclidean norm was used to define the considered class of objective functions as well as the approximation error. Only minor changes are needed to reformulate the obtained results in the terms of other norms.

## 6.2 One-Step Optimality for Bi-objective Problems

### *6.2.1 Introduction*

The problem of one-dimensional bi-objective optimization

$$\min_{x \in \mathbf{A}} \mathbf{f}(x), \ \mathbf{f}(x) = (f_1(x), f_2(x))^T, \tag{6.14}$$

is considered, where the feasible region is a bounded interval

$$\mathbf{A} = \{x : \ a \leq x \leq b\}.$$

Let us recall that theoretically the solution to problem (6.14) consists of two sets: $\mathbf{P}(\mathbf{f})_O$, the Pareto optimal solutions in the space of objectives (Pareto front), and $\mathbf{P}(\mathbf{f})_D$, the set of Pareto optimal decisions in $\mathbf{A}$. We are interested in the efficient computation of a discrete representation of $\mathbf{P}(\mathbf{f})_O$ for the non-convex objective functions.

The selection of a concept of the efficiency of computations should be preceded by the definition of a class of targeted problems. Since the class of Lipschitz continuous functions is one of the most widely used models for single-objective non-convex optimization [87, 168, 196] this model is also accepted to substantiate the construction of algorithms for the considered problem.

The worst-case optimality is a standard concept in the analysis of the algorithms' optimality with respect to a deterministic model of problems/data [7]. The problem of construction of the worst-case optimal algorithm for a discrete representation of Pareto optimal solution sets for the problems with Lipschitz continuous objectives is considered in Sections 6.1.3 and 6.1.4. The optimal passive algorithm, as shown there, can be reduced to covering the feasible region by the balls of the minimum radius, and the optimal adaptive (sequential) algorithm is coincident with the optimal passive algorithm. For the problems where the worst-case assumptions are relevant, the computation of objectives at the points uniformly distributed in the feasible region can be favorable. That conclusion is interesting from the theoretical point of view, however, in the majority of real world problems the worst-case is not very likely. The worst-case optimal search can be interpreted as an

antagonistic game in terms of the game theory: for the current point, selected by the search algorithm, an adversary defines the most inappropriate values of objective functions [35, 47]. The most inappropriate (non-informative) for optimizer values of the objective function are equal for all the points selected. When an ordinary optimization problem is considered, the assumption about a rational adversary, selecting the most inappropriate function values at all optimization steps, seems not very realistic. We assume now that the adversary is semi-rational, i.e., the strategy is optimal only at some optimization steps, but not during the entire optimization process. Such an assumption substantiates the investigation of a sequential one-step optimal algorithm.

Besides of the methodological arguments above, the implementation of the concept of one-step optimality is reasonable also because of applicability circumstances. This concept is suitable for the construction of the optimization algorithms in the cases where the number of computations of the values of objective functions is not fixed in advance. It is also compatible with the variation of the estimates of Lipschitz constant during the search.

Note that in the theory of single-objective Lipschitz global optimization, the one-step optimal univariate algorithm [163, 199] has been constructed and investigated a decade earlier than the worst-case optimal algorithm [210]. The algorithm, considered here, relates to the algorithm in Section 6.1.3, similarly as, in the single-objective case, the one-step optimal algorithm [163, 199] is related to the worst-case multi-step optimal algorithm, investigated in [210].

Although the direct practical applicability of the developed univariate algorithm is restricted, it can be useful in the construction of multivariate algorithms applying the ideas used in single-objective global optimization, e.g., space-filling curves, well developed in [193, 197, 208]. For a further enhancement of the efficiency of the univariate algorithm, a model of functions with Lipschitz first derivatives seems attractive [119], as well as the schema of geometric Lipschitz optimization [111, 185].

### 6.2.2 *Bounds for the Pareto Frontier*

Let us recall that a class of Lipschitz objective functions $F(L_k)$ is considered, i.e., $\mathbf{f}(x) = (f_1(x), f_2(x))^T$, and $f_k(x) \in F(L_k)$ if

$$|f_k(x) - f_k(t)| \leq L_k \cdot \|x - t\|, \ k = 1, 2, \tag{6.15}$$

for $x \in \mathbf{A}$, $t \in \mathbf{A}$, $L_k > 0$, $k = 1, 2$, and $\mathbf{A}$ is supposed to be a bounded closed interval.

The availability of relatively simply computable lower bounds enables a theoretical assessment of the quality of a discrete representation of the Pareto front for bi-objective Lipschitz optimization. For the different Pareto front approximation metrics we refer to [42, 60, 266, 267].

Let $\mathbf{Y}_n = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ be the set of objective vectors computed at the points $x_i \in \mathbf{A} = [a, b]$, $i = 1, \ldots, n$, $\mathbf{X}_n = \{x_1, \ldots, x_n\}$, i.e., $\mathbf{y}_i = \mathbf{f}(x_i)$, $i = 1, \ldots, n$.

It follows from (6.15) that the functions $g_k(x),\ k = 1, 2$, define the lower bounds for $f_k(x),\ [a \le x \le b]$:

$$g_k(x) = \max\ (y^k_{o\,i} - L_k(x - x_{o\,i}),\ y^k_{o\,i+1} - L_k(x_{o\,i+1} - x)), \qquad (6.16)$$
$$x_{o\,i} \le x \le x_{o\,i+1}, i = 1, \ldots, n-1,$$

where $x_{o\,i},\ i = 1, \ldots, n$, denote the increasingly ordered points $x_i$, and $\mathbf{y}_{o\,i} = (y^1_{o\,i}, y^2_{o\,i})^T$ denote the respective values of the objective functions.

To simplify the formulae below, let us assume that

$$L_k = 1,\ k = 1, 2. \qquad (6.17)$$

The assumption above, which can also be written in the form $f_k(x) \in F(1)$, is made without loss of generality since it can be satisfied by selecting appropriate scales of values of the objective functions.

**Definition 6.1** The Pareto front of the bi-objective problem

$$\min_{x_{o\,i} \le x \le x_{o\,i+1}} \mathbf{g}(x),\ \mathbf{g}(x) = (g_1(x), g_2(x))^T, \qquad (6.18)$$

is called a local Lipschitz lower bound for $\mathbf{P(f)}_O$, and it is denoted as $\mathbf{V}_i = \mathbf{V}(\mathbf{y}_{oj},\ x_{oj},\ j = i, i+1)$. The weekly Pareto optimal solutions of $\bigcup_{i=1}^{n-1} \mathbf{V}_i$ constitute the Lipschitz lower bound for $\mathbf{P(f)}_O$ which is denoted as $\mathbf{V}(\mathbf{Y}_n,\ \mathbf{X}_n)$.

**Definition 6.2** The subset of $\bigcup_{i=1}^{n}\ \{\mathbf{z} :\ \mathbf{z} \in \mathbb{R}^2,\ \mathbf{z} \ge \mathbf{y}_i\}$, which consists of weakly Pareto optimal solutions, is called an upper bound for $\mathbf{P(f)}_O$ and is denoted by $\mathbf{U}(\mathbf{Y}_n)$.

### 6.2.3 *Properties of Lipschitz Bounds*

To simplify the formulas below, let us consider the bounds for $\mathbf{f}(t),\ 0 \le t \le v$, and denote $y_1 = f_1(0),\ y_2 = f_1(v), z_1 = f_2(0),\ z_2 = f_2(v)$. The proved properties can be easily generalized to an arbitrary interval $x_{o\,i} \le x \le x_{o\,i+1}$ by re-scaling of variables. Assume that $\mathbf{f}(0)$ and $\mathbf{f}(v)$ do not dominate each other. The assumption of mutual non-dominance, without loss of generality, is specified as $z_2 \le z_1,\ y_1 \le y_2$. An illustration of the lower bounds for both objective functions and of $\mathbf{V}(\mathbf{f}(0), \mathbf{f}(v), 0, v)$ is presented in Figure 6.1, where the same data is used for the computations.

**Definition 6.3** The maximum distance between $\mathbf{V}(\mathbf{f}(0), \mathbf{f}(v), 0, v)$ and $\{\mathbf{f}(0), \mathbf{f}(v)\}$ is called tolerance of the lower Lipschitz bound and it is denoted by

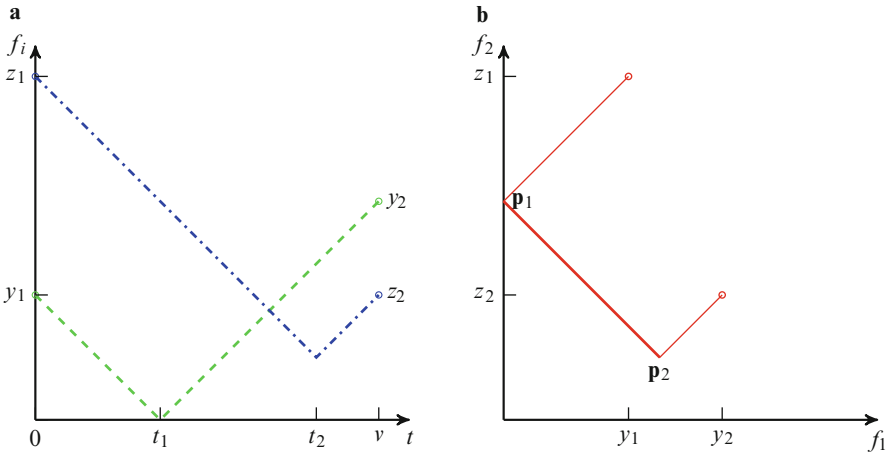


**Fig. 6.1** The lower Lipschitz bounds for: (**a**) the values of objective functions, (**b**) the Pareto set

$$\Delta(\mathbf{f}(0), \mathbf{f}(v), 0, v) =$$

$$\max\left(\min_{\mathbf{w}\in\mathbf{V}(\mathbf{f}(0),\mathbf{f}(v),0,v)} \|\mathbf{w}-\mathbf{f}(0)\|, \min_{\mathbf{w}\in\mathbf{V}(\mathbf{f}(0),\mathbf{f}(v),0,v)} \|\mathbf{w}-\mathbf{f}(v)\|\right). \tag{6.19}$$

**Lemma 6.2** *The error of approximation of* $\mathbf{P}(\mathbf{f})_O$ *by the line segment connecting* $\mathbf{f}(0)$ *and* $\mathbf{f}(v)$ *is bounded by* $\Delta(\mathbf{f}(0), \mathbf{f}(v), 0, v)$.

*Proof* The lower bounds for both objective functions in the new variables are defined by the following formulae:

$$g_1(t) = y_1 - t,\ 0 \le t \le t_1,\ g_1(t) = y_2 - (v-t),\ t_1 \le t \le v,$$
$$g_2(t) = z_1 - t,\ 0 \le t \le t_2,\ g_2(t) = z_2 - (v-t),\ t_2 \le t \le v,$$
$$t_1 = \frac{v}{2} + \frac{y_1 - y_2}{2},\ t_2 = \frac{v}{2} + \frac{z_1 - z_2}{2}. \tag{6.20}$$

The Pareto front for the problem

$$\min_{0\le t\le v} \mathbf{g}(t), \tag{6.21}$$

defines the lower Lipschitz bound for $\mathbf{P}(\mathbf{f})_O$, and is coincident with the line segment in $\mathbb{R}^2$ that connects the points $\mathbf{p}_1$ and $\mathbf{p}_2$ where

$$\mathbf{p}_1 = \begin{pmatrix} y_1 - t_1 \\ z_1 - t_1 \end{pmatrix}, \qquad \mathbf{p}_2 = \begin{pmatrix} y_2 - (v-t_2) \\ z_2 - (v-t_2) \end{pmatrix}. \tag{6.22}$$

The maximum distance between $\mathbf{V}(\mathbf{f}(0), \mathbf{f}(v), 0, v)$ and $\{(y_i, z_i)^T,\ i = 1, 2\}$ is equal to

$$\begin{aligned}\Delta(\mathbf{f}(0), \mathbf{f}(v), 0, v) = \\ = \max\left(\min_{\mathbf{w}\in\mathbf{V}(\mathbf{f}(0),\mathbf{f}(v),0,v)} \|\mathbf{w} - \mathbf{f}(0)\|,\ \min_{\mathbf{w}\in\mathbf{V}(\mathbf{f}(0),\mathbf{f}(v),0,v)} \|\mathbf{w} - \mathbf{f}(v)\|\right) \\ = \max\left(\|\mathbf{p}_1 - (y_1, z_1)^T\|,\ \|\mathbf{p}_2 - (y_2, z_2)^T\|\right). \qquad (6.23)\end{aligned}$$

Similarly as above, it is easy to show that the upper Lipschitz bound for $\mathbf{P}(\mathbf{f})_O$, which is denoted by $\tilde{\mathbf{V}}(\mathbf{f}(0), \mathbf{f}(v), 0, v)$, is a line segment parallel to the line segment representing the lower Lipschitz bound. The distance between these parallel line segments satisfies the following inequality:

$$\begin{aligned}D(\mathbf{f}(0), \mathbf{f}(v), 0, v) = \|\mathbf{p}_1 - (y_1, z_1)^T\| + \|\mathbf{p}_2 - (y_2, z_2)^T\| \\ \le 2\Delta(\mathbf{f}(0), \mathbf{f}(v), 0, v). \qquad (6.24)\end{aligned}$$

Therefore, for the points of the line segment

$$h_1(t) = y_1 + (y_2 - y_1)\cdot t,\ h_2(t) = z_1 + (z_2 - z_1)\cdot t,\ 0 \le t \le v, \qquad (6.25)$$

the following inequalities are valid

$$\min_{\mathbf{w}\in V(f(0),f(v),0,v)} \|(h_1(t), h_2(t))^T - \mathbf{w}\| \le \Delta(\mathbf{f}(0), \mathbf{f}(v), 0, v),$$

$$\min_{\mathbf{w}\in \tilde{V}(f(0),f(v),0,v)} \|(h_1(t), h_2(t))^T - \mathbf{w}\| \le \Delta(\mathbf{f}(0), \mathbf{f}(v), 0, v),$$

which imply the statement of the lemma.

**Corollary 6.1** *If* $\mathbf{f}(0)$ *and* $\mathbf{f}(v)$ *do not dominate each other, and* $\Delta(\mathbf{f}(0), \mathbf{f}(v), 0, v) = 0$*, then* $\mathbf{V}(\mathbf{f}(0), \mathbf{f}(v), 0, v)$ *and* $\tilde{\mathbf{V}}(\mathbf{f}(0), \mathbf{f}(v), 0, v)$ *are coincident with the line segment connecting the points* $\mathbf{f}(0)$*,* $\mathbf{f}(v)$*; moreover, the following equality:*

$$\mathbf{V}(\mathbf{f}(0), \mathbf{f}(v), 0, v) = \{\mathbf{y} : \mathbf{y} = \mathbf{f}(t), 0 \le t \le v\}, \qquad (6.26)$$

*holds.*

*Proof* The equality $\Delta(\mathbf{f}(0), \mathbf{f}(v), 0, v) = 0$ and inequality (6.24) imply the coincidence of $\mathbf{V}(\mathbf{f}(0), \mathbf{f}(v), 0, v)$, $\tilde{\mathbf{V}}(\mathbf{f}(0), \mathbf{f}(v), 0, v)$.

The equality $\Delta(\mathbf{f}(0), \mathbf{f}(v), 0, v) = 0$ also implies the equalities $\mathbf{p}_1 = \mathbf{f}(0)$, $\mathbf{p}_2 = \mathbf{f}(v)$, as it follows from the expression of $\Delta(\mathbf{f}(0), \mathbf{f}(v), 0, v)$ given in (6.23). Then it follows from (6.22) and (6.20) that the lower bounds for $f_i(t)$, $0 \le t \le v$, $i = 1, 2$ are defined by

$$g_1(t) = y_1 + (y_2 - y_1)\cdot t,\ g_2(t) = z_1 + (z_2 - z_1)\cdot t. \qquad (6.27)$$

It can be proved in a similar way that the upper bounds for $f_i(t)$, $0 \le t \le v$, $i = 1, 2$ are also defined by (6.27) implying that $\mathbf{f}(t)$, $0 \le t \le v$, is the line segment connecting the points $\mathbf{f}(0)$ and $\mathbf{f}(v)$.

The results of the analysis above are easily extendable for any subinterval $[x_{oi}, x_{oi+1}]$. The properties of

$$\Delta_i = \Delta(\mathbf{f}(x_{oi}), \mathbf{f}(x_{oi+1}), x_{oi}, x_{oi+1}) \text{ and } \mathbf{V}_i = \mathbf{V}(\mathbf{f}(x_{oi}), \mathbf{f}(x_{oi+1}), x_{oi}, x_{oi+1})$$

are essential in substantiating the optimization algorithm. In a special case, where $\Delta_i = 0$, the subinterval $[x_{oi}, x_{oi+1}]$ can be excluded from a further search, and $\mathbf{V}_i$ is coincident with the local ($x_{oi} \le x \le x_{oi+1}$) Pareto optimal frontier of the original multi-objective optimization problem. Generally speaking, $\mathbf{V}_i$ and $\Delta_i$, $i = 1, \ldots, n$, show the potentially reachable Pareto optimal solutions, and the quality of approximation at the current iteration.

### 6.2.4 The Implementation of One-Step Optimality

The idea of the algorithm is to tighten iteratively the Lipschitz lower bounds for the non-dominated solutions, and to indicate the subintervals of $[a, b]$ of dominated decisions which can be excluded from the further search. At the $n + 1$ iteration a subinterval $[x_{oi}, x_{oi+1}]$ is selected and subdivided by the point $x_{n+1}$ where a new objective vector $\mathbf{f}(x_{n+1})$ is computed. Let us assume that an interval is selected, and a point of subdivision should be defined. Similarly as in the previous section, the standard interval $[0, v]$ is considered to simplify the notation. The values of the objective functions at the endpoints of the interval are supposed to be known and denoted by $\mathbf{f}(0) = \mathbf{f}_1 = (y_1, z_1)^T$, $\mathbf{f}(v) = \mathbf{f}_2 = (y_2, z_2)^T$. We start the analysis from the case where $\mathbf{f}_1$ and $\mathbf{f}_2$ do not dominate each other; without loss of generality, we assume that

$$y_1 \le y_2, \; z_2 \le z_1, \tag{6.28}$$

and denote $\delta y = y_2 - y_1$, $\delta z = z_1 - z_2$.

The error of approximation of the local Pareto front is bounded by the tolerance $\Delta(\mathbf{f}_1, \mathbf{f}_2, 0, v)$. The optimal decision theory suggests to choose the point $\hat{t}$ for the current computation of the vector of objectives by minimizing the maximum of two forecasted tolerances corresponding to the subintervals, obtained by the subdivision by the chosen point. According to the worst-case paradigm, the tolerances are forecasted assuming the most unfavorable values of the objective functions at the chosen point, and the point $\hat{t}$ is defined as follows:

$$\hat{t} = \arg \min_{0 \le t \le v} \max_{\mathbf{w} \in \mathbf{W}(t)} \max \left( \Delta(\mathbf{f}_1, \mathbf{w}, 0, t), \; \Delta(\mathbf{w}, \mathbf{f}_2, t, v) \right), \tag{6.29}$$

where $\mathbf{w} = (w_1,\ w_2)^T$, and $\mathbf{W}(t)$ is a two-dimensional interval defined by the lower and upper Lipschitz bounds for the function values $\mathbf{f}(t)$.

**Theorem 6.2** *Let the inequality $\delta y \le \delta z$ be satisfied besides (6.28). Then $\hat{t} = \frac{v}{2}$ is the point of the current computation of the objective functions, since it is a minimizer of (6.29).*

*Proof* To find $\hat{t}$, the expression, supposed to minimize with respect to $t$ in (6.29), should be evaluated for various potential values of the objective functions from the intervals $\mathbf{W}(t)$ which are defined by the lower bounds (6.20), and the upper bounds for the objective functions

$$h_1(t) = y_1 + t,\ 0 \le t \le \tau_1,\ h_1(t) = y_2 + (v - t),\ \tau_1 \le t \le v,$$
$$h_2(t) = z_1 + t,\ 0 \le t \le \tau_2,\ h_2(t) = z_2 - (v - t),\ \tau_2 \le t \le v,$$
$$\tau_1 = \frac{v}{2} - \frac{y_1 - y_2}{2},\ \tau_2 = \frac{v}{2} - \frac{z_1 - z_2}{2}, \tag{6.30}$$

which are defined similarly as in (6.20).

To highlight the property essential for a further analysis of $\Delta(\cdot)$, let us substitute expressions (6.22) for $\mathbf{p}_1,\ \mathbf{p}_2$ in (6.23). The following equality is obtained:

$$\Delta(\mathbf{f}_1, \mathbf{f}_2, 0, v) = \sqrt{2}\max\{\tau_1,\ v - \tau_2\} =$$
$$\sqrt{2}\max\left\{\frac{v}{2} - \frac{\delta y}{2},\ \frac{v}{2} - \frac{\delta z}{2}\right\} = \frac{\sqrt{2}}{2}(v - \delta y), \tag{6.31}$$

where the inequality $\delta y \le \delta z$ is taken into account. Formula (6.31) can be extended to the case $\delta y \ge \delta z$ and subintervals of $[0,\ v]$ in an obvious way: $\sqrt{2}\Delta(\cdot)$ is equal to the length of the corresponding subinterval minus the smaller one out of two differences between the objective function values, computed for the endpoints of the intervals.

Both terms in (6.29) can be maximized (by the "adversary") by selecting the appropriate values of components of the vector $\mathbf{w} = (w_1, w_2)^T \in \mathbf{W}(t)$. To maximize $\Delta(\mathbf{f}_1, \mathbf{w}, 0, t)$, either the value of $w_1$ closest to $z_1$ or $w_2$ closest to $y_1$ should be chosen. Similarly, to maximize $\Delta(\mathbf{w}, \mathbf{f}_2, t, v)$, either the value of $w_1$ closest to $z_2$ or $w_2$ closest to $y_2$ should be chosen. A choice should be made taking into account the bounds (6.20) and (6.30) and trade-off between the values of $\Delta(\mathbf{f}_1, \mathbf{w}, 0, t)$ and $\Delta(\mathbf{w}, \mathbf{f}_2, t, v)$. Subsequently, we consider three different cases depending on relative values of $\delta y$ and $\delta z$.

1. Let us maximize (for an "adversary") $\Delta(\mathbf{f}_1, \mathbf{w}, 0, t)$ and $\Delta(\mathbf{w}, \mathbf{f}_2, t, v)$ with respect to $\mathbf{w} \in \mathbf{W}(t)$ assuming that $\delta z \le \frac{v}{2}$. In that case the maximizer $\hat{\mathbf{w}}(t)$, as it seen from Figure 6.2, is defined by the following equalities:

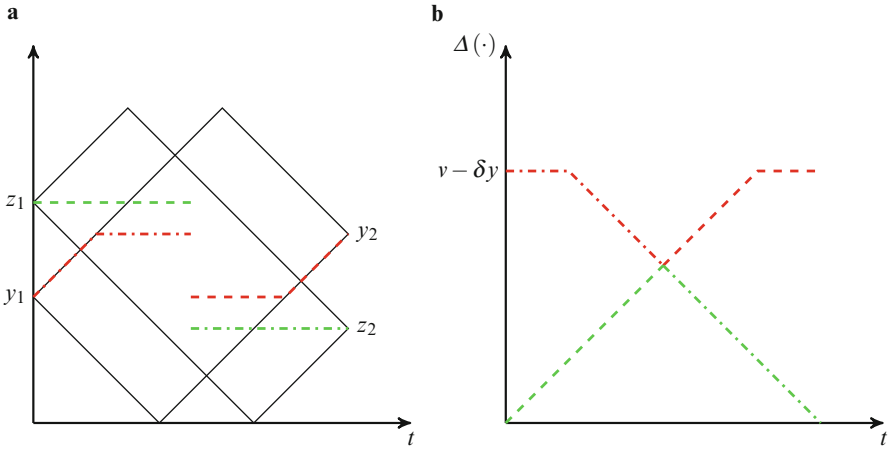


**Fig. 6.2** (**a**) The Lipschitz bounds for the case $\delta z \le \frac{v}{2}$ shown by *thin lines*, and the worst-case predicted values of objectives $\hat{w}_1(t)$ and $\hat{w}_2(t)$ shown by *green* and *red lines* correspondingly, (**b**) The graphs of $\Delta(\mathbf{f}_1, \hat{\mathbf{w}}(t), 0, t)$ and $\Delta(\hat{\mathbf{w}}(t), \mathbf{f}_2, t, v)$ are presented in *dashed* and *dot-dashed lines* correspondingly; the color indicates which of $\hat{w}_1$ and $\hat{w}_2$ is used in computation of the respective $\Delta(\cdot)$

$$\begin{cases} \hat{w}_1(t) = z_1, & 0 \le t \le \frac{v}{2}, \\ \hat{w}_1(t) = y_1, & \frac{v}{2} < t \le v - \delta y, \\ \hat{w}_1(t) = y_2 - v + t, & v - \delta y < t \le v, \end{cases} \tag{6.32}$$

$$\begin{cases} \hat{w}_2(t) = y_1 + t, & 0 \le t \le \delta y, \\ \hat{w}_2(t) = y_2, & \delta y < t \le \frac{v}{2}, \\ \hat{w}_2(t) = z_2, & \frac{v}{2} < t \le v, \end{cases} \tag{6.33}$$

and the respective maxima are equal to

$$\sqrt{2}\Delta(\mathbf{f}_1, \hat{\mathbf{w}}(t), 0, t) = \begin{cases} t, & 0 \le t \le v - \delta y, \\ v - \delta y, & v - \delta y < t \le v, \end{cases} \tag{6.34}$$

$$\sqrt{2}\Delta(\hat{\mathbf{w}}(t), \mathbf{f}_2, t, v) = \begin{cases} v - \delta y, & 0 \le t \le \delta y, \\ v - t, & \delta y < t \le v. \end{cases} \tag{6.35}$$

From (6.34) and (6.35) it follows that

$$\min_{0 \le t \le v} \max_{\mathbf{w} \in \mathbf{W}(t)} \min\{\Delta(\mathbf{f}_1, \mathbf{w}, 0, t),\ \Delta(\mathbf{w}, \mathbf{f}_2, t, v)\} = \frac{v}{2\sqrt{2}},$$

and the minimum is achieved at the point $\hat{t} = v/2$; see Figure 6.3.

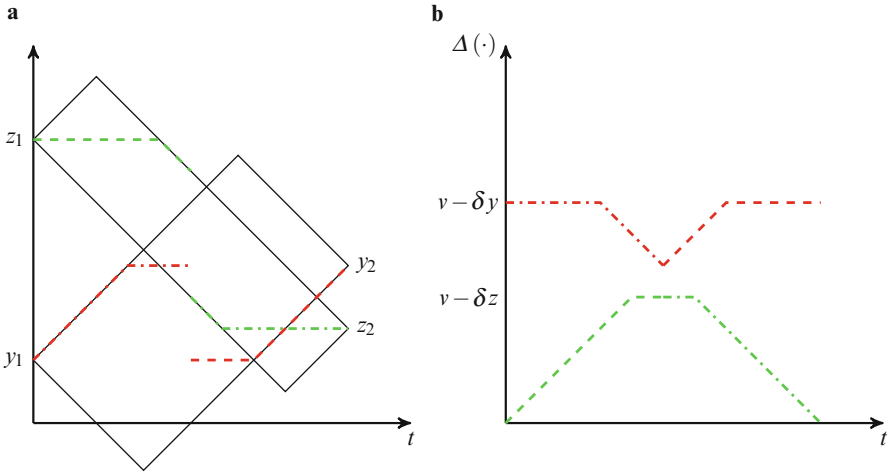


**Fig. 6.3** (**a**) The Lipschitz bounds for the case $\delta y < \frac{v}{2}, \delta z > \frac{v}{2}$ shown by *thin lines*, and the worst-case predicted values of objectives $\hat{w}_1(t)$ and $\hat{w}_2(t)$ shown by *green* and *red lines* correspondingly, (**b**) The graphs of $\Delta(\mathbf{f}_1, \hat{\mathbf{w}}(t), 0, t)$ and $\Delta(\hat{\mathbf{w}}(t), \mathbf{f}_2, t, v)$ are presented in *dashed* and *dot-dashed lines* correspondingly; the color indicates which of $\hat{w}_1$ and $\hat{w}_2$ is used in computation of the respective $\Delta(\cdot)$

2. In the case $\delta y < \frac{v}{2}, \delta z > \frac{v}{2}$ the choice (of "an adversary") of the most unfavorable $\hat{\mathbf{w}}(t)$ is similar as above, and the differences appear, as seen from Figure 6.3, only because of different relation between $\delta y$ and $\delta z$:

$$\begin{cases} \hat{w}_1(t) = z_1, & 0 \le t \le v - \delta z, \\ \hat{w}_1(t) = v + z_2 - t, & v - \delta z < t \le \frac{v}{2}, \\ \hat{w}_1(t) = \frac{y_1 + y_2}{2}, & \frac{v}{2} < t \le v - \delta y, \\ \hat{w}_1(t) = y_2 - v + t, & v - \delta y < t \le v, \end{cases} \tag{6.36}$$

$$\begin{cases} \hat{w}_2(t) = y_1 + t, & 0 \le t \le \delta y, \\ \hat{w}_2(t) = y_2, & \delta y < t \le \frac{v}{2}, \\ \hat{w}_2(t) = z_1 - t, & \frac{v}{2} < t \le \delta z, \\ \hat{w}_2(t) = z_2, & \delta z < t \le v. \end{cases} \tag{6.37}$$

The respective maxima of $\Delta(\cdot)$ are equal to

$$\sqrt{2}\Delta(\mathbf{f}_1, \hat{\mathbf{w}}(t), 0, t) = \begin{cases} t, & 0 \le t \le v - \delta z, \\ v - \delta z, & v - \delta z < t \le \frac{v}{2}, \\ t, & \frac{v}{2} < t \le v - \delta y, \\ v - \delta y, & v - \delta y < t \le v, \end{cases} \tag{6.38}$$

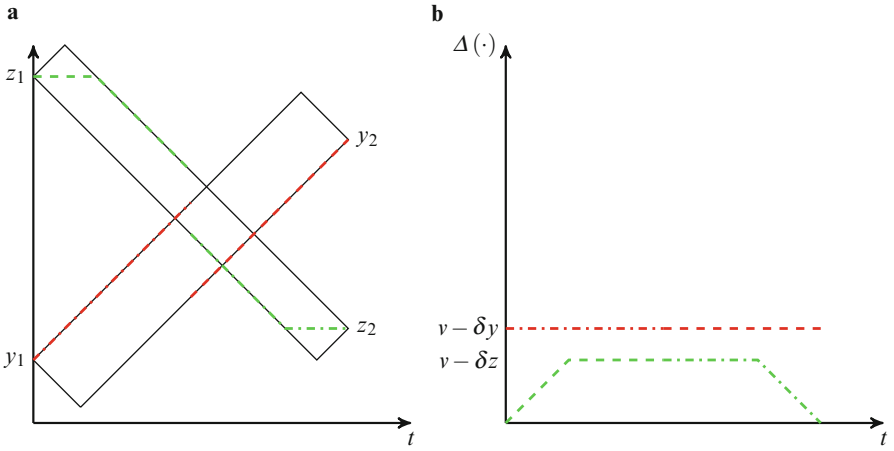


**Fig. 6.4** (**a**) The Lipschitz bounds for the case $\delta y > \frac{v}{2}$ shown by *thin lines*, and the worst-case predicted values of objectives $\hat{w}_1(t)$ and $\hat{w}_2(t)$ shown by *green* and *red lines* correspondingly, (**b**) The graphs of $\Delta(\mathbf{f}_1, \hat{\mathbf{w}}(t), 0, t)$ and $\Delta(\hat{\mathbf{w}}(t), \mathbf{f}_2, t, v)$ are presented in *dashed* and *dot-dashed lines* correspondingly; the color indicates which of $\hat{w}_1$ and $\hat{w}_2$ is used in computation of the respective $\Delta(\cdot)$

$$\sqrt{2}\Delta(\hat{\mathbf{w}}(t), \mathbf{f}_2, t, v) = \begin{cases} v - \delta y, & 0 \le t \le \delta y, \\ v - t, & \delta y < t \le \frac{v}{2}, \\ v - \delta z, & \frac{v}{2} < t \le \delta z, \\ v - t, & \delta z < t \le v. \end{cases} \tag{6.39}$$

The point of the next computation of values of the objective functions is $\tilde{t} = \frac{v}{2}$, and

$$\min_{0 \le t \le v} \max_{\mathbf{w} \in \mathbf{W}(t)} \min\{\Delta(\mathbf{f}_1, \mathbf{w}, 0, t),\ \Delta(\mathbf{w}, \mathbf{f}_2, t, v)\} = \frac{v}{2\sqrt{2}}.$$

3. Finally, the case $\delta y > \frac{v}{2}$ is considered below and illustrated in Figure 6.4:

$$\begin{cases} \hat{w}_1(t) = z_1, & 0 \le t \le v - \delta z, \\ \hat{w}_1(t) = v + z_2 - t, & v - \delta z < t \le \frac{v}{2}, \\ \hat{w}_1(t) = z_1 - t, & \frac{v}{2} < t \le \delta z, \\ \hat{w}_1(t) = z_2, & \delta z < t \le v, \end{cases} \tag{6.40}$$

$$\begin{cases} \hat{w}_2(t) = y_1 + t, & 0 \le t \le \frac{v}{2}, \\ \hat{w}_2(t) = y_2 - v + t, & \frac{v}{2} < t \le v. \end{cases} \tag{6.41}$$

The respective maxima of $\Delta(\cdot)$ are equal to

$$\sqrt{2}\Delta(\mathbf{f}_1, \hat{\mathbf{w}}(t), 0, t) = \begin{cases} t, & 0 \le t \le v - \delta z, \\ v - \delta z, & v - \delta z < t \le \frac{v}{2}, \\ v - \delta y, & \frac{v}{2} < t \le v, \end{cases} \tag{6.42}$$

$$\sqrt{2}\Delta(\hat{\mathbf{w}}(t), \mathbf{f}_2, t, v) = \begin{cases} v - \delta y, & 0 \le t \le \frac{v}{2}, \\ v - \delta z, & \frac{v}{2} < t \le \delta z, \\ v - t, & \delta z < t \le v. \end{cases} \tag{6.43}$$

The point of the next computation of values of the objective functions can be chosen arbitrarily in the interval $[0,\ v]$, and the worst-case forecasted tolerance is equal to

$$\min_{0 \le t \le v} \max_{\mathbf{w} \in \mathbf{W}(t)} \min\{\Delta(\mathbf{f}_1, \mathbf{w}, 0, t),\ \Delta(\mathbf{w}, \mathbf{f}_2, t, v)\} = \frac{v - \delta y}{\sqrt{2}}. \tag{6.44}$$

The choice of the value $\tilde{t} = \frac{v}{2}$ seems reasonable to maintain the uniformity of distribution of points for the computation of values of the objective functions; recall that the uniform distribution is worst-case optimal as shown in Section 6.1.3.

Note that $\tilde{t}$ here is slightly different from that defined in [251] where maximization in (6.29) was performed with respect to continuous $\mathbf{w}(t)$.

The statement of the theorem defines the most important part of the algorithm, i.e., the subdivision of the selected subinterval. The candidate set for the selection consists of the non-dominated subintervals where $[x_{oj},\ x_{oj+1}]$ is defined as dominated if the whole local lower Lipschitz bound of this subinterval is dominated by a solution belonging to the current $\mathbf{U}(\mathbf{Y}_n)$. The selection criterion is the tolerance $\varepsilon_j$ which for the interval with non-dominated $\mathbf{y}_{oj},\ \mathbf{y}_{oj+1}$ is defined by the Definition 6.3. If at least one of $\mathbf{y}_{oj},\ \mathbf{y}_{oj+1}$ is dominated, then $\varepsilon_j$ is defined as the maximum distance between the local lower Lipschitz bound and $\mathbf{U}(\mathbf{Y}_n)$. At the current optimization step the interval with the maximum tolerance is selected for the subdivision.

Strictly speaking, the theorem defines the subdivision in the case where both of $\mathbf{y}_{oj}$, $\mathbf{y}_{oj+1}$, corresponding to the selected interval, are non-dominated. Let us now consider the case, where at least one of the function values $\mathbf{y}_{oj},\ \mathbf{y}_{oj+1}$, is dominated. In that case the local lower Lipschitz bound (coincident with the line segment connecting the respective points $\mathbf{p}_1$ and $\mathbf{p}_2$ (6.22)) can be either non-dominated or partially dominated. In the former case the subdivision, defined by the theorem is well-founded, since implies either the (worst-case) maximum tightening of the lower Lipschitz bounds or the even most efficient indication of a dominated subinterval. The situation is slightly more complicated if the local lower Lipschitz bound, corresponding to the selected interval, is partially dominated. In that case, a dominated part of the selected interval is excluded, and the non-dominated part is subdivided as defined by the theorem replacing the unknown values of objectives by their lower bounds.

The maximum current tolerance can be used for termination the optimization algorithm when the inequality

$$\max_{1\le i\le n} \varepsilon_i \ge \varepsilon, \tag{6.45}$$

is satisfied where $\varepsilon$ is the acceptable tolerance for the approximation in question.

## The One-Step Worst-Case Optimal Algorithm

**Input:** bounds of a feasible interval $LB$, $UB$; vector of Lipschitz constants $L$; maximum number of iterations $N$; maximum allowable tolerance in estimating Pareto front $\varepsilon$;
**Initialization:** $x_1 \leftarrow LB$, $x_2 \leftarrow UB$,
$\mathbf{y}_i \leftarrow \mathbf{f}(x_i)$, $x_{oi} \leftarrow x_i$, $\mathbf{y}_{oi} \leftarrow \mathbf{f}(x_{oi})$, $i = 1, 2$,
initialize: $\mathbf{V}(\mathbf{Y}_2, \mathbf{X}_2)$, $\mathbf{U}(\mathbf{Y}_2)$, and the set of non-dominated solutions $\mathbf{P}_2$,
$n \leftarrow 2$, compute tolerances $\varepsilon_i$, $i = 1, 2$,
**while** $n < N \bigwedge \max_{1\le i\le n} \varepsilon_i > \varepsilon$
select an interval for subdivision $[x_{oj}, x_{oj+1}]$,
compute: $x_{n+1}$, $\mathbf{f}(x_{n+1})$,
compute local lower Lipschitz bounds for $[x_{oj}, x_{n+1}]$ and $[x_{n+1}, x_{oj+1}]$,
update: $\{x_i\}, \{y_i\}, \{x_{oi}\}, \{y_{oi}\}$, $\mathbf{V}(\mathbf{Y}_n, \mathbf{X}_n), \mathbf{U}(\mathbf{Y}_n)$, $\mathbf{P}_n$, $\{\varepsilon_i\}$,
$n \leftarrow n + 1$,
**end while**
**output:** $\mathbf{P}_n$.

### *6.2.5 Numerical Experiments*

The performance of the one-step worst-case optimal algorithm is demonstrated below by solving several typical test problems. The first multi-objective test considered consists of two (slightly modified) Rastrigin functions (*Rastr*) which are widely used (see, e.g., [216]) for testing single-objective global minimization algorithms:

$$\begin{aligned} f_1(x) &= (x - 0.5)^2 - \cos(18(x - 0.5)), \\ f_2(x) &= (x + 0.5)^2 - \cos(18(x + 0.5)), \ -1 \le x \le 1. \end{aligned} \tag{6.46}$$

The Lipschitz constant of both objective functions is equal to 21.

The second problem used (1.2) is referred as *Fo&Fle*. We present below its definition for a one-dimensional decision variable

$$\begin{aligned} f_1(x) &= 1 - \exp(-(x-1)^2), \\ f_2(x) &= 1 - \exp(-(x+1)^2), \\ -4 &\le x \le 4. \end{aligned} \tag{6.47}$$

The Lipschitz constant of both objective functions is equal to 1. Problem (6.47) presents a specific challenge from the point of view of global minimization. The functions $f_1(x)$ and $f_2(x)$ in (6.47) are similar to the most difficult objective function, the response surface of which is almost constant over a large part of the feasible decision region and has an unknown number of sharp spikes. The discrepancy between the model of objective functions used to substantiate the algorithm, and the actual objective functions can negatively influence the efficiency of the algorithm.

The third one-dimensional test problem by Schaffer (see [42, pp. 339–340]) is defined by the following formulas:

$$\begin{aligned} f_1(x) &= \begin{cases} -x, & \text{if } x \le 1, \\ x-2, & \text{if } 1 < x \le 3, \\ 4-x, & \text{if } 3 < x \le 4, \\ x-4, & \text{if } x > 4, \end{cases} \\ f_2(x) &= (x-5)^2, \ -1 \le x \le 8; \end{aligned} \tag{6.48}$$

to refer to this test below we use the shorthand *Schaf*. The Lipschitz constants of $f_1(x)$ and $f_2(x)$ are equal to 1 and 12, respectively. Although both objective functions of (6.48) are unimodal, $f_1(x)$ is not convex implying that the multi-objective problem is not convex. The Pareto set of (6.48) is discontinuous. The Lipschitz constants of $f_1(x)$ and $f_2(x)$ differ substantially as well as the ranges of function values. The latter property presents some challenge to the representation of the Pareto set.

The feasible objective regions of the considered test functions are shown in Figures 6.5, 6.6, and 6.7. In the figures, the Pareto front is highlighted by *thicker lines*. The selection of test problems can be summarized as follows: (6.46) is constructed generalizing typical test problems of global optimization, and two other problems, (6.47) and (6.48) are selected from a set of frequently used non-convex multi-objective test problems.

All three test problems were solved by means of the developed algorithm. The termination condition was defined as $\varepsilon_n \le 0.1$, where $\varepsilon_n$ denotes the tolerance defined in the previous section. The algorithm stopped after 90, 36, and 270 computations of the objective functions, respectively, and the numbers of found non-dominated solutions were equal to 29, 16, and 186. The relation between the number of objective function computation $N$ and the number of non-dominated solutions found $nd$, for different test problems corresponds to the structures of their sets of the Pareto optimal solutions and decisions. The length of the interval of Pareto optimal

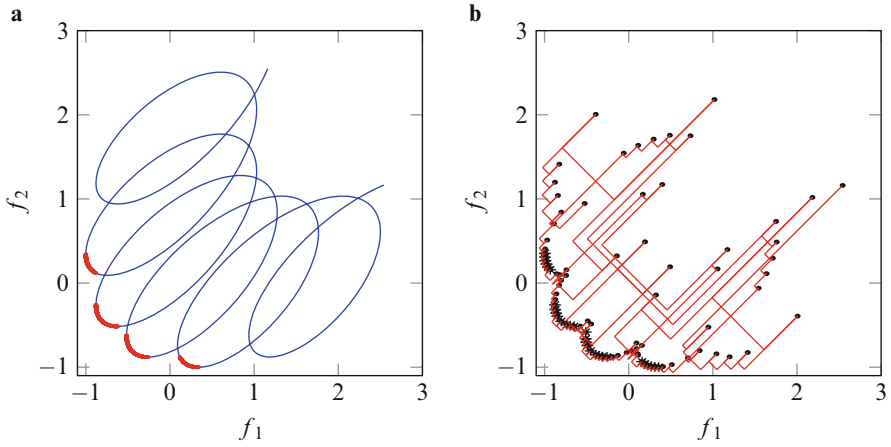


**Fig. 6.5** (**a**) The feasible objective region (*blue curve*) and the Pareto front of the *Rastr* test problem (6.38), (**b**) Illustration of the performance of the proposed algorithm where *points* denote the solutions generated, *stars* denote the found non-dominated solutions, and *line segments* show the Lipschitz local lower bounds for the Pareto set

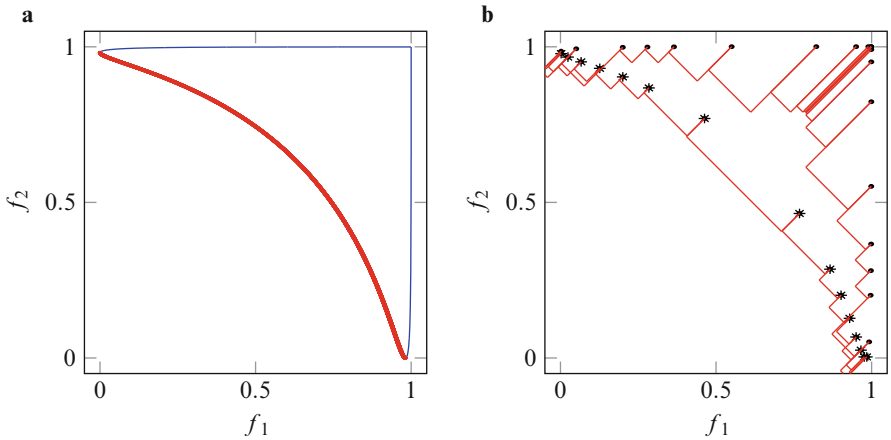


**Fig. 6.6** (**a**) The feasible objective region (*blue curve*) and the Pareto front of the *Fo&Fle* test problem (6.39), (**b**) illustration of the performance of the proposed algorithm where *points* denote the solutions generated, *stars* denote the found non-dominated solutions, and *line segments* show the Lipschitz local lower bounds for the Pareto set

decisions of the test problem *Fo&Fle* constitutes 25% of the length of the feasible optimization interval, and the Pareto front is continuous; the requested number of function evaluations was smallest, and the percentage of the found non-dominated solutions was rather high, respectively. The solution of the problem *Schaf* with the tolerance 0.1 terminated after a rather large number of function evaluations. The vast number of function evaluations was due to the relatively high (with respect

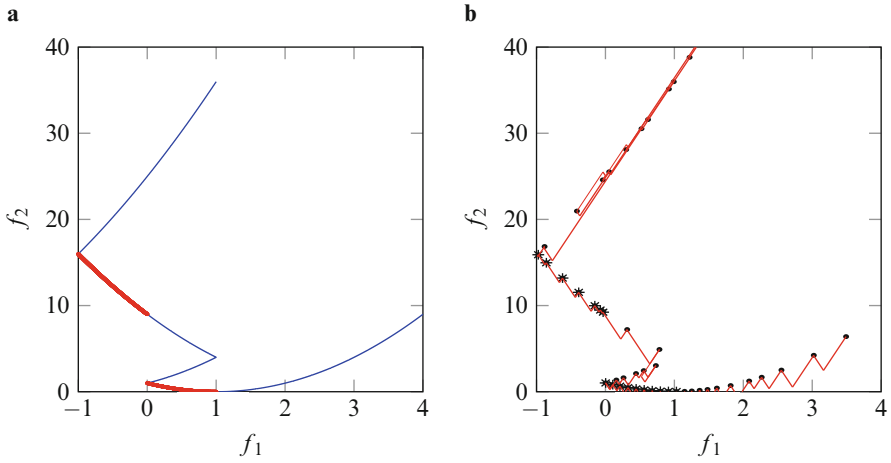


**Fig. 6.7** (**a**) The feasible objective (*blue curve*) region and the Pareto front of the *Schaf* test problem (6.48), (**b**) illustration of the performance of the proposed algorithm where *points* denote the solutions generated, *stars* denote the found non-dominated solutions, and *line segments* show the Lipschitz local lower bounds for the Pareto set

to the Lipschitz constant of $f_2(\cdot)$) requested precision in the representation of the Pareto front; in the case of the termination condition defined by the tolerance equal to 1, the algorithm stopped after $N = 47$ evaluations of the function values, and the number of non-dominated solutions found was $nd = 17$. The latter result illustrates a challenge in the rational choice of a stopping tolerance in the case of substantially different Lipschitz constants. The optimization results (with $\varepsilon_n \leq 0.1$ for (6.46) and (6.47), and $\varepsilon_n \leq 1$ for (6.48)) are also illustrated graphically in Figures 6.5, 6.6, and 6.7, where not only the solutions computed during the search, but also Lipschitz local lower bounds are shown.

