# SOFT COMPUTING FOR COMPLEX MULTIPLE CRITERIA DECISION MAKING

Ignacy Kaliszewski

Operations Research • Management Science

Springer's INTERNATIONAL SERIES

# Soft Computing For Complex Multiple Criteria Decision Making

***Recent titles in the*** **INTERNATIONAL SERIES IN OPERATIONS RESEARCH & MANAGEMENT SCIENCE**
**Frederick S. Hillier, Series Editor**, *Stanford University*

Maros/ *COMPUTATIONAL TECHNIQUES OF THE SIMPLEX METHOD*
Harrison, Lee & Neale/ *THE PRACTICE OF SUPPLY CHAIN MANAGEMENT: Where Theory and Application Converge*
Shanthikumar, Yao & Zijm/ *STOCHASTIC MODELING AND OPTIMIZATION OF MANUFACTURING SYSTEMS AND SUPPLY CHAINS*
Nabrzyski, Schopf & Węglarz/ *GRID RESOURCE MANAGEMENT: State of the Art and Future Trends*
Thissen & Herder/ *CRITICAL INFRASTRUCTURES: State of the Art in Research and Application*
Carlsson, Fedrizzi, & Fullér/ *FUZZY LOGIC IN MANAGEMENT*
Soyer, Mazzuchi & Singpurwalla/ *MATHEMATICAL RELIABILITY: An Expository Perspective*
Chakravarty & Eliashberg/ *MANAGING BUSINESS INTERFACES: Marketing, Engineering, and Manufacturing Perspectives*
Talluri & van Ryzin/ *THE THEORY AND PRACTICE OF REVENUE MANAGEMENT*
Kavadias & Loch/*PROJECT SELECTION UNDER UNCERTAINTY: Dynamically Allocating Resources to Maximize Value*
Brandeau, Sainfort & Pierskalla/ *OPERATIONS RESEARCH AND HEALTH CARE: A Handbook of Methods and Applications*
Cooper, Seiford & Zhu/ *HANDBOOK OF DATA ENVELOPMENT ANALYSIS: Models and Methods*
Luenberger/ *LINEAR AND NONLINEAR PROGRAMMING, 2nd Ed.*
Sherbrooke/ *OPTIMAL INVENTORY MODELING OF SYSTEMS: Multi-Echelon Techniques, Second Edition*
Chu, Leung, Hui & Cheung/ *4th PARTY CYBER LOGISTICS FOR AIR CARGO*
Simchi-Levi, Wu & Shen/ *HANDBOOK OF QUANTITATIVE SUPPLY CHAIN ANALYSIS: Modeling in the E-Business Era*
Gass & Assad/ *AN ANNOTATED TIMELINE OF OPERATIONS RESEARCH: An Informal History*
Greenberg/ *TUTORIALS ON EMERGING METHODOLOGIES AND APPLICATIONS IN OPERATIONS RESEARCH*
Weber/ *UNCERTAINTY IN THE ELECTRIC POWER INDUSTRY: Methods and Models for Decision Support*
Figueira, Greco & Ehrgott/ *MULTIPLE CRITERIA DECISION ANALYSIS: State of the Art Surveys*
Reveliotis/ *REAL-TIME MANAGEMENT OF RESOURCE ALLOCATIONS SYSTEMS: A Discrete Event Systems Approach*
Kall & Mayer/ *STOCHASTIC LINEAR PROGRAMMING: Models, Theory, and Computation*
Sethi, Yan & Zhang/ *INVENTORY AND SUPPLY CHAIN MANAGEMENT WITH FORECAST UPDATES*
Cox/ *QUANTITATIVE HEALTH RISK ANALYSIS METHODS: Modeling the Human Health Impacts of Antibiotics Used in Food Animals*
Ching & Ng/ *MARKOV CHAINS: Models, Algorithms and Applications*
Li & Sun/ *NONLINEAR INTEGER PROGRAMMING*
Kaliszewski/ *SOFT COMPUTING FOR COMPLEX MULTIPLE CRITERIA DECISION MAKING*
Bouyssou et al/ *EVALUATION AND DECISION MODELS WITH MULTIPLE CRITERIA: Stepping stones for the analyst*
Blecker & Friedrich/ *MASS CUSTOMIZATION: Challenges and Solutions*
Appa, Pitsoulis & Williams/ *HANDBOOK ON MODELLING FOR DISCRETE OPTIMIZATION*
Herrmann/ *HANDBOOK OF PRODUCTION SCHEDULING*
Axsäter/ *INVENTORY CONTROL, 2nd Ed.*

***** A list of the early publications in the series is at the end of the book ****

# Soft Computing For Complex Multiple Criteria Decision Making

IGNACY KALISZEWSKI

Springer

Ignacy Kaliszewski
Systems Research Institute of the
Polish Academy of Sciences
Warszawa, Poland

Library of Congress Control Number: 2005935530

ISBN-13: 978-0387-30243-0 (HB) ISBN-13: 978-0387-30177-8 (e-book)
ISBN-10: 0-387-30243-3 (HB) ISBN-10: 0-387-30177-1 (e-book)

Printed on acid-free paper.