For the majority of the multi-objective optimization algorithms, the non-dominated solutions found are the only information on the set of Pareto optimal solutions. The algorithm, based on the Lipschitz model, is advantageous, since the Lipschitz lower bounds provide the essential supplementary information on the Pareto front in question. In the case of bi-objective problems, that information can be visualized by the graph of Pareto front. An important but really challenging problem is to develop methods to visualize Lipschitz bounds of problems with a larger number of objectives.

To assess the performance of the proposed algorithm, a comparison with other algorithms is needed. The general testing methodology for multi-objective optimization algorithms conforms to the concept of competitive testing according to the discussion in [82]; for details we refer to [42, 60, 129, 265, 266]. However, the proposed algorithm is meant for the assessment of a theoretical concept rather than for practical applications. Therefore, its competitive testing would be premature according to the arguments similar to those presented in [252]. For the experimental

comparison, two black-box optimization algorithms were selected. The uniform distribution of the decisions in the feasible region (*Unif*), a very simple algorithm indeed, nevertheless, it is worst-case optimal in the class of Lipschitz functions as shown in Section 6.1.3. The P-algorithm (*P-alg*) is a one-step optimal algorithm with respect to the chosen statistical model of objective functions. The multi-objective P-algorithm was proposed in [252]; we refer for details to Section 7.3. Normally, a stationary random field model with the exponential correlation function is used for the construction of the P-algorithm, since that model is suitable for an arbitrary dimensionality of the decision space. Since we focus here on one-dimensional problems, the Wiener process (which is especially favorable for the construction of one-dimensional global optimization algorithms [216]) is selected as a statistical model of objective functions.

The proposed algorithm (*Lipsch*) has a natural termination condition defined using (6.45). However, such a termination condition is not applicable to two other algorithms considered. Therefore a simple termination condition after $N = 100$ computations of objective function values was applied to all the three algorithms.

To assess the efficiency of multi-objective optimization algorithms, various metrics are used [42, 60, 265, 266]. However, in the case of single-variable objective functions, the assessment problem is much simpler, and the use of many metrics would be redundant. From the visual analysis of the feasible objective regions, it can be concluded that the non-dominated solutions found are mainly the true Pareto optimal solutions. In Table 6.1, the number of non-dominated solutions found is denoted by *nd*, and the true Pareto solutions found are denoted by *np*; the metric called the error rate can be computed using the formula $(nd - np)/nd$. Usually the metric generational distance is used to evaluate the closeness of the found non-dominated solutions to the Pareto optimal set. In the considered case, however, only a few non-dominated solutions do not belong to the Pareto optimal set, and instead of the generational distance, we present the absolute error value

$$E = \max_{\tilde{\mathbf{Y}} \in \tilde{\mathbf{P}}} \min_{\mathbf{Y} \in \mathbf{P}(\mathbf{f})_{\mathbf{O}}} \|\tilde{\mathbf{Y}} - \mathbf{Y}\|, \tag{6.49}$$

**Table 6.1** The testing results where the optimization stops after the predefined number of objective function evaluations $N = 100$

| Test problem | *Rastr* | | | *Fo&Fle* | | | *Schaf* | | |
|---|---|---|---|---|---|---|---|---|---|
| Algorithm | *Lipsch* | *P-alg* | *Unif* | *Lipsch* | *P-alg* | *Unif* | *Lipsch* | *P-alg* | *Unif* |
| *nd* | 34 | 35 | 10 | 66 | 67 | 26 | 54 | 52 | 23 |
| *np* | 26 | 35 | 6 | 65 | 67 | 24 | 51 | 51 | 23 |
| *E* | 0.014 | 0 | 0.061 | 0 | 0 | 0 | 0.010 | 0.010 | 0 |
| *H* | 0.064 | 0.055 | 0.107 | 0.027 | 0.027 | 0.042 | 0.254 | 0.242 | 0.561 |

where $\tilde{\mathbf{P}}$ denotes the set of non-dominated solutions found. The similarly defined metric

$$H = \max_{\mathbf{Y} \in \mathbf{P}(\mathbf{f})_O} \min_{\hat{\mathbf{Y}} \in \hat{\mathbf{P}}} \|\hat{\mathbf{Y}} - \mathbf{Y}\|, \tag{6.50}$$

where $\hat{\mathbf{P}}$ denotes the set of the Pareto optimal solutions found, evaluates the largest "hole" in the discrete representation of the Pareto set. To compute approximate values of $E$ and $H$, the set $\mathbf{P}(\mathbf{f})_O$ was replaced by a discrete set $\mathbf{P}$ of the values of $\mathbf{f}(\cdot)$ computed at $10^6$ points, uniformly distributed in the set of Pareto optimal decisions. The results are presented in Table 6.1, where the error caused by such a discretization is smaller than $10^{-4}$; therefore the results are presented using three digits after the decimal point, and the values smaller than $10^{-4}$ are replaced by 0.

The results of numerical experiments, although limited, corroborate a considerable potential of ideas of the Lipschitz global optimization for a non-convex multi-objective optimization. The rich experience of the development of the single-objective optimization algorithms, based on the Lipschitz function model, can be essentially helpful in the extension of this algorithm to a multidimensional case.

### 6.2.6 Remarks

The ideas of single-objective Lipschitz global optimization are fruitfully applicable in the development of algorithms of non-convex multi-objective optimization. The passive worst-case optimal algorithm of multi-objective optimization for Lipschitz objectives is reduced to the computation of centers of the equal and minimum radius balls which constitute the optimal covering of the feasible region. The worst-case multi-objective problem essentially coincides with a worst-case single-objective problem, since the components of the worst-case vector objective function coincide up to linear scaling. The adaptive worst-case optimal algorithm, under the assumptions made, coincides with the passive worst-case optimal algorithm.

The concept of one-step optimality is suitable where the number of computations of objectives is not fixed in advance, a local Lipschitz constant is varying during the search, and antagonistic game theory model is not appropriate. This concept is implemented to construct an algorithm for bi-objective optimization problems with univariate non-convex objectives.

The convergence analysis of the algorithm would be an interesting topic for a further research. Extensions of the algorithm to a multidimensional case, and development of algorithms, based on the adaptive Lipschitz models, are interesting topics for the further research as well. The limited experimental testing shows that the one-step worst-case optimal algorithm for multi-objective univariate Lipschitz optimization is of a similar efficiency as the one-step optimal algorithm, based on a statistical model.

## 6.3 Multidimensional Bi-objective Lipschitz Optimization

### 6.3.1 Introduction

The problem of bi-objective non-convex optimization

$$\min_{\mathbf{x}\in\mathbf{A}} \mathbf{f}(\mathbf{x}),\ \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}))^T,\ \mathbf{A} \subset \mathbb{R}^d, \tag{6.51}$$

is considered, where the properties of the objective functions and the feasible region will be defined later.

The construction of an optimal algorithm in a broad class of global optimization algorithms is difficult [239]; nevertheless some partially optimal solutions usually are possible. We assume that the feasible region is a hyper-rectangle $\mathbf{A} = \{x :\ a_i \leq x_i \leq b_i,\ i = 1, \ldots, d\}$, and the objective functions are Lipschitz continuous. In the present section we consider hybridization of the branch and bound approach and the concept of one-step worst-case optimality with respect to the class of Lipschitz functions. Multi-objective branch and bound is presented in Chapter 5, and for the thorough presentation of branch and bound in global (including multi-objective) optimization we refer to [55, 87, 185, 208], and for the arguments in favor of the Lipschitz model we refer to [87, 110, 159, 168, 196].

We analyze the possibility to generalize, for the multidimensional case, the results of Section 6.2.4 concerning the univariate ($d = 1$) bi-objective optimization. The arguments in Section 6.2.4 show that in the worst-case setting the concept of sequential one-step optimality seems most appropriate from the applications point of view. To generalize a univariate algorithm for the multidimensional case we apply hyper-rectangular partitioning with diagonal approach [259]. The feasible region is sequentially partitioned into decreasing hyper-rectangles which are selected for subdivision on the base of a criterion which depends on the objective function values at the endpoints of the diagonal. Our goal is to define a criterion of the selection and the rule of the bisection according to the concept of one-step worst-case optimality. Besides of the one-step optimal (bisection) algorithm a similar trisection algorithm is developed where hyper-rectangles covering the feasible region are subdivided into three equal parts [259].

### 6.3.2 Lipschitz Bound for the Pareto Frontier

In this section we consider Lipschitz continuous functions for which the city-block metric is used in the decision space, i.e., the following inequalities are valid:

$$|f_k(\mathbf{x}) - f_k(\mathbf{z})| \leq L_k \cdot \sum_{i=1}^{d} |x_i - z_i|,\ k = 1, 2, \tag{6.52}$$

where $\mathbf{x} \in \mathbf{A}$, $\mathbf{z} \in \mathbf{A}$, $L_k > 0$, $k = 1, 2$.

The class of Lipschitz continuous functions is advantageous for constructing global minimization algorithms because of relatively simply computable lower bounds for the function values. The availability of such bounds enables a theoretical assessment of the quality of a discrete representation of the Pareto front for bi-objective Lipschitz optimization.

Let $\mathbf{a}(r) \in \mathbf{A}$ and $\mathbf{b}(r) \in \mathbf{A}$ be the endpoints of a diagonal of a hyper-rectangle $\mathbf{A}_r$; without loss of generality it is assumed that $a_i(r) < b_i(r)$, $i = 1, \ldots, d$. As follows from (6.52) the functions $g_k(\mathbf{x}, \mathbf{A}_r)$, $k = 1, 2$, define the lower bounds for $f_k(\mathbf{x})$, $\mathbf{x} \in \mathbf{A}_r$:

$$g_k(\mathbf{x}, \mathbf{A}_r) = \tag{6.53}$$
$$= \max\left(f_k(\mathbf{a}(r)) - L_k \sum_{i=1}^{d}(x_i - a_i(r)),\ f_k(\mathbf{b}(r)) - L_k \sum_{i=1}^{d}(b_i(r) - x_i)\right).$$

The Pareto front of the bi-objective problem

$$\min_{\mathbf{x} \in \mathbf{A}_r} \mathbf{g}(\mathbf{x}, \mathbf{A}_r), \tag{6.54}$$
$$\mathbf{g}(\mathbf{x}, \mathbf{A}_r) = (g_1(\mathbf{x}, \mathbf{A}_r), g_2(\mathbf{x}, \mathbf{A}_r))^T,$$

is denoted by $\mathbf{V}_r = \mathbf{V}(\mathbf{f}(\mathbf{a}(r)), \mathbf{f}(\mathbf{b}(r)), \mathbf{A}_r)$.

**Lemma 6.3** *No element of* $\mathbf{V}_r$ *is dominated by a vector* $\mathbf{f}(\mathbf{x})$, $\mathbf{x} \in \mathbf{A}_r$.

*Proof* Let us assume contrary that there exist $\mathbf{x} \in \mathbf{A}_r$ and $\mathbf{y} \in \mathbf{V}_r$ such that $\mathbf{z} = \mathbf{f}(\mathbf{x}) \succ \mathbf{y}$. Since (6.53) implies that $\mathbf{g}(\mathbf{x}, \mathbf{A}_r) \succeq \mathbf{z}$, and $\mathbf{V}_r$ is the subset of non-dominated elements of the set of $\{\mathbf{f}(\mathbf{x}) : \mathbf{x} \in \mathbf{A}_r\}$, there exists an element $\mathbf{v} \in \mathbf{V}_r$ such that

$$\mathbf{v} \succeq \mathbf{g}(\mathbf{x}, \mathbf{A}_r) \succeq \mathbf{z} \succ \mathbf{y}. \tag{6.55}$$

However, the obtained relation of dominance of $\mathbf{v}$ over $\mathbf{y}$ cannot be true since both, $\mathbf{v}$ and $\mathbf{y}$, are elements of the Pareto front $\mathbf{V}_r$. Therefore, the assumption made at the beginning of the proof is not truthful, and the proof is completed.

The following definition is a natural sequel of Lemma 6.3:

**Definition 6.4** $\mathbf{V}_r$ is called a local lower Lipschitz bound for $\mathbf{P}(\mathbf{f}, \mathbf{A})_O$.

**Definition 6.5** The Pareto front of the set $\bigcup_{r=1}^{R} \mathbf{V}_r$ is called a lower Lipschitz bound for $\mathbf{P}(\mathbf{f}, \mathbf{A})_O$ and is denoted by $\mathbf{V}(\mathbf{Y}_R, \mathbf{A}_{[R]})$, where $\mathbf{Y}_R = \{\mathbf{f}(\mathbf{a}(r)), \mathbf{f}(\mathbf{b}(r)) : r = 1, \ldots, R\}$, $\mathbf{A}_{[R]}$ denotes the set of hyper-rectangles which constitute a partition of $\mathbf{A}$, and $\mathbf{a}(r)$ and $\mathbf{b}(r)$ are the endpoints of the diagonals of the mentioned hyper-rectangles.

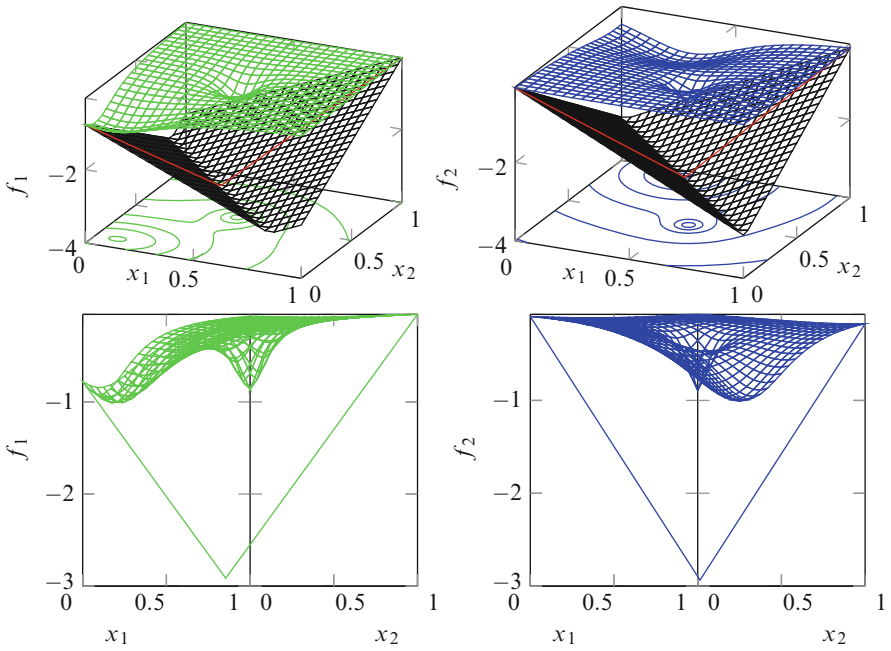


**Fig. 6.8** Illustration of two objective functions $f_1$ (*green*) and $f_2$ (*blue*) and their city-block bounding functions taking into account function values at the endpoints of the diagonal of the square $[0, 1]^2$

Let us consider the bi-objective minimization problem on a line segment $\tilde{\mathbf{A}}(r)$

$$\min_{\mathbf{x}\in\tilde{\mathbf{A}}(r)} \mathbf{g}(\mathbf{x}, \mathbf{A}_r), \tag{6.56}$$

where $\tilde{\mathbf{A}}(r)$ denotes the diagonal of $\mathbf{A}_r$. The Pareto front of (6.56) is denoted by $\tilde{\mathbf{V}}(r)$.

The properties of the objective functions $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ are illustrated in Figure 6.8. The functions are multimodal as can be seen from the surface and contour plots. We also show city-block bounding functions computed taking into account function values at the endpoints of the diagonal of the square $[0, 1]^2$ and different slopes for each objection function. If one views from a certain angle, these lower-bounding functions are seen as line segments (see lower plots). These line segments can be interpreted as bounding functions over diagonal.

Taking such bounding functions into account, the graphical representation of $\{\mathbf{g}(\mathbf{x}, \mathbf{A}_r) : \mathbf{x} \in \mathbf{A}_r\}$ can be made similarly to Section 6.2. We show such a graphical representation in Figure 6.9, where $\mathbf{f}(\mathbf{a}(r)) = \mathbf{y} = (y_1, y_2)^T$, $\mathbf{f}(\mathbf{b}(r)) = \mathbf{z} = (z_1, z_2)^T$, and $\tilde{\mathbf{V}}(r)$ is shown by a thicker line. Different slopes of bounding functions illustrate possibly different Lipschitz constants of objective functions.

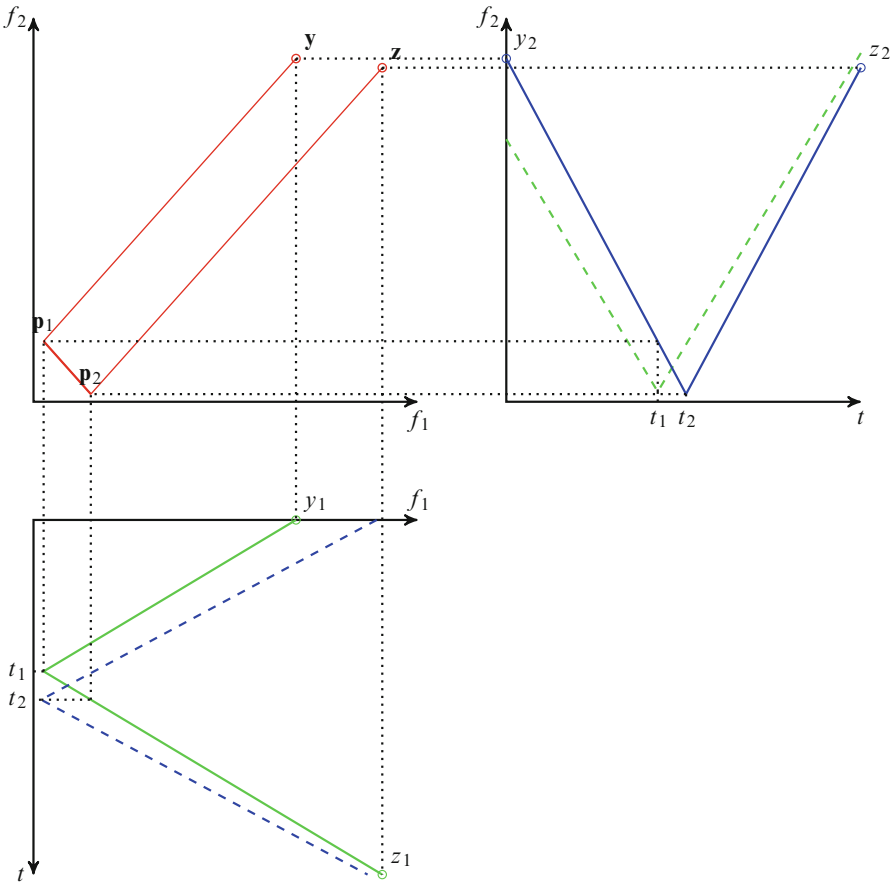


**Fig. 6.9** A graphical representation of $\{\mathbf{g}(\mathbf{x}, \mathbf{A}_r) : \mathbf{x} \in \mathbf{A}_r\}$, where $\tilde{\mathbf{V}}(r)$ is shown by *thicker line*, and $\mathbf{f}(\mathbf{a}(r)) = \mathbf{y} = (y_1, y_2)^T$, $\mathbf{f}(\mathbf{b}(r)) = \mathbf{z} = (z_1, z_2)^T$

**Lemma 6.4** *The sets* $\mathbf{V}_r$ *and* $\tilde{\mathbf{V}}(r)$ *are coincident.*

*Proof* The equality of the sets $\mathbf{G}_r = \{\mathbf{g}(\mathbf{x}, \mathbf{A}_r) : \mathbf{x} \in \mathbf{A}_r\}$ and $\tilde{\mathbf{G}}_r = \{\mathbf{g}(\mathbf{x}, \mathbf{A}_r) : \mathbf{x} \in \tilde{\mathbf{A}}(r)\}$, which is proved below, implies the proof of lemma.

The inclusion $\tilde{\mathbf{A}}_r \subset \mathbf{A}_r$ implies the inclusion $\tilde{\mathbf{G}}_r \subseteq \mathbf{G}_r$. Therefore, to prove the equality $\tilde{\mathbf{G}}_r = \mathbf{G}_r$ it is sufficient to show that an element $\mathbf{z} \in \mathbf{G}_r$ belongs also to $\tilde{\mathbf{G}}_r$.

Let us assume that $\mathbf{z} \in \mathbf{G}_r$. Then there exists such $\mathbf{x} \in \mathbf{A}_r$ that $\mathbf{z} = \mathbf{g}(\mathbf{x}, \mathbf{A}_r)$, i.e.,

$$z_k = \max\left(f_k(\mathbf{a}(r)) - L_k \sum_{i=1}^{d}(x_i - a_i(r)),\ f_k(\mathbf{b}(r)) - L_k \sum_{i=1}^{d}(b_i(r) - x_i)\right), \quad k = 1, 2. \tag{6.57}$$

Let $\tau$ be defined by the equality

$$\tau = \sum_{j=1}^{d}(x_j - a_j(r))/\sum_{j=1}^{d}(b_j - a_j(r)),$$

and $\tilde{\mathbf{x}} \in \tilde{\mathbf{A}}_r$ be defined by the equality

$$\tilde{\mathbf{x}} = \mathbf{a}(r) + \tau(\mathbf{b}(r) - \mathbf{a}(r)) \in \tilde{\mathbf{A}}(r).$$

Since the following equalities hold:

$$\sum_{i=1}^{d}(\tilde{x}_i - a_i(r)) = \sum_{i=1}^{d}(x_i - a_i(r)), \tag{6.58}$$

$$\sum_{i=1}^{d}(b_i(r) - \tilde{x}_i) = \sum_{i=1}^{d}(b_i(r) - x_i), \tag{6.59}$$

there holds also the equality $\mathbf{g}(\mathbf{x}, \mathbf{A}_r) = \mathbf{g}(\tilde{\mathbf{x}}, \mathbf{A}_r)$ which implies the inclusion $\mathbf{z} \in \tilde{\mathbf{G}}_r$.

The statement of Lemma 6.4 facilitates the computation of a local lower Lipschitz bound for $\mathbf{P}(\mathbf{f}, \mathbf{A})_O$ by means of the reduction to the computation of a lower Lipschitz bound for a similar one-dimensional problem where the domain of objectives is the diagonal of a hyper-rectangular subset of $\mathbf{A}$.

**Definition 6.6** The subset of

$$\left(\bigcup_{r=1}^{R}\{\mathbf{z}:\ \mathbf{z} \in \mathbb{R}^2,\ \mathbf{z} \geq \mathbf{f}(\mathbf{a}(r))\}\right) \bigcup \left(\bigcup_{r=1}^{R}\{\mathbf{z}:\ \mathbf{z} \in \mathbb{R}^2,\ \mathbf{z} \geq \mathbf{f}(\mathbf{b}(r))\}\right),$$

which consists of weakly Pareto optimal solutions is called a simple upper bound for $\mathbf{P}(\mathbf{f})_O$ and is denoted by $\mathbf{U}(\mathbf{Y}_R)$.

In case a tight upper bound for $\mathbf{P}(\mathbf{f})_O$ would be of interest, a combination of $\mathbf{U}(\mathbf{Y}_R)$ with an upper Lipschitz bound, defined similarly to $\mathbf{V}(\mathbf{Y}_R, \mathbf{A}_{[R]})$, could be helpful.

### *6.3.3 Properties of Local Lower Lipschitz Bound*

As follows from Lemma 6.4, $\mathbf{V}(\mathbf{f}(\mathbf{a}(r)), \mathbf{f}(\mathbf{b}(r)), \mathbf{A}_r)$, presenting a local lower Lipschitz bound for $\mathbf{P}(\mathbf{f}, \mathbf{A})_O$, is also coincident with the similar bound of a one-dimensional problem where the length of the feasible interval is equal to the length of the diagonal of $\mathbf{A}_r$. Therefore, some results concerning the properties of lower

Lipschitz bounds, proved in Section 6.2, are relevant and can be adapted to the considered here problem. Below we estimate the tightness of $\mathbf{V}(\mathbf{f}(\mathbf{a}(r)), \mathbf{f}(\mathbf{b}(r)), \mathbf{A}_r)$ using the data related only to the diagonal of $\mathbf{A}_r$.

**Definition 6.7** The precision of local lower Lipschitz bound $\mathbf{V}(\mathbf{f}(\mathbf{a}(r)), \mathbf{f}(\mathbf{b}(r)), \mathbf{A}_r)$ we assess by the tightness $\Delta(\cdot)$ which is defined by the formula

$$\Delta(\mathbf{f}(\mathbf{a}(r)), \mathbf{f}(\mathbf{b}(r)), \mathbf{A}_r) = \tag{6.60}$$
$$\max\left(\min_{\mathbf{w}\in\mathbf{V}(\mathbf{f}(\mathbf{a}(r)),\mathbf{f}(\mathbf{b}(r)),\mathbf{A}_r)} \|\mathbf{w}-\mathbf{f}(\mathbf{a}(r))\|, \min_{\mathbf{w}\in\mathbf{V}(\mathbf{f}(\mathbf{a}(r)),\mathbf{f}(\mathbf{b}(r)),\mathbf{A}_r)} \|\mathbf{w}-\mathbf{f}(\mathbf{b}(r)\|\right),$$

in case $\mathbf{f}(\mathbf{a})$ and $\mathbf{f}(\mathbf{b})$ do not dominate each other, and by the formula

$$\Delta(\mathbf{f}(\mathbf{a}(r)), \mathbf{f}(\mathbf{b}(r)), \mathbf{A}_r) = \min_{\mathbf{w}\in\mathbf{V}(\mathbf{f}(\mathbf{a}(r)),\mathbf{f}(\mathbf{b}(r)),\mathbf{A}_r)} \|\mathbf{w}-\mathbf{f}(\mathbf{a})\|, \tag{6.61}$$

otherwise, i.e., in case $\mathbf{f}(\mathbf{a}) \succ \mathbf{f}(\mathbf{b})$.

As mentioned above, the lower Lipschitz bound $\mathbf{V}(\mathbf{f}(\mathbf{a}(r)), \mathbf{f}(\mathbf{b}(r)), \mathbf{A}_r)$ is coincident with the similar bound for a vector function of one variable

$$\mathbf{F}(t) = \mathbf{f}(\mathbf{x}(t)),\ 0 \le t \le v,$$

where

$$\mathbf{x}(t) = \mathbf{a}(r) + (\mathbf{b}(r) - \mathbf{a}(r))t/v,$$
$$v = \sum_{j=1}^{d} (b_j(r) - a_j(r)).$$

If $\mathbf{F}(0) = \mathbf{y} = (y_1, y_2)^T$ and $\mathbf{F}(v) = \mathbf{z} = (z_1, z_2)^T$ do not dominate each other we assume, without loss of generality, that $z_2 \le y_2$, $y_1 \le z_1$. Otherwise we assume that $\mathbf{F}(0) \succ \mathbf{F}(v)$. The lower Lipschitz bound for the Pareto front of the problem

$$\min_{0\le t\le v} \mathbf{F}(t), \tag{6.62}$$

is coincident with the Pareto front of the bi-objective minimization problem

$$\min_{0\le t\le v} \tilde{\mathbf{F}}(t), \tag{6.63}$$

where

$$\tilde{\mathbf{F}}(t) = (\tilde{F}_1(t),\ \tilde{F}_2(t)),\ 0 \le t \le v, \tag{6.64}$$
$$\tilde{F}_i(t) = \max\{y_i - L_i t,\ z_i - L_i(v-t)\},\ i = 1, 2.$$

The Pareto front of the problem (6.63) is coincident with the line segment, the endpoints $\mathbf{p}_1$ and $\mathbf{p}_2$ of which are defined by the minima of $\tilde{F}_i(t)$, $i = 1, 2$, as illustrated by Figure 6.9. The points $\mathbf{p}_1$ and $\mathbf{p}_2$ are defined by the following formulas:

$$\mathbf{p}_1 = \begin{pmatrix} y_1 - L_1 t_1 \\ y_2 - L_2 t_1 \end{pmatrix}, \quad \mathbf{p}_2 = \begin{pmatrix} z_1 - L_1 (v - t_2) \\ z_2 - L_2 (v - t_2) \end{pmatrix}, \; t_1 \le t_2, \tag{6.65}$$

$$\mathbf{p}_1 = \begin{pmatrix} y_1 - L_1 t_2 \\ y_2 - L_2 t_2 \end{pmatrix}, \quad \mathbf{p}_2 = \begin{pmatrix} z_1 - L_1 (v - t_1) \\ z_2 - L_2 (v - t_1) \end{pmatrix}, \; t_1 > t_2, \tag{6.66}$$

where $t_i$, $i = 1, 2$ are the minimum points of $\tilde{F}^i(t)$ over the interval $[0,\ v]$:

$$t_1 = \frac{v}{2} + \frac{y_1 - z_1}{2L_1}, \; t_2 = \frac{v}{2} + \frac{y_2 - z_2}{2L_2}. \tag{6.67}$$

**Theorem 6.3** *The tightness of the local lower Lipschitz bound is defined by the formulas*

$$\Delta(\mathbf{f}(\mathbf{a}(r)), \mathbf{f}(\mathbf{b}(r)), \mathbf{A}_r) = \begin{cases} (L_1 + L_2) \max(t_1,\ (v - t_2)), \; t_1 \le t_2, \\ (L_1 + L_2) \max(t_2,\ (v - t_1)), \; t_1 > t_2. \end{cases} \tag{6.68}$$

*The error of approximation of* $\mathbf{P}(\mathbf{f}, \mathbf{A}_r)_O$ *by the line segment connecting* $\mathbf{f}(\mathbf{a}(r))$ *and* $\mathbf{f}(\mathbf{b}(r))$ *is bounded by* $\Delta(\mathbf{f}(\mathbf{a}(r)), \mathbf{f}(\mathbf{b}(r)), \mathbf{A}_r)$.

*Proof* $\Delta_r = \Delta(\mathbf{f}(\mathbf{a}(r)), \mathbf{f}(\mathbf{b}(r)), \mathbf{A}_r)$ is equal to the tightness of the local lower Lipschitz bound for the univariate problem (6.62). The latter, which is denoted by $\delta_r$, is equal to the maximum of the distances between the respective bound and points $\mathbf{F}(0)$ and $\mathbf{F}(v)$. Since the bound is coincident with the line segment connecting the points $\mathbf{p}_1$, $\mathbf{p}_2$, the following equality is valid:

$$\delta_r = \max(||\mathbf{F}(0) - \mathbf{p}_1||,\ ||\mathbf{F}(v) - \mathbf{p}_2||) = \max(||\mathbf{y} - \mathbf{p}_1||,\ ||\mathbf{z} - \mathbf{p}_2||). \tag{6.69}$$

Substitution of $\mathbf{p}_1$, $\mathbf{p}_2$ in (6.69) by their expressions (6.65), (6.66) implies (6.68).

The upper Lipschitz bound for $\mathbf{P}(\mathbf{f})_O$, which is denoted by $\tilde{\mathbf{V}}(\mathbf{F}(0), \mathbf{F}(v), 0, v)$, can be constructed similarly as the respective lower bound. $\tilde{\mathbf{V}}(\mathbf{F}(0), \mathbf{F}(v), 0, v)$ is a line segment parallel to the line segment representing the lower Lipschitz bound. The distance between these parallel line segments satisfies the following inequality:

$$D(\mathbf{F}(0), \mathbf{F}(v), 0, v) = \|\mathbf{p}_1 - (y_1, y_2)^T\| + \|\mathbf{p}_2 - (z_1, z_2)^T\| \le 2\Delta(\mathbf{F}(0), \mathbf{F}(v), 0, v).$$

Therefore, for the points of the line segment $\mathbf{h}(t) = (h_1(t),\ h_2(t))^T$ where

$$h_1(t) = y_1 + (y_2 - y_1) \cdot t, \; h_2(t) = z_1 + (z_2 - z_1) \cdot t, \; 0 \le t \le v, \tag{6.70}$$

the following inequalities are valid:

$$\min_{\mathbf{w}\in\mathbf{V}(\mathbf{F}(0),\mathbf{F}(v),0,v)} \|(h_1(t),h_2(t))^T-\mathbf{w}\| \le \Delta(\mathbf{F}(0),\mathbf{F}(v),0,v),$$

$$\min_{\mathbf{w}\in\tilde{\mathbf{V}}(\mathbf{F}(0),\mathbf{F}(v),0,v)} \|(h_1(t),h_2(t))^T-\mathbf{w}\| \le \Delta(\mathbf{F}(0),\mathbf{F}(v),0,v),$$

and imply the second statement of the theorem.

**Corollary 6.2** *If* $\mathbf{f}(\mathbf{a}(r))$ *and* $\mathbf{f}(\mathbf{b}(r))$ *do not dominate each other, and* $\Delta(\mathbf{f}(\mathbf{a}(r)),\mathbf{f}(\mathbf{b}(r)),\mathbf{A}_r)=0$*, then* $\mathbf{V}(\mathbf{f}(\mathbf{a}(r)),\mathbf{f}(\mathbf{b}(r)),\mathbf{A}_r)$ *is coincident with the line segment connecting the points* $\mathbf{f}(\mathbf{a}(r)),\mathbf{f}(\mathbf{b}(r))$*; moreover, the following equality*

$$\mathbf{V}(\mathbf{f}(\mathbf{a}(r)),\mathbf{f}(\mathbf{b}(r)),\mathbf{A}_r)=\{\phi:\phi=\mathbf{f}(\mathbf{x}),\mathbf{x}\in\mathbf{A}_r\}, \tag{6.71}$$

*holds.*

The properties of $\Delta_r=\Delta(\mathbf{f}(\mathbf{a}(r)),\mathbf{f}(\mathbf{b}(r)),\mathbf{A}_r)$ and $\mathbf{V}_r=\mathbf{V}(\mathbf{f}(\mathbf{a}(r)),\mathbf{f}(\mathbf{b}(r)),\mathbf{A}_r)$ are essential in substantiating the optimization algorithm. In a special case, where $\Delta_r=0$, the hyper-rectangle $\mathbf{A}_r$ can be excluded from the further search, and $\mathbf{V}_r$ is coincident with the local (related to $\mathbf{A}_r$) Pareto front of the original multi-objective optimization problem. Generally speaking, $\mathbf{V}_r$ and $\Delta_r$, $r=1,\ldots,R$, show the potentially reachable Pareto optimal solutions, and the quality of approximation at the current iteration.

### 6.3.4 Worst-Case Optimal Bisection

In the present section we consider the most important part of the multi-objective optimization algorithm based on the partition of the feasible region into hyper-rectangles by means of the bisection of a selected hyper-rectangle. Let $\mathbf{A}_r$ be selected according to a rule described below. We intend to define the worst-case optimal bisection of the selected hyper-rectangle. The notation used in the analysis of the bisection is explained in Figure 6.10. The worst-case optimal bisection is defined as follows:

$$(\hat{t},\hat{j})=\arg\min_{t,j}\ \max\left(\max_{\mathbf{w}}\Delta(\mathbf{f}(\mathbf{a}(r)),\mathbf{w},\mathbf{A}_{r'}),\ \max_{\mathbf{v}}\Delta(\mathbf{v},\mathbf{f}(\mathbf{b}(r)),\mathbf{A}_{r''})\right), \tag{6.72}$$

where the hyper-rectangles $\mathbf{A}_{r'}$, $\mathbf{A}_{r''}$ are obtained by the bisection of $\mathbf{A}_r$, $0\le t\le 1$ is the cutting ratio of the interval $[a_j(r),\ b_j(r)]$, $j$ defines the coordinate axis orthogonal to the cutting hyper-plane, the hyper-rectangles $\mathbf{A}_{r'}$, $\mathbf{A}_{r''}$ are obtained by the bisection of $\mathbf{A}_r$, and the feasible regions for $\mathbf{w}=\mathbf{f}(\mathbf{b}')$ and $\mathbf{v}=\mathbf{f}(\mathbf{a}'')$ depend on the cutting parameters $t$, $j$ and other information related to $\mathbf{A}_\mathbf{r}$.

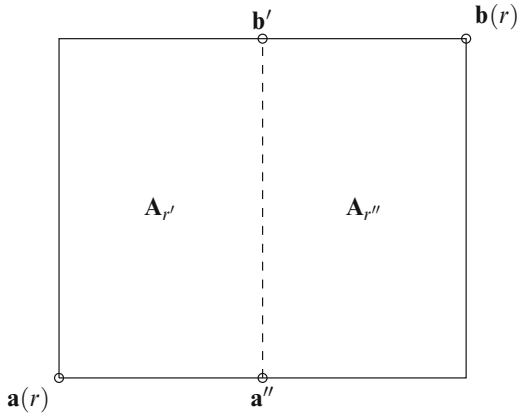


**Fig. 6.10** Bisection of a rectangle

Let us recall the assumptions made in Section 6.2.4 that $z_2 \le z_1$, $y_1 \le y_2$, and assume also that the inequality $\delta y \le \delta z$ is satisfied where $\delta y = y_2 - y_1$ and $\delta z = z_1 - z_2$. These assumptions do not reduce generality since the relevant variables can be renamed, respectively.

**Theorem 6.4** *The cutting hyper-plane of the worst-case optimal bisection is orthogonal to the longest edge of the considered hyper-rectangle. The cutting point is defined by the equality*

$$x_j = a_j + \hat{t}(b_j - a_j),$$

*where $j$ is the index of the coordinate axis corresponding to the (longest) edge $[a_j, b_j]$, and $\hat{t} = \frac{1}{2}$.*

*Proof* The equalities (6.61)–(6.67) imply the following formulas for $\Delta_r$

$$\Delta_r = \max\left(v + y_1 - z_1,\ v + z_2 - y_2\right), \tag{6.73}$$

if $\mathbf{F}(0) = (y_1, y_2)^T$ and $\mathbf{F}(v) = (z_1, z_2)^T$ do not dominate each other, and

$$\Delta_r = v + y_1 - z_1, \tag{6.74}$$

otherwise. In case $\mathbf{A}_r$ is bisected by a hyper-plane, orthogonal to the $j$-th coordinate axis, which divides $b_j(r) - a_j(r)$ with ratio $t$, the lengths of the diagonals of the obtained hyper-rectangles are equal to

$$v' = \sum_{i=1}^{d} \left(b_i(r) - a_i(r)\right) - (1-t) \cdot \left(b_j(r) - a_j(r)\right),$$

$$v'' = \sum_{i=1}^{d} (b_i(r) - a_i(r)) - t \cdot \left(b_j(r) - a_j(r)\right). \tag{6.75}$$

As follows from formulas (6.75) the minima of $v'$ and $v''$ with respect to $j$, for an arbitrary $t$, are achieved at $\hat{j}$ which corresponds to the longest edge of $\mathbf{A}_r$. On the other hand, the Lipschitz bounds for $\mathbf{w}$ and $\mathbf{v}$ in (6.72) are most restrictive if $j = \hat{j}$. Therefore the minimizer with respect to $j$ in (6.72) is equal to $j = \hat{j}$.

The computation of $\hat{t}$ involves maximization in (6.72) with respect to $\mathbf{w}$, $\mathbf{v}$ where their bounds are needed. Since $\mathbf{b}'$, $\mathbf{a}''$ are defined by the equalities

$$\begin{aligned} b'_i &= b_i(r),\ i \neq j,\ b'_j = b'_j(t) = a_j(r) + t \cdot (b_j(r) - a_j(r), \\ a''_i &= a_i(r),\ i \neq j, a''_j = a''_j(t) = a_j(r) + t \cdot (b_j(r) - a_j(r), \end{aligned} \tag{6.76}$$

the maximization bounds depend on $t$ only via $a''_j(t)$, $b'_j(t)$ in the same way as in the univariate case researched in Section 6.2.4. Therefore, the proof of this theorem is reduced to that of Theorem 6.2 in Section 6.2.4.

### 6.3.5 *Trisection of a Hyper-Rectangle*

Bisection and trisection were compared in Section 5.1. In this section we use trisection. Examples of Lipschitz bounds before and after trisection are presented in Figure 6.11. The original square and its trisection into three rectangles are shown in the upper row where diagonals are also shown. The middle row illustrates bounding functions and the lower row illustrates the Pareto bounding fronts. Lower bounds are illustrated similarly to Figure 6.9. Small circles show objective function values at the ends of diagonals and the gray lines connecting them illustrate diagonals. The currently non-dominated vectors are shown as larger circles while horizontal and vertical line segments illustrate the current approximation of the Pareto front.

### 6.3.6 *Implementation of the Algorithms*

The main idea of the algorithm is to tighten iteratively the lower Lipschitz bound, and to indicate the hyper-rectangles $\mathbf{A}_r$ which can be excluded from the further search because the whole $\mathbf{V}_r$ consists of dominated vectors [259]. The tightness of the lower Lipschitz bound $\mathbf{V}(\mathbf{Y}_R, \mathbf{A}_{[R]})$ can be assessed similarly as the local lower Lipschitz bounds but using also the information on $\mathbf{U}(\mathbf{Y}_R)$. The global tolerance for $\mathbf{P}(\mathbf{f}, \mathbf{A}_r)_O$ is denoted by $\tilde{\Delta}_r$=$\tilde{\Delta}(\mathbf{f}(\mathbf{a}(r)), \mathbf{f}(\mathbf{b}(r)), \mathbf{A}_r)$ and defined as follows:

- if $\mathbf{f}(\mathbf{a}_r)$, $\mathbf{f}(\mathbf{b}_r)$ are not dominated by the elements of $\{\mathbf{f}(\mathbf{a}_i), \mathbf{f}(\mathbf{b}_i), i = 1, \ldots, R\}$, then $\tilde{\Delta}_r = \Delta_r$,

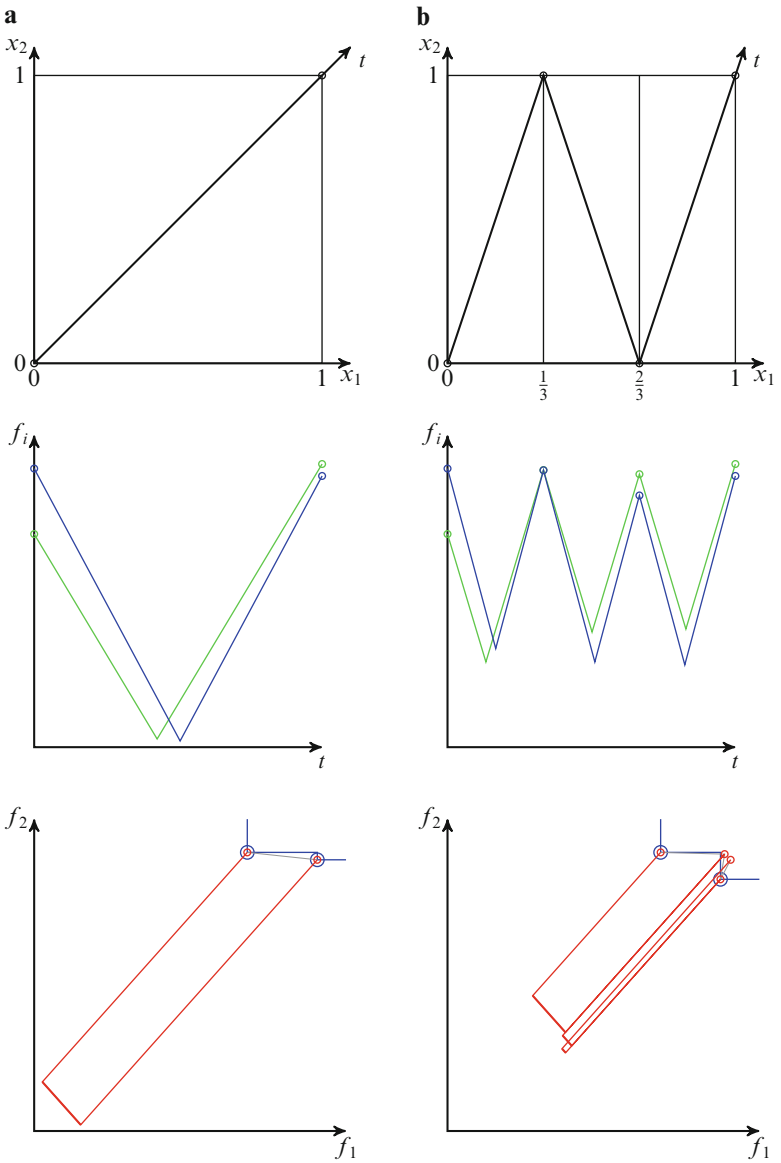


**Fig. 6.11** Illustration of bounds: (**a**) before, and (**b**) after trisection

- if all vectors belonging to $\mathbf{V}_r$ are dominated by some of elements of $\{\mathbf{f}(\mathbf{a}_i), \mathbf{f}(\mathbf{b}_i), i = 1, \ldots, R\}$, then $\tilde{\Delta}_r = 0$,
- in other cases the line segment $\mathbf{V}_r$ intersects with $\mathbf{U}(\mathbf{Y}_R)$, and

$$\tilde{\Delta}_r = \max_{\mathbf{w} \in \mathbf{V}_r} \min_{\mathbf{u} \in \mathbf{U}(\mathbf{Y}_R)} \|\mathbf{w} - \mathbf{u}\|. \tag{6.77}$$

At the $n + 1$ iteration a hyper-rectangle $\mathbf{A}_{\hat{r}}$ is selected for partition, where

$$\hat{r} = \arg\max_r \ \tilde{\Delta}_r. \tag{6.78}$$

The algorithm is terminated when

$$\max_r \ \tilde{\Delta}_r < \varepsilon$$

or after the predefined number of function evaluations.

The algorithm can be described in the following steps:

**Step 1** Initialization: start with the feasible region as the first candidate hyper-rectangle. Compute the function values at the ends of diagonal and the lower Lipschitz bound, compose the first approximation of the Pareto front.

**Step 2** Repeat remaining **Steps 3–7** while the predefined number of function evaluations is not exceeded.

**Step 3** Choose the candidate hyper-rectangle $\mathbf{A}_r$ with the largest $\tilde{\Delta}_r$. If such $\max \tilde{\Delta}_r$ is smaller than $\varepsilon$, stop.

**Step 4** Subdivide the chosen hyper-rectangle through the longest direction.

**Step 5** Compute the function values at new endpoints of diagonals and the lower Lipschitz bounds.

**Step 6** Update the current approximation of the Pareto front.

**Step 7** Store new candidate hyper-rectangles.

### *6.3.7 Numerical Examples*

Let us illustrate the performance of the algorithm using three numerical examples. We compare the results with that from [55], therefore two example problems from there are considered. The results of the algorithms are measured using hyper-volume and uniformity of distribution criteria. A hyper-volume criterion measures the volume (area in bi-objective case) of the space dominated by approximation of the Pareto front and restricted by a given reference point. The larger the hyper-volume the better approximation of the Pareto front. A uniformity of the distribution measures how uniformly the points approximating the Pareto front are distributed. We use the same estimation of uniformity as in [55]:

$$UD(\tilde{\mathbf{P}}) = \sqrt{\sum_{i=1}^{k}(d_i - d)^k},\ d = \frac{1}{k}\sum_{i=1}^{k} d_i, \tag{6.79}$$

where $d_i$ is the minimum distance from the point in objective space indexed by $i$ to the other points in $\mathbf{P}$. More uniform distribution is preferable and this corresponds to smaller values of this criterion.

The first problem from [55] is defined as follows:

$$\min_{\mathbf{x}\in[0,2]^2} \mathbf{f}(\mathbf{x}), f_1(\mathbf{x}) = x_1,\ f_2(\mathbf{x}) = \min(|x_1 - 1|, 1.5 - x_1) + x_2 + 1. \tag{6.80}$$

The Lipschitz constant of a function corresponding to the city-block metric is the maximum of infinity norm of the gradient over the considered feasible region [157]. Since, $\nabla f_1 = (1, 0)^T$ and $\nabla f_2 = (\pm 1, 1)^T$, the Lipschitz constants for this example problem are $L_1 = 1$ and $L_2 = 1$.

To get a number of function evaluations close to 500 as in [55] we choose $\varepsilon = 0.025$. The algorithm stops after 435 function evaluations. The number of points in the approximation of the Pareto front is 92 and the hyper-volume of the approximation using the reference point $(2, 3)$ is 3.60519 (99.5% of the hyper-volume of the true Pareto front). Using the similar number of function evaluations we get the larger number of points in the approximation and the larger hyper-volume than one using non-uniform covering method described in [55]. The results are illustrated in Figure 6.12 where space of objectives is shown in Figure 6.12a, and space of variables in Figure 6.12b. *Blue circles* represent non-dominated vectors and *blue lines* illustrate the approximation of the Pareto front. Two *black line*

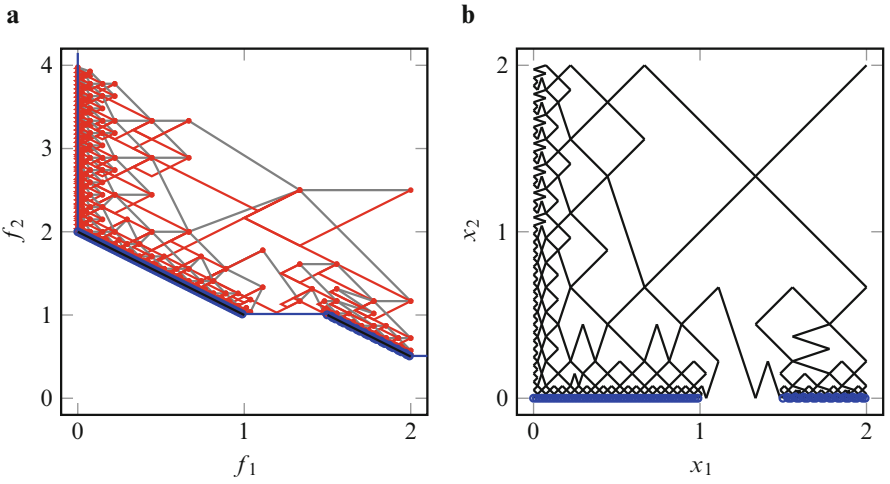


**Fig. 6.12** Illustration of the results optimizing problem (6.80): (**a**) Space of objectives, (**b**) Space of variables

segments represent the true Pareto front. Hyper-rectangles of the final subdivision are illustrated as diagonals in Figure 6.12b. In Figure 6.12a they are illustrated as *gray dots* showing objective function values at the ends of the diagonal, *gray line* connecting them, and *red lines* illustrating lower bounds similarly to Figure 6.9. It can be seen that the hyper-rectangles close to Pareto optimal decisions are smaller and diagonals form non-uniform net filling the space of variables.

The second problem from [55] is defined by the following formulas:

$$\min_{\mathbf{x}\in[0,1]^2} \mathbf{f}(\mathbf{x}), f_1(\mathbf{x}) = (x_1 - 1)x_2^2 + 1,\ f_2(\mathbf{x}) = x_2. \tag{6.81}$$

Since $\nabla f_1 = (x_2^2,\ 2x_2(x_1 - 1))^T$ and $\nabla f_2 = (0, 1)^T$, the appropriate Lipschitz constants over $[0, 1]^2$ are $L_1 = 2$ and $L_2 = 1$.

To get a number of function evaluations close to 500 as in [55] we choose $\varepsilon = 0.063$. The algorithm stops after 498 function evaluations. The number of points in the approximation of the Pareto front is 68 and hyper-volume of the approximation using the reference point $(1, 1)$ is 0.308484. Similarly to problem (6.80) we get the larger number of points in the approximation and the bigger hyper-volume than ones using the non-uniform covering method described in [55]. Uniformity of the point distribution in the approximation calculated according to (6.79) is 0.174558 which is smaller (better) than that for non-uniform covering method [55].

The results are illustrated in Figure 6.13. The *black curve* represents the true Pareto front. Again, the hyper-rectangles close to Pareto optimal decisions are smaller and their diagonals form non-uniform net filling the space of variables.

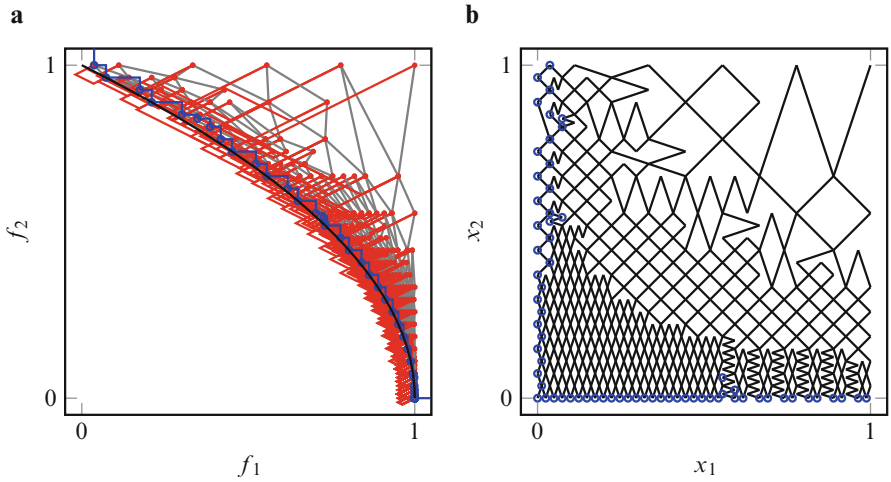


**Fig. 6.13** Illustration of the results optimizing problem (6.81): (**a**) Space of objectives, (**b**) Space of variables

**Table 6.2** Summary of the results of numerical examples and comparison to results in [55]

| Method | NFE | $\varepsilon$ | NGEN | NP | HV | UD |
|---|---|---|---|---|---|---|
| *Problem (6.80)* | | | | | | |
| Genetic algorithm | 500 | | 500 | 221 | 3.27 | |
| Monte Carlo | 500 | | | 22 | 3.38 | |
| non-uniform covering | 490 | 0.07 | | 36 | 3.42 | |
| Multi-objective trisection | 435 | 0.025 | | 92 | 3.61 | |
| *Problem (6.81)* | | | | | | |
| Genetic algorithm | 500 | | 500 | 104 | 0.312 | 1.116 |
| Monte Carlo | 500 | | | 67 | 0.300 | 1.277 |
| non-uniform covering | 515 | 0.0675 | | 29 | 0.306 | 0.210 |
| Multi-objective trisection | 498 | 0.063 | | 68 | 0.308 | 0.175 |

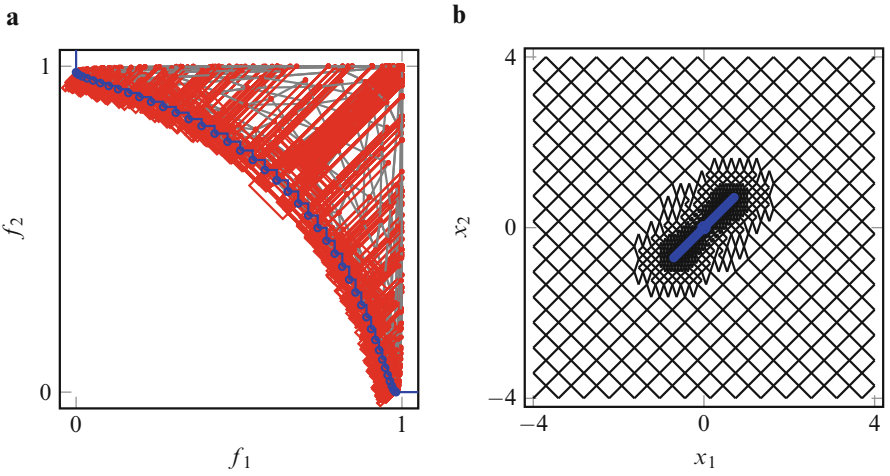


**Fig. 6.14** Illustration of the results optimizing problem (1.5): (**a**) Space of objectives, (**b**) Space of variables

The results are summarized in Table 6.2. Here, NP shows the number of non-dominated solutions in approximation of the Pareto front, HV represents the hyper-volume, and UD uniformity of the point distribution calculated according to (6.79). The results of genetic algorithm, Monte Carlo, and non-uniform covering from [55] are given. The table illustrates that the trisection algorithm outperforms all others on Problem (6.80). For Problem (6.81) trisection algorithm outperforms others by means of uniformity of the point distribution, but the genetic algorithm produces more points in approximation and a little bit better hyper-volume.

As a third example the problem (1.2) was solved. The algorithm stops after 1176 function evaluations when $\varepsilon = 0.1$ providing 46 points in the approximation of the Pareto front. The results are illustrated in Figure 6.14 where one can see non-uniform net of diagonals filling the space of variables with tighter coverage close to Pareto optimal decisions and reasonable uniformity of the point distribution in the approximation of the Pareto front.

### *6.3.8 Remarks*

In this section we discussed the multidimensional bi-objective optimization algorithm based on the branch and bound approach and trisection of hyper-rectangles. The bounding fronts are based on Lipschitz conditions for the objective functions. Numerical examples confirm the usefulness of the approach. After stopping the algorithm the hyper-rectangles close to Pareto optimal decisions are relatively small and their diagonals form non-uniform net in the space of variables.

# Chapter 7
# Statistical Models Based Algorithms

## 7.1 Introduction

Multi-objective optimization problems with expensive objective functions are typical in engineering design, where time-consuming computations are involved for modeling technological processes. Frequently, the available software implements an algorithm to compute the values of objective functions, but neither details of implementation nor analytical properties of the objective functions are known. Nevertheless, the continuity of the objective functions can be normally assumed. The complexity of the computational model implies not only the expensiveness of the objective function but also the uncertainty in its properties, so that other analytical properties of $\mathbf{f}(\mathbf{x})$, besides the continuity, cannot be substantiated. Such unfavorable, from the optimization point of view, properties of $\mathbf{f}(\mathbf{x})$ as non-differentiability, non-convexity, and multimodality cannot be excluded. Difficulties of the black-box global optimization of expensive functions are well known from the experience gained in the single-objective case.

The substantiation of an optimization algorithm based on the ideal black-box model would be hardly possible. To construct algorithms rationally, a hypothetic behavior of objective functions in question should be guessed. Therefore, the concept of black-box should be revisited. We will consider mathematical models of some transparency complemented black-box for which the term gray box seemingly would be suitable. However, following widely accepted tradition we will use the term black-box. Actually, we assume that the analytical properties of the objective functions are not available but some very general assumptions concerning the uncertainty of their behavior can be made. In the deterministic approach, typically an assumption is made which enables the construction of underestimates for function values, e.g., the assumption on Lipschitz continuity of the considered objective function enables the construction of Lipschitz underestimates. For a brief review of Lipschitz optimization, we refer to Section 4.2 and for the details to [87]. Another approach, based on statistical models of objective functions, is validated



P.M. Pardalos et al., *Non-Convex Multi-Objective Optimization*, Springer
Optimization and Its Applications 123, DOI 10.1007/978-3-319-61007-8_7

in [242]; for a brief review of single-objective optimization methods based on statistical models, we refer to Section 4.3. Since the statistical models and the utility theory-based global optimization is a well theoretically justified approach to the single-objective global optimization of expensive functions, it seems worthwhile to generalize that approach also to the multi-objective case.

Let us recall the notation of the considered problem

$$\min_{\mathbf{x}\in\mathbf{A}} \mathbf{f}(\mathbf{x}),\ \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(x), \ldots, f_m(\mathbf{x}))^T,\ \mathbf{A} \subset \mathbb{R}^d, \tag{7.1}$$

where the vector objective function $\mathbf{f}(\mathbf{x})$ is defined over a simple feasible region, e.g., for the concreteness it can be assumed that $\mathbf{A}$ is a hyper-rectangle. We aim at numerical approximation of the set of Pareto optimal solutions. Frequently, in the starting optimization phase, a rough approximation of the whole Pareto optimal set is of interest; in the intermediate phase, a subset of the Pareto optimal set of interest is intended to be approximated more precisely; finally, a specific Pareto optimal solution is sought. A similar strategy is also justified in single-objective global optimization: starting from a uniform search over the feasible region, concentrating the search in prospective subregions, and finishing with a local algorithm chosen according to the local properties of the objective function. Here, we consider the search for a discrete representation of the whole set of Pareto optimal solutions.

The situations favorable for statistical methods based single-objective optimization methods are discussed in Section 4.3. The multi-objective optimization methods based on statistical models are aimed at the problems with similar properties, namely, at the problems with black-box, multimodal, and expensive objectives. Let us recall that objective functions are supposed to be expensive if the computation of the (vector) value of $\mathbf{f}(\mathbf{x})$ is time consuming.

## 7.2 Statistical Model

To validate, from the decision theory perspective, the selection of a site for current computation/observation of the vector of objectives, a model of objective functions is needed. We consider here an approach based on statistical models of the objective functions. For the consideration of functions under uncertainty, the stochastic function models are developed in the probability theory. Assuming such a classical probabilistic model we assume that the considered objective functions are random realizations of the chosen stochastic function. However, the algorithms considered below can also be interpreted in terms of more general statistical models constructed in [242], using the ideas of subjective probabilities that are discussed from all angles in [58] and [182]. To facilitate the implementation of the corresponding algorithms, the Gaussian stochastic functions normally are chosen for statistical models.

The accepted for the statistical models of separate objectives $f_j(\mathbf{x})$ stochastic functions comprise a vector-valued Gaussian random field $\Xi(\mathbf{x})$ which is accepted for the statistical model of $\mathbf{f}(\mathbf{x})$. In many real world applied problems as well as in

the test problems, the objectives are not (or weakly) interrelated. Correspondingly, components of $\Xi(\mathbf{x})$ are supposed to be independent. The correlation between the components of $\Xi(\mathbf{x})$ could be included into the model; however, it would imply some numerical and statistical inference problems requiring a further investigation.

It is assumed that a priori information about the expected behavior (a form of variation over $\mathbf{A}$) of objective functions is not available. The heuristic assumption on the lack of a priori information is formalized as an assumption that $\xi_i(\mathbf{x})$, $i = 1, \ldots, m$, are homogeneous isotropic random fields, i.e., that their mean values $\mu_i$ and variances $\sigma_i^2$ are constants, and that the correlation between $\xi_i(\mathbf{x}_j)$ and $\xi_i(\mathbf{x}_k)$ depends only on $||\mathbf{x}_j - \mathbf{x}_k||_D$. The choice of the exponential correlation function $\rho(t) = \exp(-ct)$, $c > 0$, is motivated by the fact that, in the one-dimensional case, such a correlation function ensures Markovian property, and by the positive previous experience in use of the stochastic models with the exponential correlation function for the construction of single-objective global optimization; see, e.g., the monograph [139].

The parameters of the statistical model should be estimated using data on $\mathbf{f}(\cdot)$; since the components of $\Xi(\mathbf{x})$ are assumed to be independent, the parameters for each $\xi_j(\mathbf{x})$ can be estimated separately. In the further described implementation, mean values and variances of $\xi_j(\mathbf{x})$ are estimated using their values at random points generated uniformly over $\mathbf{A}$. Arithmetic means are used as the estimates of mean values, and sample variances are used as the estimates of variances. The choice of the value of the correlation function parameter $c$ depends on the scale of variables; in the case of $\mathbf{A}$, scaled to the unit hyper-cube, $c$ is defined by the equation $c\sqrt{d} = 7$. Such a choice is motivated by a supposed uncertainty in the behavior of the objective functions: the correlation between random field values at the maximum distant points is assumed vanishing as $10^{-3}$; on the other hand, the random field values at the closely located points are assumed reasonably correlated, i.e., it is supposed that the correlation coefficient is about 0.5 in the case the mutual distance constitutes 10% of the maximum distance in $\mathbf{A}$.

## 7.3 Multi-Objective P-Algorithm

In this section, a single-objective global optimization algorithm, based on a statistical model of multimodal functions, namely the P-algorithm, is generalized for the case of black-box multi-objective optimization. For the axiomatic definition of statistical models of multimodal black-box functions and the single-objective P-algorithm, we refer to [242, 243].

The minimization is considered at the $n + 1$-st minimization step. The points where the objective functions were computed are denoted by $\mathbf{x}_i$, $i = 1, \ldots, n$, and the corresponding objective vectors are denoted by $\mathbf{y}_i = \mathbf{f}(\mathbf{x}_i)$; $\mathbf{y}_i = (y_{i1}, \ldots, y_{im})$. Based on the discussion in the previous section, the vector-valued Gaussian random field

$$\Xi(\mathbf{x}) \in \mathbb{R}^m, \ \mathbf{x} \in \mathbf{A} \subset \mathbb{R}^d, \tag{7.2}$$

is accepted as a model of the vector objective function. In the frame of that model, an unknown vector of objectives $\mathbf{f}(\mathbf{x})$, $\mathbf{x} \neq \mathbf{x}_i$, $i = 1, \ldots, n$, is interpreted as a random vector whose distribution is defined by the conditional distribution function of $\Xi(\mathbf{x})$

$$\Pi_x^n(\mathbf{t}) = P\{\Xi(\mathbf{x}) \leq \mathbf{t} | \Xi(\mathbf{x}_i) = \mathbf{y}_i,\ i = 1, \ldots, n\}. \tag{7.3}$$

The choice of the current observation point, i.e., of the point where to compute the vector of objectives at the current minimization step, is a decision under uncertainty. The statistical model (7.2) represents the uncertainty in a result of the decision. With respect to the statistical model (7.2), the choice of the current observation point $\mathbf{x}_{n+1} \in \mathbf{A}$ means a choice of a distribution function from the set of distribution functions $\Pi_x^n(\cdot)$, $\mathbf{x} \in \mathbf{A}$. To justify the choice, the methodology of rational decision making under uncertainty can be applied. If the preference of choice satisfies the rationality principles, then there exists a unique (to within a linear transformation) utility function $U(\cdot)$ compatible with the preference of choice [58]:

$$\Pi_x^n(\cdot) \succ \Pi_z^n(\cdot) \text{ iff } \int_{-\infty}^{\infty} \ldots \int_{-\infty}^{\infty} U(\mathbf{t}) \cdot d\Pi_x^n(\mathbf{t}) \geq \int_{-\infty}^{\infty} \ldots \int_{-\infty}^{\infty} U(\mathbf{t}) \cdot d\Pi_z^n(\mathbf{t}). \tag{7.4}$$

The problem of selection of a utility function for the case of single-objective minimization ($\min_{\mathbf{x} \in \mathbf{A}} f(\mathbf{x})$) was considered in [243]. The classical axioms of rational decision making (see, e.g., [58, 182]) are reformulated in terms of the rationality of search for the global minimum in [243], where it has also been shown that the following utility function

$$u(t) = 1, \text{for } t \leq y^n, \text{and } u(t) = 0, \text{for } t > y^n \tag{7.5}$$

is compatible with these axioms; $y^n$ denotes an improvement threshold: a new function value at the current step is considered an improvement if it is smaller than $y^n$; $y^n < y_i$, $i = 1, \ldots, n$; $y_i = f(\mathbf{x}_i)$ are the function values computed at previous minimization steps. The average utility defined using the utility function (7.5) means the improvement probability. The corresponding algorithm is called the P-algorithm since the point of the computation of the objective function at the current minimization coincides with the maximizer of the improvement probability

$$\mathbf{x}_{n+1} = \arg\max_{\mathbf{x} \in \mathbf{A}} \mathbf{P}\{\xi(\mathbf{x}) \leq y^n | \xi(\mathbf{x}_1) = y_1, \ldots, \xi(\mathbf{x}_n) = y_n\}. \tag{7.6}$$

The selection of the threshold in the starting and middle phases of minimization should be considerably below the current record as discussed in [216]. For an investigation of the influence of the vanishing gap, between the threshold and the current record, to the asymptotic convergence rate of the one-dimensional P-algorithm, we refer to [31].

Since the components of the random field $\Xi(\mathbf{x})$ are assumed independent, it seems also natural to assume the corresponding utilities independent where independence means the factorability of the utility function

$$U(\mathbf{t}) = u_1(t_1) \cdot u_2(t_2) \cdot \ldots \cdot u_m(t_m). \tag{7.7}$$

Assuming that the structure of $u(\cdot)$ is coincident with that in (7.5), we obtain that $U(\mathbf{t}) = 1$ for $\mathbf{t} \leq \mathbf{y}^n$, and $U(\mathbf{t}) = 0$ otherwise; here, a vector non-dominated by $\mathbf{y}_i$, $i = 1, \ldots, n$, can be selected as a natural threshold:

$$\mathbf{y}^n = (y_1^n, y_2^n, \ldots, y_m^n)^T.$$

The corresponding P-algorithm of multi-objective optimization computes the current vector of objectives at the point

$$\mathbf{x}_{n+1} = \arg\max_{\mathbf{x} \in \mathbf{A}} \mathbf{P}\{\Xi(\mathbf{x}) \leq \mathbf{y}^n | \Xi(\mathbf{x}_1) = \mathbf{y}_1, \ldots, \Xi(\mathbf{x}_n) = \mathbf{y}_n\}. \tag{7.8}$$

The implementation of the multi-objective P-algorithm is similar to that of the single-objective P-algorithm developed in [243]. Since the components of the vector random field are assumed independent, the probability in (7.8) is computed as the product of probabilities for the respective (scalar valued) random fields

$$\begin{aligned}
&\mathbf{P}\{\Xi(\mathbf{x}) \leq \mathbf{y}^n | \Xi(\mathbf{x}_1) = \mathbf{y}_1, \ldots, \Xi(\mathbf{x}_n) = \mathbf{y}_n\} \\
&= \prod_{i=1}^{m} \mathbf{P}\{\xi_i(\mathbf{x}) \leq y_i^n | \xi_i(\mathbf{x}_1) = y_{1i}, \ldots, \xi_i(\mathbf{x}_n) = y_{ni}\} \\
&= \prod_{i=1}^{m} \int_{-\infty}^{y_i^n} \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{(m_i(\mathbf{x}|\cdot) - t)^2}{2s_i^2(\mathbf{x}|\cdot)}\right\} dt, \\
&= \prod_{i=1}^{m} G(y_i^n,\, m_i(\mathbf{x}|\cdot),\, s_i(\mathbf{x}|\cdot)),
\end{aligned} \tag{7.9}$$

where $G(\cdot, m, s)$ denotes the Gaussian (with the mean value $m$ and the standard deviation $s$) cumulative distribution function; $m_i(\mathbf{x}|\cdot)$ and $s_i^2(\mathbf{x}|\cdot)$ denote the conditional mean and conditional variance of the random field $\xi_i(\mathbf{x})$ at the point $\mathbf{x}$.

The time needed to compute the value of (7.9) consists mainly of the time of computing the conditional means and conditional variances. The computation of $m_i(\mathbf{x}|\cdot)$ and $s_i(\mathbf{x}|\cdot)$ at the $n+1$ minimization step is reduced to the inversion of the $n \times n$ matrix, multiplication of the inverted matrix with the $n$-dimensional vectors, and the scalar multiplication of $n$-dimensional vectors. The operation of the highest complexity is matrix inversion; in the case of different correlations functions of the components of the random field, this complexity is assessed as $O(m \times n^3)$. The complexity of other computations is $O(m \times n^2)$. The number of computations of the values of (7.9), performed for planning of the current step, depends on the minimization algorithm, applied in (7.8). In the experiments described below, a simple version of multistart was used; the details are presented in Section 7.5.2. An operation of the greatest complexity at the current minimization step is the

inversion of the $n \times n$ correlation matrix, but this operation is performed only once, and its result is used in the repeated computations of $m_i(\mathbf{x}|\cdot)$ and $s_i(\mathbf{x}|\cdot)$ during the maximization in (7.8). From the point of view of implementation, the multi-objective P-algorithm is a simple generalization of the single-objective P-algorithm; the implementation of the latter is discussed in detail in [216] and [239].

## 7.4 Multi-Objective $\pi$-Algorithm

In this section, a recently proposed approach [250] constructing multi-objective optimization algorithms, induced by the rational decision theory, is considered. The properties, to be satisfied by a rational decision concerning the current optimization step, are postulated. These properties are inherent for several well-known single-objective optimization algorithms, e.g., for the P-algorithm [243]. The proposed approach, from a new more general perspective, substantiates the single-objective P-algorithm. For the multi-objective optimization, this approach not only constitutes a new, more general, substantiation of the known algorithms but also facilitates construction of a family of algorithms, similar in a sense to the multi-objective P-algorithm.

### *7.4.1 A New Approach to Single-Objective Optimization*

Let us consider the current minimization step, where $n$ function values have been computed at the previous steps: $y_i = f(\mathbf{x}_i)$, $i = 1, \ldots, n$. A rational choice of a point for the next computation of the objective function value cannot be performed without the assessment of the uncertainty in the result of the computation. The only objective information on $f(\cdot)$ is $\mathbf{x}_i$, $y_i$, $i = 1, \ldots, n$. Besides that objective information, normally some subjective information is available, e.g., the experience of solution of similar problems in the past. As shown in [242], very general assumptions on the rational perception of uncertainty imply a random variable model for the objective function value to be computed, i.e. those assumptions imply that a random variable $\xi_x$, $\mathbf{x} \in \mathbf{A}$, $\mathbf{x} \neq \mathbf{x}_i$, $i = 1, \ldots, n$, is acceptable as a statistical model of the unknown value of the objective function $f(\mathbf{x})$. We refer to [216, 239] for the axiomatic construction of a computationally simple statistical model of objective functions. In case a stochastic function is chosen as a statistical model of $f(\cdot)$, the corresponding random variable is defined by the conditional distribution of this stochastic function.

Let us consider the choice of a point for the current computation of the objective function value. Such a choice in the black-box situation is a decision under uncertainty, and the rational decision theory [58] can be applied to make the choice rationally. The theory suggests to make a decision by maximizing the average utility. To compute the average utility, a utility function is needed besides of a statistical

model of uncertainty. A utility function corresponding to the conception of global optimization is proposed in [243]. However, a natural extension of the axioms proposed in [243] to the multi-objective case is difficult.

Any characterization of a random variable normally includes a location parameter (e.g., the mean value) and a spread parameter (e.g., the standard deviation); we use a minimal description of $\xi_x$ by these two parameters denoted by $m(\mathbf{x})$ and $s(\mathbf{x})$. The dependence of both parameters on the information available at the current optimization step $(\mathbf{x}_i, y_i, i = 1, \ldots, n)$ will be included into the notation where needed. Assume that the average utility of computation of the current objective function value at the point $\mathbf{x}$ at the $n$ step $V_{n+1}(\cdot)$ depends on $\mathbf{x}$ only via $m(\mathbf{x})$ and $s(\mathbf{x})$:

$$V_{n+1}(m(\mathbf{x}), s(\mathbf{x}), y^n) = \int_{-\infty}^{\infty} u(t) d\Pi_x^1(t), \tag{7.10}$$

where the threshold for the improvement is also shown as a variable influencing the average utility. The subscript can be omitted in the cases where general properties of the average utility are considered independently of the number of steps of the search. The shorthand $v_{n+1}(\mathbf{x})$ is usable where only the dependence of the average utility on $\mathbf{x}$ is of interest.

The following assumptions on $V(\cdot)$ express the invariance of the utility with respect to the scales of the objective function values:

$$\begin{aligned} V(m(\mathbf{x}) + c, s(\mathbf{x}), y^n + c) &= V(m(\mathbf{x}), s(\mathbf{x}), y^n), \\ V(m(\mathbf{x}) \cdot C, s(\mathbf{x}) \cdot C, y^n \cdot C) &= V(m(\mathbf{x}), s(\mathbf{x}), y^n),\ C > 0, \end{aligned} \tag{7.11}$$

which seem rational because of arbitrariness of scales of variables in the algorithmic implementation of the mathematical model of considered problem. Since the minimization problem is considered, it is desirable to find a possibly small objective function value at every iteration; therefore, we postulate that

$$m < \mu \text{ implies } V(m, s, y) > V(\mu, s, y). \tag{7.12}$$

The postulated properties are inherent for several well-known optimization algorithms as shown in [52, 248].

**Theorem 7.1** *The function that satisfies assumptions (7.11) is of the following structure*

$$V(m(\mathbf{x}), s(\mathbf{x}), y^n) = \pi\left(\frac{y^n - m(\mathbf{x})}{s(\mathbf{x})}\right). \tag{7.13}$$

*Moreover, if assumption (7.12) is satisfied, then $\pi(\cdot)$ is an increasing function.*

*Proof* The substitution of $-y^n$ for $c$ in the equality $V(m(\mathbf{x}) + c, s(\mathbf{x}), y^n + c) = V(m(\mathbf{x}), s(\mathbf{x}), y^n)$ results in

$$V(m(\mathbf{x}), s(\mathbf{x}), y^n) = V(m(\mathbf{x}) - y^n, s(\mathbf{x}), 0). \tag{7.14}$$

The substitution of $1/s(\mathbf{x})$ for $C$ in (7.14) and the second equality in (7.11) yields the following equality

$$V(m(\mathbf{x}), s(\mathbf{x}), y^n) = U\left(\frac{m(\mathbf{x}) - y^n}{s(\mathbf{x})}, 1, 0\right). \tag{7.15}$$

Now, equality (7.13) is obtained simply by denoting $\pi(z) = V(-z, 1, 0)$. Assumption (7.12) obviously implies that $\pi(z)$ is an increasing function of $z$.

The theorem substantiates the construction of the global optimization algorithm the $n+1$ iteration of which is defined by the solution of the following maximization problem

$$\mathbf{x}_{n+1} = \max_{\mathbf{x}\in\mathbf{A}} \pi\left(\frac{y^n - m(\mathbf{x}|\mathbf{x}_i, \mathbf{y}_i,\ i = 1, \ldots, n)}{s(\mathbf{x}|\mathbf{x}_i, \mathbf{y}_i,\ i = 1, \ldots, n)}\right). \tag{7.16}$$

Formula (7.16) is coincident with that derived in [243], where the assumption on the Gaussian distribution of $\xi(\mathbf{x})$ is made. In that case, $\pi(\cdot)$ is the Gaussian distribution function, and the utility of the current computation is interpreted as the improvement probability. For single-objective optimization, the theorem generalizes the known P-algorithm. For multi-objective optimization, such a generalization is more important since it provides more flexibility in constructing relevant algorithms.

### *7.4.2 The Generalization to the Multi-Objective Case*

Recently several papers have been published which propose multi-objective optimization algorithms that generalize single-objective optimization algorithms based on statistical models of objective functions [53, 101, 105, 106, 142, 224, 252]. The numerical results included there show the relevance of the proposed algorithms to the problems of multi-objective optimization with black-box expensive objectives. We present here a new idea for constructing relevant algorithms.

A multi-objective minimization problem can be stated almost identically to the single-objective problem considered in the previous subsection:

$$\min_{\mathbf{x}\in\mathbf{A}} \mathbf{f}(\mathbf{x}),\ \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}))^T,\ \mathbf{A} \subset \mathbb{R}^d, \tag{7.17}$$

however, the concept of solution in this case is more complicated. For the definitions of the solution to a multi-objective optimization problem with nonlinear objectives, we refer to Chapter 1.

In the case of multi-objective optimization, a vector objective function $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}))^T$ is considered. The same arguments, as in the case of single-objective optimization, corroborate the applicability of statistical models. The assumptions on black-box information and expense of the objective functions together with the standard assumptions of rational decision making imply the acceptability of a family of random vectors $\Xi(\mathbf{x}) = (\xi_1(\mathbf{x}), \ldots, \xi_m(\mathbf{x}))^T$, $x \in \mathbf{A}$, as a statistical model of $\mathbf{f}(\mathbf{x})$. Similarly, the location and spread parameters of $\xi_i(\mathbf{x})$, denoted by $m_i(\mathbf{x})$, $s_i(\mathbf{x})$, $i = 1, \ldots, r$, are essential in the characterization of $\xi_i(\mathbf{x})$. For a more specific characterization of $\Xi(\mathbf{x})$, e.g., by a multidimensional distribution of $\Xi(\mathbf{x})$, the available information usually is insufficient. If the information on, e.g., the correlation between $\xi_i(\mathbf{x})$ and $\xi_j(\mathbf{x})$ were available, the covariance matrix could be included into the statistical model. However, here we assume that the objectives are independent, and the spread parameters are represented by a diagonal matrix $\Sigma(\mathbf{x})$ whose diagonal elements are equal to $s_1, \ldots, s_m$. Similarly to the case of single-objective optimization, we assume that the utility of choice of the point for the current computation of the vector value $\mathbf{f}(\mathbf{x})$ has the following structure

$$v_{n+1}(\mathbf{x}) = V_{n+1}(\mathbf{m}(\mathbf{x}), \Sigma(\mathbf{x}), \mathbf{y}^n), \tag{7.18}$$

where $\mathbf{m}(\mathbf{x}) = (m_1(\mathbf{x}), \ldots, m_m(\mathbf{x}))^T$, and $\mathbf{y}^n$ denotes a vector desired to improve.

At the current optimization step, a point for computing the value of $\mathbf{f}(\mathbf{x})$ is sought by an optimization algorithm which maximizes $v_{n+1}(\mathbf{x})$ which should be invariant with respect to the scales of data. Such a rationality assumption can be expressed by the following properties of $V(\cdot)$:

$$\begin{aligned}
&V(\mathbf{m}(\mathbf{x}) + \mathbf{c}, \Sigma(\mathbf{x}), \mathbf{y}^n + \mathbf{c}) = V(\mathbf{m}(\mathbf{x}), \Sigma(\mathbf{x}), \mathbf{y}^n),\ \mathbf{c} = (c_1, \ldots, c_m)^T,\\
&V(\mathbf{C} \cdot \mathbf{m}(\mathbf{x}), \mathbf{C} \cdot \Sigma(\mathbf{x}), \mathbf{C} \cdot \mathbf{y}^n) = V(\mathbf{m}(\mathbf{x}), \Sigma(\mathbf{x}), \mathbf{y}^n),\\
&\mathbf{C} = \begin{pmatrix} C_1 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & C_m \end{pmatrix},\ C_i > O.
\end{aligned} \tag{7.19}$$

Since the minimization problem is considered, it is desirable to find a vector of objectives with possibly small values at every iteration; therefore, we postulate that for $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_m)^T$, where $\mu_i \geq m_i$, $i = 1, \ldots, r$, and at least one inequality is strict, the following inequality is valid

$$V(\mathbf{m}, \Sigma, \mathbf{y}) > V(\boldsymbol{\mu}, \Sigma, \mathbf{y}). \tag{7.20}$$

**Theorem 7.2** *The function that satisfies assumptions (7.19) is of the following structure:*

$$V(\mathbf{m}(\mathbf{x}), \Sigma(\mathbf{x}), \mathbf{y}^n) = \pi\left(\frac{y_1^n - m_1(\mathbf{x})}{s_1(\mathbf{x})}, \ldots, \frac{y_m^n - m_m(\mathbf{x})}{s_m(\mathbf{x})}\right). \tag{7.21}$$

*Moreover, if assumption (7.20) is satisfied, then $\pi(z_1, \ldots, z_m)$ is an increasing function of its variables.*

*Proof* The proof repeats the main steps of the proof of Theorem 7.1 replacing the operations with scalar variables, where necessary, by the operations with vectors/matrices.

The substitution of $-\mathbf{y}^n$ for $\mathbf{c}$ in the equality

$$V(\mathbf{m}(\mathbf{x}) + \mathbf{c}, \Sigma(\mathbf{x}), \mathbf{y}^n + \mathbf{c}) = V(\mathbf{m}(\mathbf{x}), \Sigma(\mathbf{x}), \mathbf{y}^n)$$

results in

$$V(\mathbf{m}(\mathbf{x}), \Sigma(\mathbf{x}), \mathbf{y}^n) = V(\mathbf{m}(\mathbf{x}) - \mathbf{y}^n, \Sigma(\mathbf{x}), 0). \tag{7.22}$$

The substitution of $\Sigma(\mathbf{x})^{-1}$ for $\mathbf{C}$ in (7.22) and the second equality in (7.11) gives the following equality:

$$V(\mathbf{m}(\mathbf{x}), \Sigma(\mathbf{x}), \mathbf{y}^n) = U\left(\Sigma(\mathbf{x})^{-1} \cdot (\mathbf{m}(\mathbf{x}) - \mathbf{y}^n), \mathbf{I}, 0\right). \tag{7.23}$$

Now, equality (7.21) is obtained simply by denoting $\pi(z_1, \ldots, z_m) = V(-\mathbf{z}, \mathbf{I}, 0)$. Assumption (7.20) obviously implies that $\pi(\mathbf{z})$ is an increasing function of $z_i$.

Theorem 7.2 states that a rational choice of a point for the current computation of objectives is the maximization problem of aggregated objectives $(y_i^n - m_i(\mathbf{x}))/s_i(\mathbf{x})$. Such a conclusion is not surprising since the implementation of optimal, in some sense, single-objective optimization algorithms normally involves the optimization of an auxiliary function which formalizes the concept of optimality. The previously developed multi-objective P-algorithm [252] uses a special case of scalarization, where $\mathbf{P}(\cdot)$ means a probability that the Gaussian random vector $\Xi(\mathbf{x})$ dominates $\mathbf{y}^n$:

$$\mathbf{P}\left(\frac{y_1^n - m_1(\mathbf{x})}{s_1(\mathbf{x})}, \ldots, \frac{y_m^n - m_m(\mathbf{x})}{s_m(\mathbf{x})}\right) = \prod_{i=1}^{r} \Phi\left(\frac{y_i^n - m_i(\mathbf{x})}{s_i(\mathbf{x})}\right),$$

$$\Phi(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-t^2}{2}\right) dt. \tag{7.24}$$

The substantiation of rationality of various scalarizations opens a broad potentiality of the development of multi-objective optimization algorithms based on statistical models of objective functions. However, the investigation of compatibility of a priori information on the properties of objective functions with particular scalarization methods is needed to realize the mentioned potentiality. This algorithm was called the $\pi$-algorithm in [250].

## 7.5 Experimental Assessment

### *7.5.1 Methodological Problems*

Theoretical and algorithmic achievements of multi-objective optimization have implied also the expansion of respective applications. Among applied multi-objective optimization problems, expensive multimodal black-box problems are rather frequent. However, they constitute still a relatively little researched subfield of multi-objective optimization and deserve more attention from researchers. Since the statistical models based single-objective optimization algorithms well correspond to the challenges of single-objective global optimization of expensive black-box functions, they were generalized to the multi-objective case. As shown in the previous sections, the theoretical generalization is rather straightforward. Some experimental investigation was performed to find out how much the generalization corresponds to the expectations of their suitability for multi-objective optimization.

General methodological concepts of testing and comparison of mathematical programming algorithms and software are well developed; see [131]. The methodology, called Competitive Testing in [82], should normally be applied for the comparison of the well-established algorithms. This methodology is also extended for testing and comparison of multi-objective optimization algorithms; see, e.g., [42, 60, 135, 266, 267]. In the case of the well-researched classes of problems (e.g., convex multi-objective optimization), this methodology is universally applicable, only the selection of test functions should be specially selected taking into account the properties of the considered sub-class of problems, e.g., considered in [66, 129]. The tests, based on special cases of real world applied problems, can be very useful for evaluating the efficiency of the respective algorithms; see, e.g., [154] where multi-objective portfolio problems are used for testing the algorithms aimed to distribute solutions uniformly in the Pareto optimal set.

However, the standard testing methodology is not well suitable for the algorithms considered in this chapter. The first difficulty is caused by the main feature of the targeted problems: they are supposed to be expensive. Therefore, a solution, found by an optimization algorithm applied, normally is rather rough. An optimization algorithm is as much useful as much its application aids a decision maker in making a final decision in the conditions of uncertainty reduced because of the application of the algorithm. The quantitative assessment of such a criterion of an algorithm is difficult.

The second difficulty in applying the competitive testing is caused by the novelty of the approach and relatively little researched problems in question. We cite a recognized authority in multi-objective optimization [42, p. 320]: “When a new and innovative methodology is initially discovered for solving a search and optimization problem, a visual description is adequate to demonstrate the working of the proposed methodology. . . . it is important to establish, in the mind of the reader, a picture of the new suggested procedure.” In other words, here the so-called scientific

testing (see [82]) seems most suitable. The main goal of the scientific testing is to check whether the implementation of an algorithm possesses the expected properties.

The efficiency of the single-objective P-algorithm is the result of a rational distribution of observation points: the density of the points is higher in promising subsets of the feasible region. Correspondingly, it can be expected that the statistical models based multi-objective optimization algorithms distribute the observation points more densely closer to the efficient points in the feasible decision region implying a denser distribution of the solutions close to the Pareto front. That property can be considered heuristically by means of visualization as well as by computing some relevant criteria.

In the competitive testing, the termination conditions of the considered algorithms should be similar, aimed at high precision of the found solution. In the case of minimization of expensive black-box functions, application of the termination conditions, defined by convergence criteria, is not relevant since termination normally is conditioned by the computation time limit. Indeed, in real world problems with expensive functions the termination depends on concrete circumstances. Usually the admissible number of computations of objective function values is much smaller than that needed to achieve the convergence criteria used, e.g., for optimization of functions defined by analytical formulas. To define the testing conditions adequate to real world optimizations situations, the termination conditions should be defined by the agreement of researchers and practitioners. In the scientific testing, the termination conditions are appropriate which enable highlighting the properties of an algorithm in question.

In the situations where the competitive testing is not appropriate, nevertheless a comparison of the performance of the considered algorithm with a standard algorithm is desirable. The choice of the latter is problematic. Optimization algorithms, especially heuristic ones, depend on numerous parameters. The values of those parameters usually are tuned to achieve efficiency in some typical situations, most frequently, supposing to find a solution with high precision. However, the parameter values tuned to the latter case can be very different from the values relevant to the optimization of expensive functions where optimization is terminated after relatively small number of computations of function values. The substantiation of relevant values of parameters for the standard algorithm requires experimental investigation which is normally out of scope of the author of the considered algorithm. A possible way out from this difficulty is a choice of the uniform search as a standard algorithm. The latter is parameter-free, and it is optimal in the worst case setting as shown in Section 6.1.

### *7.5.2 Test Functions*

Bi-objective problems with one and two variables were chosen for the experiments to enable visual analysis of the results.

Some experiments were performed using objective functions of a single variable. Experimentation with one-dimensional problems was extensive during the development of the single-objective methods based on statistical models; see, e.g., [139, 208, 216]. These test functions have been used also to demonstrate the performance of the Lipschitz model based algorithms in Section 6.2.5: *Rastr* (6.46), *Fo&Fle* (6.47), and *Schaf* (6.48). The feasible objective regions with the highlighted Pareto front of the considered test functions are shown in Figures 6.5, 6.6, and 6.7.

Two bi-objective test problems of two variables are chosen for the experimentation. The test problems of two variables are chosen similarly to the choice of one-dimensional problems: the first multi-objective test problem is composed using a typical test problem for a single-objective global optimization, and the second one is chosen from the set of functions frequently used for testing multi-objective algorithms. The first test function *Shek* (1.6) is composed of two Shekel functions which are frequently used for testing global optimization algorithms, see, e.g., [216]. A rather simple multimodal case is intended to be considered, so the number of minimizers of both objectives is selected equal to two. The objective functions are represented by contour lines in Figure 1.3. The second problem *Fo&Fle*, (1.5), is especially difficult from the point of view of global minimization, since the functions $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ in (1.5) are similar to the most difficult objective function whose response surface is comprised of a flat plateau over a large part of the feasible decision region, and of the unknown number of sharp spikes. The estimates of parameters of the statistical model of (1.5) are biased towards the values that represent the "flat" part of response surface. The discrepancy between the statistical model and the modeled functions can negatively influence the efficiency of the statistical models based algorithms.

The selection of test problems can be summarized as follows: two problems, (6.46) and (1.6), are constructed generalizing typical test problems of global optimization, and two other problems, (6.48) and (1.5), are selected from a set of non-convex multi-objective test problems. The former problems are well represented by the considered above statistical model, and the latter ones are not. Both objective functions of problem (1.5) are especially difficult for global optimization, and their properties do not correspond to the properties predictable using the statistical model.

### *7.5.3 Experiments with the P-Algorithm*

The MATLAB implementations of the considered algorithms were used for experimentation. The results of minimization obtained by the uniform random search are presented for the comparison. Minimization was stopped after 100 computations of objective function values.

The sites for the first 50 computations of $\mathbf{f}(\mathbf{x})$ were chosen by the P-algorithm randomly with a uniform distribution over the feasible region. That data were used to

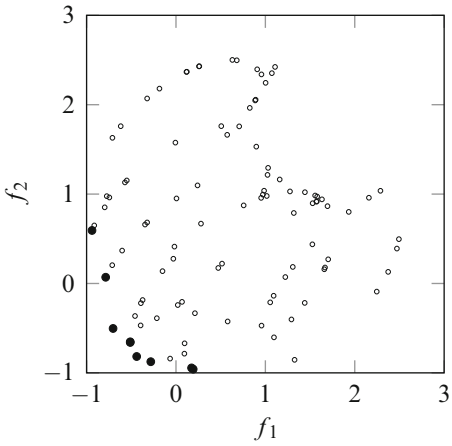


**Fig. 7.1** The points in the feasible objective region for problem *Rastr* (6.46) generated by the Monte Carlo method; non-dominated solutions are denoted by *thicker points*

estimate the parameters of the statistical model as well as in planning of the next 50 observations according to (7.8). The maximization of the improvement probability was performed by a simple version of multistart. The values of the improvement probability (7.9) were computed at 1000 points, generated randomly with uniform distribution over the feasible region. A local descent was performed from the best point using the codes from the MATLAB Optimization Toolbox.

Let us start from the comments about the results of minimization of the functions of one variable. In the experiments with objective functions of a single variable, the algorithms can be assessed with respect to the solutions found in the true Pareto front, while in the case of several variables normally the approximate solutions are solely available for the assessment of the algorithms.