Printed in the United States of America.

9 8 7 6 5 4 3 2 1 SPIN 11052739

springer.com

*To*
*Ewa.*

*To*
*Dominika,*
*Kacper,*
*Agnieszka.*

# Contents

# List of Figures

# List of Tables

# Preface

*"Daprima importa sappere di che cosa si tratta."*
(*"The first thing is to know what the talk is about"*.)

Attributed to **Vilfredo Pareto**.

This book results from my continuous and deep interest in multiple criteria decision making (MCDM). Eleven years ago I wrote in my previous monograph: *"This work results from my interest in the field of vector optimization. I stumbled first upon this subject in 1982 [ ... ]. I was attracted then by a gap between vector optimization used to serve as a formal model for multiple objective decision problems and the decision problems themselves, the gap nonexistent in scalar optimization. Roughly speaking, vector optimization provides methods for ranking decisions according to a partial order whereas decision making requires a linear ordering of decisions."* This declaration is still valid and nothing needs to be changed.

To be more specific, this book is a fruit of my dissatisfaction with the current state-of-the-art of MCDM. MCDM is a branch of science, whose declared ultimate goal is to provide practical tools. However, we cannot say, and this is regrettable, that all present MCDM methods and algorithms are in popular use by those who make complex decisions and for that purpose are in need of methodological or computational support. And this is despite the number of scientists involved in research in the field, thought to be well over one thousand persons active worldwide, and the cumulative number of relevant publications, assessed to be in a range of thousands. I have acquired a sharper view on the matter since when holding, in addition to my scientific affiliation, some advisory positions in business.

It seems that the key word to understand the situation is "involvement". Moreover, it is the complexity of interfaces between users and MCDM methods rather than the involvement of the methods that is to be blamed for limited MCDM popularity in real-life decision making. To illustrate and to put in a broader perspective what I mean by an interface let me recall the famous Black-Scholes close-end formula for pricing stock options. The methodology behind the Black-Scholes development is involved indeed and calls for knowledge of advanced stochastic calculus. However, the interface - in this case the formula itself - can be easily programmed and calculated in a spreadsheet or in a pocket calculator with financial built-in functions. This is how knotty mathematics can become accessible to a wide spectrum of lay users.

Though the notion of close-end formulas is not relevant to MCDM, lay and broad use of MCDM will be not possible without simple, low computing-intensive methods. This, I claim, is the main challenge in MCDM for the coming years.

This book is a modest step towards meeting this challenge. When writing it my governing rule was to use only as much formalism as can be immediately consumed in practical schemes and support tools.

The book summarizes my research performed in the Systems Research Institute of the Polish Academy of Science in the period after the publication of my first monograph on MCDM in 1994.

The title for the book was selected, on purpose, to be slightly intriguing. I hope, however, that the contents of the book fully justifies it.

*Warsaw, June 2005.*

# Notation

| | | |
|---|---|---|
| $\mathcal{R}^k$ | – | $k-$ dimensional real space ($\mathcal{R}$ for $k=1$), |
| $\mathcal{X}$, $\mathcal{Y}$ | – | spaces, |
| $A, ..., Z$ | – | sets, |
| $R^k_+$ | – | nonnegative orthant of $\mathcal{R}^k$ , |
| $x, y$ | – | elements of a space or a set , |
| $x_i, y_i$ | – | $i-$ th component of elements $x, y$ , |
| $\{x, y, \, ... \}$ | – | set composed of elements $x, y, \, ...$ , |
| $\emptyset$ | – | empty set, |
| $\subseteq$ | – | set inclusion, |
| $\subset$ | – | proper set inclusion, |
| $\cup$ | – | union of sets, |
| $\cap$ | – | intersection of sets, |
| $\setminus$ | – | difference of sets, |
| $+\, ,\, -$ | – | algebraic summation and subtraction of scalars, vectors, or sets, |
| int( $\cdot$ ) | – | interior of a set, |
| $\mid \cdot \mid$ | – | cardinality of a set, |
| $\|\| \cdot \|\|$ | – | norm, |
| $dist(y, A)$ | – | distance from element $y$ to set $A,\ dist(y, A) = \min_{y' \in A} \|\|y - y'\|\|$ |

$yy'$ – scalar product; for $y \in \mathcal{R}^k,\ y' \in \mathcal{R}^k,\ yy' = \sum_{i=1}^{k} y_i y_i'$.

# Chapter 1

# INTRODUCTION

*"There is more religion in men's science than there is science in their religion."*

**Henry David Thoreau**,
*A Week on the Concord and Merrimac Rivers.*

In a loose formulation a decision making problem could be stated as follows:

$$\textit{given a set of alternatives,}\\ \textit{choose a feasible alternative, which according to}\\ \textit{decision making circumstances is the most preferred.} \tag{1.1}$$

To survive in risky environments and to quell the nature, human beings possess brains capable of making decisions the vast majority of which are correct. Otherwise *homo sapiens* would perhaps still dwell in wild.

Every day people make myriads of decisions. When there is no time for reflection, decisions are made subconsciously and automatically. When time for reflection is limited, decisions can be made consciously but still quite automatically. Finally, there are also decisions, which result from thorough, time-spanned analysis.