The feasible objective region of problem (6.46) is presented for the visual analysis in Figure 6.5. The (one hundred) points in the feasible objective region, generated by the method of random uniform search (RUS), are shown in Figure 7.1; the non-dominated solutions found (*thicker points*) do not represent the Pareto front well. The P-algorithm was applied to that problem with two values of the threshold vector. In Figure 7.2, the trial points in the feasible objective region are shown, and non-dominated points are denoted by *thicker points*. The left-side figure shows the results obtained with the threshold vector equal to $(-1,-1)^T$ (35 non-dominated points found), and the right-hand side figure shows the results obtained with the threshold vector equal to $(-0.75,-0.75)^T$ (51 non-dominated points found). In the first case, the threshold is the ideal point; that case is similar to the case of a single-objective minimization, where the threshold is considerably below the current record. Presumably, for such a case the globality of the search strategy prevails and implies the uniformity (over the Pareto front) of the distribution of the non-dominated solutions found. For the threshold closer to the Pareto front, some localization of observations can be expected in the sense of increased density of the non-dominated points closer to the threshold vector. Figure 7.2 illustrates the realization of the hypothesized properties.

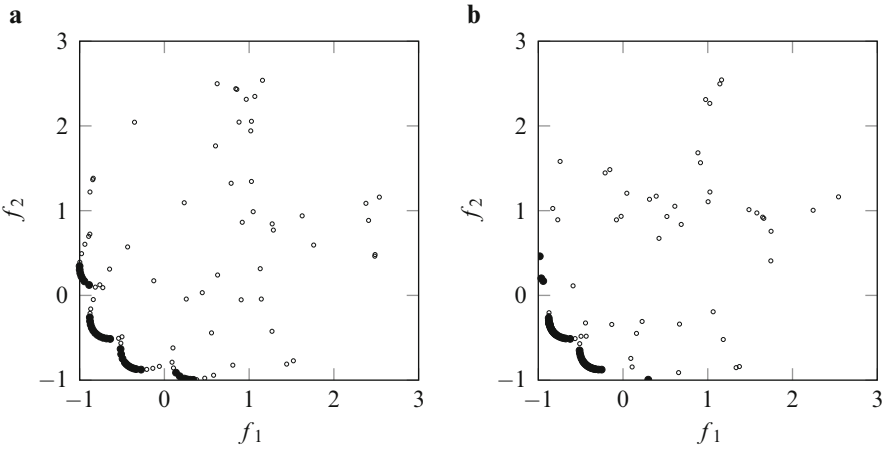


**Fig. 7.2** The points generated by the P-algorithm in the feasible objective region for problem *Rastr* (6.46); non-dominated solutions are denoted by *thicker points*, (**a**) the case of search with the threshold vector equal to $(-1,-1)^T$, and (**b**) the case of search with the threshold vector equal to $(-0.75,-0.75)^T$

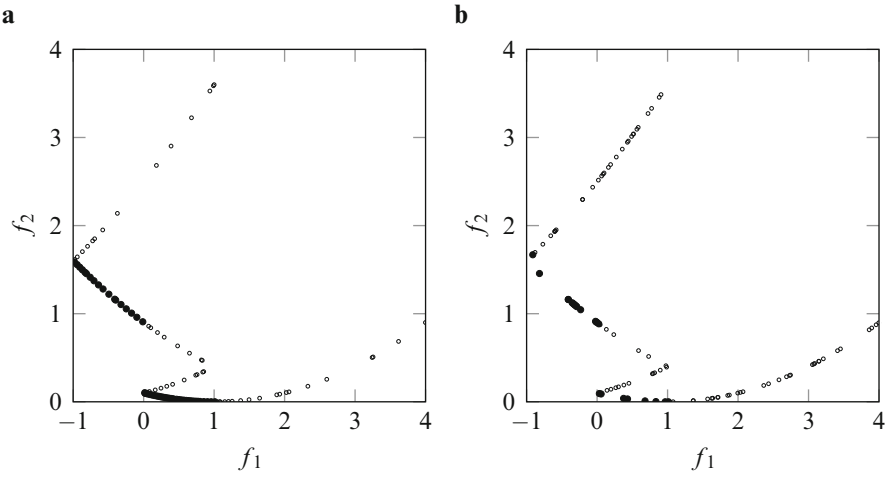


**Fig. 7.3** The points generated in the objective space for the problem *Schaf* (6.48), non-dominated solutions are denoted by *thicker points*, (**a**) by the P-algorithm (with the threshold vector equal to $(-1,0)^T$), (**b**) the points generated by RUS

Since test problem (6.48) is frequently used in experimental assessment of multi-objective algorithms, we will not comment its properties. The results of an experiment with test problem (6.48) are presented in Figure 7.3 for the visual analysis. The search strategy of the P-algorithm, using the ideal vector for the threshold, seems similar to that for problem (6.46).

Several metrics have been proposed in recent publications for the comparison of multi-objective algorithms; for the comprehensive list and discussion, we refer to [42]. Generally speaking, it is aimed to assess the quality of approximations of the Pareto front and the efficiency of algorithms, used to compute these approximations. Here, we consider only the approximation quality. The performance of the RUS is routinely assessed by statistical methods; the mean values and standard deviations of the considered metrics are estimated and presented below. The performance of the P-algorithm is assessed similarly. Recall that the first 50 observation points by the P-algorithm are selected randomly with a uniform distribution over $\mathbf{A}$; therefore, each optimization process is different, and the results are random.

Both algorithms (P-algorithm and RUS) have been run for 1000 times for test problems (6.46) and (6.48). The minimization was stopped after 100 observations, and the following performance criteria were computed: NN—the number of non-dominated points, NP—the number of points in the true Pareto front, MD—the maximum "hole" length in the Pareto front, and DP—the maximum distance between the found non-dominated solutions outside the Pareto front and the Pareto front; in all cases, the distances correspond to the Euclidean norm. MD is computed as the maximum of two maxima: the maximum distance between the nearest neighboring solutions found in the Pareto front, and the maximum distance between the limit points of the Pareto front and their nearest neighbors in the set of the found Pareto solutions. NP shows how many observations were successful in the sense of hitting the Pareto front. The difference between NN and NP gives the estimate of the number of "wrong" solutions, i.e., the number of non-dominated solutions outside the Pareto front. The so-called error ratio (NN-NP)/NN is a frequently used metric for the assessment of multi-objective algorithms. The metric MD characterizes the spread of solutions over the Pareto front very well. The metric DP characterizes error in evaluating the Pareto front implied by the "wrong" solutions. The mean values and standard deviations of the listed above metrics are presented in Table 7.1.

The experiments with the test problems, where objective functions depend on two decision variables, are performed in a similar way as with the one-variable tests above. The feasible objective region and the Pareto front of problem *Shekel* (1.6) are shown in Figure 1.3. The feasible objective region is an image of mesh points of the $101 \times 101$ rectangular Cartesian grid in the feasible decision region. While the points in the feasible decision region are distributed uniformly, solutions in the feasible objective region are distributed not uniformly at all. From the density of

**Table 7.1** The mean values and standard deviations of the performance criteria of the P-algorithm and the RUS for problems (6.46) and (6.48)

| Algorithm | P-algorithm | | | | RUS | | | |
|---|---|---|---|---|---|---|---|---|
| Problem | Problem (6.46) | | Problem (6.48) | | Problem (6.46) | | Problem (6.48) | |
| NP | 31.4 | 2.3 | 41.9 | 3.0 | 6.9 | 2.5 | 22.4 | 4.2 |
| NN | 34.3 | 2.4 | 43.5 | 3.0 | 9.9 | 2.3 | 24.3 | 4.1 |
| MD | 0.095 | 0.011 | 0.099 | 0.015 | 0.29 | 0.05 | 0.36 | 0.11 |
| DP | 0.038 | 0.014 | 0.013 | 0.009 | 0.18 | 0.18 | 0.06 | 0.06 |

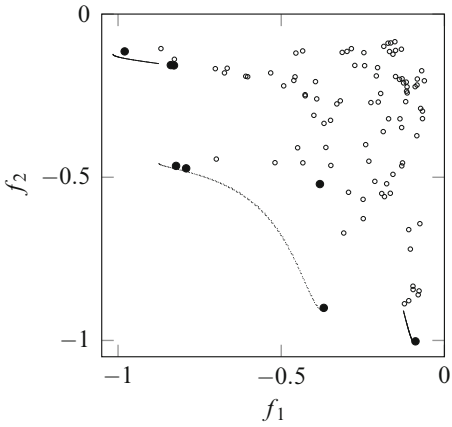


**Fig. 7.4** The points generated by the RUS for the problem *Shekel* (1.6); non-dominated solutions are denoted by *thicker points*

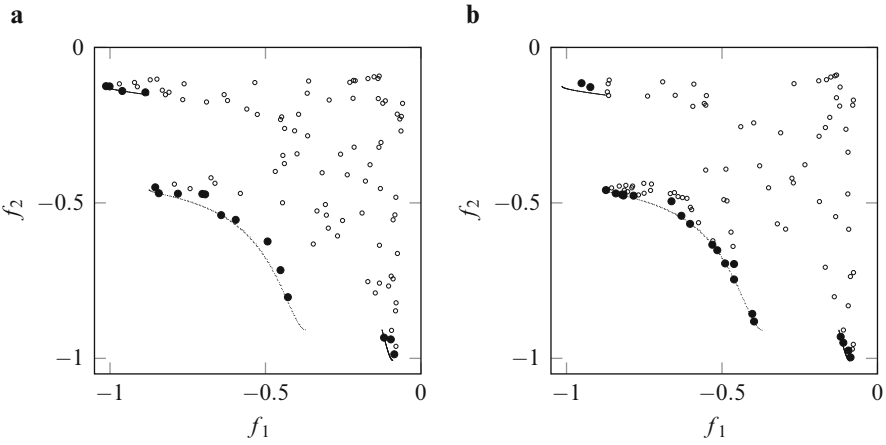


**Fig. 7.5** The points generated by the P-algorithm in the objective feasible region of problem *Shekel* (1.6), non-dominated solutions are denoted by *thicker points*, the *line* indicates the Pareto front, the threshold is equal to (**a**) $(-1, -1)^T$, (**b**) $(-0.6, -0.6)^T$

points over various subsets of the feasible objective region, it can be seen that a relatively small number of points are in a close vicinity of the Pareto front. The solutions generated by the RUS are presented in Figure 7.4; the figure corroborates the conclusion which might be drawn from Figure 1.3: a large number of uniformly distributed points in the feasible decision region is needed to get at least the right impression about the form of the south-west part of the feasible objective region, yet more to get an acceptable approximation of the Pareto optimal front.

Figure 7.5 shows the approximations found by the P-algorithm; the left side corresponds to the threshold vector $(-1, -1)^T$, and the right side corresponds to the threshold $(-0.6, -0.6)^T$. The threshold vector $(-1, -1)^T$ is relatively far from the Pareto front and corresponds to the ideal vector. It can be expected that, in this

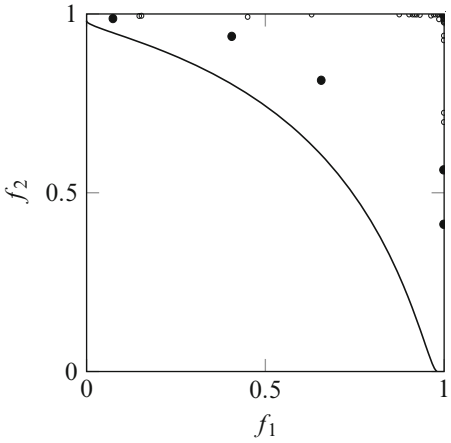


**Fig. 7.6** The points generated by the RUS for the problem *Fo&Fle* (1.5); non-dominated solutions are denoted by *thicker points*, and the *line* indicates the Pareto front

case, the non-dominated solutions will be nearly uniformly distributed close to the Pareto front. The left-hand side of Figure 7.5 corroborates the expectation and gives rather a good general impression about the feasible objective region. The threshold $(-0.6,\ -0.6)^T$ is relatively close to the Pareto front indicating the Pareto solutions, nearest to the threshold, as the focus of interest. Indeed, the non-dominated solutions found are distributed more densely close to the central part of the Pareto front. The approximation of a favorable subset of the Pareto front (like, e.g., shown on the right side of Figure 7.5) in many applications is even more important than the approximation of the whole front. However, the quantitative assessment of such approximations is problematic because of ambiguity in definition of the favorable subset.

Similar experiments have been done with the problem (1.5). The results, obtained by the RUS, and presented in Figure 7.6, illustrate the property of the problem commented in Section 7.5.2: the values of both objective functions are close to one over the large subset of **A**. Correspondingly, the generated in the feasible objective region points are crowded in the vicinity of $(1,\ 1)^T$.

The non-dominated points found by the P-algorithm, in the case of the threshold equal to the ideal vector, $(0,\ 0)$, are distributed similarly as expected, i.e., the distribution is not significantly different from the uniform one (Figure 7.7). But in the case of the threshold vector $(0.6,\ 0.6)^T$, the increase of the density of non-dominated solutions close to the center of the Pareto front does not actualize. Contrary to the expectations, increase of the density is seen near to the limit points of the Pareto front (Figure 7.7). Such a localization can be explained by the estimates of the parameters of the statistical model obtained using the values of the objective functions. The estimates of the mean values are much closer to 1 than to 0, and the estimates of the variances are relatively small; the average of 1000 values of the estimates computed using 50 observations have been obtained as follows: $\bar{\mu}_1 = 0.9496$, $\bar{\mu}_2 = 0.9512$, $\bar{\sigma}_1^2 = 0.0269$, $\bar{\sigma}_2^2 = 0.0339$. With respect to such

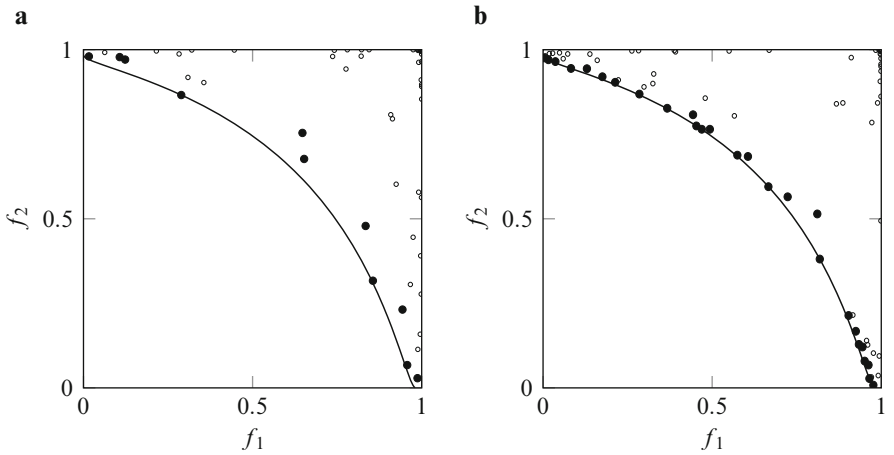


**Fig. 7.7** The points generated by the P-algorithm in the objective feasible region of problem *Fo&Fle* (1.5), non-dominated solutions are denoted by *thicker points*, the *line* indicates the Pareto front, the threshold is equal to (**a**) $(0, 0)^T$, (**b**) $(0.6,\ 0.6)^T$

estimates, the values of the objective functions close to the limit points of the Pareto front seem as incredibly good and attract further observations in their vicinity.

The metrics used for the quantitative assessment of precision of the Pareto front approximation in the case of one-dimensional objective functions are, except of NN, not suitable in the case of larger dimensionality of the decision space. In the former case, points really belonging to the Pareto front are used for the approximation, and in the latter case are used non-dominated points, normally not belonging to the Pareto front. To estimate, how close to the Pareto front are the found non-dominated points, the generational distance (GD) is used [42]. GD is computed as the maximum of distances between the found non-dominated solutions and their closest neighbors from the Pareto front. The epsilon indicator (EI) is a metric suggested in [267] which integrates measures of the approximation precision and spread: it is the min max distance between the Pareto front and the set of the found non-dominated solutions

$$\mathrm{EI} = \max_{1 \le i \le K} \min_{1 \le j \le N} ||Z_i - F_j||, \tag{7.25}$$

where $F_j$ are the non-dominated solutions found by the considered algorithm, and $\{Z_i,\ i = 1, \ldots, K\}$ is the set of points well representing the Pareto front, i.e., $Z_i$ are sufficiently densely and uniformly distributed over the Pareto front.

The mean values and standard deviations of the considered metrics in case of the solution of (1.6) by the P-algorithm (the threshold vector equal to $(-1,\ -1)^T$), and by RUS are presented in Table 7.2. The statistics of the approximation metrics for problem (1.5) are also presented in Table 7.2; the P-algorithm has been run with the threshold vector equal to (0, 0).

**Table 7.2** The mean values and standard deviations of the performance criteria of the P-algorithm and the random uniform method for problems (1.6) and (1.5)

| Algorithm | P-algorithm | | | | RUS | | | |
|---|---|---|---|---|---|---|---|---|
| Problem | Problem (1.6) | | Problem (1.5) | | Problem (1.6) | | Problem (1.5) | |
| NN | 15.7 | 2.0 | 9.87 | 1.4 | 12.1 | 3.0 | 6.6 | 1.6 |
| GD | 0.070 | 0.051 | 0.015 | 0.0061 | 0.23 | 0.081 | 0.11 | 0.068 |
| EI | 0.13 | 0.053 | 0.20 | 0.034 | 0.26 | 0.056 | 0.33 | 0.097 |

To assess how many observations are needed to RUS to achieve the average values of the metrics EI and GD reached by the P-algorithm after 100 observations, RUS has been tested with the various numbers of maximum trials in the termination condition. It has been shown that RUS, while solving problem (1.6), needs about 600 observations to reach average values of EI (0.1316) and GD (0.0959), similar to those (see Table 7.2) of the P-algorithm after 100 observations. The averaged values of EI and GD, equal to 0.2331 and 0.0810 correspondingly, are reached by the RUS in minimization of (1.5) terminated after 200 observations; in that case, Monte Carlo method needs only twice as many observations to match up the P-algorithm with respect to EI and GD.

### *7.5.4 Experiments with the π-Algorithm*

A version of the bi-objective $\pi$-algorithm has been implemented as described in Section 7.4.2. A product of two arctangents was used for $\pi(\cdot)$. Then the $n+1$ step of the $\pi$-algorithm is defined as the following optimization problem

$$\mathbf{x}_{n+1} = \arg \max_{\mathbf{x}\in\mathbf{A}} \arctan\left(\frac{y_1^n - m_1(\mathbf{x})}{s_1(\mathbf{x})} + \frac{\pi}{2}\right) \cdot \arctan\left(\frac{y_2^n - m_2(\mathbf{x})}{s_2(\mathbf{x})} + \frac{\pi}{2}\right), \tag{7.26}$$

where the information collected at previous steps is taken into account when computing $m_i(\mathbf{x}) = m_i(\mathbf{x}|\mathbf{x}_j, \mathbf{y}_j, j = 1, \ldots, n)$ and $s_i(\mathbf{x}) = s_i(\mathbf{x}|\mathbf{x}_j, \mathbf{y}_j, j = 1, \ldots, n)$. The maximization in (7.26) was performed by a simple version of multistart: from the best of 1000 points, generated randomly with uniform distribution over the feasible region, a local descent was performed using the codes from the MATLAB Optimization Toolbox. By this implementation, we wanted to check whether the function $\arctan(\cdot) \cdot \arctan(\cdot)$ chosen rather arbitrarily could be as good as the Gaussian cumulative distribution function for constructing statistical model-based multi-objective optimization algorithms. The experimentation with this version of the algorithm can be helpful also in selecting the most appropriate statistical model for a further development where two alternatives seem competitive: a Gaussian random field versus a statistical model, based on the assumptions of subjective probability [216].

Some experiments have been done for the comparison of the $\pi$-algorithm with the multi-objective P-algorithm described in Section 7.3. The optimization results by RUS from Section 7.5.3 are included to highlight the properties of the selected test problems. The results obtained by a multi-objective genetic algorithm (the MATLAB implementation in [80]) are also provided for the comparison. Two examples are presented and commented; we think that extensive competitive testing would be premature, as argued in Section 7.5.1.

Since the considered approach is oriented to expensive problems, we are interested in the quality of the result obtained computing a modest number of the values of objectives. Following the concept of experimentation above, a termination condition of all the considered algorithms was defined by the maximum number of computations of the objective function values, equal to 100. The parameters of the statistical model, needed by the $\pi$-algorithm, have been estimated using a sample of $\mathbf{f}(\mathbf{x})$ values, chosen similarly to the experiments with the P-algorithm: the sites for the first 50 computations of $\mathbf{f}(\mathbf{x})$ were chosen randomly with a uniform distribution over the feasible region; the obtained data were used not only for estimating parameters but also in planning of the next 50 observations according to (7.26).

An important parameter of the $\pi$-algorithm is $\mathbf{y}^n$. The vector $\mathbf{y}^n$ should be not dominated by the known values $\mathbf{y}_1, \ldots, \mathbf{y}_n$. A heuristic recommendation is to select $\mathbf{y}^n$ at a possibly symmetric site with respect to the global minima of objectives. We have selected the values of $\mathbf{y}^n$ used in the P-algorithm above: $\mathbf{y}^n = (-0.6, -0.6)$ in the case of problem (1.6), and $\mathbf{y}^n = (0.6, 0.6)$ in the case of problem (1.5). Typical results are illustrated in Figure 7.8.

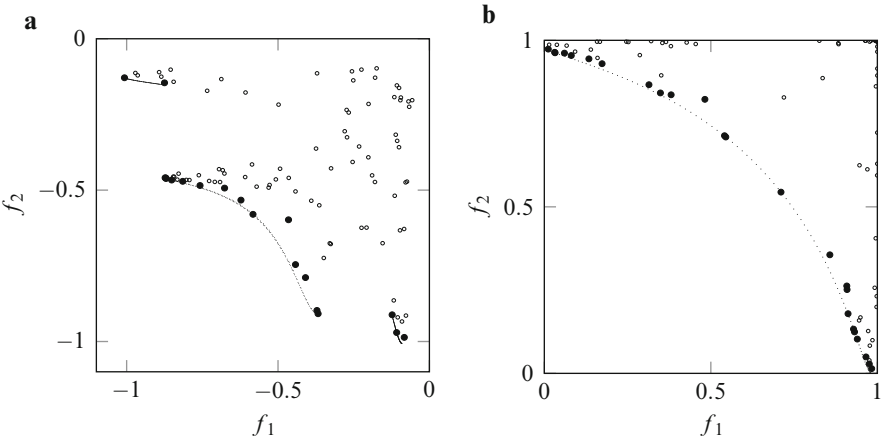


**Fig. 7.8** The points generated by the $\pi$-algorithm in the objective feasible region, non-dominated solutions are denoted by *thicker points*, a *line* indicates the Pareto front, (**a**) problem *Shekel* (1.6), (**b**) problem *Fo&Fle* (1.5)

**Table 7.3** The mean values and standard deviations of the performance criteria of the considered algorithms for problems (1.6) and (1.5)

| Algorithm | $\pi$-Algorithm | | | | P-algorithm | | | |
|---|---|---|---|---|---|---|---|---|
| Problem | Problem (1.6) | | Problem (1.5) | | Problem (1.6) | | Problem (1.5) | |
| NN | 20.2 | 3.10 | 19.6 | 2.4 | 15.7 | 2.0 | 9.87 | 1.4 |
| GD | 0.044 | 0.027 | 0.055 | 0.028 | 0.070 | 0.051 | 0.015 | 0.0061 |
| EI | 0.13 | 0.074 | 0.14 | 0.047 | 0.13 | 0.053 | 0.20 | 0.034 |
| Algorithm | GA | | | | RUS | | | |
| Problem | Problem (1.6) | | Problem (1.5) | | Problem (1.6) | | Problem (1.5) | |
| NN | 9.3 | 2.5 | 8.2 | 2.2 | 12.1 | 3.0 | 6.6 | 1.6 |
| GD | 0.30 | 0.079 | 0.18 | 0.098 | 0.23 | 0.081 | 0.11 | 0.068 |
| EI | 0.38 | 0.078 | 0.55 | 0.21 | 0.26 | 0.056 | 0.33 | 0.097 |

The following metrics, used for the quantitative assessment of the precision of Pareto front approximation, have been evaluated: the number of non-dominated solutions found (NN), the generational distance (GD), and the epsilon indicator (EI). The mean values and standard deviations of the considered metrics are presented in Table 7.3. Besides of the results of experimentation with the $\pi$-algorithm, the results of the P-algorithm, the uniform random search, and the genetic algorithm are presented for the comparison. Since all the considered algorithms are randomized, the statistical estimates of the considered metrics are presented.

Almost all the results of the $\pi$-algorithm are somewhat better than those of the P-algorithm. The only exception is GD in the case of problem (1.5). However, the isolated comparison of two numbers here is insufficient, since NN of the $\pi$-algorithm in this case is double of that of the P-algorithm. The experimental results corroborate the acceptability of the departure from the Gaussian model.

In the second half of Table 7.3, similar data of the experiments with the RUS and the genetic algorithm (GA) is presented for the comparison. As mentioned above, functions (1.6) and (1.5) are somewhat similar to the worst case Lipschitz objective functions. Therefore, it is interesting to assess the performance of the worst case optimal algorithm for these problems. RUS is a randomized approximation of the optimal worst case algorithm which computes the values of the objective functions at the centers of balls optimally covering the feasible region; see Chapter 6. The results of experiments with RUS, where the search was stopped after 100 computations of the objectives, are included into Table 7.3. The performance of RUS is obviously worse than that of the $\pi$-algorithm and of the P-algorithm. In Section 7.5.3, it is also reported how many computations are needed to RUS to achieve the average values of the metrics EI and GD comparable with those by the P-algorithm after 100 observations. The RUS, while solving problem (1.6), needs about 600 observations to reach average values of EI (0.13) and GD (0.096), which are close to those of the P-algorithm presented in the Table 7.2. The averaged values of EI and GD, equal to 0.2331 and 0.0810 correspondingly, are reached by the RUS

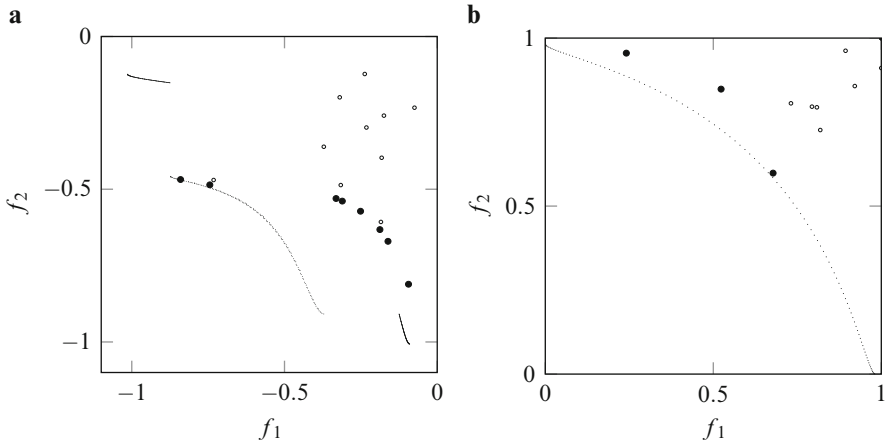


**Fig. 7.9** The solutions generated by GA at the last (fifth) iteration, non-dominated solutions are denoted by *thicker points*, a *line* indicates the Pareto front, (**a**) problem *Shekel* (1.6), (**b**) problem *Fo&Fle* (1.5)

for (1.5) after 200 observations; in that case, the RUS needs only twice as many observations to match up the P-algorithm with respect to EI and GD. As it could be expected from the theory in Chapter 6, the relative performance of the uniform search was better in the case where the objective functions are more similar to the worst case ones.

The experiments with GA are reported to illustrate the hardness of the considered black-box multi-objective expensive global optimization problems, by demonstrating that the conventional algorithms are inappropriate here. The "Pareto Genetic Algorithm" from the book oriented to practical applications [80] was chosen for the experimentation. The intrinsic parameters of GA have been selected as recommended by the authors in [80]. The problem relevant parameters, the population size, and the number of iterations were chosen equal to 20 and 5 correspondingly, taking into account the termination condition of the other algorithms considered in our investigation. The values of metrics presented in the Table 7.3 are obtained as averages of the 200 independent runs of the algorithm. The typical results presented in Figure 7.9 illustrate large values of EI and GD.

Even with 10 times larger number of computations of the objective functions values, where the population size was equal to 100 and the number of iterations was equal to 10, the average values of the considered metrics for GA were worse than those of the $\pi$-algorithm in Table 7.3. For the problem (1.6), the average values of EI and GD were 0.27 and 0.23 correspondingly; for the problem (1.5), those values were 0.29 and 0.10 correspondingly.

## 7.6 Discussion and Remarks

The approach based on statistical models of objective functions facilitates construction of a family of multi-objective optimization algorithms, similar in some sense to the single-objective P-algorithm. The constructed multi-objective algorithms are aimed at the problems with multimodal, black-box, and expensive objectives. The proposed multi-objective algorithms can be implemented as a rather simple modification of its predecessor—the single-objective algorithms, and possess some advantages of the latter. The results of experimentation with small size test problems are promising. The presented results of the preliminary numerical experiments corroborate the theoretical conjecture that the proposed approach is favorable in the field of black-box expensive global optimization.

# Chapter 8
# Probabilistic Bounds in Multi-Objective Optimization

## 8.1 Introduction

Randomization is one of the most important ideas used in the construction of heuristic methods for multi-objective optimization. Mathematically substantiated stochastic methods for non-convex multi-objective optimization attracted interest of researchers quite recently. A natural idea is to generalize single-objective optimization methods to the multi-objective case, and a prospective candidate is the well-developed method based on the statistical methods of extremal values. A brief review of this approach, known as a branch and probability bound (BPB) method is presented in Section 4.4. Some statistical procedures which are well known in single-objective optimization can be extended to multi-objective problems using scalarization. In this chapter, a version of multi-objective BPB method is described and discussed; this method is an extension of the BPB method developed for the case of a single-objective function in [237], see also [239, Section 2.6.1]. The considered extension is based on the Tchebycheff scalarization method briefly discussed in Chapter 2.

Let us recall that by means of the Tchebycheff scalarization a multi-objective optimization problem

$$\min_{\mathbf{x}} \in \mathbf{A}\mathbf{f}(\mathbf{x}),\ \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(x), \ldots, f_m(\mathbf{x}))^T,\ \mathbf{A} \subset \mathbb{R}^d, \tag{8.1}$$

is reduced to the single-objective optimization problem where the objective function is defined using the following formula:

$$g(\mathbf{x}, \mathbf{w}) = \max_{i=1,\ldots,m} w_i(f_i(\mathbf{x}) - u_i), \tag{8.2}$$

$$w_i \geq 0 \quad \text{for} \quad i = 1, \ldots, m \quad \text{and} \quad \sum_{i=1}^{m} w_i = 1\,, \tag{8.3}$$



P.M. Pardalos et al., *Non-Convex Multi-Objective Optimization*, Springer Optimization and Its Applications 123, DOI 10.1007/978-3-319-61007-8_8

and $\mathbf{u}$ is a so-called utopian vector

$$u_i < \min f_i(\mathbf{x}) \text{ for } i = 1, \ldots, m.$$

The minimizer of (8.2) is a weakly Pareto optimal decision of the respective multi-objective optimization problem. Similarly, by the modified versions of the Tchebycheff scalarization, defined by (2.5) and (2.6), exclusively the Pareto optimal decisions can be found.

More precisely this can be formulated as follows [135]. Let $\mathbf{y} \in \mathbf{P}(\mathbf{f})_O$ be any Pareto optimal objective vector, then it corresponds to a minimum of an objective function (2.5) defined by the modified Tchebycheff scalarization. The parameters of the objective function (2.5) should satisfy the following requirements: $\rho \geq 0$ should be sufficiently small, and weights should be defined by the following formulas: $w_i = \beta/(y_i - u_i)$, $i = 1, \ldots, m$, where $\mathbf{u}$ is a utopian vector. On the other hand, the set $\{\mathbf{f}(\mathbf{x}_\mathbf{w}) : \ w_i \geq 0, \ \sum_{i=1}^m w_i = 1\}$ is coincident with $\mathbf{P}(\mathbf{f})_O$, where $\mathbf{x}_\mathbf{w}$ is an optimal decision of the scalarized problem corresponding to the vector of weights $\mathbf{w}$. Let us recall that when the original Tchebycheff scalarization (8.2) is used, the weakly Pareto optimal solutions can be obtained besides of Pareto optimal solutions.

## 8.2 Statistical Inference About the Minimum of a Function

Let $f(\mathbf{x})$ be a function given on a feasible region $\mathbf{A}$ and let $\mathbf{x}_1, \ldots, \mathbf{x}_n$, be identically distributed points in $\mathbf{A}$. Note that in the present section the estimation of the minimum of a scalar-valued function is considered. Further, we provide estimates for $\mathsf{m} = \min_{\mathbf{x}\in\mathbf{A}} f(\mathbf{x})$, and show how to construct confidence intervals for $\mathsf{m}$.

From the sample $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ we pass to the sample $\mathbf{Y} = \{y_1, \ldots, y_n\}$ consisting of the values $y_j = f(\mathbf{x}_j)$ of the objective function $f(\mathbf{x})$ at the points $\mathbf{x}_j$ $(j = 1, \ldots, n)$. The sample $\mathbf{Y}$ is independent, and its underlying cumulative distribution function is given by

$$G(t) = \Pr\{\mathbf{x} \in \mathbf{A} \ : \ f(\mathbf{x}) < t\} = \int_{f(\mathbf{x})<t} P(d\mathbf{x}) \,, \tag{8.4}$$

where $P(\cdot)$ is the distribution of points $\mathbf{x}_j \in \mathbf{A}$.

Denote by $\eta$ a random variable which has cumulative distribution function $G(t)$, and by $y_{1,n} \leq \ldots \leq y_{n,n}$ the order statistics corresponding to the sample $\mathbf{Y}$. The parameter $\mathsf{m} = \min_{\mathbf{x}\in\mathbf{A}} f(\mathbf{x})$ is at the same time the lower endpoint of the random variable $\eta$, i.e., $\mathsf{m} = \operatorname{ess\,inf} \eta$. That is, $\mathsf{m}$ is such that $G(\mathsf{m}) = 0$ and $G(\mathsf{m} + \varepsilon) > 0$ for any $\varepsilon > 0$.

For a very wide class of functions $f(\mathbf{x})$ and distributions $P$, the cumulative distribution function $G(\cdot)$ can be shown to have the following representation for $t \simeq \mathsf{m}$:

$$G(t) = c(t - \mathsf{m})^\alpha + \mathrm{o}((t - \mathsf{m})^\alpha), \ t \downarrow \mathsf{m}. \tag{8.5}$$

This representation is valid for some positive constants $c$ and $\alpha$; more generally, $c = c(t)$ is a slowly varying function for $t \simeq \mathsf{m}$ but the results cited below are also valid for this slightly more general case. The value of $c$ is irrelevant but the value of $\alpha$, which is called "tail index," is important. We shall assume that the value of $\alpha$ is known. As discussed below this can always be considered true in the algorithms we consider.

Several good estimates of $\mathsf{m}$ are known for given $\alpha$, see [239, Section 2.4]. We shall use one of them, the optimal linear estimator based on the use of $k$ order statistics. This estimator has the form

$$\widehat{\mathsf{m}}_{n,k} = c \sum_{i=1}^{k} [u_i / \Gamma(i + 2/\alpha)]\, y_{i,n}\,, \tag{8.6}$$

where $\Gamma(\cdot)$ is the Gamma-function,

$$u_i = \begin{cases} (\alpha + 1)\,, & \text{for } i = 1\,, \\ (\alpha - 1)\Gamma(i), & \text{for } i = 1, \dots, k-1\,, \\ (\alpha - \alpha k - 1)\Gamma(k)\,, & \text{for } i = k\,, \end{cases}$$

$$1/c = \begin{cases} \sum_{i=1}^{k} 1/i\,, & \text{for } \alpha = 2, \\ \frac{1}{\alpha - 2}\left(\alpha\Gamma(k+1)/\Gamma(k+2/\alpha) - 2/\Gamma(1+2/\alpha)\right), & \text{for } \alpha \neq 2\,. \end{cases}$$

Under the assumption (8.5), for given $k$ and $\alpha$, the estimator $\widehat{\mathsf{m}}_{n,k}$ is consistent, asymptotically (as $n \to \infty$) unbiased, and $E(\widehat{\mathsf{m}}_{n,k} - \mathsf{m})^2$, its asymptotic mean squared error, has maximum possible rate of decrease in the class of all consistent estimates including the maximum likelihood estimator of $\mathsf{m}$, see [239].

The estimation procedure outlined above can be generalized to the case when the original sample $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ is stratified rather than independent. Assume that the distribution $P(\cdot)$ is uniform on $\mathbf{A}$ and the set $\mathbf{A}$ is split into $m$ subsets of equal volume. Assume also that in each subset we generate $l$ independent uniformly distributed points. The sample size is then $n = ml$. Under the assumption $l > k$, exactly the same estimator $\widehat{\mathsf{m}}_{n,k}$ can again be used, see [238]. The accuracy of this estimator is slightly better than the accuracy of the same estimator for the case of independent sample, see [239, Section 3.2].

If the sample $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ is independent, then the following confidence interval for $\mathsf{m}$ has asymptotic (as $n \to \infty$) confidence level $1 - \delta$:

$$[y_{1,n} - (y_{k,n} - y_{1,n})/c_{k,\delta}, y_{1,n}]\,, \quad \text{where} \quad c_{k,\delta} = \left[1 - (1-\delta)^{1/k}\right]^{-1/\alpha} - 1\,. \tag{8.7}$$

## 8.3 Conditions on the Intersection of Pareto Fronts

Now we return to the analysis of the multi-objective optimization problem (8.1). For any $\mathbf{S} \subseteq \mathbf{A}$, define the function

$$L_{\mathbf{S}}(\mathbf{w}) = \min_{\mathbf{x}\in\mathbf{S}} g(\mathbf{x};\mathbf{w}) = \min_{\mathbf{x}\in\mathbf{S}} \max_{i=1,\dots,m} \{w_i(f_i(\mathbf{x}) - u_i)\};$$

this function is defined on the set of weight vectors $\mathbf{w} = (w_1,\dots,w_m)$ satisfying (8.3). To specify the Pareto front corresponding to a subset of the decision space $\mathbf{S}$ we will use the notation $\mathbf{P}(\mathbf{f},\mathbf{S})_O$.

**Lemma 8.1** *Let $\mathbf{y}^* = (y_1^*,\dots,y_m^*) \in \mathbf{P}(\mathbf{f},\mathbf{A})_O$ be any Pareto optimal objective vector, $\mathbf{u} = (u_1,\dots,u_m)$ be any utopian vector, and $\rho \geq 0$ be sufficiently small. Define the vector of weights $\mathbf{w} = (w_1,\dots,w_m)$ by $w_i = \beta/(y_i^* - u_i)$ for $i = 1,\dots,m$, where $\beta$ is the normalizing factor providing the equality $\sum_{i=1}^m w_i = 1$. Then*

$$\mathbf{y}^* = \mathbf{f}(\mathbf{x}^*_{\mathbf{w},\rho}),$$

*where*

$$\mathbf{x}^*_{\mathbf{w},\rho} = \arg\min_{\mathbf{x}\in\mathbf{A}} g(\mathbf{x},\mathbf{w},\rho), \tag{8.8}$$

$$g(\mathbf{x},\mathbf{w},\rho) = \max_{i=1,\dots,m} [w_i(f_i(\mathbf{x}) - u_i)] + \rho \sum_{i=1}^m (f_i(\mathbf{x}) - u_i). \tag{8.9}$$

*Moreover, if $\rho > 0$, then each vector $\mathbf{f}(\mathbf{x}^*_{\mathbf{w},\rho})$ belongs to the Pareto front $\mathbf{P}(\mathbf{f},\mathbf{A})_O$ for an arbitrary vector of weight $\mathbf{w}$.*

Lemma 8.1 is a combination of Theorems 3.4.5 and 3.4.6 in [135], see also [98]. This Lemma gives a complete characterization of the Pareto front $\mathbf{P}(\mathbf{f},\mathbf{A})_O$. If $\rho = 0$, then the set of $\mathbf{f}(\mathbf{x}^*_{\mathbf{w},0})$ can include weakly Pareto optimal solution corresponding to some $\mathbf{w}$, where $\mathbf{x}^*_{\mathbf{w},0}$ is a minimizer of $g(\mathbf{x},\mathbf{w},0)$.

**Theorem 8.1** *Let $\mathbf{S}$ be a subset of $\mathbf{A}$ such that $L_{\mathbf{S}}(\mathbf{w}) > L_{\mathbf{A}}(\mathbf{w})$ for all weight vectors $\mathbf{w} = (w_1,\dots,w_m)$. Then the Pareto front $\mathbf{P}(\mathbf{f},\mathbf{A})_O$ has an empty intersection with the Pareto front $\mathbf{P}(\mathbf{f},\mathbf{S})_O$; that is, $\mathbf{P}(\mathbf{f},\mathbf{A})_O \cap \mathbf{P}(\mathbf{f},\mathbf{S})_O = \emptyset$.*

*Proof* Assume that $L_{\mathbf{S}}(\mathbf{w}) > L_{\mathbf{A}}(\mathbf{w})$ for all weight vectors $\mathbf{w} = (w_1,\dots,w_m)$. We will show that for any objective vector $\mathbf{s}^* = (s_1^*,\dots,s_m^*) \in \mathbf{P}(\mathbf{f},\mathbf{S})_O$ there exists a vector from $\mathbf{Y} = \{\mathbf{y} :\ \mathbf{y} = \mathbf{f}(\mathbf{x}),\ \mathbf{x} \in \mathbf{A}\}$ which dominates $\mathbf{s}^*$ (in fact, we will find a vector which strictly dominates $\mathbf{s}^*$).

For the chosen $\mathbf{s}^*$, compute the vector of weights $\mathbf{w} = (w_1,\dots,w_m)$ as formulated in Lemma 8.1. All weights are positive (that is, $w_i > 0$ for all $i = 1,\dots,m$) and

$$L_{\mathbf{S}}(\mathbf{w}) = \max_{i=1,\dots,m} \{w_i(s_i^* - u_i)\} = \min_{\mathbf{x}\in\mathbf{S}} \max_{i=1,\dots,m} \{w_i(f_i(\mathbf{x}) - u_i)\}.$$

By Lemma 8.1, we have $\mathbf{s}^* = \mathbf{f}(\mathbf{x}^*_{\mathbf{w},\mathbf{S}})$, where

$$\mathbf{x}^*_{\mathbf{w},\mathbf{S}} = \arg\min_{\mathbf{x}\in\mathbf{S}} \max_{i=1,\dots,m} \{w_i(f_i(\mathbf{x}) - u_i)\},$$

so that $s^*_i = f_i(\mathbf{x}^*_{\mathbf{w},\mathbf{S}})$ for $i = 1, \dots, m$.

Let $j \in \{1, 2, \dots, m\}$ be such that

$$w_j(s^*_j - u_j) = \min_{\mathbf{x}\in\mathbf{S}} \max_{i=1,\dots,m} \{w_i(f_i(\mathbf{x}) - u_i)\}.$$

For all $i \in \{1, 2, \dots, m\}$, set

$$v_i = w_j(s^*_j - u_j)/(s^*_i - u_i) \text{ and } \widetilde{w}_i = v_i / \sum_{l=1}^{m} v_l.$$

The new vector of weights $\widetilde{\mathbf{w}} = (\widetilde{w}_1, \dots, \widetilde{w}_m)$ is constituted of the positive components, and for all $i = 1, \dots, m$ we have

$$\widetilde{w}_i(s^*_i - u_i) = \text{const} \geq L_{\mathbf{S}}(\widetilde{\mathbf{w}}) = \min_{\mathbf{x}\in\mathbf{S}} \max_{l=1,\dots,m} \{\widetilde{w}_l(f_l(\mathbf{x}) - u_l)\}. \tag{8.10}$$

On the other hand, by the main assumption we have

$$L_{\mathbf{S}}(\widetilde{\mathbf{w}}) > L_{\mathbf{A}}(\widetilde{\mathbf{w}}) = \min_{\mathbf{x}\in\mathbf{A}} \max_{l=1,\dots,m} \{\widetilde{w}_l(f_l(\mathbf{x}) - u_l)\} = \max_{l=1,\dots,m} \{\widetilde{w}_l(f_l(\mathbf{x}^*_{w,\mathbf{A}}) - u_l)\}, \tag{8.11}$$

where

$$\mathbf{x}^*_{w,\mathbf{A}} = \arg\min_{\mathbf{x}\in\mathbf{A}} \max_{l=1,\dots,m} \{\widetilde{w}_l(f_l(\mathbf{x}) - u_l)\}.$$

Set $\widetilde{\mathbf{y}} = (\widetilde{y}_1, \dots, \widetilde{y}_m) = \mathbf{f}(\mathbf{x}^*_{w,\mathbf{A}})$ so that $\widetilde{y}_i = f_i(\mathbf{x}^*_{w,\mathbf{A}})$ for $i = 1, \dots, m$. Combining (8.10) and (8.11), we have for all $i = 1, \dots, m$:

$$\widetilde{w}_i(s^*_i - u_i) > \max_{l=1,\dots,m} \{\widetilde{w}_l(\widetilde{y}_l - u_l)\} \geq \widetilde{w}_i(\widetilde{y}_i - u_i). \tag{8.12}$$

Since $\widetilde{w}_i > 0$ for all $i$, the inequalities (8.12) are equivalent to the inequalities $s^*_i > \widetilde{y}_i$ for all $i = 1, \dots, m$. This means that the objective vector $\widetilde{y} \in \mathbf{Y}$ strictly dominates the vector $\mathbf{s}^*$. □

Theorem 8.1 implies the importance of estimation of the function $L_{\mathbf{A}}(\mathbf{w})$ for the construction of a branch and probabilistic bound algorithm for multi-objective optimization.

## 8.4 Upper and Lower Estimates for the Pareto Front

In this section, we visualize the behavior of the function

$$L_{\mathbf{A}}(\mathbf{w}) = \min_{\mathbf{x}\in\mathbf{A}} g(\mathbf{x}, \mathbf{w})$$

and the estimates of this function which are based on the use of (8.6) and (8.7). Two test problems are used below. The first test problem is ideal for the application of the statistical method described in Section 8.2. The second test function represents/models problems aimed by the bi-objective optimization method.

Visualization of the function $L_{\mathbf{A}}(\mathbf{w})$ corresponding to more than two objectives is difficult, and we thus assume $m = 2$; that is, $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}))$. In this case, we set $\mathbf{w} = (w, 1-w)$ and consider the function $L_{\mathbf{A}}(\mathbf{w}) = L_{\mathbf{A}}(w)$ depending on one variable only, $w \in [0, 1]$. We also assume that $d = 2$ and $\mathbf{x} = (x_1, x_2) \in \mathbf{A} = [0, 1] \times [0, 1]$.

As the first test problem, we consider (1.3) where both objectives are quadratic functions. The sets of Pareto optimal solutions and Pareto optimal decisions are presented in Figure 1.1.

The second multi-objective problem (1.6) is composed of two Shekel functions which are frequently used for testing of single-objective global optimization algorithms. The sets of Pareto optimal solutions and Pareto optimal decisions are presented in Figure 1.3.

In Figures 8.1 and 8.2, we show the following estimates of $L_{\mathbf{A}}(w)$, for different $w \in [0, 1]$:

(a): $y_{1,n}$, the minimal order statistic corresponding to the sample $\{y_j = g(\mathbf{x}_j, w); j = 1, \ldots, n\}$;
(b): $\widehat{\mathsf{m}}_{n,k}$ constructed by the formula (8.6);
(c): $y_{1,n} - (y_{k,n} - y_{1,n})/c_{k,\delta}$, the lower end of the confidence interval (8.7).

In Figure 8.3 we illustrate the precision of these estimates for $L_{\mathbf{S}}(w)$ where $\mathbf{S}$ is a subset of $\mathbf{A}$ defined in the capture.

The sample size $n$ is chosen to be $n = 300$, the number $k$ of order statistics used is $k = 4$ and $\delta = 0.05$. Any increase of $n$ (as well a slight increase of $k$) leads to an improvement of the precision of the estimates. However, to observe any visible effect of the improvement one needs to significantly increase $n$, see [239] and especially [238] for related discussions.

For each $w$, the minimal order statistic $y_{1,n}$ is an upper bound for the value of the minimum $L_{\mathbf{A}}(w) = \min_{\mathbf{x}\in\mathbf{A}} g(\mathbf{x}, w)$ so that it is not a very good estimator. Similarly, $y_{1,n} - (y_{k,n} - y_{1,n})/c_{k,\delta}$, the lower end of the confidence interval (8.7), is not a good estimator of $L_{\mathbf{A}}(\mathbf{w})$ as by the definition it is an upper bound $L_{\mathbf{A}}(\mathbf{w})$ in the majority of cases. The estimator $\widehat{\mathsf{m}}_{n,k}$ is always in-between the above two bounds, so that we always have $y_{1,n} \le \widehat{\mathsf{m}}_{n,k} \le y_{1,n} - (y_{k,n} - y_{1,n})/c_{k,\delta}$ (this can be proved theoretically). We have found that the estimator $\widehat{\mathsf{m}}_{n,k}$ is rather precise in the chosen test problems.

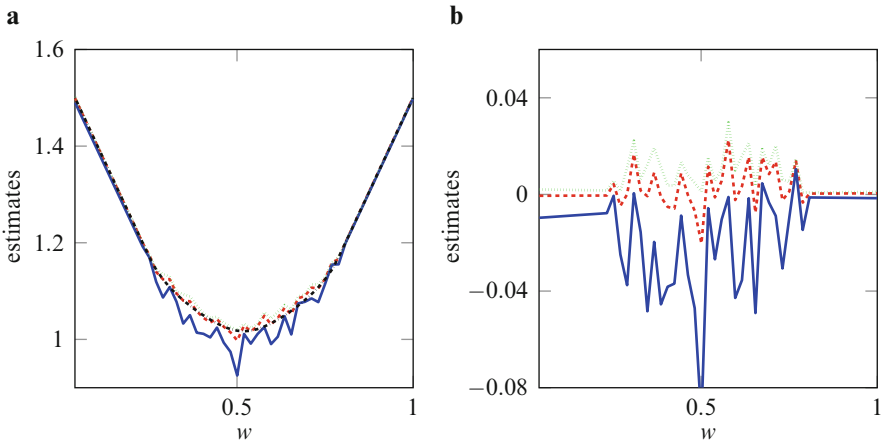


**Fig. 8.1** Illustration for the test problem (1.3) with values $w \in [0, 1]$ on the horizontal axis; (**a**) exact values of the function $L_{\mathbf{A}}(w)$ (*black, dash-dot*), its estimates using formula (8.6) with $k = 4$ (*red, dash*), its estimates using the minimal order statistic only (*green, dots*), lower end of the confidence interval (8.7) (*blue, solid line*) with $k = 4$, (**b**) the differences between the three estimates plotted and $L_{\mathbf{A}}(w)$

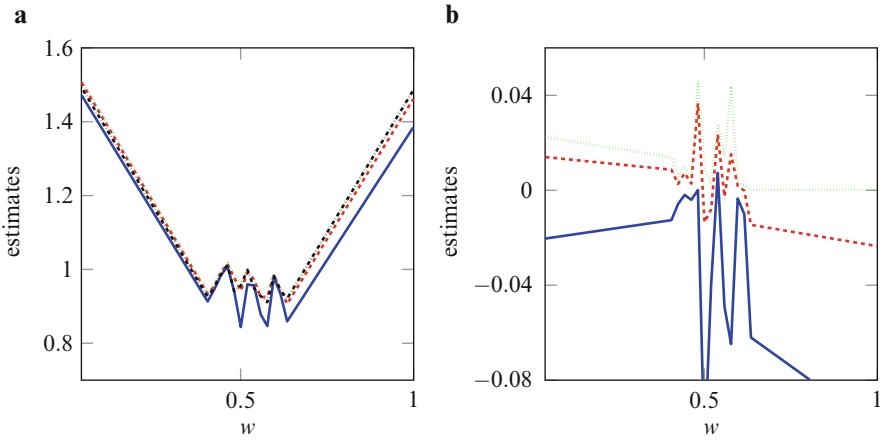


**Fig. 8.2** Illustration for the test problem (1.6) similar to Figure 8.1

## 8.5 Branch and Probability Bound Methods

For a single-objective optimization, branch and bound optimization methods are widely known. They are frequently based on the assumption that the objective function $f(\mathbf{x})$ satisfies the Lipschitz condition; see Section 4.2. These methods consist of several iterations, each includes the three following stages:

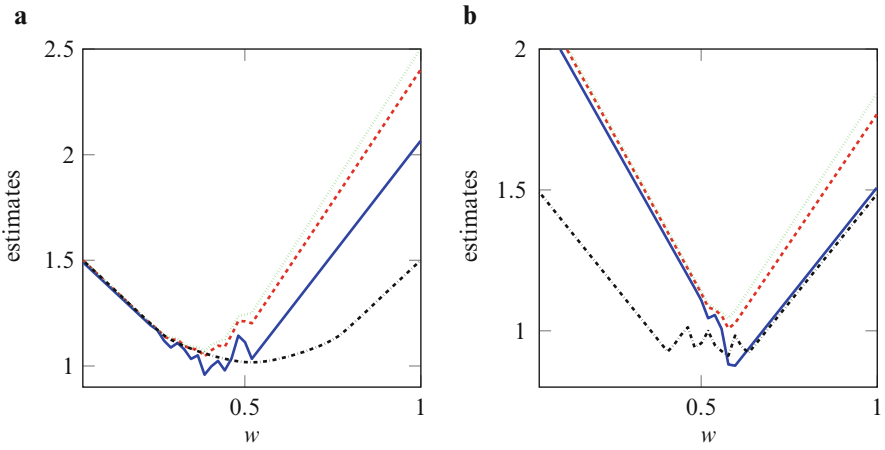


**Fig. 8.3** Estimates for the subsets of feasible regions: (**a**) test problem (1.3) with the region $x_1 \geq 0.5$, $x_2 \geq 0.5$, (**b**) test problem (1.6) with the region $x_1 \geq 0.75$, $x_2 \leq 1/3$

(i) branching of the optimization set into a tree of subsets,
(ii) making decisions about the prospectiveness of the subsets for further search, and
(iii) selection of the subsets that are recognized as prospective for further branching.

To make a decision at stage (8.5) prior information about $f(\mathbf{x})$ and values of $f(\mathbf{x})$ at some points in $\mathbf{A}$ are used, deterministic lower bounds (often called "underestimates") for the infimum of $f(\mathbf{x})$ on the subsets of $\mathbf{A}$ are constructed, and those subsets $\mathbf{S} \subset \mathbf{A}$ are rejected for which the lower bound for $\mathsf{m}_S = \inf_{\mathbf{x}\in\mathbf{S}} f(\mathbf{x})$ does not exceed an upper bound $\hat{f}^*$ for $\mathsf{m} = \min_{\mathbf{x}\in\mathbf{A}} f(\mathbf{x})$. (The minimum among evaluated values of $f(\mathbf{x})$ in $\mathbf{A}$ is a natural upper bound $\hat{f}^*$ for $\mathsf{m}$.)

The branch and bound techniques are among the best deterministic techniques developed for single-objective global optimization. These techniques are naturally extensible to multi-objective case as shown in Chapter 5. In the case of single-objective optimization, deterministic branch and bound techniques have been generalized in [238] and [237] to the case where the bounds are stochastic rather than deterministic, and are constructed on the base of statistical inferences about the minimal value of the objective function. The corresponding methods are called branch and probability bound methods. In these methods, statistical procedures for testing the hypothesis $H_0 : M_S \leq \hat{f}^*$ are applied to make a decision concerning the prospectiveness of a set $\mathbf{S} \subset \mathbf{A}$ at stage (ii). Rejection of the hypothesis $H_0$ corresponds to the decision that the global minimum $\mathsf{m} = \min_{\mathbf{x}\in\mathbf{A}} f(\mathbf{x})$ cannot be reached in $\mathbf{S}$. Unlike the deterministic decision rules such rejection may be false. This may result that the global maximizer is lost. However, an asymptotic level for the probability of the false rejection can be controlled and it will be fixed.

As for stages (i) and (iii) above they can be implemented in exactly the same fashion as in the deterministic branch and bound methods. For the case when the structure of $\mathbf{A}$ is not too complicated the following technique has been proved in [238] to be convenient and efficient. At each $j$-th iteration we first isolate in the search domain $\mathbf{A}_j$ a subdomain $\mathbf{S}_{j1}$ with the center at the point corresponding to the record value of $f(\mathbf{x})$. The point corresponding to the record value of $f(\mathbf{x})$ over $\mathbf{S}_j \backslash \mathbf{S}_{j1}$ is the center of a subdomain $\mathbf{S}_{j2}$. Similar subdomains $\mathbf{S}_{ji}$ $(i = 1, \ldots, I)$ are isolated until either $\mathbf{A}_i$ is not covered, or the hypothesis is not rejected that the global maximum can occur in the residual set $\mathbf{S}_j / \cup_{i=1}^{I} \mathbf{S}_{ji}$ (the hypothesis can be verified by the procedure described below). The search domain $\mathbf{S}_{j+1}$ of the next $(j+1)$-th iteration is naturally to be either

$$\mathbf{S}^{(j)} = \cup_{i=1}^{I} \mathbf{S}_{ji}$$

or a hyper-rectangle covering $\mathbf{S}^{(j)}$ or a union of disjoint hyper-rectangles covering $\mathbf{S}^{(j)}$.

In multi-objective optimization, the branch and probabilistic bound methods can be formulated in exactly the same manner except the main hypothesis to be tested is

$$L_{\mathbf{S}}(\mathbf{w}) < L_{\mathbf{A}}(\mathbf{w}) \quad \text{for} \quad \text{all} \quad \mathbf{w} \tag{8.13}$$

rather than $\min_{\mathbf{x} \in \mathbf{S}} f(\mathbf{x}) > \min_{\mathbf{x} \in \mathbf{A}} f(\mathbf{x})$, as in the case of single-objective optimization.

Since we do not know both $L_{\mathbf{S}}(\mathbf{w})$ and $L_{\mathbf{A}}(\mathbf{w})$, we need to replace these functions with estimates. Deterministic branch and bound methods would require the use of an underestimate (minorant) for $L_{\mathbf{S}}(\mathbf{w})$ and overestimate (majorant) for $L_{\mathbf{A}}(\mathbf{w})$. Assuming we have a sample of points $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ in $\mathbf{A}$, construction of an overestimate for $L_{\mathbf{A}}(\mathbf{w})$ is easy: this is simply the record value $\min_{j=1,\ldots,n} g(\mathbf{x}_j, w)$, see the discussion above. It is well known that a uniform distribution of $w$ in the interval [0, 1] does not necessarily imply uniform distribution of the respective non-dominated points in the Pareto front. To obtain an approximately uniform distribution, a method of adaptive weights, e.g., the method proposed in [154] can be applied here; let us note, however, that an efficient hybridization of these methods is not a trivial task. The performance of the considered method can depend also on the selection of the utopian vector. This problem is also considered for our further research.

Assume now we have a sample of points $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ in $\mathbf{S}$ and the corresponding sample $\{y_j = g(\mathbf{x}_j, w); j = 1, \ldots, n\}$. If we do not require our algorithm to be theoretically exact, then for a given $w$ we can use the estimate $\widehat{\mathsf{m}}_{n,k}$ defined in (8.6). As discussed above this estimate is rather precise. However, it is an estimate but not an underestimate. If we want to be more precise in our statistical conclusions, we need to use $y_{1,n} - (y_{k,n} - y_{1,n})/c_{k,\delta}$, the lower end of the confidence interval (8.7). For each $w$, this estimate becomes an underestimate of $L_{\mathbf{S}}(w)$ with probability approximately equal $1 - \delta$ (for large $n$).

The branch and probability bound methods based on the above ideas and the statistical procedures described above are both practically efficient for small or moderate dimensions $d$ (say, $d \leq 5$) and theoretically justified in the sense that under general assumptions concerning $f(\mathbf{x})$ they asymptotically converge with a given probability which can be chosen close to 1. Also, the algorithms are rather simple and could be easily realized as computer routines.

## 8.6 Visualization

For the expensive black-box multi-objective optimization problems it seems reasonable to hybridize a computer aided algorithmic search with an interactive human heuristic. Visualization is very important in perception of relevant information by a human expert [48, 122, 260, 263]. In this section we investigate possibilities of the visualization of scarce information on the Pareto front using a statistical model of the considered problem.

The following problem of bi-objective optimization is considered:

$$\min_{\mathbf{x}\in\mathbf{A}} \mathbf{f}(\mathbf{x}), \quad \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}))^T, \tag{8.14}$$

where the properties of $\mathbf{f}(\mathbf{x})$ and of the feasible region $\mathbf{A} \subseteq \mathbb{R}^d$ are specified later on. We are interested in the approximation and visualization of $\mathbf{P}(\mathbf{f})_O$ using scarce information obtained in the initial/exploration phase of optimization. The necessity of the exploration phase follows from the assumption on the black-box objectives. The human heuristic abilities can be advantageous here in perception of scarce information gained during the exploration. The restriction of information scarcity is implied by the assumption on expensiveness of the objectives. The further search can be rationally planned by the optimizer depending on the results of the exploration. Visualization is expected to aid the perception of the available results.

The exploratory phase assumes that we have values of the objective functions at some number of random points in $\mathbf{A}$ which are independent and uniformly distributed. This exploration method can be seen as an analog of a popular heuristic decision by throwing a coin in the case of a severe uncertain decision situation. Moreover, the uniform distribution of points in the feasible region is the worst-case optimal algorithm for the multi-objective optimization of Lipschitz objectives; see Chapter 6. Although in the latter case the uniformity is understood in the deterministic sense, the random uniform distribution of points is a frequently used simply implementable approximation of the deterministic one. Now we have to extract, useful for the further search, information from the available data, i.e., from a set of $\mathbf{x}_i$, $\mathbf{y}_i = \mathbf{f}(\mathbf{x}_i)$, $i = 1, \ldots, n$.

In single-objective global optimization, some information on the global minimum of $f(\mathbf{x})$ can be elicited from the sample $z_i = f(\mathbf{x}_i)$, where $\mathbf{x}_i$ are independent random points, by means of the methods of statistics of extremes; see Section 4.4.

Let $\mathbf{x}_i \in \mathbf{A}$, $i = 1, \ldots, n$, be a sample of random points uniformly distributed in $\mathbf{A}$. The objective function values computed at these points are denoted by $\mathbf{y}_i = \mathbf{f}(\mathbf{x}_i)$. The subset of $\{\mathbf{y}_i,\ i = 1, \ldots, n\}$, constituted of weakly non-dominated points, is denoted by $\mathbf{P} = \{\mathbf{p}_i,\ i = 1, \ldots, m\}$. The points in $\mathbf{P}$ present some approximation of an upper bound for $\mathbf{P}(\mathbf{f})_O$. However, normally a lower bound is of real interest. To construct a statistical estimate of the latter we apply the weighted Tchebycheff scalarization (discussed in the previous section) of the considered problem replacing the vector-valued function $\mathbf{f}(\mathbf{x})$ by a family of scalar-valued functions

$$g(\mathbf{x}, w) = \max\{w(f_1(\mathbf{x}) - u_1),\ (1 - w)(f_2(\mathbf{x}) - u_2)\},\ 0 \le w \le 1, \tag{8.15}$$

where $\mathbf{u} = (u_1, u_2)^T$ is the utopian vector:

$$u_i < \min_{\mathbf{x} \in \mathbf{A}} f_i(\mathbf{x}),\ i = 1, 2.$$

For a given $w$, the application of the scalarization (8.15) to the available objective function values $\mathbf{y}_i$, $i = 1, \ldots, n$, produces the sample

$$z_i(w) = \max\{w(y_{i1} - u_1),\ (1 - w)(y_{i2} - u_2)\},\ i = 1, \ldots, n. \tag{8.16}$$

Formulas (8.6) and (8.7) applied to the sample (8.16) give the dependent on $w$ estimates and confidence intervals of minima of the scalarized problem. Now the statistics concerning the sample of scalar random variables $z_i(w)$ should be mapped to the two-dimensional space of objectives. However, to construct such a mapping we need some facts concerning the weighted Tchebycheff scalarization.

Let the elements of $\mathbf{P}$ be ordered with respect to the first component: $p_{i1} < p_{i+1\,1}$.

**Theorem 8.2** *The following equalities are valid:*

$$\{\mathbf{y}_{j(w)},\ 0 \le w \le 1\} = \{\mathbf{p}_i,\ i = 1, \ldots, m\}, \tag{8.17}$$

where $j(w) = \arg \min\limits_{1 \le i \le n} z_i(w)$,

$$w_i = \arg \min_{0 \le w \le 1} \max\left(w(p_{i1} - u_1),\ (1 - w)(p_{i2} - u_2)\right) = \frac{p_{i2} - u_2}{p_{i1} - u_1 + p_{i2} - u_2}, \tag{8.18}$$

$$\mathbf{y}_{j(w)} = \begin{cases} \mathbf{p}_1\,,\ w_1 \le w_2^*, \\ \mathbf{p}_k\,,\ w_k^* \le w \le w_{k+1}^*,\ 2 \le k \le m - 1, \\ \mathbf{p}_m\,,\ w_m^* \le w \le w_m, \end{cases} \tag{8.19}$$

where $w_k^* = \dfrac{p_{k-1\,2} - u_2}{p_{k1} - u_1 + p_{k-1\,2} - u_2}$. (8.20)

*Proof* The equalities in the statement of the theorem are reformulations of known properties of the method of weighted Tchebycheff scalarization, which are convenient for our investigation.

Formula (8.17) means that a solution found by the method of weighted Tchebycheff scalarization is Pareto optimal, possibly including weakly Pareto optimal solutions. Let us assume contrary that for any $w$, the equality $\mathbf{y}_i = \mathbf{y}_{j(w)}$ is valid where $\mathbf{y}_i$ is a strictly dominated solution. Since $\mathbf{y}_i$ is a strictly dominated solution, there exists $\mathbf{p}_k \succ \mathbf{y}_i$ for which the inequalities

$$p_{ks} - u_i < y_{is} - u_i,\ s = 1, 2, \tag{8.21}$$

are valid. The latter inequalities contradict to the equality $j(w) = i$ since $j(w)$ is the minimizer in (8.17). The latter contradiction means that the assertion $\mathbf{y}_i = \mathbf{y}_{j(w)}$ cannot be true.

Since the inequalities $p_{i1} - u_1 > 0$, $p_{i2} - u_2 > 0$, and $0 \le w \le 1$ are valid, the equality

$$w(p_{i1} - u_1) = (1 - w)(p_{i2} - u_2),$$

is the sufficient minimum condition which implies the formula for $w_i$ given in (8.18).

We start the derivation of (8.18) with proving the inequality $w_i > w_{i+1}$. Since the function $t/(t + v)$, $t > 0$, $v > 0$ is increasing in $t$ and decreasing in $v$, the following inequalities are valid:

$$\frac{t_2}{t_2 + v_2} < \frac{t_1}{t_1 + v_2} < \frac{t_1}{t_1 + v_1},\ 0 < t_1 < t_2,\ 0 < v_2 < v_1.$$

The substitution of $t_1,\ v_1$ by $p_{i1} - u_1, p_{i2} - u_2$, and $t_2,\ v_1$ by $p_{i+1\,1} - u_1, p_{i+1\,2} - u_2$ yields $w_i > w_{i+1}$.

A Pareto optimal solution $\mathbf{p}_k$ is found using the method of weighted Tchebycheff scalarization not only for $w = w_k$ but also for an interval of $w$ values. Taking into account the established monotonicity of $w_i$ with respect to $i$, a threshold value $w_k^*$, $w_{k+1} < w_k^* < w_k$, can be found such that

$$j(w) = k \text{ for } w_{k+1}^* < w \le w_k^*.$$

The expression for $w_k^*$

$$w_k^* = \frac{p_{k\,2} - u_2}{p_{k\,2} - u_2 + p_{k+1\,1} - u_1}, \tag{8.22}$$

follows from the equalities for the solutions of the corresponding scalar problems related to the neighboring Pareto optimal solutions $\mathbf{p}_k$ and $\mathbf{p}_{k+1}$:

$$\max\left(w(p_{k1} - u_1),\ (1 - w)(p_{k2} - u_2)\right) = \max\left(w(p_{k+1\,1} - u_1),\ (1 - w)(p_{k+1\,2} - u_2)\right).$$

The derived formula (8.22) coincides with (8.20) in the last part of the statement of the theorem.

The visualization of the approximation of the Pareto front can be obtained on the base of the estimates related to the solutions of the scalar problems

$$z^*(w) = \min_{1 \le i \le n} z_i(w). \tag{8.23}$$

As follows from the theorem, the minimum in (8.23) for $w^*_{k+1} < w \le w^*_k$ corresponds to the weakly non-dominated vector $\mathbf{p}_k$. As a representative of the interval $(w^*_{k+1}, w^*_k]$ we select $w_k$. The visualization of the estimates is problematic since the estimates are single-dimensional but should be presented in the two-dimensional solution space. Moreover, the sample $z_i(w)$ is constructed as weighted Tchebycheff distances from $\mathbf{u}$, but the estimates are obtained as scalar increments to $z^*(w)$.

To find the point in the objective space, which corresponds to the estimator $z^*(w_k)$, the vector-valued increment $\Delta\mathbf{p}_k$ should be added to $\mathbf{p}_k$, where $\Delta\mathbf{p}_k$ represents the scalar increment $\delta_k = z^*(w_k) - \widehat{Z}_{n,k}(w_k)$. The term "represents" here means the equality of the weighted Tchebycheff norm of $\Delta\mathbf{p}_k$ and $-\delta_k$ which implies the following expression of $\Delta\mathbf{p}_k$:

$$\Delta\mathbf{p}_k = (\delta_k/w_k,\ \delta_k/(1-w_k))^T.$$

The confidence interval is visualized similarly. Thus the points in the objective space, which mark the estimator and the lower bound for the Pareto front related to $w_k$ are located at the line segment connecting the utopian point and the point marking $\mathbf{p}_k$.

Two test problems are considered to illustrate the proposed method of visualization of the Pareto front; these test problems are considered also in [253]. As the first test problem we consider a problem with two quadratic objective functions (1.3). The second multi-objective problem is composed of two Shekel functions which are frequently used for testing of single-objective global optimization algorithms (1.6).

The feasible objective regions and Pareto fronts for both test problems are depicted in Figures 1.1 and 1.3. The density of points in the objective feasible regions shows the "thickness" of the close vicinity of the Pareto set in the decision space. The "thickness" here means relative quantity of points in the close vicinity of the set of Pareto optimal decisions. The probability to get a point in a "thin" vicinity of the Pareto set by the uniform distribution of points in the feasible region is small. From Figures 1.1 and 1.3 it is clear that the second problem is more difficult than the first one, and that the estimation of the central part of the Pareto front of the second test problem is more difficult than the estimation of its outside parts.

A hundred random points were generated with uniform distribution in the feasible region, and the values of objective functions are computed at these points. The samples of the vector values were mapped into scalar samples by means of the weighted Tchebycheff method. The estimators of the minima and the confidence

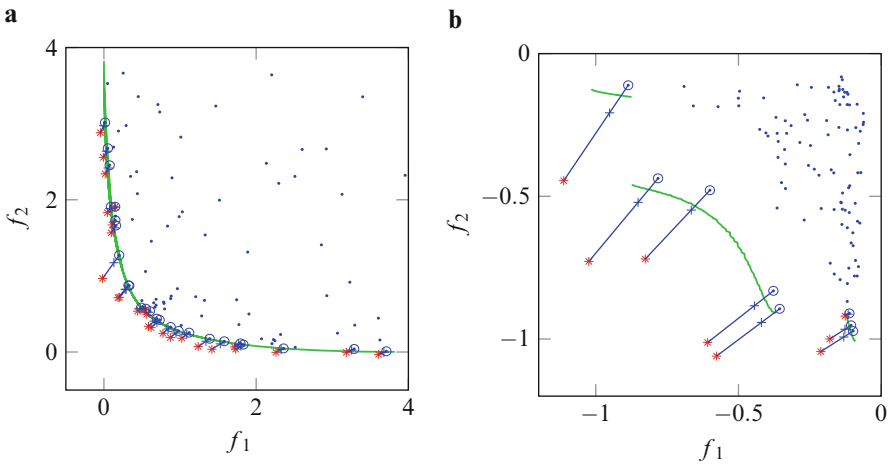


**Fig. 8.4** The visualization of the estimates of Pareto front, *line* shows the true Pareto front, the explanation of markers is given in the text, (**a**) the first test problem (1.3), (**b**) the second test problem (1.6)

**Table 8.1** The values of $w_i$, $i = 1, \ldots, 24$, for the first test problem

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.67 | 0.66 | 0.63 | 0.62 | 0.62 | 0.61 | 0.58 | 0.55 | 0.54 | 0.51 | 0.50 | 0.49 |
| 0.48 | 0.47 | 0.46 | 0.44 | 0.43 | 0.41 | 0.39 | 0.38 | 0.37 | 0.34 | 0.31 | 0.29 |

intervals were computed as described above using the following parameters of the model: $\alpha = 1$, $\delta = 0.1$, $k = 3$. The visualization of the obtained estimates is presented in Figure 8.4. The *points* denote the vector values of the objective functions presented for the analysis. The *circles* denote the non-dominated solutions. The *line segments* show the confidence intervals for the Pareto optimal solutions: the upper bound coincides with a respective non-dominated point and the lower bound is denoted by a *star*. The *crosses* inside the *intervals* denote the estimators of the Pareto optimal solutions.

The number of non-dominated solutions found for the first test problem is equal to 24. The values of $w_i$, $i = 1, \ldots, 24$, as shown in Table 8.1, are distributed rather uniformly between the extremal values. The respective confidence intervals for the Pareto optimal solutions are rather short. The information used to visualize the Pareto front seems comprehensive and reliable.

The number of non-dominated solutions found for the second test problem is equal to 6. The values of $w_i$, $i = 1, \ldots, 6$, as shown in Table 8.2, are not distributed uniformly between the extremal values. The respective confidence intervals for the Pareto optimal solutions are rather long. The information used for the visualization of the Pareto front seems to be not comprehensive and not very reliable. Such a result is hardly unexpected: in the description of the second test problem, it was mentioned that the vicinity of the set of Pareto optimal decisions is thin. Therefore,

**Table 8.2** The values of $w_i$, $i = 1, \ldots, 8$, for the second test problem

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0.60 | 0.55 | 0.52 | 0.44 | 0.43 | 0.40 | 0.39 | 0.389 |

the probability to hit this vicinity is small, and hundred random points are not sufficient for a reliable estimation of the Pareto front. This conclusion can be made without a priori knowledge of the unfavorable property of the problem but by visual analysis of the figure (of course, the line showing the true Pareto front would be absent).

However, the previous figure gives also valuable information for further analysis. The estimate of the minimum of $f_2(\cdot)$ is rather reliable. To obtain information about the Pareto front in the part of interest but of great uncertainty the minimization of $g(\mathbf{x}, w)$ can be recommended with $w$ chosen by interpolating respective values from Table 8.2. For example, a solution with approximately equal values of $f_1(\cdot)$ and $f_2(\cdot)$ can be of interest, to have a suitable trade-off for the equally important objectives. Such a solution can be expected in the part of the Pareto front of great uncertainty somewhere between the third and the forth (from links) points. Therefore the minimization of $g(\mathbf{x}, w)$ would be reasonable with $w = (0.52+0.44)/2 = 0.48$. The starting point was chosen as a middle point of the line segment connecting the third and the forth efficient points: $\mathbf{x}_o = (0.5287,\ 0.5240)^T$. The respective objectives are equal to $\mathbf{f}(\mathbf{x}_o) = (-0.5159,\ -0.5579)^T$. The Nelder–Mead algorithm was applied to the minimization of $z(\mathbf{x}, 0.48)$. A direct search algorithm was chosen because of the assumption that the objectives are of the black-box type. The stopping condition was defined by the maximum permissible number of iterations equal to 10. The found solution was $\mathbf{x}^* = (0.5037,\ 0.4740)^T$, $\mathbf{f}(\mathbf{x}^*) = (-0.5036,\ -0.6581)^T$.

If more reliable estimation of the whole Pareto front would be of interest, then we could recommend either an extension of the uniform random search or the method of branch and probability bounds.

## 8.7 Discussion and Remarks

In this chapter we have presented very recent results on generalization of the method of branch and probability bound to problems of multi-objective optimization. Although theoretical substantiation of such a generalization does not seem complicated, a broad experimental investigation is still needed till clear description of favorable application areas of this method will be possible. From the limited experience the research on hybridization of the considered method with visualization of estimates of the Pareto front seems promising, as well as the application of statistical estimates of the Pareto front for controlling and terminating heuristic search methods.

# Part III
# Applications

Most of the optimization problems arising in practice have several objectives which have to be optimized simultaneously. These problems include different problems in engineering, economy, management science, medicine, etc., see, for example, [2, 13, 56, 113, 123, 174, 181, 219, 262]. In this part we discuss some of application problems in multi-objective optimization, but it is impossible to mention all of them since there are huge numbers of such applications. In the following chapters we discuss some application of non-convex multi-objective optimization in more detail.

Let us briefly mention some applications of multi-objective optimization related to the topics discussed in this book. The web access problem was studied in [54]. It is supposed in the problem that one wants to retrieve a list of records from the world-wide web. The problem is to minimize the total cost and the total delay and maximize the overall probability of the result. Two models are studied, cost model and reward model, with respect to the problem. In the cost model, an order is sought where the expected overall cost is minimized. In the latter model, it is assumed that a constant known reward should be collected if at least one source returns a correct answer. Then a schedule of maximum expected reward is sought. The authors showed that the second problem is NP-hard and developed approximation algorithms for those problems. We discussed complexity of such problems in Section 3.1.

The portfolio selection problem in the traditional mean–variance formulation [126] has been used in finance for the last half century. According to this theory, in the model the risk is measured with variance thus generating a quadratic programming model. This model has been frequently criticized as not consistent with axiomatic models of preferences for choice under risk. It is usually assumed in the model that at least one of the following hypotheses should be verified: the utility function of the investor is quadratic, the returns of the stocks have normal distributions. However, quadratic utility functions are not consistent in economics. According to the second hypothesis, negative returns are very unlikely. It has been shown that stock returns have asymmetrical or leptokurtic distribution based on empirical tests [36]. On the other hand, the multidimensional nature of the problem has been emphasized by researchers [14, 90, 102, 201]. During the portfolio

evaluation, decision makers face several criteria including return criterion, risk criterion, liquidity criterion, and size criteria. Based on these criteria, the portfolio selection problem is usually formulated as a multi-objective optimization problem. Several authors have employed multi-objective methods for the portfolio selection problem [22, 89, 268]. A bibliographic survey about multi-objective optimization in finance can be found in [205].

Capital budgeting is the problem of determining whether or not investment projects such as building a new plant or investing in a long-term venture are worthwhile. In other words, the capital budgeting decision is to select a subset of available projects that gives the highest earnings to the company while it does not exceed a certain budget. Traditionally capital budgeting concerns a single-objective function which is usually in the form of a maximization of company's revenue. Capital budgeting process was first formulated as an optimization problem in [226]. Capital budgeting with multiple objectives was studied in [5, 116, 215]. Typical capital budgeting model with multiple criteria is usually expressed as a multi-objective knapsack problem discussed in Section 3.1. A time-dependent capital budgeting problem was presented as a time-dependent multi-objective knapsack problem in [104]. The dynamic programming approach [107] can be applied for finding the set of all efficient solutions of this problem when all cost coefficients and the budget are integer [103].

Industrial problems of cell formation arising in group technology and industrial engineering aim to grouping of machines into manufacturing cells and parts into product families so that each family is processed mainly within one cell [61, 70, 71]. Although these problems involve conflicting objectives [149, 227], most of the literature considers either the single-objective version of the problem or an aggregation of the objectives into a single one [45, 61, 125], for example, weighting of the criteria is used in [115, 124, 209, 214]. Rare exceptions explicitly consider multiple objectives using multi-objective evolutionary algorithms [6, 21, 46, 144, 223], multi-objective scatter search [9], multi-objective tabu search [118], or branch and bound [264]. An example bi-criterion problem of cell formation is given in Section 1.3 and a branch and bound algorithm for this problem is presented in Section 5.2.

# Chapter 9
# Visualization of a Set of Pareto Optimal Decisions

## 9.1 Introduction

Applications of optimization methods for optimal design are most efficient when combined with the human engineer's insight. Visualization is a technique which makes it easier for an engineer to understand the design space and interpret a design found by an optimization algorithm [255]. There are many techniques aimed at visualization of Pareto optimal solutions, especially for the bi-objective problems where a set of solutions, usually referred to as a Pareto front, is a curve in a two-dimensional solution space. The problem of visualization of the sets of optimal decisions is researched not so thoroughly. However, the visualization of a set of Pareto optimal decisions can significantly aid the choice of an appropriate engineering design. In this chapter, we present a case study of a process optimization where the application of a visualization technique was very helpful in the analysis of appropriate design variables.

## 9.2 A Design Problem

An optimal design of a process of chemical engineering is considered. Pressure swing adsorption (PSA) is a cyclic adsorption process for gas separation and purification. PSA systems have the potential of achieving a higher productivity for $CO_2$ capture than alternative separation processes [180], such as absorption. An efficient and cost-competitive PSA unit is one that achieves high levels of purity and recovery of the product [57]. Therefore, a bi-objective optimization in search for an appropriate design is applicable.

To apply an optimization-based design method, a mathematical model of the system is needed. The development of an appropriate mathematical model is crucial for the success of the optimization aided design. However, here we do not go into



P.M. Pardalos et al., *Non-Convex Multi-Objective Optimization*, Springer
Optimization and Its Applications 123, DOI 10.1007/978-3-319-61007-8_9

the technological details and their mathematical description. Since we focus on the visualization of the potential decisions, only few aspects of the mathematical model in question, which are important for the considered visualization problem, will be mentioned. For the technological aspects we refer to [18, 57, 180]. PSA processes are governed by partial differential algebraic equations. The simulation of a PSA process is computationally challenging, and the task to perform PSA simulation can be very time consuming; a single simulation can take minutes, hours, or even days.

In the case study considered here, the number of design parameters (variables of the respective optimization problem) was 6, and they were re-scaled so that the feasible region was reduced to the unit hyper-cube. The number of objectives was 2. The values of the objective functions are scaled to the intervals [0, 1]. A mathematical model based on simplified governing equations was used; see [18, 57, 263] for details. The simulation time for a single design ranged between 10 min and an hour, depending on the design parameters [263].

## 9.3 Visualization of the Optimization Results

Visualization in the considered case study was applied at the final stage of decision making. A multi-objective optimization algorithm (NSGA-II) has been applied by the experts in the process engineering, and complete computation results were recorded. The thorough analysis of the available data was requested to substantiate the choice of the design variables.

The application of the chosen multi-objective algorithm to the problem considered resulted in computing of $N = 1584$ two-dimensional vectors of objectives $\mathbf{y}_i$ at the points $\mathbf{x}_i = (x_{i1}, \ldots, x_{i6})^T$, $i = 1, \ldots, N$, the components of which belong to the unit interval: $0 \leq x_{ij} \leq 1$ where the index $j$, $(1 \leq j \leq 6)$, denotes the $j$-th component of the $i$-th vector.

The subset of $\mathbf{y}_i$, $i = 1, \ldots, N$, constituted of non-dominated points represents the Pareto front of the considered problem $\mathbf{P}(\mathbf{f})_O$; it consists of $N_P$=179 two-dimensional vectors; the corresponding subset of $\mathbf{x}_i$, $i = 1, \ldots, N$, is denoted by $\mathbf{X}_P$. A graphical drawing of the Pareto front is the standard presentation of results of a bi-objective optimization problem. By a visual analysis of the drawing a decision maker can choose an appropriate trade-off between the objectives. Since the Pareto front is represented by a finite number of points, the representation precision crucially depends on the number and distribution of points. The drawing of the Pareto front of the considered problem is presented in Figure 9.1a. The points are distributed rather densely and uniformly over the whole Pareto front but there is a discontinuity at the beginning of the upper part of the graph which indicates some neighborhood of minimum of $f_1(\cdot)$. The very precise computation of the boundaries of this discontinuity does not seem important since the most interesting part of the Pareto front is that including the kink where trade-off between the objectives seems favorable.

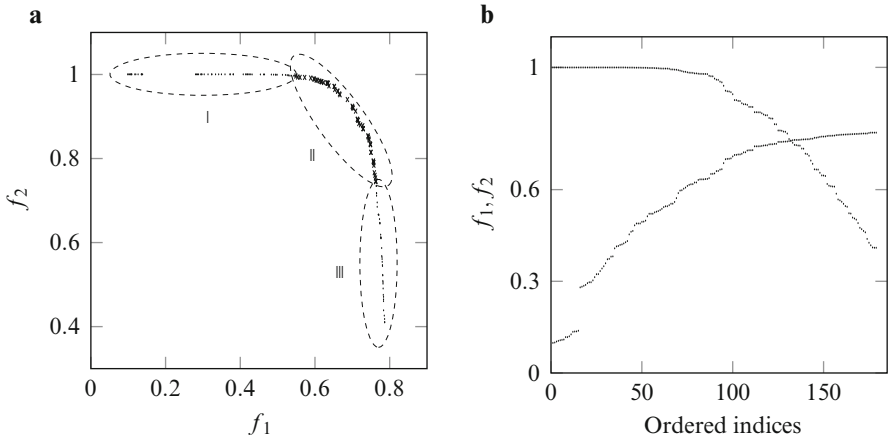


**Fig. 9.1** (**a**) Pareto front of the considered problem, (**b**) The graphs of $f_1(\mathbf{x}_i)$ and $f_2(\mathbf{x}_i)$, $\mathbf{x}_i \in \mathbf{X}_P$ with respect to the reordered indices

For a better identification of this part of the Pareto front a graph of $f_1(\mathbf{x}_i)$ and $f_2(\mathbf{x}_i)$, $\mathbf{x}_i \in \mathbf{X}_P$ is presented in Figure 9.1b where the horizontal axis is for the indices reordered according to the increase of $f_1(\mathbf{x}_i)$, $\mathbf{x}_i \in \mathbf{X}_P$.

A kink can be observed in the curve corresponding to $f_2(\mathbf{x}_i)$ for $i \approx 90$ where the horizontal behavior switches to a downward trend. It is therefore of interest to explore the solutions corresponding to indices in the interval [80, 139].

By a visual analysis of the graphs in Figure 9.1, an appropriate Pareto solution can be selected as well as the decision $\mathbf{x} \in \mathbf{X}_P$ which corresponds to the selected Pareto solution. However, such a choice is not always satisfactory since it does not pay respect to such properties of the corresponding decision as, e.g., the location of the selected decision vector in the feasible region $\mathbf{A}$. The analysis of the location of the set of efficient points in $\mathbf{A}$ can be especially valuable in cases of structural properties of the considered set important for the decision making. For example, some subsets of $\mathbf{A}$ might not be forbidden but may be unfavorable, and that property may not be easy to introduce into a mathematical model. The analysis of the properties of the set of efficient points can enable the discovery of latent variables, a relation between which essentially defines the Pareto front.

Numerous statistical methods are developed for the estimation of the impact of particular variables to the considered response. Since the optimization problem is bi-objective, the Pareto front and the set of the non-dominated decisions are single-dimensional manifolds; in other words, the latter is a curve in the six-dimensional unit cube. In principle, the parametric description of such a curve could be derived using a least-squares technique. However, the numerical solution of general nonlinear least-squares problems is difficult [256]. Therefore, an exploratory analysis of the available data seems reasonable here since it may highlight important specific properties of the problem in question.

To get an idea of the location of $\mathbf{X}_P$ in the six-dimensional unit cube, a multidimensional scaling-based algorithm has been applied to the two-dimensional visualization of a set of six-dimensional points which consists of $\mathbf{X}_P$ and of the cube vertices. The multidimensional scaling (MDS)- based visualization seems most suitable here since well applicable for highlighting the structure of data sets with up to 10 variables and 1000 points [20, 37]. Note that the application of the MDS-based visualization method [128] to the optimal design of a distillation process was very useful to highlight the location of a low-dimensional set of points of interest with respect to the vertices of the nine-dimensional hyper-cube [260]. The methods of MDS compute, for a set of points in multidimensional space, its image in a space of low dimensionality; naturally, for the visual analysis an image in the two-dimensional space is used. The main idea of the methods is to find the points in the two-dimensional space whose inter-distances would be as close as possible to the inter-distances between the points in the multidimensional space. Only the inter-distances are taken into account, therefore the images, obtained by the MDS methods, are invariant to translations and rotations of the data sets. For the same reason, the position and orientation of the image in the plane can also be selected rather arbitrary. The coordinate axes in the plane of images are abstract, though in special cases their interpretation in terms of the variables of the original problem can be found. The location of the set (curve) of Pareto optimal decisions with respect to the original coordinate system can aid discovery of most influential variables and their interplay, in case those exist. To highlight the location of the set of Pareto optimal decisions, it is visualized together with the vertices of the hyper-cube the edges of which are parallel to the coordinate axes in the original parameter space.

Before visualizing and analyzing the data of the considered problem, the two-dimensional image of the set of vertices of the six-dimensional unit hyper-cube is presented in Figure 9.2 to illustrate the space where the image of interest will be located. Note that in the figure, the symmetries specific for the original data are preserved. The images of subsets of $2^k$, $k = 1, 2, \ldots, 6$, vertices also create the structures similar to the original ones.

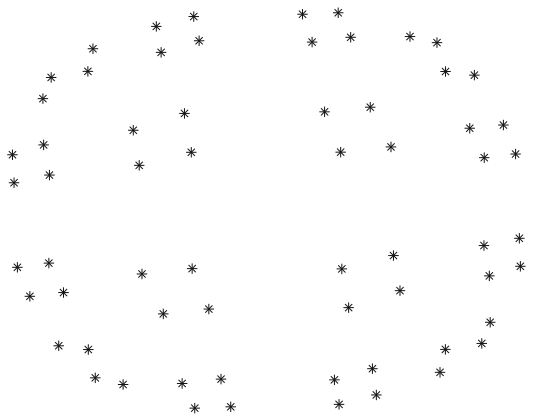

**Fig. 9.2** Two-dimensional image of vertices of the six-dimensional hyper-cube

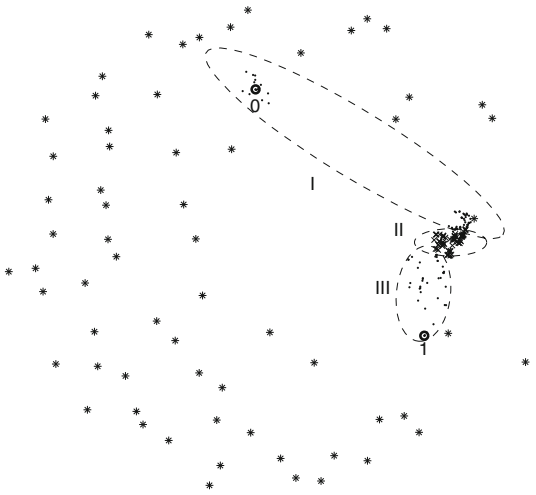


**Fig. 9.3** Two-dimensional image of the discrete representation of the set of Pareto optimal decisions, and the vertices of the six-dimensional hyper-cube; the *markers* 0 and 1 show the location of minimizers of $f_1(\cdot)$ and $f_2(\cdot)$, respectively

The set of six-dimensional vectors to be visualized consists of the vectors, respective to the vertices of the hyper-cube, and of $\mathbf{X}_P$. The number of points in the set is equal to 243 where 179 points belong to $\mathbf{X}_P$ and 64 vectors represent vertices of the hyper-cube $\mathbf{A}$. The image obtained using the visualization method [128] is presented in Figure 9.3. The images of points of $\mathbf{X}_P$ are shown by *circles* (o) and the vertices of the hyper-cube are shown by *stars* (*). The advantage of the method is that a compact set of points is shown in the foreground: the image of $\mathbf{X}_P$ and the closest vertices are seen in the foreground of Figure 9.3. The image of $\mathbf{X}_P$ occupies a relatively small part of the figure where big empty spots exist. These spots represent an empty space in the original hyper-cube. Indeed, $\mathbf{X}_P$ occupies a small part of the hyper-cube since $\mathbf{X}_P$ approximates the set of Pareto optimal decisions, which in turn is a one-dimensional manifold (a curve). The latter fact is a property of a bi-objective optimization problem where the set of Pareto optimal solutions (Pareto front) is a (possibly discontinuous) curve, and correspondingly the set of Pareto optimal decisions is a set of the same type.

As it is clearly seen in Figure 9.3 the set $\mathbf{X}_P$ consists of two subsets. The smaller subset is in some vicinity of the minimizer of the objective function $f_1(\cdot)$. The second, larger, subset is located closely to the line segment connecting two vertices in Figure 9.5b. The other vertices of the hyper-cube are considerably further from that subset of $\mathbf{X}_P$. The considered vertices of the hyper-cube are $(0, 0, 1, 0, 0, 1)^T$ and $(0, 1, 1, 0, 0, 1)^T$. From the graph it can be guessed that the most significant variable for the selection of a Pareto decision is $x_2$. The values of other variables can be chosen close to the boundary values $x_1 = x_4 = x_5 = 0$, and $x_3 = x_6 = 1$.

## 9.4 The Analysis of Exploratory Guess

The visualization applied to the available data gives a reason to guess the variable $x_2$ as the most significant variable defining a point on the Pareto front. However, the visual exploratory approach does not provide reliable conclusions, and a further analysis is necessary either to prove or reject this guess.

If the guess above is correct, a clearly expressed dependency between the value of $x_2$ and the position of the corresponding point on the Pareto front should exist. To indicate such a dependency in Figure 9.4a, the values of $x_2$ are shown depending on the index of a Pareto optimal solution where the latter are sorted according to the increase of $f_1(\mathbf{x}_i)$.

The linear dependency between $x_2$ and the index of a Pareto optimal decision is clearly seen in Figure 9.4a for the indices which belong to the interval (20, 170), which is much longer than the interval of interest indicated above. Since the points of the discrete representation are distributed over the Pareto front quite densely and uniformly, all characteristics of interest can be presented as functions of the index as an independent variable. However, such a parametric description of the problem data has a disadvantage: the independent variable has no interpretation in the engineering terms of problem formulation.

A variable $t$ varying along the line in Figure 9.4a seems well suited for use as an independent variable for parametric description of the data of interest. The value of $t$ can be interpreted as a value of $x_2$ smoothed along the Pareto front. The values of $t$ in the interval $0.15 \leq t \leq 0.4$ correspond to the kink of the Pareto front. The relationship between $f_1(\mathbf{x}_i)$, $f_2(\mathbf{x}_i)$ and the corresponding value of $t$ is presented by Figure 9.4b. The graphs of $x_2(t)$ and $x_4(t)0.15 \leq t \leq 0.4$ are presented in Figure 9.5a.

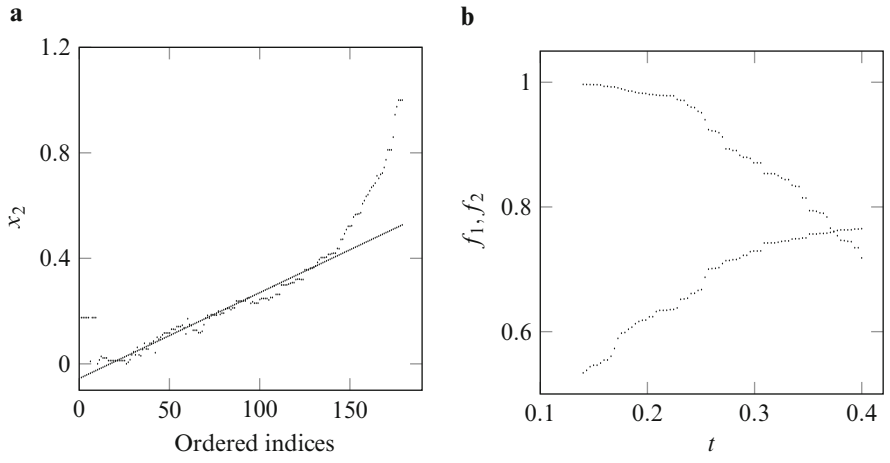


**Fig. 9.4** (**a**) The dependency of $x_2$ on indices of ordered points in the Pareto front, (**b**) The graphs of $f_1(\mathbf{x}(t))$ and $f_2(\mathbf{x}(t))$, where the values of $t$ correspond to the kink of the Pareto front

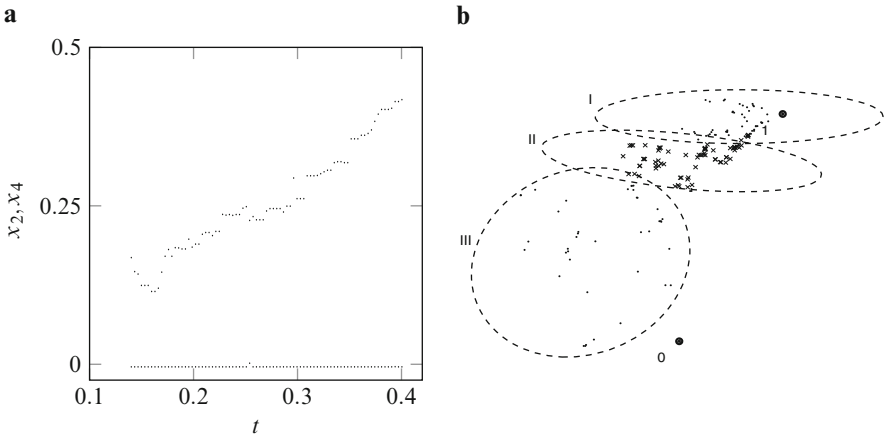


**Fig. 9.5** (**a**) The dependency of $x_2$ (*increasing curve*) and of $x_4$ (*lower graph*) on $t$, $0.15 \leq t \leq 0.4$, (**b**) Two-dimensional image of the discrete representation of a part of the set of Pareto optimal decisions and the vertices of the six-dimensional hyper-cube where the *markers* 0 and 1 show the locations of the closest vertices

To highlight the location of the most interesting part of the Pareto optimal decisions corresponding to the kink of the Pareto front, the part of image of Figure 9.2 is presented in Figure 9.5b. The images of the closest vertices are marked by 0 and 1.

Before the visualization-based analysis, the engineers expected a different impact of the design variables to the trade-off between the Pareto optimal solutions. We refer to [263] where a priori opinion and the final conclusions are discussed in engineering terms.

## 9.5 Remarks

The visualization of the set of Pareto optimal decisions can be a valuable tool to aid the selection of the final most appropriate design from the promising competing designs identified by a bi-objective optimization method. It may enable a decision maker to better understand the relation between Pareto optimal decisions and also the place of most appropriate designs in the overall design space. A version of the method described in this chapter was applied to multi-objective optimization of the design of a biosensor [11].

# Chapter 10
# Multi-Objective Optimization Aided Visualization of Business Process Diagrams

## 10.1 Introduction

Graphs are very popular models of many research subjects. Graphical presentation of a problem is advantageous for heuristic perception and understanding of relations between the objects considered. On the other hand, many efficient algorithmic techniques are available to attack mathematically stated graph problems. Therefore, graph models are especially useful where heuristic abilities of a human user in the formulation of a problem are combined with its algorithmic solution in the interactive mode. In the present chapter we consider the graph models of business processes which, in the literature on the management of business processes, are called business process diagrams (BPDs). To be more precise, a problem of drawing aesthetically pleasing layouts of BPDs is considered. The research of this problem was motivated by the fact that the aesthetic layouts are not only well readable but also most informative and practical [12].

The considered business process management methodology is oriented to managers and consultants either designing a new Small/Medium Enterprise (SME) or searching for the possibilities to improve an existing one. The Business Process Modeling Notation (BPMN) is accepted as a standard for drawing the business process diagrams [146]. The graph drawing aesthetics is comprehensively discussed, e.g., in [15, 171]. Although the problem of graph drawing attracts many researchers, and plenty of publications are available, special cases of that problem frequently cannot be solved by straightforward application of the known methods and algorithms. We cite [172]: "Few algorithms are designed for a specific domain, and there is no guarantee that the aesthetics used for generic layout algorithms will be useful for the visualization of domain-specific diagrams."

The original problem is reduced to a problem of the combinatorial multi-objective optimization. We start with the analysis of the attitude of the potential users of the supposed algorithms towards the relative importance of the



P.M. Pardalos et al., *Non-Convex Multi-Objective Optimization*, Springer
Optimization and Its Applications 123, DOI 10.1007/978-3-319-61007-8_10

widely acceptable criteria of aesthetics. The latter are used as the objectives of the multi-objective optimization problem which formalizes the problem of BPD drawing as an optimization problem.

General rules for drawing BPDs are defined by the standards of BPMN. These standards, however, leave sufficient freedom for choosing the techniques of BPD drawing that are most relevant to the supposed conditions of business process management. We consider in the present chapter the methods for drawing BPDs aimed at the application by the business process management consultants who advise managers of small and medium enterprises planned to be established or undergoing the re-organization.

## 10.2 Visualization of Sequence Flow

A special graph drawing problem, related to the visualization of business process diagrams, is stated as an optimization problem. It is requested to find aesthetically looking paths between the given pairs of vertices of a grid, where the vertices represent the business process flow objects, the paths represent the sequence flow, and the aesthetic criteria are formulated quantitatively. The sites of flow objects on the grid are fixed, and the sequence flow is defined. We state the problem as a problem of combinatorial multi-objective optimization, where the objectives correspond to the generally recognized criteria of aesthetics. To find a solution precisely, the algorithm of linear binary programming CPLEX was applied. For a faster solution, supposed for an interactive mode, a heuristic algorithm is developed. The experimental comparison of the mentioned algorithms is performed to substantiate the applicability of the latter.

To enable a user to completely percept the information presented by the observable part of the BPD, the sub-charts which comprise up to 30 shapes are drawn. A navigation tool aids a user to specify the sub-chart of interest. To denote the flow objects in the considered diagrams, the notation of BPMN, namely the rectangles, the rhombuses, and the circles, is used where these shapes represent the processes, the gateways, and the events correspondingly. The shapes are located in the pool lanes. The sequence flow is represented by the lines constituted of orthogonal straight line segments.

We assume that the location of geometric shapes, that represent business process objects, is fixed on a plane, and a sequence flow is defined. The lines, that represent the sequence flow, should be drawn aiming to obtain an aesthetically pleasing layout. For the general concept of aesthetic graph drawing we refer to [12, 15, 100, 171–173]. Here we reduce this problem to a combinatorial multi-objective optimization.

As the main criteria of aesthetics considered in the literature, e.g., in [12, 100, 173], are: the total length of connectors, the number of crossings, the number of bends, the uniformity of the distribution of shapes in the area of the drawing, and the compatibility of the process sub-flows with the generalized top-down and

left-right direction. However, the influence of these criteria to the perception of the information presented by the considered BPD is different, and depends on the user of the BPD. To implement a visualization method, which ensures the effective perception of a BPD, the attitude of potential users towards the listed above criteria is important. The generalized attitude of a group of the potential users was elicited by means of a psychological experiment [93]. Subsequently, the elicited information is used in the statement of the multi-objective optimization problem the solution of which defines the considered sequence flow by the orthogonal connectors.

### *10.2.1 A Brief Overview of Single-Objective Algorithms Aimed at Visualization of Sequence Flow*

Since we focus on the sequence flow visualization, the differences of the flow objects can be ignored, and a single rectangular shape can be used below to represent the flow objects. Visualization of the graphs, where vertices are drawn as rectangles connected by piecewise vertical and horizontal lines, is commonly used. As the examples, Entity Relationship and UML diagrams can be mentioned among others. The problem of drawing graphs, with the rectangular images of vertices, and with the edges composed of the pieces of vertical and horizontal lines, is considered in many papers. Depending on a supposed application area, the algorithms should satisfy different requirements. Some algorithms, efficient from the point of view of general complexity theory, are described in [117, 232]. A comprehensive review of algorithms oriented to the routing of paths to interconnect the pins on the circuit blocks or pads at the chip boundary is presented in [32]. The general purpose routing algorithms are classified in three groups, namely the maze, line-search, and A*-search groups. Different versions of these algorithms are proposed and investigated with the focus on the asymptotic complexity estimates and on the application in the chip design. However, from the point of view of the BPD drawing, the criteria of aesthetics prevail the criteria important in technological applications emphasized in [32]. In a recent paper [231] a brief review of the available algorithms and software, for the construction of orthogonal connectors, is presented. The experimental testing performed by these authors has shown that some of the available software packages, although provide the automatic orthogonal connector routing, produce the routes which may overlap other objects in the diagram. Popular software packages Microsoft Visio 2007 and Concept Draw Pro5 provide the object-avoiding orthogonal connector routing, but in both cases the aesthetic criteria, such as minimizing distance or number of segments, are not taken into account. We cite the conclusion made in the introduction of [231]: “in all current tools that we are aware of, automatic routing of orthogonal connectors uses ad-hoc heuristics that lead to aesthetically unpleasing routes and unpredictable behavior.” Agreeing with the latter conclusion as well as with the remarks cited in the introductory Section 10.1 we find the developing of new domain-specific algorithms reasonable.

### *10.2.2 Description of the Problem*

The problem of visualization of a sequence flow was posed as a problem of combinatorial multi-objective optimization in [93], where the objectives correspond to the aesthetic criteria. The results of a psychological experiment, described in [93], substantiate the selection of aesthetic criteria that are most important for the potential users. The stated problem was attacked in [93] by a metaheuristic ant colony optimization algorithm. The solutions, found by means of that algorithm in reasonable time, were assessed as acceptable for applications. Nevertheless, the following reason motivated a further investigation: usually a few non-dominated solutions were found. Therefore, a hypothesis seems likely that there exist other Pareto optimal solutions, but they were not found by the metaheuristic algorithm used. To test that hypothesis all the global optima of the criteria in question should be found with a guarantee. To this end, the corresponding single-objective optimization problems were stated in the form of binary-linear optimization, and the CPLEX algorithm [38] was applied to solve them. A combination of CPLEX with the scalarization technique is also used to solve the multi-objective optimization problem in question. However, such a combination, although well suitable to solve small size problems, fails in the case of larger problems because of long computing time. A heuristic algorithm was proposed applicable to the problems of size sufficient for applications.

An example of elementary BPD is presented in Figure 10.1a. In geometric terms, it is requested to draw paths that consist of horizontal and vertical line segments, and connect the given geometric shapes (circles, rhombuses, and rectangles) in the plane. The shapes are located in "swim lanes," at the centers of cells of a rectangular grid, and the paths are requested to consist of horizontal and vertical segments with the ends at circle markers, located on the boarders between "swim lanes," as shown in Figure 10.1b. In terms of the graph theory we are interested in the paths between the given vertices of a graph, defined by a rectangular grid [92] of the type presented in Figure 10.1b. We search here for the paths with the minimum total length, minimum total number of bends, and minimum neighborhood; we refer to [93] for a detailed discussion on the criteria of path desirability. The argumentation presented there substantiates the consideration of the problem of aesthetic drawing of BPDs by means of the methods for multi-objective graph optimization.

Although some similarity is obvious between the considered problem and the classical bi-objective path problem [64, 77], the known methods for the latter do not seem applicable to the former one. This remark also holds for the similarity of the considered problem with routing problems in electronic design [32].

To decrease the computational burden, a reduced version of the graph is considered where the pivot vertices correspond to that of the rectangular grid, and the intermediate vertices correspond to the middle points of cell sides; they are represented by filled circle markers and not filled circle markers in Figure 10.2, respectively. Only the intermediate vertices can be the origin and sink of a path.

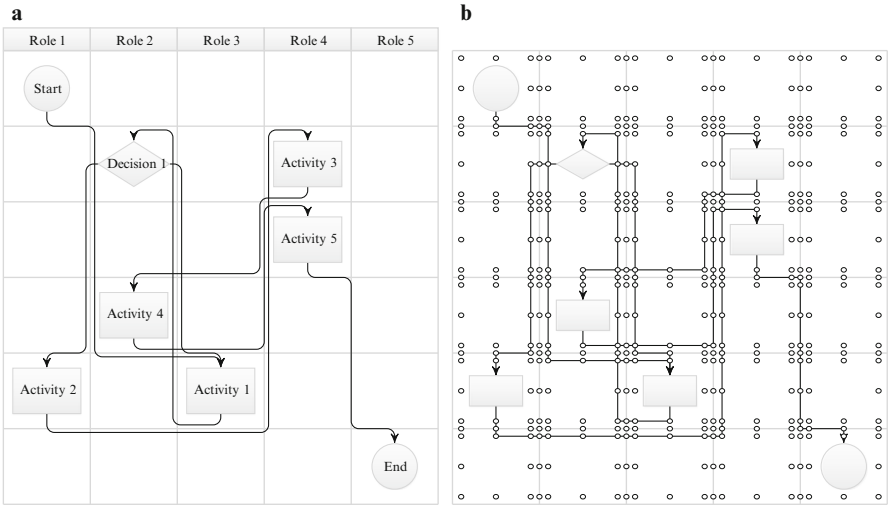


**Fig. 10.1** (**a**) An elementary example of BPD, (**b**) Grid model

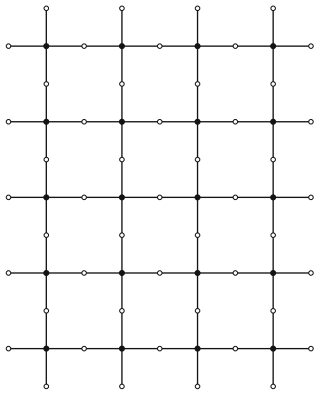

**Fig. 10.2** An example of the reduced graph

The length of an edge between the pivot and intermediate vertices is supposed to be of unit length. The paths in the full graph can be reconstructed from the paths in reduced graphs by the method described in [91].

### *10.2.3 Binary-Linear Model*

To state a multi-objective optimization problem mathematically, we have to introduce variables that define the considered (reduced) graph. Let $p$ denote the number of rows, and $n$ denote the number of columns. The pivot vertices are marked by a double index $ij$ that indicates the crossing of the $i$-th row and $j$-th

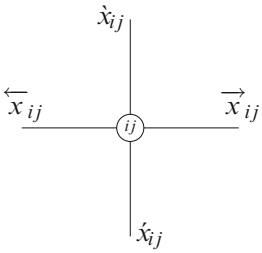


**Fig. 10.3** The notation of a node and of edges

column. The intermediate vertices are indicated by two neighboring pivot vertices. A path is defined by assigning value 1 to the indexed variable $x$, related to the edge which belongs to the path; the values of the variables related to edges not belonging to the path in question are equal to zero. The variable $x$ is indexed as follows: $\grave{x}_{ij}$ and $\acute{x}_{ij}$ are related to the top and bottom adjacent edges of the vertex $ij$, respectively; see Figure 10.3. Similarly $\overleftarrow{x}_{ij}$ and $\overrightarrow{x}_{ij}$ are related to the right and left adjacent edges. The values of $z_{ij}$ mark the path as follows: $z_{ij} = 1$, if the vertex $ij$ is on the path, and $z_{ij} = 0$, if it is not on the path. The values of the introduced variables should satisfy the following equalities:

$$\acute{x}_{ij} - \grave{x}_{i+1,j} = 0,\ \overleftarrow{x}_{i,j+1} - \overrightarrow{x}_{ij} = 0, \tag{10.1}$$

$$\acute{x}_{ij} + \grave{x}_{ij} + \overleftarrow{x}_{ij} + \overrightarrow{x}_{ij} - 2z_{ij} = 0, \tag{10.2}$$

$$\overleftarrow{x}_{i1} = 0,\ \overrightarrow{x}_{in} = 0,\ \grave{x}_{1j} = 0,\ \acute{x}_{mj} = 0, \tag{10.3}$$

$$i = 1, \ldots, p,\ j = 1, \ldots, n.$$

Note that the zero length edges in (10.3) are redundant; but they are included into the model to unify the adjacency of all the pivot vertices.

A path is specified by the start and sink vertices, which are of the intermediate type; such vertices are presented in the mathematical model by the balance equalities as follows:

$$\acute{x}_{ij} + \grave{x}_{i+1,j} = 1, \tag{10.4}$$

if the vertex is located on the $j$-th column between $i$ and $i + 1$ rows, and

$$\overleftarrow{x}_{i,j+1} + \overrightarrow{x}_{ij} = 1, \tag{10.5}$$

if the vertex is located on the $i$-th row between $j$ and $j + 1$ columns.

Before formulating a multi-objective optimization problem, let us note that the important criterion of the total number of bends cannot be reduced to the criteria considered in the classical bi-objective path problem (see, e.g., [64]). We start the analysis from the single path and single objective which is the path length. The shortest path can be found as a solution of the binary-linear programming problem,

where the objective function is

$$\sigma(\mathbf{X}) = \sum_{i=1}^{p} \sum_{j=1}^{n} \left( \overleftarrow{x}_{ij} + \overrightarrow{x}_{ij} + \acute{x}_{ij} + \grave{x}_{ij} \right), \tag{10.6}$$

and the constraints are defined by (10.1)–(10.5), where $\mathbf{X}$ denotes the vector composed of the model variables $x_{ij}$, $z_{ij}$, $i = 1, \ldots, p$, $j = 1, \ldots, n$. The shortest path in such a graph problem could be alternatively found by, e.g., Dijkstra's algorithm which is obviously efficient for such problems. However, we are interested in problems with several paths, in the assessment of which other criteria besides the length should also be taken into account. As the next step, we extend the problem to the case of minimization of the total length of several paths. To define $K$ paths, $K$ replicas of $\mathbf{X}$ should be considered: $\mathbf{X}^k$, $k = 1, \ldots, K$; the upper index here and onward indicates the number of the path. The components of $\mathbf{X}^k$ should satisfy the equations similar to (10.1)–(10.5). The latter binary-linear model includes $5 \times m \times n \times K$ variables and $(3 \times p \times n + p + n) \times K$ equalities. In the model with several paths $X$ denotes the vector of all the variables: $\mathbf{X} = (\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^K)^T$.

To assess the number of bends of a path in the graph model defined by (10.1)–(10.5), we introduce variables $b_{ij}$ which take value 1, in case the path has a bend at the vertex $ij$, and value 0, otherwise. The following equalities and inequalities imply the requested property of $b_{ij}$:

$$\begin{aligned} \overleftarrow{x}_{ij} + \overrightarrow{x}_{ij} - b_{ij} \geq 0,\ \acute{x}_{ij} + \grave{x}_{ij} - b_{ij} \geq 0, \\ \overleftarrow{x}_{ij} + \acute{x}_{ij} - 2b_{ij} \leq 1,\ \overleftarrow{x}_{ij} + \grave{x}_{ij} - 2b_{ij} \leq 1, \\ \overrightarrow{x}_{ij} + \acute{x}_{ij} - 2b_{ij} \leq 1,\ \overrightarrow{x}_{i,j} + \grave{x}_{ij} - 2b_{ij} \leq 1. \end{aligned} \tag{10.7}$$

Let us extend the one-path model complementing vector $\mathbf{X}$ by components $b_{ij}$, and the system of equations (10.1)–(10.5) complementing by (10.7). Then the number of bends of the path, which satisfies (10.1)–(10.5) and (10.7), is equal to

$$\beta(\mathbf{X}) = \sum_{i=1}^{p} \sum_{j=1}^{n} b_{ij}. \tag{10.8}$$

Similarly as above, the $K$ paths model is extended to enable computation of the total number of bends. Summing up, we formulate the problem of aesthetic drawing of $K$ paths as a bi-objective minimization problem with linear objectives

$$\sigma(\mathbf{X}) = \sum_{k=1}^{K} \sigma(\mathbf{X}^k),\ \beta(\mathbf{X}) = \sum_{k=1}^{K} \beta(\mathbf{X}^k), \tag{10.9}$$

and linear equality and inequality constraints defined above. The size of the problem is defined by the number of variables equal to $6 \times p \times n \times K$ as well as the number of equality and inequality constraints equal to $3 \times p \times n + p + n$ and $4 \times p \times n \times K$, respectively.

The considered above criteria for assessing a set of $K$ paths, i.e., the total length and the total number of bends (10.9), are sums of these criteria defined independently for each path. However, the decomposition of the problem into $K$ independent smaller problems here is not an appropriate optimization method because of the following reason. The paths, that are optimal if considered independently, can be infeasible because of the excessive number of replicas of the same edge included into different paths. Restriction for the number of such replicas is implied by the breadth of the boarders between "swim lanes" in the original full graph model as shown in Figure 10.1: the number of replicas of each edge should not exceed 3. The latter requirement supplements the model defined in (10.1)–(10.9) with $4 \times n \times p$ inequality constraints

$$\overleftarrow{h}_{ij} \le \tau,\ \overrightarrow{h}_{ij} \le \tau,\ \acute{v}_{ij} \le \tau,\ \grave{v}_{ij} \le \tau, \tag{10.10}$$

$$\overleftarrow{h}_{ij} = \sum_{k=1}^{K} \overleftarrow{x}_{ij}^{k},\ \overrightarrow{h}_{ij} = \sum_{k=1}^{K} \overrightarrow{x}_{ij}^{k},$$

$$\acute{v}_{ij} = \sum_{k=1}^{K} \acute{x}_{ij}^{k},\ \grave{v}_{ij} = \sum_{k=1}^{K} \grave{x}_{ij}^{k},$$

$$i = 1, \ldots, p,\ j = 1, \ldots, n,$$

where the bound $\tau$ to the number of replicas of edges included into different paths is normally chosen equal to 2, and, in the complicated cases, $\tau$ is chosen equal to 3.

### *10.2.4 Optimization Problems of Interest*

The results in [93] related to multi-objective optimization of paths in the graphs, used as models of BPD, imply the hypothesis that for the frequent in practice instances there exist solutions that are optimal with respect to both considered criteria: the length of path and the number of bends. To test such a hypothesis, the investigation of the following optimization problems is essential

$$\min \sigma(\mathbf{X}), \tag{10.11}$$

$$\min \beta(\mathbf{X}), \tag{10.12}$$

$$\min(\sigma(\mathbf{X}),\ \beta(\mathbf{X}))^{T}, \tag{10.13}$$

where the feasible region is defined by (10.1)–(10.7) and (10.10) with $\tau = 2$ for simple instances, and $\tau = 3$ for more complicated instances.

The first and second optimization problems are single-objective, and their minimizers are coincident if the Pareto front of two-objective problem (10.13)

consists of a single solution. Otherwise, the weighted sum scalarization method could be applicable to solve (10.13) by means of minimizing $\phi(\mathbf{X})$,

$$\min \phi(\mathbf{X}) = w \cdot \sigma(\mathbf{X}) + (1 - w) \cdot \beta(\mathbf{X}),\ 0 < w < 1, \tag{10.14}$$

where the value of $w$ is important only in the case of the incorrect hypothesis on the single-point Pareto front. In that case, $w = \frac{w_2}{w_1 + w_2}$ is a reasonable choice, where $w_1$ is the sum of Manhattan distances between the vertices to be connected, and $w_2$ is the minimum number of bends of the connectors which ignore all the restrictions; $w_1$ and $w_2$ are easily computable. This value of $w$ corresponds to the approximately equal importance of both objectives, i.e., length and bends of paths.

To test the hypothesis stated above, it is enough to compare the minima of (10.11) and (10.12) with the values of $\sigma(\mathbf{X})$ and $\beta(\mathbf{X})$ at the minimum point of (10.14). As an alternative to the weighted sum scalarization a method of lexicographic optimization, e.g., [154] could be applied, however, in such a case single-objective optimization problem should be solved two times.

### *10.2.5 Optimization by Heuristic Methods*

As shown in the previous sections, the guaranteed solutions of instances of the considered problem can be obtained by means of an algorithm of binary-linear programming. However, the binary-linear problems of interest involve large number of variables and restriction; see Table 10.1. It is clear, that in the case of larger problems, the solution time by that method can be too long for interactive systems. Therefore, the development of heuristic algorithms for this problem is important. In the discussion above, we have mentioned that the algorithm based on the idea of ant colony optimization [92, 93] was not quite satisfactory. One of the possible challenges for application of this algorithm is related to difficulties in assigning to an edge a proper value of "heuristic attractiveness" with respect to the criterion "number of bends."

**Table 10.1** The size of the binary-linear programming problem depending on the size of grid $p \times n$ and the number of paths $K$, where $V$ denotes the number of vertices, $E$—the number of edges, $NV$—the number of variables, $NE$—the number of equality constraints, and $NI$—the number of inequality constraints

| $p$ | $n$ | $K$ | $V$ | $E$ | $NV$ | $NE$ | $NI$ |
|---|---|---|---|---|---|---|---|
| 6 | 5 | 8 | 30 | 72 | 1440 | 101 | 960 |
| 6 | 6 | 8 | 36 | 84 | 1728 | 120 | 1152 |
| 8 | 8 | 10 | 64 | 144 | 3840 | 208 | 2560 |
| 10 | 9 | 17 | 90 | 200 | 9180 | 289 | 6120 |
| 10 | 10 | 12 | 100 | 220 | 7200 | 320 | 4800 |
| 12 | 12 | 14 | 144 | 312 | 12,096 | 456 | 8064 |

As potential alternatives we consider two following heuristic algorithms. The first algorithm is an extension of the classical shortest path algorithm for the specific multi-objective problem considered here. The second algorithm is a version of metaheuristic, called Harmony Search [67], which is claimed efficient in some recent publications. Harmony Search is a random search algorithm where moves are interpreted in musical terms. The similar in many respects random search algorithm, called Evolutionary Strategy, was proposed about 40 years ago (see, e.g., [187]) and was shown efficient in indeed many applications.

The problem considered is similar to the shortest path problem which is a classical problem of computer science. Here the vertices of the grid, presented in Figure 10.2, are vertices of the considered graph. The lengths of edges, the number of which is equal to $4 \times n \times p$, is assumed equal to 1. Several paths should be found to connect the vertices which represent the shapes, and the appropriateness of a solution is assessed by the characteristics of the set of found paths. As the start and sink vertices of the paths in question can be only the intermediate vertices of the grid. Although the shortest path problem can be efficiently solved by, e.g., Dijkstra's algorithm, this and other similar algorithms directly are not appropriate here. Our problem is more complicated, since we are interested not only in the total length of paths; the total number of bends, and the number of paths sharing the same edges, are also important. Nevertheless, the similarity between our problem and the shortest path problem induces an idea to construct a heuristic algorithm including the shortest path algorithm as a constituent.

The objective of total length of paths and the objective of total number of bends are not independent, and their dependency suggests an idea to apply the scalarization of the considered multi-objective problem. We propose here a scalarization converting the considered bi-objective optimization problem to a (single-objective) shortest path problem for a modified graph with the same set of vertices but with larger set of edges. The original graph is complemented by the edges that connect the vertices located on the same horizontal and vertical lines. The edges of the original graph are characterized by their length. The similar characteristic of edges of the modified graph is called weight, to mark clearly that the edges of the modified graph are considered. The weight of the complementary edges, adjacent to the vertices $(ij)$ and $(ik)$, is defined as $\delta(|j-k|)$, $i = 1, \ldots, p$, as well as the weight of edges adjacent to the vertices $(ji)$ and $(ki)$, is defined as $\delta(|j-k|)$, $i = 1, \ldots, n$, where $\delta(1) = 1$, $\delta(i) < \delta(j)+\delta(k)$, $i = j+k, j > 0, k > 0$. The function $\delta(\cdot)$, used in the computations described below, was chosen empirically. The weight of the shortest path between the vertices $ij$ and $kl$, $i \neq k$, $j \neq l$, which in the complemented graph consists of two edges, is equal to $\delta(|i-k|) + \delta(|j-l|)$. For those vertices, several paths of the length $|i-k| + |j-l|$ can exist, and all of them are the shortest paths in the original graph. The sequences of vertices found by a shortest path algorithm in the modified graph can be easily complemented by vertices at either horizontal or vertical line segments to reconstruct paths in the original graph. The search for shortest paths in the modified graph is also a heuristic search for a solution of a bi-objective problem stated for the original graph: a path of smaller weight corresponds to a path in the original graph constituted by longer sub-paths and therefore with smaller number of bends.

A solution of the considered problem is a vector of two components: the total length of paths (in the original graph) and the total number of bends. However, the solution found using the proposed algorithm, although possibly optimal with respect to both criteria, can be not feasible since an edge (called onwards critical) can be included into more than $\tau$ paths. To maintain the feasibility, the shortest path problems are solved repeatedly with additional changing constraints related to the critical edges in the current best solution. From the set of paths, that share a critical edge, one is randomly selected to be changed by forbidding the inclusion of the critical edge in that path. Algorithmically such a taboo is easily implementable by assigning the infinite value to the weight of the critical edge with respect to the selected path. The algorithm terminates when a feasible solution is found. Such a solution is optimal subject to taboo defined above. Alternatively the algorithm terminates by the condition of exhausting a maximum admissible number of calls of the shortest path algorithm. In the latter case, a message is printed asking to increase $\tau$.

The heuristic algorithm, described above, is based on the specific properties of the problem in question. The adaptation of metaheuristic algorithms to this problem is a complicated task, as follows from [93]. Nevertheless, one more heuristic was tested, namely the Harmony Search (HS) heuristic, which, as mentioned above, recently was claimed efficient for various problems. A version of HS has been developed for the scalarized bi-objective optimization problem (10.13). The original HS algorithm is used for the minimization of an objective function $f(\mathbf{Z})$ of real-valued variables $z_i$ where $\mathbf{Z} = (z_1, \ldots, z_K)^T$ is interpreted as an improvisation. The main idea of the algorithm is in the search of a mostly aesthetic improvisation recalling improvisations in the Harmony Memory (HM) $\mathbf{Z}^1, \ldots, \mathbf{Z}^{MS}$, where $MS$ is the size of HM; $1 \leq MS \leq 100$. HS is based on three operators (memory consideration; pitch adjustment; random consideration) which are characterized by the parameters described below. The memory consideration rate (MCR) defines the probability to select a value of $z_i$ from those stored in HM; $0.7 \leq MCR \leq 0.99$. The values of variables not selected by the operator of memory consideration are generated randomly. The latter case (the probability of which is equal to 1-MCR) is referred to as random consideration. The pitch adjustment means a random bias (with probability PAR) of the values of $z_i$ generated by the operators considered above. HM is updated in case $f(\mathbf{Z})$ at the newly generated $\mathbf{Z}$ is smaller than $\max_{1 \leq j \leq MS} f(\mathbf{Z}^j)$. The termination condition is defined by the maximum number of improvisations (iterations) $NI$. For the details of the algorithm we refer to [3, 4].

HS seems attractive to be adapted to harmonize paths in graphs according to the aesthetic criteria discussed above. A variable $z_i$ is defined here as a pair of start and sink vertices. The set of values of a variable consists of all feasible paths between the specified start and sink vertices. An improvisation $\mathbf{Z}$ is represented by a set of paths which implement the requested connections. HM is initialized by generating $MS$ shortest paths for each $z_i$. The operators of memory consideration and random consideration are similar to those of the original algorithm. The operator of pitch adjustment is implemented as a replacement of the path, corresponding to the selected $z_i$. The selected path is replaced by a shortest path found under the taboo for an edge of the selected path. If the path contains a critical edge, then this edge is declared taboo. Otherwise, a taboo edge is chosen at random.

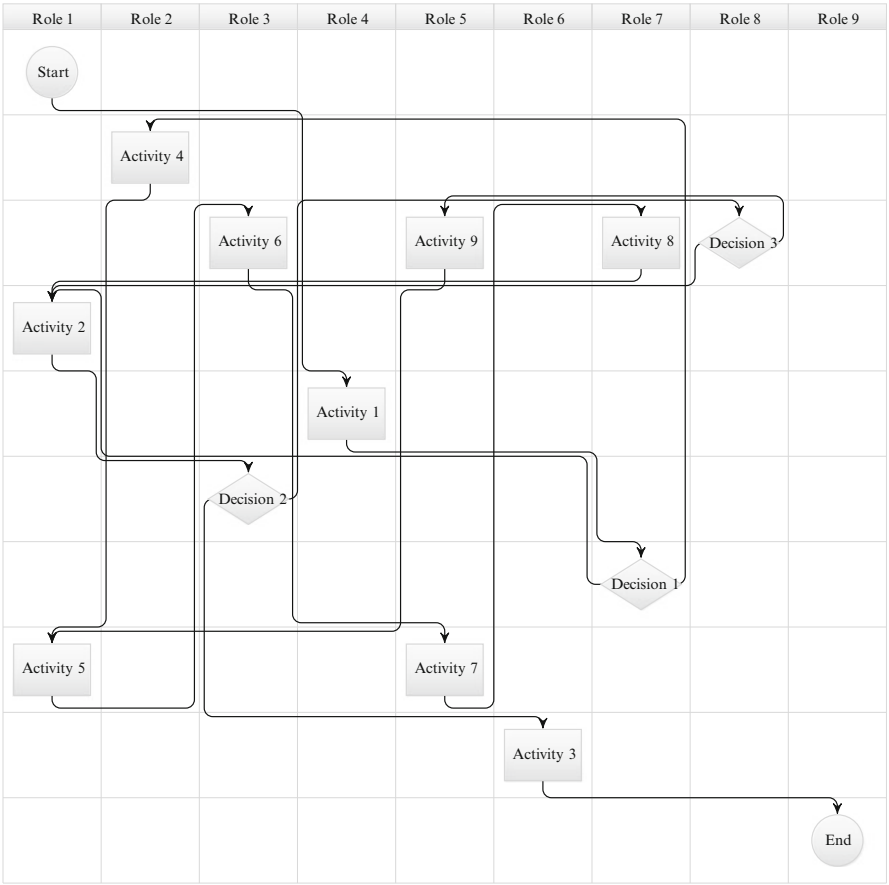


**Fig. 10.4** The second example of BPD

### *10.2.6 Numerical Experiments*

The performance of the proposed algorithms has been tested using two real life problems and a sample of randomly generated data. The general input data contains: the size of the grid, the types and the positions of shapes, and the links between shapes. The respective optimization problems have been formulated, and the described above algorithms have been applied; the values of method's parameters are presented below. A computer with Intel i7-4702MQ CPU 2.20 GHz, 8 GB RAM was used for experiments.

The first test problem, presented in Figure 10.1, is indeed an elementary example of BPD. The second problem, which is characterized as a complicated one, is presented in Figure 10.4. These figures present not only the input data but also the found optimal solutions. For these test problems the hypothesis on the single

**Table 10.2** Results for the practical problems

| | Test problem 1 | | | Test problem 2 | | |
|---|---|---|---|---|---|---|
| | $\sigma(\mathbf{X}_0)$ | $\beta(\mathbf{X}_0)$ | Time (s) | $\sigma(\mathbf{X}_0)$ | $\beta(\mathbf{X}_0)$ | Time (s) |
| LP | 33 | 14 | 0.33 | 103 | 24 | 21.22 |
| RSP | 33.39 | 14 | 0.17 | 103.47 | 24 | 2.17 |
| HS | 33.27 | 14.22 | 30.14 | 104.66 | 25.70 | 120.35 |

**Table 10.3** The mean values and standards of $\beta(\mathbf{X}_0)$ and $\sigma(\mathbf{X}_0)$ where $\mathbf{X}_0$ is the ideal point

| | | $n = 6$, $p = 6$, $K = 8$ | $n = 8$, $p = 8$, $K = 10$ | $n = 10$, $p = 10$, $K = 12$ | $n = 12$, $p = 12$, $K = 14$ |
|---|---|---|---|---|---|
| $\sigma(\mathbf{X}_0)$ | Mean | 27.8947 | 49.8462 | 75.7000 | 112.0000 |
| | Std | 4.7007 | 7.4710 | 13.0560 | 18.4145 |
| $\beta(\mathbf{X}_0)$ | Mean | 11.0526 | 15.0000 | 18.6400 | 22.5581 |
| | Std | 1.3744 | 1.1662 | 1.7815 | 1.7086 |

Pareto optimal point has been proved to be true: the results of minimization of objectives (10.11), (10.12) and of the aggregated objective (10.14) by the CPLEX algorithm [38] have shown that the corresponding values of $\sigma(\cdot)$ and $\beta(\cdot)$ are coincident. These values are presented in Table 10.2, line LP. The averaged values of the objective functions computed at the points found by the considered heuristic algorithms are also presented in Table 10.2. The heuristic algorithms are randomized, therefore their averaged results of 100 runs are presented. A version of the heuristic algorithm of randomized search for shortest paths (RSP) was used where $\delta(i) = \sqrt{i}$ and the maximum admissible number of calls of the shortest path algorithm was equal to $K \cdot 10$ ($K$ denotes the number of links). The HS algorithm with the following parameters was used for these experiments: $NI = 10{,}000$, $MS = 100$. The values of parameters of the heuristic algorithms have been chosen experimentally: the increase of these values imply increase of computing time without decrease of average errors.

To assess the performance of the considered algorithms they have been tested using 100 randomly generated test problems. The optimization problems were related to BPD where shapes were located randomly on the grid of size $n \times p$ ($6 \times 6, 8 \times 8, 10 \times 10$, and $12 \times 12$) with the exception of "Start" and "Finish" that were located at their specific corners. The number of rectangles was equal to $n - 2$, and the number of rhombuses was equal to 2. Hundred randomly generated problems have been solved by all the algorithms considered.

The solution to the problem (10.14) found by the binary-linear programming algorithm was coincident with the ideal point for all the test data generated (Table 10.3). Let us note that the ideal points are constituted of the minima of both objectives found by the same algorithm. Thus, in all the cases the hypothesis of a single Pareto optimal solution has been proved truth.

**Table 10.4** The mean values and standards of errors of heuristic algorithms

| | | $n = 6$, $p = 6$, $K = 8$ | $n = 8$, $p = 8$, $K = 10$ | $n = 10$, $p = 10$, $K = 12$ | $n = 12$, $p = 12$, $K = 14$ |
|---|---|---|---|---|---|
| *RSP* | | | | | |
| $\Delta\sigma$ | Mean | 0.0789 | 0.1538 | 0.1200 | 0.0930 |
| | Std | 0.3588 | 0.8806 | 0.4798 | 0.3661 |
| $\Delta\beta$ | Mean | 1.2105 | 1.1538 | 1.3600 | 1.0930 |
| | Std | 0.8433 | 0.8806 | 0.8514 | 0.9714 |
| *HS* | | | | | |
| $\Delta\sigma$ | Mean | 0.3400 | 1.0408 | 0.6154 | 0.4186 |
| | Std | 1.4230 | 1.2576 | 1.5175 | 1.5466 |
| $\Delta\beta$ | Mean | 0.4200 | 0.2653 | 0.7692 | 1.3023 |
| | Std | 0.6091 | 0.6382 | 0.9316 | 0.9395 |

**Table 10.5** Running time in seconds

| | $n = 6$, $p = 6$, $K = 8$ | $n = 8$, $p = 8$, $K = 10$ | $n = 10$, $p = 10$, $K = 12$ | $n = 12$, $p = 12$, $K = 14$ | $n = 14$, $p = 14$, $K = 16$ | $n = 16$, $p = 16$, $K = 18$ | $n = 18$, $p = 18$, $K = 20$ | $n = 20$, $p = 20$, $K = 22$ |
|---|---|---|---|---|---|---|---|---|
| LP | 1.38 | 4.14 | 53.27 | 129.84 | – | – | – | – |
| RSP | 0.11 | 0.46 | 1.36 | 3.30 | 6.46 | 12.54 | 22.29 | 36.87 |
| HS | 13.02 | 26.06 | 45.99 | 89.11 | 157.06 | 130.12 | 270.69 | 377.88 |

The same randomly generated problems have been solved also by means of RSP and HS. The errors of the found estimates $\Delta\sigma$ and $\Delta\beta$ have been recorded for every randomly generated problem. The mean values and standard deviations of errors are presented in Table 10.4.

The running time of the considered algorithms for various problem sizes is presented in Table 10.5. As follows from Tables 10.3 and 10.5 the performance of the proposed algorithm RSP is similar to that of CPLEX adapted to the problem in question. However, computing time of RSP is much shorter; RSP is applicable to the problems of size not appropriate to CPLEX.

The results of numerical experiments have shown the truth of the hypothesis on a single Pareto optimal solution at least for the test problems used here. The single Pareto optimal solution of the considered problems was coincident with the ideal point, i.e., the minima of both objectives are components of the single Pareto optimal solution. It is interesting to mention that such a property is intrinsic for the worst-case multi-objective global optimization problem as shown in Chapter 6. The specially developed heuristic algorithm (RSP) has found sufficiently precise solutions of test problems in acceptable time; it is also applicable to large problems. The experimentation with HS has corroborate the conclusion from the results of [93]: the adaptation of a general metaheuristic to the solution of the considered problem is difficult.

### *10.2.7 Discussion and Remarks*

A problem of aesthetical drawing of BPDs is an interesting multi-objective optimization problem for special graphs. The problems of medium size are solvable in acceptable time using the CPLEX algorithm applied to their scalarized versions, stated as binary-linear programming problems. The heuristic algorithm proposed finds acceptably accurate solutions to medium size problems and is applicable to large problems.

The experiments with randomly generated test problems, similar to those in some real world applications, have corroborated the hypothesis that the ideal point is a single Pareto optimal solution to problems of this kind. An interesting theoretical problem emerges to find the mathematical conditions implying this phenomenon.

## 10.3 Multi-Objective Allocation of Shapes

### *10.3.1 A Problem of the Allocation of Shapes*

The algorithms considered in the previous section are aimed to solve one subproblem of visualization of Business Process Diagrams. The other subproblem is allocation of shapes [222, 258]. Let us recall that a business process diagram consists of elements (e.g., activities, events, and gateways) which should be drawn according to the rules of Business Process Modeling Notation. The elements of diagrams are drawn as shapes which are allocated in a pool divided by the (vertical) swimlanes according to the function or role. We consider a restricted set of shapes constituted by a rectangle, a rhombus, and a circle. In this section we are interested only in the allocation of shapes, i.e., we ignore the interpretation of the diagram in terms of the visualized business process.

The input for the problem of the drawing a diagram is several lists containing shapes which should be allocated in the same swimlanes. The input data also defines the pairs of shapes which should be connected. It is requested to allocate shapes in swimlanes, and the swimlanes with regard to each other aiming at aesthetical appeal of the drawing. The problem is reduced to a problem of multi-objective combinatorial optimization. In this section the bi-objective problem is considered taking into account two objectives: the length of connectors and compatibility of the process flow with the top-down left-right direction.

To formulate a mathematical optimization problem we consider graphs the vertices of which are located at the nodes of a rectangular grid and the edges are composed of segments of horizontal and vertical lines. The shapes are attributed to the swimlanes in the description of a business process. The problem considered here is to find the mutual disposition of the swimlanes and to allocate the shapes to nodes inside of swimlanes. Subsequently another multi-objective optimization problem is solved where the edges of the graph are sought which represent aesthetically pleasing drawings of paths between the shapes as described in the previous section.

### *10.3.2 Allocation of Shapes by Multi-Objective Optimization*

A problem of allocation of the shapes (flow objects) in business process diagram was formulated above as a multi-objective optimization problem for graphs. In this problem it is requested to allocate shapes in swimlanes, and the swimlanes with regard to each other aiming at aesthetical appeal of the drawing. Let us denote the number of rows by $n_r$ and the number of columns by $n_c$. The connectors show the sequence flow: two flow objects are connected if one directly precedes another. The shapes are allocated in such a way so that the connected shapes were close to each other and that the flow of connections would be directed from left to right and from top to bottom. Two objectives are simultaneously optimized:

- Minimization of the total length of connectors: the sum of city-block distances between connected shapes is minimized.
- Minimization of the number of right down flow violations: the number of times the preceding shape in the connection is not higher than and is to the right from the following shape is minimized.

The data of the problem consists of the roles (or functions) the shapes belong to and the list of connections. Let us denote the number of shapes by $n$ and the roles corresponding to shapes by $\mathbf{d}$, where $d_i,\ i = 1, \ldots, n$ define the role number of each shape. The connections are defined by $n_k \times 2$ matrix $\mathbf{K}$ whose rows define connecting shapes where $k_{i1}$ precedes $k_{i2}$.

The shapes belonging to the same role should be shown in the same column (swimlane), however, the columns may be permuted. Therefore, part of decision variables define assignment of roles to columns. Let us denote the assignment of roles to columns by $\mathbf{y}$ which is a permutation of $(1, \ldots, n_c)$ and $y_i$ defines the column number of the $i$th role. Another part of decision variables define assignment of shapes to rows. Let us denote this assignment by $\mathbf{x}$, where $x_i$ defines the row number of the $i$th shape.

We define the objectives as follows. The length of orthogonal connector cannot be shorter than the city-block distance between the connected points. Therefore, we model the potential length of a connector as the city-block distance between shapes. The total length of connectors is calculated as

$$f_1(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n_k} |x_{k_{i1}} - x_{k_{i2}}| + |y_{d_{k_{i1}}} - y_{d_{k_{i2}}}|.$$

The number of right down flow violations is calculated as

$$f_2(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n_k} v_d(k_{i1}, k_{i2}) + v_r(k_{i1}, k_{i2}),$$

where down flow violation is

$$v_d(i, j) = \begin{cases} 1, \text{ if } x_i \geq x_j, \\ 0, \text{ otherwise,} \end{cases}$$

and right flow violation is

$$v_r(i,j) = \begin{cases} 1, \text{ if } y_{d_i} > y_{d_j}, \\ 0, \text{ otherwise.} \end{cases}$$

The connection of two shapes in the same row violates down flow since the bottom or side of preceding shape connects to the top of the following shape.

In such a definition the objective functions are separable into two parts, one is dependent only on the decision variables $\mathbf{x}$ and another on $\mathbf{y}$:

$$\begin{aligned} f_1(\mathbf{x},\mathbf{y}) &= f_{1x}(\mathbf{x}) + f_{1y}(\mathbf{y}), \\ f_{1x}(\mathbf{x}) &= \sum_{i=1}^{n_k} |x_{k_{i1}} - x_{k_{i2}}|, \\ f_{1y}(\mathbf{y}) &= \sum_{i=1}^{n_k} |y_{d_{k_{i1}}} - y_{d_{k_{i2}}}|, \\ f_2(\mathbf{x},\mathbf{y}) &= f_{2x}(\mathbf{x}) + f_{2y}(\mathbf{y}), \\ f_{2x}(\mathbf{x}) &= \sum_{i=1}^{n_k} v_d(k_{i1}, k_{i2}), \\ f_{2y}(\mathbf{y}) &= \sum_{i=1}^{n_k} v_r(k_{i1}, k_{i2}). \end{aligned}$$

Therefore, the problem can be decomposed into two: find the non-dominated solutions ($f_{1x}$,$f_{2x}$) and the non-dominated solutions ($f_{1y}$,$f_{2y}$). The non-dominated decisions of two problems are then aggregated and the non-dominated decisions of the whole problem are retained. The number of decisions of the second problem is much smaller than that of the first problem. We approach the first problem by using a branch and bound algorithm. The second problem may be solved by a similar algorithm or even using enumeration of all decisions.

The number of decisions of the first problem is

$$\prod_{i=1}^{n_c} \frac{n_r!}{(n_r - n_i)!},$$

where $n_i$ is the number of shapes assigned to $i$th role. The number of decisions of the second problem is $n_c!$. For example, if we have three roles, there are 4 objects in one role and 6 objects in each other two roles, and we want to fit the diagram in 7 rows, the number of decisions of the second problem is $3! = 6$ and the number of decisions of the first problem is

$$\frac{7!}{3!} \times 7! \times 7! = 21{,}337{,}344{,}000$$

which is a very large number.

### 10.3.3 Branch and Bound Algorithm for Shape Allocation

As it was described in the previous subsection, the number of decisions of the second problem is much smaller than that of the first problem. Therefore, we solve the second problem by enumeration of all decisions while the first problem is solved using the branch and bound algorithm. In the case the number of columns is larger, branch and bound algorithm can be used for the second problem as well.

The assignment of shapes to rows is denoted by $\mathbf{x}$, where $x_i$ defines the row number of the $i$th shape. Two shapes of the same role cannot be assigned to the same row:

$$x_i \neq x_j,\ i \neq j,\ d_i = d_j,$$

where $d_i$ defines the role number of the $i$th shape.

Therefore the optimization problem is

$$\min \mathbf{f}(\mathbf{x}),$$

$$f_1(\mathbf{x}) = \sum_{i=1}^{n_k} |x_{k_{i1}} - x_{k_{i2}}|, \tag{10.15}$$

$$f_2(\mathbf{x}) = \sum_{i=1}^{n_k} v_d(k_{i1}, k_{i2}), \tag{10.16}$$

$$\text{s.t. } x_i \neq x_j,\ i \neq j,\ d_i = d_j, \tag{10.17}$$

$$x_i \in \{1, \ldots, n_r\}.$$

A set of decisions (subset of feasible decisions) may correspond to a partial decision where only some shapes are assigned to rows. Therefore, the partial decision is represented by the assignment $\mathbf{x}$ of $n' < n$ shapes to rows. Let us denote a bounding vector for the objective functions as

$$\mathbf{b}(\mathbf{x}, n') = \left( \sum_{i=1}^{n_k} c_1(i, \mathbf{x}, n'),\ \sum_{i=1}^{n_k} c_2(i, \mathbf{x}, n') \right),$$

where $c_1(i, \mathbf{x}, n')$ and $c_2(i, \mathbf{x}, n')$ denote the contribution of the $i$th connector to the bounds. When both connecting shapes are assigned in the partial decision, the direct contribution of the connector can be computed. In the contrary case, the most favorable contribution may be estimated.

The bounds for objective functions of allocation of the shapes to rows in business process diagrams (10.15) and (10.16) may be computed as follows. The total length of connectors cannot be smaller than the length of connectors among already assigned shapes. Similarly the number of down flow violations cannot be smaller than violated by already assigned shapes. Such bounds involve only the direct contribution:

$$c_1^1(i,\mathbf{x},n') = \begin{cases} |x_{k_{i1}} - x_{k_{i2}}|, \text{ if } k_{i1} \le n',\ k_{i2} \le n', \\ 0, \qquad\qquad \text{otherwise,} \end{cases}$$

$$c_2^1(i,\mathbf{x},n') = \begin{cases} 1, \text{ if } k_{i1} \le n',\ k_{i2} \le n',\ x_{k_{i1}} \ge x_{k_{i2}}, \\ 0, \text{ otherwise.} \end{cases}$$

A single ideal vector may be used as a bounding vector

$$\mathbf{b}^1(\mathbf{x},n') = \left(b_1^1(\mathbf{x},n'),\ b_2^1(\mathbf{x},n')\right) \tag{10.18}$$

composed of two lower bounds for each objective function:

$$b_1^1(\mathbf{x},n') = \sum_{i=1}^{n_k} c_1^1(i,\mathbf{x},n'),$$

$$b_2^1(\mathbf{x},n') = \sum_{i=1}^{n_k} c_2^1(i,\mathbf{x},n').$$

Let us investigate a possible contribution of later assigned shapes. If the connected shapes belong to the same role, then the vertical distance between them cannot be zero because two shapes cannot be assigned to the same row in the same column, see (10.17). Therefore, connectors contribute at least one to the vertical distance if they connect two shapes of the same role:

$$c_1^2(i,\mathbf{x},n') = \begin{cases} |x_{k_{i1}} - x_{k_{i2}}|, \text{ if } k_{i1} \le n',\ k_{i2} \le n', \\ 1, \qquad\qquad \text{if } k_{i1} > n' \text{ or } k_{i2} > n',\ d_{k_{i1}} = d_{k_{i2}}, \\ 0, \qquad\qquad \text{otherwise.} \end{cases}$$

In such a case the bounding vector may be

$$\mathbf{b}^2(\mathbf{x},n') = \left(b_1^2(\mathbf{x},n'),\ b_2^1(\mathbf{x},n')\right), \tag{10.19}$$

where

$$b_1^2(\mathbf{x},n') = \sum_{i=1}^{n_k} c_1^2(i,\mathbf{x},n').$$

If one of the connected shapes is already assigned, a favorable contribution of the connector may be estimated by looking at available places for the other connected shape:

$$c_1^3(i,\mathbf{x},n') = \begin{cases} |x_{k_{i1}} - x_{k_{i2}}|, & \text{if } k_{i1} \le n',\ k_{i2} \le n', \\ \min_{x \neq x_j,\ d_j = d_{k_{i2}}} |x_{k_{i1}} - x|, & \text{if } k_{i1} \le n',\ k_{i2} > n', \\ \min_{x \neq x_j,\ d_j = d_{k_{i1}}} |x - x_{k_{i2}}|, & \text{if } k_{i1} > n',\ k_{i2} \le n', \\ 1, & \text{if } k_{i1} > n' \text{ and } k_{i2} > n', d_{k_{i1}} = d_{k_{i2}}, \\ 0, & \text{otherwise,} \end{cases}$$

$$c_2^3(i,\mathbf{x},n') = \begin{cases} 1, \text{ if } k_{i1} \le n',\ k_{i2} \le n',\ x_{k_{i1}} \ge x_{k_{i2}}, \\ 1, \text{ if } k_{i1} \le n',\ k_{i2} > n',\ \nexists x > x_{k_{i1}} :\ x \neq x_j,\ d_j = d_{k_{i2}}, \\ 1, \text{ if } k_{i1} > n',\ k_{i2} \le n',\ \nexists x < x_{k_{i2}} :\ x \neq x_j,\ d_j = d_{k_{i1}}, \\ 0, \text{ otherwise.} \end{cases}$$

The bounding vector involving these contributions

$$\mathbf{b}^3(\mathbf{x},n') = \left(b_1^3(\mathbf{x},n'),\ b_2^3(\mathbf{x},n')\right), \tag{10.20}$$

where

$$b_1^3(\mathbf{x},n') = \sum_{i=1}^{n_k} c_1^3(i,\mathbf{x},n'),$$

$$b_2^3(\mathbf{x},n') = \sum_{i=1}^{n_k} c_2^3(i,\mathbf{x},n'),$$

is better than one defined in (10.19), but it is more expensive to compute.

The most favorable contribution of the connector to the vertical distance is zero. It is achieved when the connected shapes belong to different roles and could be assigned to the same row. However, in such a situation the contribution of the connector to down flow violation should be one because bottom or side of one shape connects to the top of the other shape. On the contrary, the most favorable contribution of the connector to down flow violation is zero when preceding shape is higher than the following shape ($x_{k_{i1}} < x_{k_{i2}}$). However, in such a situation the vertical distance between the shapes would be at least one. Taking this into account a bounding front may be built:

$$\mathbf{B}^4(\mathbf{x},n') = \left\{\left(b_1^2(\mathbf{x},n') + j,\ b_2^1(\mathbf{x},n') + n_j - j\right) :\ j = 0,\ldots,n_j\right\}, \tag{10.21}$$

where $n_j$ is the number of connectors where at least one of the shapes is not assigned in the partial decision and the shapes belong to different roles:

$$n_j = \left|\left\{i : k_{i1} > n' \text{ or } k_{i2} > n',\ d_{k_{i1}} \neq d_{k_{i2}},\ i = 1,\ldots,n_k\right\}\right|.$$

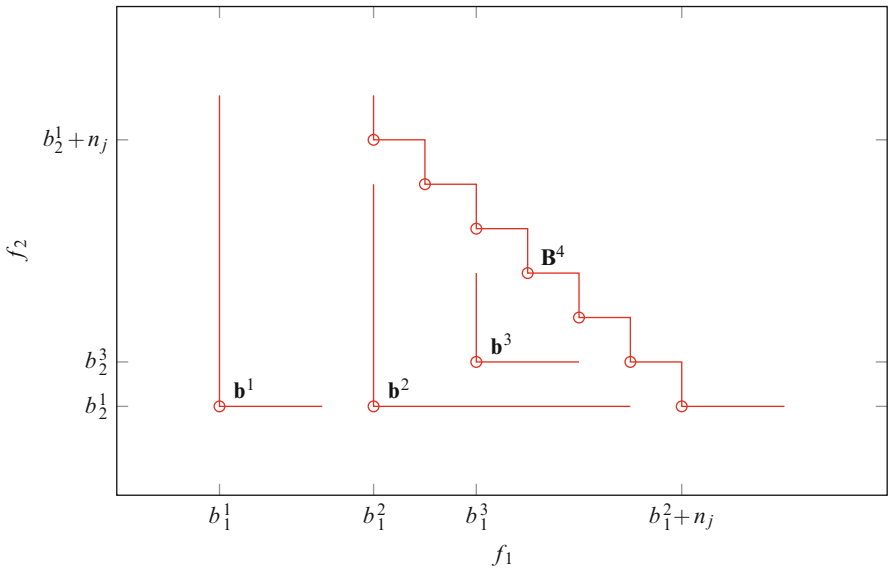


**Fig. 10.5** Bounding vectors and bounding front in shape allocation problem

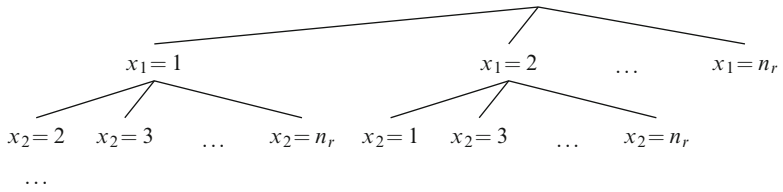


**Fig. 10.6** Search tree of branch and bound for the allocation of shapes in business process diagrams

Bounding vectors $\mathbf{b}^1(\mathbf{x}, n')$, $\mathbf{b}^2(\mathbf{x}, n')$, $\mathbf{b}^3(\mathbf{x}, n')$, and bounding front $\mathbf{B}^4(\mathbf{x}, n')$ are illustrated in Figure 10.5. The bounding front $\mathbf{B}^4$ is better (tighter) than the bounding vectors, $\mathbf{b}^3$ is better than $\mathbf{b}^2$ which is better than $\mathbf{b}^1$.

We build a branch and bound algorithm for the multi-objective problem for allocation of the shapes in business process diagrams using the depth first selection. The search tree of the branch and bound for allocation of the shapes is illustrated in Figure 10.6. The levels of the tree represent different shapes. The branches of the tree represent assignment of the flow objects to rows of business process diagram. Of course the shape cannot be assigned to the row where another shape of the same role (swimlane) is already assigned.

The algorithm for multi-objective problem for allocation of the shapes in business process diagrams can be outlined in the following steps:

**Step 1** *Initialization*: Form the first possible assignment of shapes to rows in $\mathbf{x}$. Set $n' \leftarrow n + 1$.

**Step 2** Repeat while $n' > 0$

- *Evaluation of complete decision*: If the current decision is complete ($n' \geq n$)
  - Set $n' \leftarrow n$.
  - Compute

$$f_{1x}(\mathbf{x}) = \sum_{i=1}^{n_k} |x_{k_{i1}} - x_{k_{i2}}|,$$

$$f_{2x}(\mathbf{x}) = \sum_{i=1}^{n_k} v_d(k_{i1}, k_{i2}).$$

  - If no decisions in the current approximation $\mathbf{S}$ of the efficient set dominate the current decision $\mathbf{x}$, add it to the approximation:

$$\text{If} \quad \nexists \mathbf{a} \in \mathbf{S} : \mathbf{a} \succ \mathbf{x}, \ \text{then} \ \ \mathbf{S} \leftarrow \mathbf{S} \cup \{\mathbf{x}\}.$$

  - If there are decisions in the current approximation $\mathbf{S}$ of the efficient set dominated by the current decision, remove them from the approximation:

$$\mathbf{S} \leftarrow \mathbf{S} \setminus \{\mathbf{a} \in \mathbf{S} : \mathbf{x} \succ \mathbf{a}\}.$$

- Otherwise
  - *Bounding*: Compute $\mathbf{b}^1(\mathbf{x}, n')$ using (10.18), $\mathbf{b}^2(\mathbf{x}, n')$ using (10.19), $\mathbf{b}^3(\mathbf{x}, n')$ using (10.20), or $\mathbf{B}^4(\mathbf{x}, n')$ using (10.21).
  - *Pruning*: If $\mathbf{b}^1(\mathbf{x}, n')$, $\mathbf{b}^2(\mathbf{x}, n')$, $\mathbf{b}^3(\mathbf{x}, n')$, or every $\mathbf{b} \in \mathbf{B}^4(\mathbf{x}, n')$ is dominated by a decision from the current approximation $\mathbf{S}$ of the efficient set, reduce $n'$.
- *Branching or retracting* (depth first search): Update $x_{n'}$ by an available number and increase $n'$ or reduce $n'$ if there are no further numbers available.

**Step 3** Find non-dominated solutions $\mathbf{Q}$ of the second problem.

**Step 4** Aggregate non-dominated solutions of two problems,

$$\begin{aligned} f_1(\mathbf{x}, \mathbf{y}) &= f_{1x}(\mathbf{x}) + f_{1y}(\mathbf{y}), \\ f_2(\mathbf{x}, \mathbf{y}) &= f_{2x}(\mathbf{x}) + f_{2y}(\mathbf{y}), \end{aligned} \quad \mathbf{x} \in \mathbf{S}, \ \mathbf{y} \in \mathbf{Q},$$

and retain non-dominated decisions of the whole problem.

**Table 10.6** Data of example business process diagram

| Flow objects | | |
|---|---|---|
| No. | Role | Name |
| 1 | Role 1 | Event 1 |
| 2 | Role 1 | Activity 1 |
| 3 | Role 1 | Gateway 1 |
| 4 | Role 1 | Activity 4 |
| 5 | Role 1 | Activity 5 |
| 6 | Role 1 | Activity 7 |
| 7 | Role 2 | Event 2 |
| 8 | Role 2 | Activity 3 |
| 9 | Role 2 | Gateway 4 |
| 10 | Role 2 | Activity 6 |
| 11 | Role 2 | Gateway 3 |
| 12 | Role 2 | Event 4 |
| 13 | Role 3 | Activity 8 |
| 14 | Role 3 | Activity 2 |
| 15 | Role 3 | Gateway 2 |
| 16 | Role 3 | Event 3 |

| Connections | | |
|---|---|---|
| $i$ | $k_{i1}$ | $k_{i2}$ |
| 1 | 1 | 2 |
| 2 | 2 | 3 |
| 3 | 3 | 4 |
| 4 | 3 | 5 |
| 5 | 7 | 8 |
| 6 | 6 | 9 |
| 7 | 10 | 9 |
| 8 | 5 | 11 |
| 9 | 9 | 12 |
| 10 | 11 | 12 |
| 11 | 5 | 13 |
| 12 | 13 | 9 |
| 13 | 14 | 11 |
| 14 | 14 | 13 |
| 15 | 8 | 15 |
| 16 | 15 | 11 |
| 17 | 14 | 15 |
| 18 | 16 | 14 |

### *10.3.4 Computational Experiments*

We performed computational experiments on problems of business process diagrams. A computer with Intel i7-2600 CPU 3.40 GHz, 8 GB RAM, and Ubuntu 12.10 Linux was used for experiments. The branch and bound algorithm has been implemented in C/C++ and built with g++ 4.7.2 compiler.

The data of the first example business process are given in Table 10.6. There are three roles in this business process, 4 shapes are in one role, and 6 shapes are in each other two roles. The smallest number of rows to fit the business process diagram is 6. In this case, the number of different assignments of roles to columns is 6 and shapes to rows is 186,624,000. In the case of 7 rows the number is indicated in Section 10.3.2 and is 21,337,344,000.

Branch and bound leads to enumeration of all possible decisions in the worst case—when the bounds do not help to discard any sets of decisions. It is interesting to investigate how far the practical case is from the worst case. Therefore, we compare the results of the developed branch and bound algorithm with complete enumeration, where all possible assignments of shapes to rows are enumerated and non-dominated are retained. The computational time ($t$) and the number of function evaluations (NFE) for the algorithms solving the example problem are presented in Table 10.7. NFE for complete enumeration coincide with computed numbers of possible different assignments. Branch and bound is approximately 70 times faster

**Table 10.7** Experimental comparison of the branch and bound algorithm and complete enumeration

| Number | Branch and bound | | Complete enumeration | | Speed-up | |
|---|---|---|---|---|---|---|
| of rows | $t$, s | NFE | $t$, s | NFE | $t$ | NFE |
| 6 | 0.15 | 2374407 + 6 | 10.96 | 186624000 + 6 | 73 | 79 |
| 7 | 1.44 | 28118029 + 6 | 1112.63 | 21337344000 + 6 | 773 | 759 |
| 8 | 9.49 | 192605603 + 6 | 32476 | 682795008000 + 6 | 3422 | 3545 |

**Table 10.8** Pareto optimal decisions of the example business process diagram problem with 7 rows

| **x** | | | | | | | | | | | | | | | | **y** | | | **f(x, y)** | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | $x_8$ | $x_9$ | $x_{10}$ | $x_{11}$ | $x_{12}$ | $x_{13}$ | $x_{14}$ | $x_{15}$ | $x_{16}$ | $y_1$ | $y_2$ | $y_3$ | $f_1$ | $f_2$ | Figure |
| 1 | 2 | 3 | 6 | 4 | 5 | 2 | 3 | 6 | 4 | 5 | 7 | 5 | 3 | 4 | 2 | 1 | 3 | 2 | 33 | 1 | 10.8c |
| 1 | 2 | 3 | 5 | 4 | 6 | 2 | 3 | 6 | 4 | 5 | 7 | 5 | 3 | 4 | 2 | 1 | 3 | 2 | 31 | 2 | |
| 1 | 2 | 3 | 6 | 4 | 5 | 2 | 3 | 6 | 5 | 4 | 7 | 5 | 3 | 4 | 2 | 1 | 3 | 2 | 30 | 3 | |
| 1 | 2 | 3 | 5 | 4 | 6 | 2 | 3 | 6 | 5 | 4 | 7 | 5 | 3 | 4 | 2 | 1 | 3 | 2 | 28 | 4 | 10.8b |
| 1 | 2 | 3 | 5 | 4 | 6 | 2 | 3 | 6 | 7 | 4 | 5 | 5 | 3 | 4 | 2 | 1 | 3 | 2 | 26 | 6 | |
| 1 | 2 | 3 | 5 | 4 | 6 | 2 | 3 | 6 | 7 | 4 | 5 | 5 | 4 | 3 | 2 | 1 | 3 | 2 | 25 | 8 | |
| 1 | 2 | 3 | 5 | 4 | 6 | 2 | 3 | 6 | 7 | 4 | 5 | 5 | 3 | 4 | 2 | 1 | 2 | 3 | 25 | 8 | |
| 1 | 2 | 3 | 5 | 4 | 6 | 2 | 3 | 6 | 7 | 4 | 5 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 24 | 10 | 10.8a |

for a 6 rows problem, approximately 770 times faster for a 7 rows problem, and approximately 3400 times faster for 8 rows problem. This enables the solution of the considered problem in an acceptable time.

Both algorithms produce the same sets of Pareto optimal decisions as expected. Pareto optimal decisions of the example problem with 7 rows are given in Table 10.8. The Pareto fronts for the problems with different numbers of rows are illustrated in Figure 10.7. Pareto optimal solutions of the problem with 6 rows are dominated by that of the problems with more rows. This means that the problem with 6 rows is too restrictive. Many Pareto optimal solutions of the problems with 7 and 8 rows coincide.

Business process diagrams corresponding to non-dominated decisions of the example multi-objective problem of shape allocation with 7 and 8 rows are shown in Figure 10.8. We do not consider the problem of drawing orthogonal connectors in this section. In these diagrams connectors are drawn following the rules of the software developed by OrgSoft. The connectors may be drawn in several horizontal and vertical lanes located in the middle and on the sides of cells. Bends were rounded to enhance visibility and clarity. The number of crossings was minimized when drawing connectors.

The diagram corresponding to the shortest total length decision with objective vector $(24, 10)$ is shown in Figure 10.8a. It can be seen that connected flow objects are placed nearby, often in the same row what violates direction down. The diagrams corresponding to the smallest number of violations decision in 7 rows

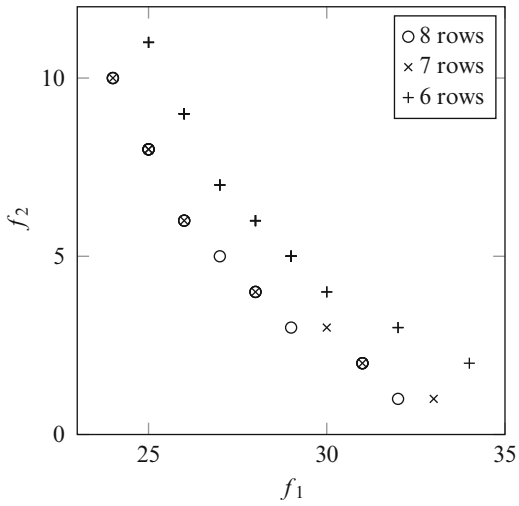


**Fig. 10.7** Pareto fronts for the multi-objective optimization problems for allocation of the shapes in the example business process diagram

with objective vector (33, 1) and in 8 rows with objective vector (32, 1) are shown in Figure 10.8c, d. There is only one violation coming from the last column to the previous one violating flow right. All the other connectors come down and in the same swimlane or directing to the right. The additional row enabled the move of flow objects in swimlane Role 2 and Role 3 down by one row and rearrangement of flow objects in swimlane Role 1 to save total length. The diagram corresponding to a non-dominated intermediate decision with the objective vector (28, 4) is shown in Figure 10.8b. Balance between flow violations and the total length can be seen.

We continue our experiments with a middle size business process diagram shown in Figure 10.9. The example diagram contains some rather long connectors as well as some sequence flows not compatible with the favorable top-down left-right direction. Such a diagram was used for the psychological experiment described in [93, 261] as an example of a diagram of medium appeal of the considered criteria. The data of the problem are given in Table 10.9. Role 1 corresponds to Department, Role 2 to Management Department, Role 3 to Law and Personnel Department, Role 4 to Department related to document, Role 5 to Management Department, Role 6 to Personnel. Two roles are related to the Management Department since in the considered example the functions of planning (director) and organizing (accountant, etc.) are explicitly separated. There are six roles in this business process, there is one role with 5 shapes, two with 4 shapes, and three with 2 shapes. The smallest number of rows to fit the business process diagram is 5. Even with this number of rows the number of different assignments of shapes to rows is 13,824,000,000.

The results of the experimental investigation are presented in Table 10.10. The computational time ($t$) and the number of function evaluations (NFE) for the branch and bound algorithm are shown. It can be seen that the use of bounding vector $\mathbf{b}^2(\mathbf{x}, n')$ (10.19) considerably improves the number of function evaluations and

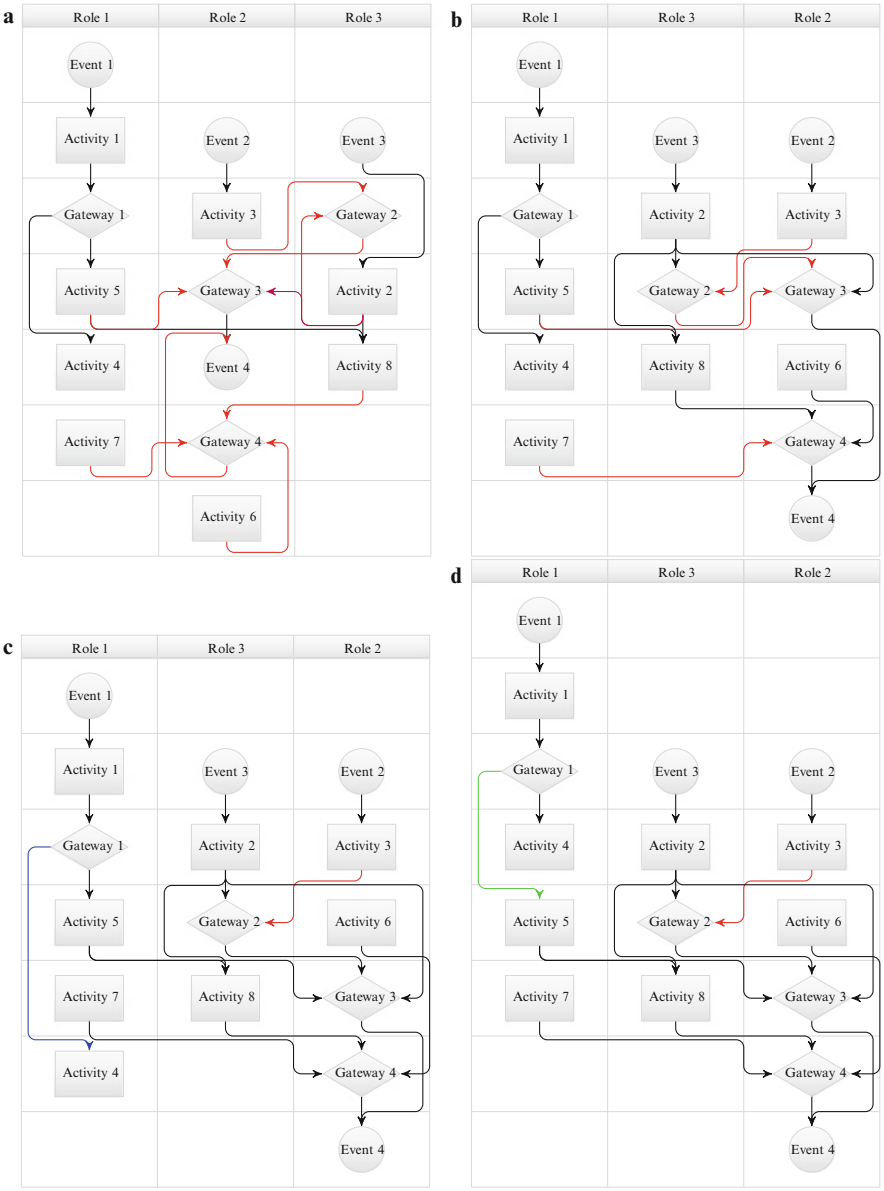


**Fig. 10.8** Business process diagrams corresponding to non-dominated decisions of the example multi-objective problem of shape allocation: (**a**) the shortest total length (24, 10), (**b**) intermediate solution (28, 4), (**c**) the smallest number of violations with 7 rows (33, 1), (**d**) the smallest number of violations with 8 rows (32, 1). Flow violations are shown in *red* and double violation (violating both right and down flow) is shown in *purple*, reduction of length is shown comparing *blue* and *green* connections

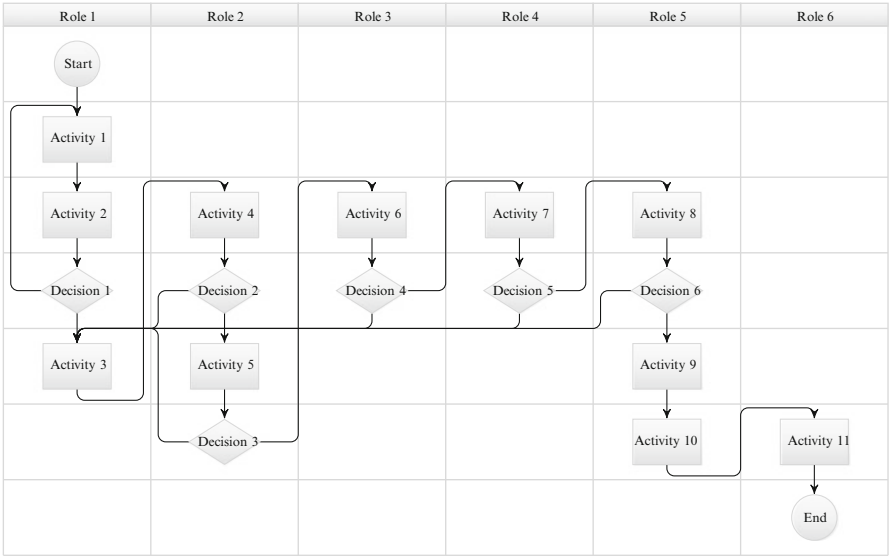


**Fig. 10.9** An example of a business process diagram

time compared to $\mathbf{b}^1(\mathbf{x}, n')$ (10.18). The use of bounding vector $\mathbf{b}^3(\mathbf{x}, n')$ (10.20) improves the number of function evaluations a bit further, but increases computational time since it is more expensive to compute. The use of bounding front $\mathbf{B}^4(\mathbf{x}, n')$ (10.21) considerably improves the number of function evaluations and computational time. Using the bounding front reduces computational time up to almost 100 times compared to using the bounding vector.

The Pareto fronts for problems with different number of rows are illustrated in Figure 10.10. Pareto solutions of the problem with 5, 6, and 7 rows are dominated by that of the problems with more rows. This means that the problem with less than 8 rows is too restrictive. The non-dominated solution with the objective vector $(77, 3)$ appears when more than 12 rows are allowed.

The diagram corresponding to the shortest total length decision with objective vector $(33, 15)$ is shown in Figure 10.11. Connected flow objects are placed nearby each other, but flow violations are quite often. The diagram corresponding to the decision with the smallest number of flow violations $(77, 3)$ is shown in Figure 10.12. The desirable flow in top-down left-right direction is clearly visible, only unavoidable violations appear due to the cycles in the process. However, this diagram requires a larger number of rows than the one with the shortest total length.

One of the non-dominated decisions with the objective vector $(38, 11)$ is illustrated in Figure 10.13. Similarly to the smaller problem a balance between flow violations and the total length of connectors can be seen.

**Table 10.9** Data of middle size business process diagram

| Flow objects | | | |
|---|---|---|---|
| No. | Role | Name | Description |
| 1 | Role 1 | Start | Start |
| 2 | Role 1 | Activity 1 | Responsible person prepares document project |
| 3 | Role 1 | Activity 2 | Head of dep. revises document |
| 4 | Role 1 | Decision 1 | Is approved? |
| 5 | Role 1 | Activity 3 | Responsible person redirects document for sequential approval |
| 6 | Role 2 | Activity 4 | Linguist revises document |
| 7 | Role 2 | Decision 2 | Is approved? |
| 8 | Role 2 | Activity 5 | Accountant revises document |
| 9 | Role 2 | Decision 3 | Is approved? |
| 10 | Role 3 | Activity 6 | Head of dep. revises document |
| 11 | Role 3 | Decision 4 | Is approved? |
| 12 | Role 4 | Activity 7 | Head of dep. revises document |
| 13 | Role 4 | Decision 5 | Is approved? |
| 14 | Role 5 | Activity 8 | Director revises document |
| 15 | Role 5 | Decision 6 | Is approved? |
| 16 | Role 5 | Activity 9 | Referent registers document |
| 17 | Role 5 | Activity 10 | Referent redirects document for familiarization |
| 18 | Role 6 | Activity 11 | Employees familiarize & sign in registry book |
| 19 | Role 6 | End | |

Connections

| $i$ | $k_{i1}$ | $k_{i2}$ |
|---|---|---|
| 1 | 1 | 2 |
| 2 | 2 | 3 |
| 3 | 3 | 4 |
| 4 | 4 | 2 |
| 5 | 4 | 5 |
| 6 | 5 | 6 |

| $i$ | $k_{i1}$ | $k_{i2}$ |
|---|---|---|
| 7 | 6 | 7 |
| 8 | 7 | 5 |
| 9 | 7 | 8 |
| 10 | 8 | 9 |
| 11 | 9 | 5 |
| 12 | 9 | 10 |

| $i$ | $k_{i1}$ | $k_{i2}$ |
|---|---|---|
| 13 | 10 | 11 |
| 14 | 11 | 5 |
| 15 | 11 | 12 |
| 16 | 12 | 13 |
| 17 | 13 | 5 |
| 18 | 13 | 14 |

| $i$ | $k_{i1}$ | $k_{i2}$ |
|---|---|---|
| 19 | 14 | 15 |
| 20 | 15 | 5 |
| 21 | 15 | 16 |
| 22 | 16 | 17 |
| 23 | 17 | 18 |
| 24 | 18 | 19 |

**Table 10.10** Results of the experimental investigation

| $n_r$ | $\mathbf{b}^1(\mathbf{x}, n')$, (10.18) | | $\mathbf{b}^2(\mathbf{x}, n')$, (10.19) | | $\mathbf{b}^3(\mathbf{x}, n')$, (10.20) | | $\mathbf{B}^4(\mathbf{x}, n')$, (10.21) | |
|---|---|---|---|---|---|---|---|---|
| | $t$, s | NFE | $t$, s | NFE | $t$, s | NFE | $t$, s | NFE |
| *Example problem* | | | | | | | | |
| 6 | 0.15 | 237,440 | 0.06 | 930,469 | 0.08 | 704,048 | 0.01 | 187,343 |
| 7 | 1.44 | 28,118,029 | 0.33 | 6,074,083 | 0.49 | 4,576,302 | 0.04 | 656,290 |
| 8 | 9.49 | 192,605,603 | 1.54 | 29,855,682 | 2.26 | 23,101,727 | 0.18 | 2,593,238 |
| *Middle size problem* | | | | | | | | |
| 5 | 0.87 | 11,846,524 | 0.14 | 2,292,133 | 0.19 | 2,110,564 | 0.04 | 518,681 |
| 6 | 8.76 | 87,341,601 | 0.86 | 15,097,449 | 1.26 | 14,111,040 | 0.21 | 2,993,714 |
| 7 | 20.10 | 267,553,983 | 2.21 | 40,710,474 | 3.40 | 38,251,546 | 0.52 | 7,370,189 |
| 8 | 37.05 | 473,246,383 | 3.41 | 64,644,742 | 5.37 | 60,846,181 | 0.83 | 11,886,008 |
| 9 | 76.96 | 997,982,630 | 6.72 | 128,330,033 | 10.66 | 120,741,102 | 1.31 | 18,437,102 |
| 10 | 193.69 | 1,946,020,628 | 13.17 | 257,442,963 | 21.22 | 243,423,005 | 3.23 | 47,220,762 |
| 11 | 394.98 | 3,386,280,514 | 25.03 | 487,597,206 | 39.77 | 464,519,182 | 8.50 | 131,752,014 |
| 12 | 751.75 | 5,496,804,470 | 46.13 | 949,050,115 | 76.33 | 914,075,489 | 24.45 | 397,440,621 |
| 13 | 1175.44 | 8,072,969,995 | 58.66 | 1,201,936,218 | 97.00 | 1,145,782,878 | 15.70 | 236,090,687 |
| 14 | 1845.78 | 11,516,056,991 | 85.80 | 1,774,663,616 | 143.27 | 1,695,153,806 | 23.48 | 353,554,807 |
| 15 | 2746 | 15,764,528,221 | 120.29 | 2,493,528,143 | 204.61 | 2,385,705,518 | 33.12 | 498,906,138 |
| 16 | 3825 | 20,848,903,023 | 161.08 | 3,363,454,730 | 270.26 | 3,222,389,040 | 44.81 | 672,931,502 |
| 17 | 5182 | 26,788,986,132 | 209.02 | 4,388,173,880 | 352.65 | 4,208,888,470 | 57.97 | 876,392,519 |
| 18 | 6817 | 33,597,007,137 | 263.56 | 5,570,100,374 | 445.95 | 5,347,708,471 | 73.53 | 1,109,646,700 |
| 19 | 8670 | 41,280,000,441 | 330.56 | 6,912,015,181 | 526.19 | 6,272,785,312 | 92.77 | 1,373,682,420 |

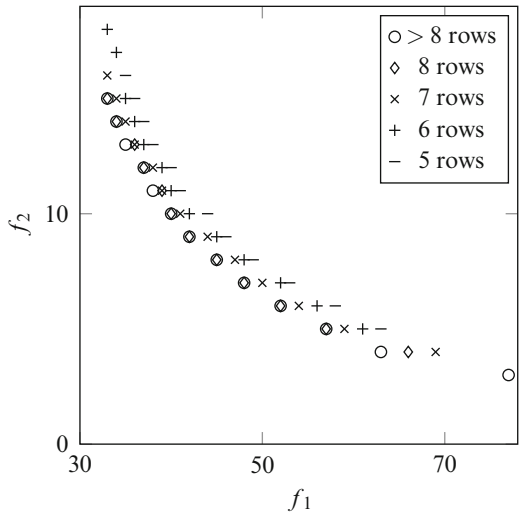


**Fig. 10.10** Pareto fronts for the multi-objective optimization problems for allocation of the shapes in a middle size business process diagram

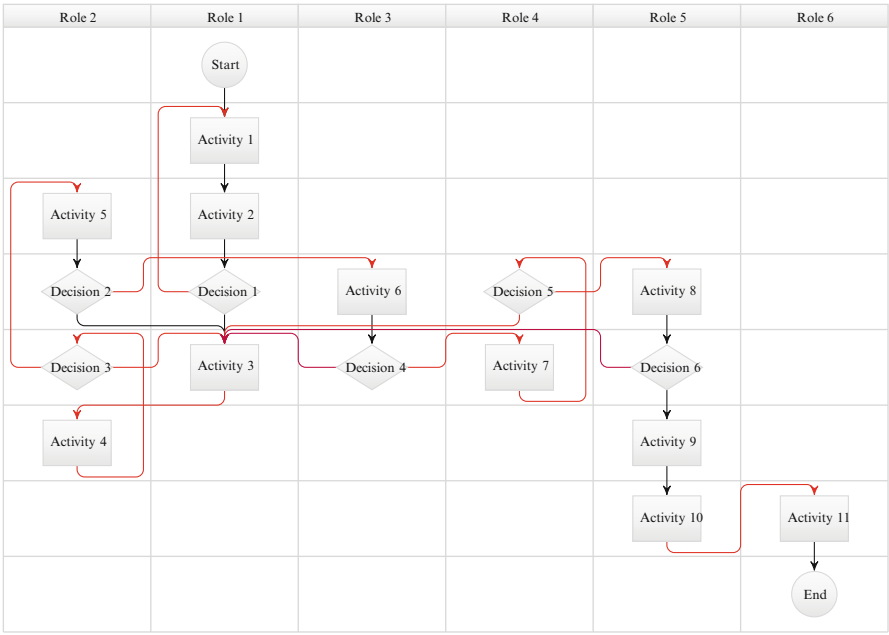


**Fig. 10.11** Diagram corresponding to the decision with the shortest total length $(33, 15)$ of the multi-objective problem of shape allocation in a middle size business process diagram, flow violations are shown in *red* and double violations (violating both right and down flow) are shown in *purple*

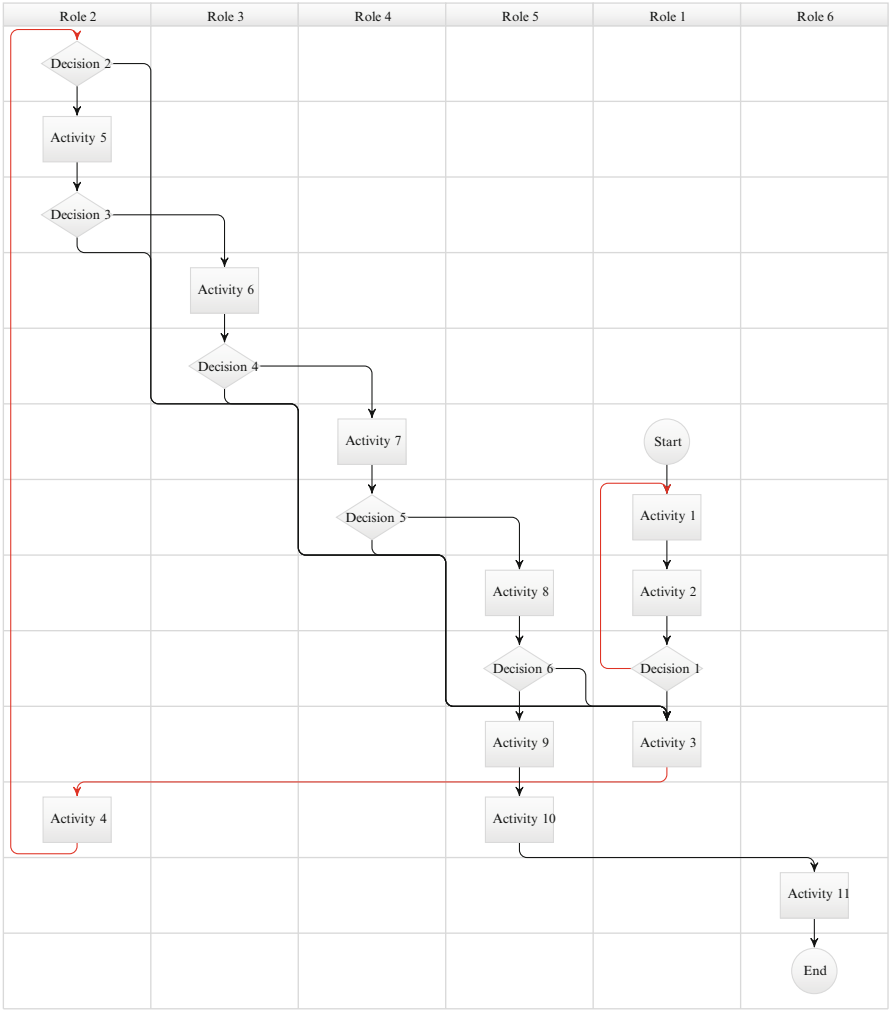


**Fig. 10.12** Diagram corresponding to the decision with the smallest number of flow violations (77, 3) of the multi-objective problem of shape allocation in a middle size business process diagram

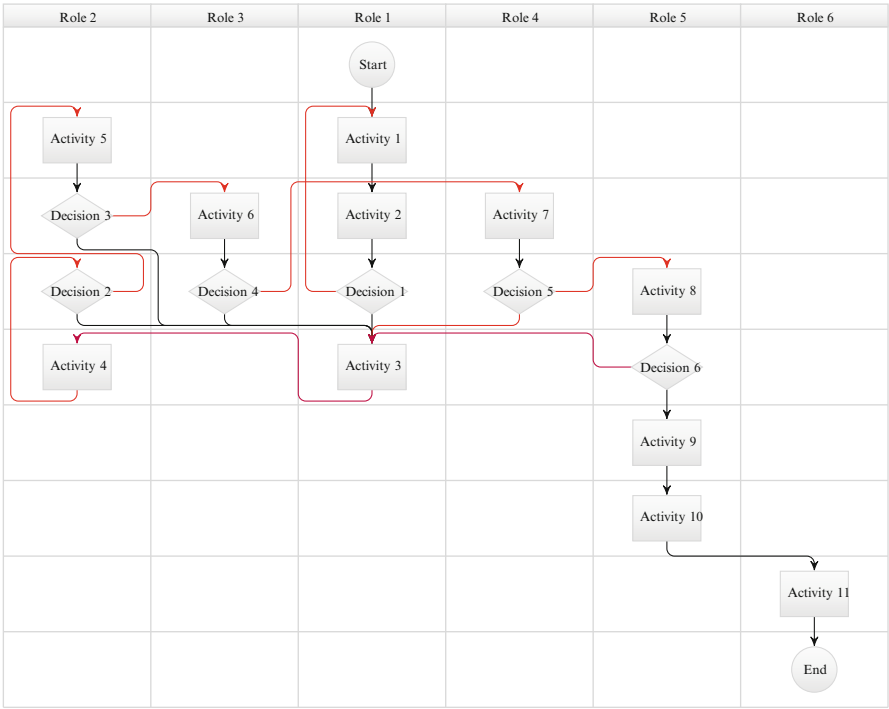


**Fig. 10.13** Diagram corresponding to a non-dominated intermediate decision with the objective vector (38, 11) of the multi-objective problem of shape allocation in a middle size business process diagram

# References


1. Aarts, E., Korst, J.: Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing. Wiley, New York (1989)
2. Agarwal, S.: Portfolio Selection Using Multi-Objective Optimization. Palgrave Macmillan, Basingstoke (2017). doi:10.1007/978-3-319-54416-8
3. Al-Betar, M.A., Doush, I.A., Khader, A.T., Awadallah, M.A.: Novel selection schemes for harmony search. Appl. Math. Comput. **218**(10), 6095–6117 (2012)
4. Al-Betar, M.A., Khader, A.T., Geem, Z.W., Doush, I.A., Awadallah, M.A.: An analysis of selection methods in memory consideration for harmony search. Appl. Math. Comput. **219**(22), 10753–10767 (2013)
5. Ansoff, H.I.: Corporate Strategy. Penguin, Harmondsworth (1968)
6. Arkat, J., Hosseini, L., Farahani, M.H.: Minimization of exceptional elements and voids in the cell formation problem using a multi-objective genetic algorithm. Exp. Syst. Appl. **38**(8), 9597–9602 (2011). doi:10.1016/j.eswa.2011.01.161
7. Arora, S., Barak, B.: Computational Complexity a Modern Approach. Cambridge University Press, Cambridge (2009)
8. Bäck, T.: Evolutionary Algorithms in Theory and Praxis. Oxford University Press, Oxford (1996)
9. Bajestani, M.A., Rabbani, M., Rahimi-Vahed, A., Khoshkhou, G.B.: A multi-objective scatter search for a dynamic cell formation problem. Comput. Oper. Res. **36**(3), 777–794 (2009). doi:10.1016/j.cor.2007.10.026
10. Baritompa, W.: Customizing methods for global optimization—a geometric viewpoint. J. Glob. Optim. **3**(2), 193–212 (1993)
11. Baronas, R., Žilinskas, A., Litvinas, L.: Optimal design of amperometric biosensors applying multi-objective optimization and decision visualization. Electrochim. Acta **211**, 586–594 (2016). doi:10.1016/j.electacta.2016.06.101
12. Battista, G.D., Eades, P., Tamassia, R., Tollis, I.G.: Graph Drawing, Algorithms for the Visualisation of Graphs. Prentice Hall, Upper Saddle River (1999)
13. Battiti, R., Brunato, M.: The LION Way. Machine Learning Plus Intelligent Optimization, LIONlab, University of Trento, Italy (2014)
14. Bell, D.E., Raiffa, H.: Risky choice revisited. In: Bell, D.E., Raiffa, H., Tversky, A. (eds.) Decision Making: Descriptive, Normative, and Prescriptive Interactions, pp. 99–112. Cambridge University Press, Cambridge (1988)
15. Bennett, Ch., Ryall, J., Spalteholz, L., Geoch, A.: The aesthetics of graph visualization. In: Cunningham, D.W., Meyer, G., Neumann, L. (eds.) Computational Aesthetics in Graphics, Visualization, and Imaging, pp. 1–8. Springer, Berlin (2007)




P.M. Pardalos et al., *Non-Convex Multi-Objective Optimization*, Springer Optimization and Its Applications 123, DOI 10.1007/978-3-319-61007-8

16. Benson, H.P.: Multi-objective optimization: Pareto optimal solutions, properties. In: Floudas, C.A., Pardalos, P.M. (eds.) Encyclopedia of Optimization, pp. 2478–2481. Springer, Boston (2009). doi:10.1007/978-0-387-74759-0_426
17. Benson, H.P.: An outcome space algorithm for optimization over the weakly efficient set of a multiple objective nonlinear programming problem. J. Glob. Optim. **52**(3), 553–574 (2012). doi:10.1007/s10898-011-9786-y
18. Biegler, L., Jiang, L., Fox, G.: Recent advances in simulation and optimal design of pressure swing adsorption systems, separation. Sep. Purif. Rev. **33**(1), 1–39 (2005)
19. Blum, L., Cucker, F., Shub, M., Smale, S.: Complexity and Real Computation. Springer, Berlin (1998)
20. Borg, I., Groenen, P.: Modern Multidimensional Scaling: Theory and Applications. Springer, Berlin (1997)
21. Boulif, M., Atif, K.: A new fuzzy genetic algorithm for the dynamic bi-objective cell formation problem considering passive and active strategies. Int. J. Approx. Reason. **47**(2), 141–165 (2008). doi:10.1016/j.ijar.2007.03.003
22. Bouri, A., Martel, J.M., Chabchoub, H.: A multi-criterion approach for selecting attractive portfolio. J. Multi-Criteria Decis. Anal. **11**(4–5), 269–277 (2002). doi:10.1002/mcda.334
23. Branke, J., Deb, K., Miettinen, K., Słowiński, R. (eds.): Multiobjective optimization: interactive and evolutionary approaches. In: Dagstuhl Seminar on Practical Approaches to Multi-Objective Optimization, Schlosss Dagstuhl, 10–15 Dec 2006. Lecture Notes in Computer Science, vol. 5252. Springer, Berlin (2008)
24. Brusco, M.J., Stahl, S.: Branch-and-Bound Applications in Combinatorial Data Analysis. Springer, New York (2005)
25. Buhmann, M.: Radial Basis Functions. Cambridge University Press, Cambridge (2003)
26. Burachik, R.S., Kaya, C.Y., Rizvi, M.M.: A new scalarization technique to approximate Pareto fronts of problems with disconnected feasible sets. J. Optim. Theory Appl. **162**(2), 428–446 (2014). doi:10.1007/s10957-013-0346-0
27. Butz, A.R.: Space filling curves and mathematical programming. Inf. Control **12**, 319–330 (1968)
28. Calvin, J.: Average performance of passive algorithms for global optimization. J. Math. Anal. Appl. **191**, 608–617 (1995)
29. Calvin, J.M.: Lower bounds on complexity of optimization of continuous functions. J. Complex. **20**, 773–795 (2004)
30. Calvin, J., Žilinskas, A.: On convergence of the P-algorithm for one-dimensional global optimization of smooth functions. J. Optim. Theory Appl. **102**, 479–495 (1999)
31. Calvin, J., Žilinskas, A.: A one-dimensional P-algorithm with convergence rate $o(n^{-3+\delta})$ for smooth functions. J. Optim. Theory Appl. **106**, 297–307 (2000)
32. Chen, H.Y., Chang, Y.W.: Global and detailed routing. In: Wang, L.T., Chang, Y.W., Cheng, K.T. (eds.) Electronic Design Automation: Synthesis, Verification, and Testing, pp. 687–749. Elsevier/Morgan Kaufmann, Amsterdam (2008)
33. Chinchuluun, A., Pardalos, P.M.: A survey of recent developments in multiobjective optimization. Ann. Oper. Res. **154**(1), 29–50 (2007)
34. Chinchuluun, A., Yuan, D., Pardalos, P.: Optimality conditions and duality for nondifferentiable multiobjective fractional programming with generalized convexity. Ann. Oper. Res. **154**(1), 133–147 (2007)
35. Chinchuluun, A., Pardalos, P., Migdalas, A., Pitsoulis, L.: Pareto Optimality, Game Theory and Equilibria. Springer Science-Business Media, New York (2008)
36. Cloquette, J.F., Gérard, M., Hadhri, M.: An empirical analysis of Belgian daily returns using GARCH models. Brussels Econ. Rev. **148**, 513–534 (1995)
37. Cox, T., Cox, M.: Multidimensional Scaling. Chapman and Hall, Boca Raton (2001)
38. CPLEX for MATLAB Toolbox. http://www-03.ibm.com/software/products/en/ibmilogcpleoptistud (2003)
39. Curry, D.M., Dagli, C.H.: Computational complexity measure for many-objective optimization problems. Proc. Comput. Sci. **36**, 185–191 (2014)

40. Das, I., Dennis, J.E.: A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. Struct. Multidiscip. Optim. **14**(1), 63–69 (1997)
41. Das, I., Dennis, J.E.: Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems. SIAM J. Optim. **8**(3), 631–657 (1998)
42. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. Wiley, Chichester (2009)
43. Diakonikolas, I.: Approximation of multiobjective optimization problems. Ph.D. Thesis, Columbia University (2011)
44. Diakonikolas, I., Yannakakis, M.: Small approximate Pareto sets for bi-objective shortest paths and other problems.SIAM J. Comput. **39**(4), 1340–1371 (2009). doi:10.2478/v10006-007-0023-2
45. Dimopoulos, C.: A review of evolutionary multiobjective optimization applications in the area of production research. In: Congress on Evolutionary Computation, 2004 (CEC2004), vol. 2, pp. 1487–1494 (2004). doi:10.1109/CEC.2004.1331072
46. Dimopoulos, C.: Explicit consideration of multiple objectives in cellular manufacturing. Eng. Optim. **39**(5), 551–565 (2007). doi:10.1080/03052150701351631
47. Du, D., Pardalos, P.: Minimax and Applications. Kluwer Academic Publishers, Dordrecht (1995)
48. Dzemyda, G., Kurasova, O., Žilinskas, J.: Multidimensional Data Visualization: Methods and Applications. Springer, New York (2013). doi:10.1007/978-1-4419-0236-8
49. Ehrgott, M.: Approximation algorithms for combinatorial multicriteria optimization problems. Int. Trans. Oper. Res. **7**(1), 5–31 (2000)
50. Eichfelder, G.: Adaptive Scalarization Methods in Multiobjective Optimization. Springer, Berlin (1986)
51. Eichfelder, G.: An adaptive scalarization method in multiobjective optimization. SIAM J. Optim. **19**(4), 1694–1718 (2009)
52. Elsakov, S.M., Shiryaev, V.I.: Homogeneous algorithms for multiextremal optimization. Comput. Math. Math. Phys. **50**(10), 1642–1654 (2010)
53. Emmerich, M., Giannakoglou, K., Naujoks, B.: Single- and multi-objective evolutionary optimization assisted by gaussian random field metamodels. IEEE Trans. Evol. Comput. **10**(4), 421–439 (2006)
54. Etzioni, O., Hanks, S., Jiang, T., Karp, R.M., Madani, O., Waarts, O.: Efficient information gathering on the Internet. In: Proceedings of 37th Conference on Foundations of Computer Science, pp. 234–243 (1996). doi:10.1109/SFCS.1996.548482
55. Evtushenko, Yu.G., Posypkin, M.A.: Nonuniform covering method as applied to multicriteria optimization problems with guaranteed accuracy. Comput. Math. Math. Phys. **53**(2), 144–157 (2013)
56. Fernández, P., Pelegrín, B., Lančinskas, A., Žilinskas, J.: New heuristic algorithms for discrete competitive location problems with binary and partially binary customer behavior. Comput. Oper. Res. **79**, 12–18 (2017). doi:10.1016/j.cor.2016.10.002
57. Fiandaca, G., Fraga, E., Brandani, S.: A multi-objective genetic algorithm for the design of pressure swing adsorption. Eng. Optim. **41**(1), 833–854 (2009)
58. Fishburn, P.: Utility Theory for Decision Making. Wiley, New York (1970)
59. Floudas, C.A., Pardalos, P.M.: Encyclopedia of Optimization, 2nd edn. Springer, Boston (2009)
60. Fonseca, C., Fleming, P.: On the performance assessment and comparison of stochastic multiobjective optimizers. In: Ebeling, W., Rechenberg, I., Schwefel, H.P., Voigt, H.M. (eds.) Parallel Problem Solving from Nature, Berlin, September 22–26. Lecture Notes in Computer Science, vol. 1141, pp. 584–593. Springer, Berlin (1996)
61. Fontes, D.B.M.M., Gaspar-Cunha, A.: On multi-objective evolutionary algorithms. In: Zopounidis, C., Pardalos, P.M. (eds.) Handbook of Multicriteria Analysis. Applied Optimization, vol. 103, pp. 287–310. Springer, Berlin/Heidelberg (2010). doi:10.1007/978-3-540-92828-7_10

62. Galperin, E.A.: The cubic algorithm. J. Math. Anal. Appl. **112**(2), 635–640 (1985)
63. Galperin, E.A.: Precision, complexity, and computational schemes of the cubic algorithm. J. Optim. Theory Appl. **57**, 223–238 (1988)
64. Garey, M., Johnson, D.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, New York (1979)
65. Gass, S., Saati, T.: The computational algorithm for the parametric objective function. Naval Res. Log. Q. **2**, 39–45 (1955)
66. Gaviano, M., Kvasov, D.E., Lera, D., Sergeyev, Ya.D.: Algorithm 829: software for generation of classes of test functions with known local and global minima for global optimization. ACM Trans. Math. Softw. **29**(4), 469–480 (2003)
67. Geem, Z.W., Kim, J.H., Loganathan, G.V.: A new heuristic optimization algorithm: harmony search. Simulation **76**, 60–68 (2001)
68. Geoffrion, A.M.: Proper efficiency and the theory of vector maximization. J. Math. Anal. Appl. **22**(3), 618–630 (1968). doi:10.1016/0022-247X(68)90201-1
69. Goldberg, D.: Genetic Algorithms and Their Applications. Addison-Wesley, Reading (1989)
70. Goldengorin, B., Krushinsky, D., Slomp, J.: Flexible PMP approach for large-size cell formation. Oper. Res. **60**(5), 1157–1166 (2012). doi:10.1287/opre.1120.1108
71. Goldengorin, B., Krushinsky, D., Pardalos, P.M.: Cell Formation in Industrial Engineering: Theory, Algorithms and Experiments. Springer, New York (2013)
72. Gourdin, E., Hansen, P., Jaumard, B.: Global Optimization of Multivariate Lipschitz Functions: Survey and Computational Comparison. Cahiers du GERAD (1994)
73. Gutmann, H.M.: A radial basis function method for global optimization. J. Glob. Optim. **19**, 201–227 (2001)
74. Haimes, Y., Lasdon, L., Wismer, D.: On a bicriterion formulation of the problem of integrated system identification and system optimization. IEEE Trans. Syst. Man. Cybern. **1**, 296–297 (1971)
75. Hansen, E.: A globally convergent interval method for computing and bounding real roots. BIT Numer. Math. **18**(4), 415–424 (1978). doi:10.1007/BF01932020
76. Hansen, E.: Global optimization using interval analysis—the multi-dimensional case. Numer. Math. **34**(3), 247–270 (1980). doi:10.1007/BF01396702
77. Hansen, P.: Bicriterion path problems. In: Fandel, G., Gal, T. (eds.) Multiple Criteria Decision Making: Theory and Applications. Lecture Notes in Economics and Mathematical Systems, vol. 177, pp. 109–127. Springer, New York (1980)
78. Hansen, P., Jaumard, B.: Lipshitz optimization. In: Horst, R., Pardalos, P.M. (eds.) Handbook of Global Optimization, vol. 1, pp. 407–493. Kluwer Academic Publishers, Dordrecht (1995)
79. Hansen, E., Walster, G.W.: Global Optimization Using Interval Analysis, 2 edn. Marcel Dekker, New York (2004)
80. Haupt, R., Haupt, S.: Practical Genetic Algorithms, 2nd edn. Wiley-Interscience, Hoboken (2004)
81. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)
82. Hooker, J.: Testing heuristics: we have it all wrong. J. Heuristics **1**, 33–42 (1995)
83. Horst, R.: A general class of branch-and-bound methods in global optimization with some new approaches for concave minimization. J. Optim. Theory Appl. **51**, 271–291 (1986)
84. Horst, R., Pardalos, P.M. (eds.): Handbook of Global Optimization, vol. 1. Kluwer Academic Publishers, Dordrecht (1995)
85. Horst, R., Tuy, H.: On the convergence of global methods in multiextremal optimization. J. Optim. Theory Appl. **54**, 253–271 (1987)
86. Horst, R., Tuy, H.: Global Optimization: Deterministic Approaches. Springer, Berlin (1996)
87. Horst, R., Pardalos, P.M., Thoai, N.: Introduction to Global Optimization. Kluwer Academic Publishers, Dordrecht (2000)
88. Huang, D., Allen, T.T., Notz, W., Zheng, N.: Global optimization of stochastic black-box systems via sequential kriging meta-models. J. Glob. Optim. **34**, 441–446 (2006)

89. Hurson, C., Zopounidis, C.: On the use of multi-criteria decision aid methods to portfolio selection. J. Euro-Asian Manag. **1**(2), 69–94 (1995)
90. Jacquillat, B.: Les modèles d'évaluation et de sélection des valeurs mobilières: panorama des recherches américaines. Analyse Financière **11**, 68–88 (1972)
91. Jančauskas, V., Žilinskas, A.: A multi-objective optimization aided algorithm for search of aesthetically looking paths in grids. In: ICNNAI 2012: Neural Networks and Artificial Intelligence, Proceedings of the 7th International Conference, pp. 209–214. BSUIR, Minsk (2012)
92. Jančauskas, V., Kaukas, G., Žilinskas, A., Žilinskas, J.: On multi-objective optimization aided visualization of graphs related to business process diagrams. In: Čaplinskas, A., Dzemyda, G., Lupeikienė, A., Vasilecas, O. (eds.) Local Proceedings and Materials of Doctoral Consortium of the Tenth International Baltic Conference on Databases and Information Systems, pp. 71–80. Žara, Vilnius (2012)
93. Jančauskas, V., Mackutė-Varoneckienė, A., Varoneckas, A., Žilinskas, A.: On the multi-objective optimization aided drawing of connectors for graphs related to business process management. Commun. Comput. Inform. Sci. **319**, 87–100 (2012)
94. Jaumard, B., Herrmann, T., Ribault, H.: An On-line Cone Intersection Algorithm for Global Optimization of Multivariate Lipschitz Functions. Cahiers du GERAD, vol. 95, issue 7 (1995)
95. Jones, D.: A taxonomy of global optimization methods based on response surfaces. J. Glob. Optim. **21**, 345–383 (2001)
96. Jones, D.R., Perttunen, C.D., Stuckman, B.E.: Lipschitzian optimization without the Lipschitz constant. J. Optim. Theory Appl. **79**(1), 157–181 (1993). doi:10.1007/BF00941892
97. Kaliszewski, I.: A modified weighted Tchebychev metric for multiple objective programming. Comput. Oper. Res. **14**(4), 315–323 (1987)
98. Kaliszewskii, I.: A theorem on nonconvex functions and its application to vector optimization. Eur. J. Oper. Res. **80**, 439–445 (1995)
99. Karlin, S.: Mathematical Methods and Theory in Games, Programming, and Economics, vol. I + II. Pergamon Press, London (1959)
100. Kaufmann, M., Wagner, D.: Graph Drawing, Algorithms for the Visualisation of Graphs. Springer, Berlin (2001)
101. Keane, A., Scalan, J.: Design search and optimization in aerospace engineering. Phil. Trans. R. Soc. A **365**, 2501–2529 (2007)
102. Khoury, N., Martel, J.M., Veilleux, M.: Méthode multicritère de sélection de portefeuilles indiciels internationaux. L'Actualité économique **69**(1), 171–190 (1993)
103. Klamroth, K., Wiecek, M.M.: Dynamic programming approaches to the multiple criteria knapsack problem. Naval Res. Logist. **47**(1), 57–76 (2000). doi:10.1002/(SICI)1520-6750(200002)47:1<57::AID-NAV4>3.0.CO;2-4
104. Klamroth, K., Wiecek, M.M.: Time-dependent capital budgeting with multiple criteria. In: Haimes, Y.Y., Steuer, R.E. (eds.) Research and Practice in Multiple Criteria Decision Making: Proceedings of the XIVth International Conference on Multiple Criteria Decision Making (MCDM) Charlottesville, Virginia, USA, 8–12 June 1998, pp. 421–432. Springer, Berlin/Heidelberg (2000). doi:10.1007/978-3-642-57311-8_36
105. Knowles, J.: ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. IEEE Trans. Evol. Comput. **10**(1), 50–66 (2006)
106. Knowles, J., Corne, D., Reynolds, A.: Noisy multiobjective optimization on a budget of 250 evaluations. Lect. Notes Comput. Sci. **5467**, 36–50 (2009)
107. Kostreva, M.M., Wiecek, M.M.: Time dependency in multiple objective dynamic programming. J. Math. Anal. Appl. **173**(1), 289–307 (1993). doi:dx.doi.org/10.1006/jmaa.1993.1067
108. Kushner, H.: A versatile stochastic model of a function of unknown and time-varying form. J. Math. Anal. Appl. **5**, 150–167 (1962)
109. Kvasov, D.E., Sergeyev, Ya.D.: Multidimensional global optimization algorithm based on adaptive diagonal curves. Comput. Math. Math. Phys. **43**(1), 40–56 (2003)
110. Kvasov, D.E., Sergeyev, Ya.D.: Lipschitz gradients for global optimization in a one-point-based partitioning scheme. J. Comput. Appl. Math. **236**(16), 4042–4054 (2012). doi:10.1016/j.cam.2012.02.020

111. Kvasov, D.E., Sergeyev, Ya.D.: Univariate geometric Lipschitz global optimization algorithms. Numer. Algebra Control Optim. **2**(1), 69–90 (2012). doi:10.3934/naco.2012.2.69
112. Kvasov, D.E., Pizzuti, C., Sergeyev, Ya.D.: Local tuning and partition strategies for diagonal GO methods. Numer. Math. **94**(1), 93–106 (2003)
113. Lančinskas, A., Fernández, P., Pelegín, B., Žilinskas, J.: Improving solution of discrete competitive facility location problems. Optim. Lett. **11**(2), 259–270 (2017). doi:10.1007/s11590-015-0930-3
114. Land, A.H., Doig, A.G.: An automatic method of solving discrete programming problems. Econometrica **28**(3), 497–520 (1960). doi:10.2307/1910129
115. Lee, S.D., Chen, Y.L.: A weighted approach for cellular manufacturing design: minimizing intercell movement and balancing workload among duplicated machines. Int. J. Protein Res. **35**(4), 1125–1146 (1997). doi:10.1080/002075497195588
116. Lee, S.M., Lerro, A.J.: Capital budgeting for multiple objectives. Financ. Manag. **3**(1), 58–66 (1974)
117. Lee, D., Yang, C., Wong, C.: Rectilinear paths among rectilinear obstacles. Discret. Appl. Math. **70**(3), 185–216 (1996)
118. Lei, D., Wu, Z.: Tabu search for multiple-criteria manufacturing cell design. Int. J. Adv. Manuf. Technol. **28**, 950–956 (2006). doi:10.1007/s00170-004-2441-8
119. Lera, D., Sergeyev, Ya.D.: Acceleration of univariate global optimization algorithms working with Lipschitz functions and Lipschitz first derivatives. SIAM J. Optim. **23**(1), 508–529 (2013)
120. Liang, Z., Huang, H., Pardalos, P.: Efficiency conditions and duality for a class of multiobjective fractional programming problems. J. Glob. Optim. **53**, 447–471 (2003)
121. Little, J.D.C., Murty, K.G., Sweeney, D.W., Karel, C.: An algorithm for the traveling salesman problem. Oper. Res. **11**(6), 972–989 (1963). doi:10.1287/opre.11.6.972
122. Lotov, A., Miettinen, K.: Visualizing the Pareto frontiers. Lect. Notes Comput. Sci. **5252**, 213–243 (2008)
123. Lu, Q., Chen, Y., Zhang, X.: Smart Grids: Multi-Objective Optimization in Dispatching. Walter de Gruyter GmbH & Company KG, Berlin (2017)
124. Malakooti, B., Yang, Z.: Multiple criteria approach and generation of efficient alternatives for machine-part family formation in group technology. IIE Trans. **34**, 837–846 (2002). doi:10.1023/A:1015557007084
125. Mansouri, S.A., Husseini, S.M., Newman, S.: A review of the modern approaches to multi-criteria cell design. Int. J. Prod. Res. **38**(5), 1201–1218 (2000). doi:10.1080/002075400189095
126. Markowitz, H.: Portfolio selection. J. Financ. **7**(1), 77–91 (1952). doi:10.1111/j.1540-6261.1952.tb01525.x
127. Martínez, M., Sanchis, J., Blasco, X.: Global and well-distributed Pareto frontier by modified normalized normal constraint methods for bicriterion problems. Struct. Multidiscip. Optim. **34**, 197–209 (2007)
128. Mathar, R., Žilinskas, A.: On global optimization in two dimensional scaling. Acta Appl. Math. **33**, 109–118 (1993)
129. Mathar, R., Žilinskas, A.: A class of test functions for global optimization. J. Global Optim. **5**, 195–199 (1994)
130. Mayne, D.Q., Polak, E.: Outer approximation algorithm for nondifferentiable optimization problems. J. Optim. Theory Appl. **42**(1), 19–30 (1984)
131. McGeoch, C.: Experimental analysis of algorithms. In: Pardalos, P.M., Romeijn, H.E. (eds.) Handbook of Global Optimization, vol. 2, pp. 489–514. Kluwer Academic Publishers, Boston/Dordrecht/London (2002)
132. Meewella, C.C., Mayne, D.Q.: An algorithm for global optimization of Lipschitz continuous functions. J. Optim. Theory Appl. **57**(2), 307–323 (1988)
133. Meewella, C.C., Mayne, D.Q.: An efficient domain partitioning algorithms for global optimization of rational and Lipschitz continuous functions. J. Optim. Theory Appl. **61**(2), 247–270 (1989)

134. Messac, A., Ismail-Yahaya, A., Mattson, C.A.: The normalized normal constraint method for generating the Pareto frontier. Struct. Multidiscip. Optim. **25**, 86–98 (2003)
135. Miettinen, K.: Nonlinear Multiobjective Optimization. Springer, Berlin (1999)
136. Mladineo, R.H.: An algorithm for finding the global maximum of a multimodal, multivariate function. Math. Program. **34**(2), 188–200 (1986)
137. Mladineo, R.H.: Convergence rates of a global optimization algorithm. Math. Program. **54**(1–3), 223–232 (1992)
138. Mockus, J.: Multiextremal Problems in Design (in Russian). Nauka, Moscow (1976)
139. Mockus, J.: Bayesian Approach to Global Optimization. Kluwer Academic Publishers, Dordrecht (1988)
140. Monin, A., Yaglom, A.: Statistical Hydrodynamics, vols. 1 and 2 (in Russian). Nauka, Moscow (1965/1967)
141. Moore, R.E.: Methods and Applications of Interval Analysis, vol. 2. SIAM, Philadelphia (1979)
142. Nakayama, H.: Sequential Approximate Multiobjective Optimization Using Computational Intelligence. Springer, Berlin (2009)
143. Nemirovsky, A., Yudin, D.: Problem Complexity and Method Efficiency in Optimization. Wiley, Chichester (1983)
144. Neto, A.R.P., Filho, E.V.G.: A simulation-based evolutionary multiobjective approach to manufacturing cell formation. Comput. Ind. Eng. **59**(1), 64–74 (2010). doi:10.1016/j.cie.2010.02.017
145. Nevzorov, V.: Records: Mathematical Theory. American Mathematical Society, Providence (2001)
146. Owen, M., Jog, R.: BPMN and business process management. http://www.bpmn.org (2003)
147. Papadimitriou, C.: Computational Complexity. Addison-Wesley, Reading (1994)
148. Papadimitriou, C.H., Yannakakis, M.: On the approximability of trade-offs and optimal access of Web sources. In: Proceedings of 41st Annual Symposium on Foundations of Computer Science 2000, pp. 86–92 (2000). doi:10.1109/SFCS.2000.892068
149. Papaioannou, G., Wilson, J.M.: The evolution of cell formation problem methodologies based on recent studies (1997–2008): review and directions for future research. Eur. J. Oper. Res. **206**(3), 509–521 (2010). doi:10.1016/j.ejor.2009.10.020
150. Pardalos, P.: Complexity in Numerical Optimization. World Scientific, Singapore (1993)
151. Pardalos, P.: Approximation and Complexity in Numerical Optimization. Kluwer, Dordrecht (2000)
152. Pardalos, P.M., Romeijn, H.E. (eds.): Handbook of Global Optimization, vol. 2. Kluwer Academic Publishers, Dordrecht (2002)
153. Pardalos, P.M., Siskos, Y., Zopounidis, C.: Advances in Multicriteria Analysis. Springer, New York (2010)
154. Pardalos, P.M., Steponavice, I., Žilinskas, A.: Pareto set approximation by the method of adjustable weights and successive lexicographic goal programming. Optim. Lett. **6**, 665–678 (2012)
155. Pareto, V.: Cours d'Economie Politique. Droz, Geneva (1896)
156. Pascoletti, A., Serafini, P.: Scalarization vector optimization problems. J. Optim. Theory Appl. **42**(4), 499–524 (1984)
157. Paulavičius, R., Žilinskas, J.: Analysis of different norms and corresponding Lipschitz constants for global optimization in multidimensional case. Inf. Technol. Control **36**(4), 383–387 (2007)
158. Paulavičius, R., Žilinskas, J.: Improved Lipschitz bounds with the first norm for function values over multidimensional simplex. Math. Model. Anal. **13**(4), 553–563 (2008)
159. Paulavičius, R., Žilinskas, J.: Simplicial Global Optimization. Springer, New York (2014). doi:10.1007/978-1-4614-9093-7
160. Paulavičius, R., Žilinskas, J.: Simplicial Lipschitz optimization without the Lipschitz constant. J. Glob. Optim. **59**(1), 23–40 (2014). doi:10.1007/s10898-013-0089-3

161. Paulavičius, R., Žilinskas, J.: Advantages of simplicial partitioning for Lipschitz optimization problems with linear constraints. Optim. Lett. **10**(2), 237–246 (2016). doi:10.1007/s11590-014-0772-4
162. Paulavičius, R., Žilinskas, J., Grothey, A.: Investigation of selection strategies in branch and bound algorithm with simplicial partitions and combination of Lipschitz bounds. Optim. Lett. **4**, 173–183 (2010). doi:10.1007/s11590-009-0156-3
163. Pijavskii, S.: An algorithm for finding the absolute extremum of a function (in Russian). USSR Comput. Math. Math. Phys. **12**, 57–67 (1972)
164. Pinter, J.: Extended univariate algorithms for *n*-dimensional global optimization. Computing **36**(1), 91–103 (1986)
165. Pinter, J.: Globally convergent methods for *n*-dimensional multiextremal optimization. Optimization **17**, 187–202 (1986)
166. Pinter, J.: Branch-and-bound algorithms for solving global optimization problems with Lipschitzian structure. Optimization **19**(1), 101–110 (1988)
167. Pinter, J.: Continuous global optimization software: a brief review. Optika **52**, 1–8 (1996)
168. Pintér, J.D.: Global Optimization in Action: Continuous and Lipschitz Optimization: Algorithms, Implementations and Applications. Kluwer Academic Publishers, Dordrecht (1996)
169. Piyavskii, S.A.: An algorithm for finding the absolute minimum of a function (in Russian). Theory Optim. Solut. **2**, 13–24 (1967)
170. Piyavskii, S.A.: An algorithm for finding the absolute extremum of a function. Zh. Vychisl. Mat. Mat. Fiz **12**(4), 888–896 (1972)
171. Purchase, H.C.: Metrics for graph drawing aesthetics. J. Vis. Lang. Comput. **13**, 501–516 (2002)
172. Purchase, H.C., McGill, M., Colpoys, L., Carrington, D.: Graph drawing aesthetics and the comprehension of UML class diagrams: an empirical study. In: Proceedings of the 2001 Asia-Pacific Symposium on Information Visualisation, pp. 129–137. Australian Computer Society, Darlinghurst (2001)
173. Purchase, H.C., Carrington, D., Allder, J.A.: Empirical evaluation of aesthetics-based graph layout. Empir. Softw. Eng. **7**(3), 233–255 (2002)
174. Rana, K.: Resource Aware Load Balancing Using Multi Objective Optimization in Cloud. LAP LAMBERT Academic Publishing, Saarbrücken (2017)
175. Rastrigin, L.A.: Statistical Models of Search (in Russian). Nauka, Moscow (1968)
176. Regis, R., Shoemaker, C.: Constrained global optimization of expensive black box functions using radial basis functions. J. Glob. Optim. **31**, 153–171 (2005)
177. Regis, R., Shoemaker, C.: A stochastic radial basis method for the global optimization of expensive functions. INFORMS J. Comput. **19**(4), 497–509 (2007)
178. Ritter, K.: Average-Case Analysis of Numerical Problems. Lecture Notes in Mathematics, vol. 1733. Springer, New York (2000)
179. Ruíz-Canales, P., Rufián-Lizana, A.: A characterization of weakly efficient points. Math. Program. **68**(1–3), 205–212 (1995). doi:10.1007/BF01585765
180. Ruthven, D.: Principles of Adsorption and Adsorption Processes. Wiley-Interscience, New York (1984)
181. Saule, C., Giegerich, R.: Pareto optimization in algebraic dynamic programming. Algorithms Mol. Biol. **10**(1), 22 (2015). doi:10.1186/s13015-015-0051-7
182. Savage, L.: The Foundations of Statistics. Wiley, London (1954)
183. Sayin, S.: Measuring the quality of discrete representation of efficient sets in multiple objective mathematical programming. Math. Program. **87 A**, 543–560 (2000)
184. Schaefer, R.: Foundation of Global Genetic Optimization. Springer, Berlin (2007)
185. Scholz, D.: Deterministic Global Optimization: Geometric Branch-and-Bound Methods and Their Applications. Springer, Berlin (2012)
186. Schwefel, H.P.: Numerische Optimierung von Computer Modellen. Birkhäuser, Basel (1977)
187. Schwefel, H.P.: Nonlinear Optimization of Computer Models. Wiley-Blackwell, Hoboken (1981)

188. Serafini, P.: Some considerations about computational complexity for multi objective combinatorial problems. Lect. Notes Econ. Math. Syst. **294**, 222–232 (1986)
189. Sergeyev, Ya.D.: An information global optimization algorithm with local tuning. SIAM J. Optim. **5**(4), 858–870 (1995)
190. Sergeyev, Ya.D.: A one-dimensional deterministic global minimization algorithm. Comput. Math. Math. Phys. **35**(5), 553–562 (1995)
191. Sergeyev, Ya.D.: An efficient strategy for adaptive partition of n-dimensional intervals in the framework of diagonal algorithms. J. Optim. Theory Appl. **107**(1), 145–168 (2000)
192. Sergeyev, Ya.D.: Efficient partition of n-dimensional intervals in the framework of one-point-based algorithms. J. Optim. Theory Appl. **124**(2), 503–510 (2005)
193. Sergeyev, Ya.D., Kvasov, D.E.: Global search based on efficient diagonal partitions and a set of Lipschitz constants. SIAM J. Optim. **16**, 910–937 (2006). doi:10.1137/040621132
194. Sergeyev, Ya.D., Kvasov, D.E.: Diagonal Global Optimization Methods (In Russian). FizMatLit, Moscow (2008)
195. Sergeyev, Ya.D., Kvasov, D.E.: Lipschitz global optimization and estimates of the Lipschitz constant. In: Chaoqun, M., Lean, Y., Dabin, Z., Zhongbao, Z. (eds.) Global Optimization: Theory, Methods and Applications, I, pp. 518–521. Global Link Publications, Hong Kong (2009)
196. Sergeyev, Ya.D., Kvasov, D.E.: Lipschitz global optimization. In: Cochran, J.J., Cox, L.A., Keskinocak, P., Kharoufeh, J.P., Smith, J.C. (eds.) Wiley Encyclopaedia of Operations Research and Management Science, vol. 4, pp. 2812–2828. Wiley, Hoboken (2011)
197. Sergeyev, Ya.D., Strongin, R.G., Lera, D.: Introduction to Global Optimization Exploiting Space-Filling Curves. Springer, New York (2013)
198. Shepp, L.A.: The joint density of the maximum and its location for a Wiener process with drift. J. Appl. Probab. **16**, 423–427 (1976)
199. Shubert, B.O.: A sequential method seeking the global maximum of a function. SIAM J. Numer. Anal. **9**(3), 379–388 (1972)
200. Skelboe, S.: Computation of rational interval functions. BIT Numer. Math. **14**(1), 87–95 (1974). doi:10.1007/BF01933121
201. Spronk, J., Hallerbach, W.: Financial modelling: where to go? With an illustration for portfolio management. Eur. J. Oper. Res. **99**(1), 113–125 (1997). doi:10.1016/S0377-2217(96)00386-4
202. Stein, M.: Interpolation of Spatial Data. Springer, Berlin (1999)
203. Steuer, R.: Multiple Criteria Optimization: Theory, Computation, and Applications. Wiley, Hoboken (1986)
204. Steuer, R., Choo, E.U.: An interactive weighted tchebychev procedure for multiple objective programming. Math. Program. **26**(3), 326–344 (1983)
205. Steuer, R.E., Na, P.: Multiple criteria decision making combined with finance: a categorized bibliographic study. Eur. J. Oper. Res. **150**(3), 496–515 (2003). doi:10.1016/S0377-2217(02)00774-9
206. Strongin, R.G.: Numerical Methods of Multiextremal Minimization (in Russian). Nauka, Moscow (1978)
207. Strongin, R.G.: Algorithms for multi-extremal mathematical programming problems employing the set of joint space-filling curves. J. Glob. Optim. **2**, 357–378 (1992)
208. Strongin, R.G., Sergeyev, Ya.D.: Global Optimization with Non-Convex Constraints: Sequential and Parallel Algorithms. Kluwer Academic Publishers, Dordrecht (2000)
209. Su, C.T., Hsu, C.M.: Multi-objective machine-part cell formation through parallel simulated annealing. Int. J. Prod. Res. **36**(8), 2185–2207 (1998). doi:10.1080/002075498192841
210. Sukharev, A.: On optimal strategies of search for an extremum (in Russian). USSR Comput. Math. Math. Phys. **11**(4), 910–924 (1971)
211. Sukharev, A.: Best strategies of sequential search for an extremum (in Russian). USSR Comput. Math. Math. Phys. **12**(1), 35–50 (1972)
212. Sukharev, A.: A sequentially optimal algorithm for numerical integration. J. Optim. Theory Appl. **28**(3), 363–373 (1979)

213. Tarapata, Z.: Selected multicriteria shortest path problems: an analysis of complexity, models and adaptation of standard algorithms. Int. J. Appl. Math. Comput. Sci. **17**(2), 269–287 (2007). doi:10.2478/v10006-007-0023-2
214. Tavakkoli-Moghaddam, R., Ranjbar-Bourani, M., Amin, G., Siadat, A.: A cell formation problem considering machine utilization and alternative process routes by scatter search. J. Intell. Manuf. **23**, 1127–1139 (2012). doi:10.1007/s10845-010-0395-2
215. Thanassoulis, E.: Selecting a suitable solution method for a multi objective programming capital budgeting problem. J. Bus. Financ. Account. **12**(3), 453–471 (1985). doi:10.1111/j.1468-5957.1985.tb00846.x
216. Törn, A., Žilinskas, A.: Global optimization. Lect. Notes Comput. Sci. **350**, 1–252 (1989)
217. Traub, J.F., Wasilkowski, G.W., Wozniakowski, H.: Information, Uncertainty, Complexity. Addison-Wesley, Reading (1983)
218. Traub, J., Wasilkowski, G., Wozniakowski, H.: Information-Based Complexity. Academic, London (1988)
219. Turan, A.: Multi Objective Optimization of Structures Under Multiple Loads. LAP LAMBERT Academic Publishing, Saarbrücken (2017)
220. Ulungu, E.L., Teghem, J.: Multi-objective combinatorial optimization problems: a survey. J. Multi-Criteria Decis. Anal. **3**(2), 83–104 (1994). doi:10.1002/mcda.4020030204
221. van Laarhoven, P., Aarts, E.: Simulated Annealing: Theory and Applications. Springer, Dordrecht (1987)
222. Varoneckas, A., Žilinskas, A., Žilinskas, J.: Multi-objective optimization aided to allocation of vertices in aesthetic drawings of special graphs. Nonlinear Anal. Modell. Control **18**(4), 476–492 (2013)
223. Venugopal, V., Narendran, T.: A genetic algorithm approach to the machine-component grouping problem with multiple objectives. Comput. Ind. Eng. **22**(4), 469–480 (1992). doi:10.1016/0360-8352(92)90022-C
224. Wagner, T., Emmerich, M., Deutz, A., Ponweiser, W.: On expected-improvement criteria for model-based multi-objective optimization. Lect. Notes Comput. Sci. **6238**, 718–727 (2010)
225. Wang, S.: Second-order necessary and sufficient conditions in multiobjective programming. Numer. Funct. Anal. Optim. **12**(1–2), 237–252 (1991). doi:10.1080/01630569108816425
226. Weingartner, H.M.: Mathematical Programming and the Analysis of Capital Budgeting Problems. Markham Publishing and Co., Chicago (1967)
227. Wemmerlov, U., Johnson, D.J.: Empirical findings on manufacturing cell design. Int. J. Prod. Res. **38**(3), 481–507 (2000). doi:10.1080/002075400189275
228. Wierzbicki, A.: Basic properties of scalarization functionals for multi-objective optimization. Math. Oper. Stat. Ser. Optim. **8**(1), 669–691 (1977)
229. Wood, G.R.: Multidimensional bisection applied to global optimisation. Comput. Math. Appl. **21**(6–7), 161–172 (1991)
230. Wood, G.R.: The bisection method in higher dimensions. Math. Program. **55**, 319–337 (1992)
231. Wybrow, M., Marriott, K., Stuckey, P.: Orthogonal connector routing. Lect. Notes Comput. Sci. **5849**, 219–231 (2010)
232. Yang, S.D., Lee, D., Wong, C.: Rectilinear path problems among rectilinear obstacles revisited. SIAM J. Comput. **24**, 457–472 (1995)
233. Zabinsky, Z.: Stochastic Adaptive Search for Global Optimization. Kluwer, Dordrecht (2003)
234. Zarepisheh, M., Pardalos, P.M.: An equivalent transformation of multi-objective optimization problems. Ann. Oper. Res. **249**(1), 5–15 (2017). doi:10.1007/s10479-014-1782-4
235. Zhang, B.P., Wood, G., Baritompa, W.: Multidimensional bisection: the performance and the context. J. Global Optim. **3**(3), 337–358 (1993)
236. Zhigljavsky, A.: Mathematical Theory of Global Random Search (in Russian). Leningrad University Press, Leningrad (1985)
237. Zhigljavsky, A.: Branch and probability bound methods for global optimization. Informatica **1**(1), 125–140 (1990)
238. Zhigljavsky, A.: Theory of Global Random Search. Kluwer, Dordrecht (1991)
239. Zhigljavsky, A., Žilinskas, A.: Stochastic Global Optimization. Springer, Dordrecht (2008)

240. Žilinskas, A.: On one-dimensional multimodal optimization. Izvestija Acad. Nauk USSR Eng. Cybern. (in Russian) **4**, 71–74 (1976)
241. Žilinskas, A.: Algorithm as 133: optimization of one-dimensional multimodal functions. J. R. Stat. Soc. Ser. C: Appl. Stat. **27**(3), 367–385 (1978)
242. Žilinskas, A.: Axiomatic approach to statistical models and their use in multimodal optimization theory. Math. Program. **22**, 104–116 (1982)
243. Žilinskas, A.: Axiomatic characterization of a global optimization algorithm and investigation of its search strategies. Oper. Res. Lett. **4**, 35–39 (1985)
244. Žilinskas, A.: Global Optimization: Axiomatic of Statistical Models, Algorithms, Applications (in Russian). Mokslas, Vilnius (1986)
245. Žilinskas, J.: Reducing of search space of multidimensional scaling problems with data exposing symmetries. Inf. Technol. Control **36**(4), 377–382 (2007)
246. Žilinskas, J.: Branch and bound with simplicial partitions for global optimization. Math. Model. Anal. **13**(1), 145–159 (2008). doi:10.3846/1392-6292.2008.13.145-159
247. Žilinskas, A.: On similarities between two models of global optimization: statistical models and radial basis functions. J. Glob. Optim. **48**, 173–182 (2010)
248. Žilinskas, A.: On strong homogeneity of two global optimization algorithms based on statistical models of multimodal objective functions. Appl. Math. Comput. **218**(16), 8131–8136 (2012)
249. Žilinskas, A.: On the worst-case optimal multi-objective global optimization. Optim. Lett. **7**, 1921–1928 (2013)
250. Žilinskas, A.: On the statistical models-based multi-objective optimization. In: Rassias, T., Floudas, C., Butenko, S. (eds.) Optimization in Science and Engineering: In Honor of the 60th Birthday of Panos M. Pardalos, pp. 597–610. Springer Science+Business Media, New York (2014)
251. Žilinskas, A.: A one-step optimal algorithm for bi-objective univariate optimization. Optim. Lett. **8**, 1945–1960 (2014)
252. Žilinskas, A.: A statistical model-based algorithm for black-box multi-objective optimization. Int. J. Syst. Sci. **45**(1), 82–93 (2014)
253. Žilinskas, A., Zhigljavsky, A.: Branch and probability bound methods in multi-objective optimization. Optim. Lett. **10**(2), 341–353 (2016). doi:10.1007/s11590-014-0777-z
254. Žilinskas, A., Žilinskas, J.: Branch and bound algorithm for multidimensional scaling with city-block metric. J. Glob. Optim. **43**, 357–372 (2009). doi:10.1007/s10898-008-9306-x
255. Žilinskas, A., Žilinskas, J.: Optimization based visualization. In: Floudas, C.A., Pardalos, P.M. (eds.) Encyclopedia of Optimization, pp. 2785–2791. Springer, New York (2009)
256. Žilinskas, A., Žilinskas, J.: Interval arithmetic based optimization in nonlinear regression. Informatica **21**(1), 149–158 (2010)
257. Žilinskas, A., Žilinskas, J.: A hybrid global optimization algorithm for non-linear least squares regression. J. Glob. Optim. **56**(2), 265–277 (2013). doi:10.1007/s10898-011-9840-9
258. Žilinskas, J., Žilinskas, A.: Bounding fronts in multi-objective combinatorial optimization with application to aesthetic drawing of business process diagrams. In: Batsyn, M.V., Kalyagin, V.A., Pardalos, P.M. (eds.) Models, Algorithms and Technologies for Network Analysis: From the Third International Conference on Network Analysis, pp. 127–139. Springer International Publishing, Cham (2014). doi:10.1007/978-3-319-09758-9_11
259. Žilinskas, A., Žilinskas, J.: Adaptation of a one-step worst-case optimal univariate algorithm of bi-objective Lipschitz optimization to multidimensional problems. Commun. Nonlinear Sci. Numer. Simul. **21**(1–3), 89–98 (2015). doi:10.1016/j.cnsns.2014.08.025
260. Žilinskas, A., Fraga, E.S., Mackutė, A.: Data analysis and visualisation for robust multi-criteria process optimisation. Comput. Chem. Eng. **30**(6–7), 1061–1071 (2006)
261. Žilinskas, A., Mackutė-Varoneckienė, A., Varoneckas, A.: Weighting criteria of aesthetic attractiveness of layouts of business process diagrams. In: STOPROG 2012: Stochastic Programming for Implementation and Advanced Applications, pp. 142–147. Technika, Vilnius (2012)

262. Žilinskas, J., Lančinskas, A., Guarracino, M.R.: Application of multi-objective optimization to pooled experiments of next generation sequencing for detection of rare mutations. PloS One **9**(9), 1–9 (2014). doi:10.1371/journal.pone.0104992
263. Žilinskas, A., Fraga, E., Beck, J., Varoneckas, A.: Visualization of multi-objective decisions for the optimal design of a pressure swing adsorption system. Chemom. Intell. Lab. Syst. **142**, 151–158 (2015)
264. Žilinskas, J., Goldengorin, B., Pardalos, P.M.: Pareto-optimal front of cell formation problem in group technology. J. Glob. Optim. **61**(1), 91–108 (2015). doi:10.1007/s10898-014-0154-6
265. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms—a comparative study. In: Eiben, A. (ed.) Conference on Parallel Problem Solving From Nature, pp. 292–301. Springer, Amsterdam (1998)
266. Zitzler, E., Deb, K., Thiele, L.: Comparison of multi objective evolutionary algorithms: empirical results. Evol. Comput. **8**(2), 173–195 (2000)
267. Zitzler, E., Thiele, L., Laummanns, M., Fonseca, C.M., da Fonseca, V.G.: Performance assessment of multiobjective optimizers: an analysis and review. IEEE Trans. Evol. Comput. **7**(2), 117–132 (2003)
268. Zopounidis, C.: Multicriteria decision aid in financial management. Eur. J. Oper. Res. **119**(2), 404–415 (1999). doi:10.1016/S0377-2217(99)00142-3
269. Zopounidis, C., Pardalos, P.M.: Handbook of Multicriteria Analysis. Springer, Berlin (2010)

# Index





P.M. Pardalos et al., *Non-Convex Multi-Objective Optimization*, Springer Optimization and Its Applications 123, DOI 10.1007/978-3-319-61007-8