Nowadays decision making is a field of active research in neurology, psychology, and behavioral science. If there is no need to decide instantaneously and the context is intricate, decision making becomes also a branch of applied mathematics and computer science, which is where this book belongs.

We still do not know how we make decisions. Although the advances in brain research are spectacular, there are few firm hints how to mimic brain analytical capacity. Reactions to simple stimuli can be correctly foreseen. But how decisions are made in more involved settings remains a mystery. Consequently, there are no general rules available on how to construct automated decision making devices able to replace human beings in their creative capacity. At best we can be provided with tools for supporting decision making by suitable data structuring and data processing. The need for such a support is evident in the class of problems we are concerned with in this book. We are concerned with problems in which each *feasible alternative* is evaluated against a set of at least two quality *criteria*. Such problems are known as *multiple criteria decision making* (MCDM) problems.

As any decision making process, MCDM processes are composed of four phases:

- problem intelligence,
- model design,
- solution choice,
- problem review,

(cf. Figure 1.1). This book concentrates on the third phase, which is technically the most involved.

More specifically, this book concentrates on *complex problems*. By complex problems we mean problems, which to be efficiently handled need to be captured by a formal model and investigated with the help of optimization methods.

The algorithmic approach to MCDM problems provides algorithms supposed to mimic the *decision maker* (DM) if he behaves consistently with a number of assumptions. There is an inherent trap in this approach, because verification of the DM consistency with respect to assumptions is a problem at least as involved as the decision making problem itself. To give an example, it is quite common to assume that the DM is consistent with the assumption of transitivity, which states that if the DM prefers alternative $a$ to alternative $b$, and alternative $b$ to alternative $c$, then necessarily he prefers $a$ to $c$. Many examples can be given that it is not always the case. But as decisions have to be taken this or that way,

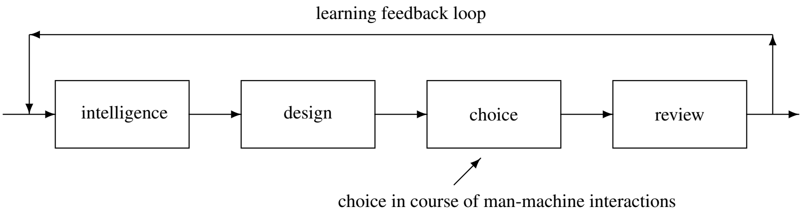


*Figure 1.1.* Four phases of decision making processes.

some DMs facing a complex decision making problem are tempted to make use of available decision making algorithms without paying much attention to the underlying assumptions.

In simple instances of decision making problems with a few alternatives DMs are often (but by no means always) quite convinced which alternative is the most preferred. Thus, in such cases, there is no need for using any decision making algorithm.

In other problems complexities can be of various sorts, ranging from problems where feasible alternatives given in the form of a list are just too numerous for their mutual relations to be fully perceived by the DM, to highly structured and mathematically modeled problems. A sort of decision making support is then required.

The strong form of the algorithmic approach assumes that the DM selects the most preferred alternative by evaluating alternatives against a scoring system defined by a function. Moreover, it is assumed that such a function, called *value function*, is known in an explicit form. Given the value function the decision making process reduces to finding a feasible alternative which maximizes that function. But the existence of a value function is guaranteed only by fulfillment of some rigid assumptions about DM's preferences, whose verification is a problem in itself.

In a real business environment it is hard to imagine the DM consenting to undergo a test for his consistency with a set of assumptions. Findings of tests conducted in laboratory environments on volunteers or students, when transferred to a practical setting can be misleading or even not valid. The fact that consistency of the assumptions cannot be tested and therefore assumed, puts a question mark on the practical usefulness of the strong form of the algorithmic approach to decision making.

Another, and by no means insignificant, problem is the construction of the explicit value function, a process which calls for a far reaching cooperation between the DM and an analyst. It can hardly be believed that the real DM (say, a company manager) will reveal his preferences in full. This might have a devastating effect on his ego and career because as soon as he permitted his role to be reduced to a set of rules known to the public, he would become vulnerable to critics and vultures. He could even fear for his job position.

For these reasons MCDM methods have drifted over time from the strong form of the algorithmic approach to softer schemes. This has been happening with healthy competition from other soft decision making methodologies such as neural networks, fuzzy and rough sets, artificial intelligence, and last but not least, heuristics.

The soft form of the algorithmic approach to MCDM is *interactive decision making* (interactive MCDM) and nowadays MCDM problems are usually tackled in this way. With this approach MCDM problems are solved via a sequence of "man-machine" interactions. At one interaction the DM expresses his partial preferences (the "man" phase) and, using the underlying formal model of the problem, a feasible alternative is selected which fits those partial preferences best (the "machine" phase).

In interactive MCDM the solution to a decision making problem is the selection of the most preferred alternative which the DM arrives at by a "tour", i.e. sequence of trial alternatives. During such a tour the DM interacts with the model while directing himself by evaluations of trial alternatives and by his preferences reflecting his current state of mind. Overall DM's preference structure remains hidden even at the end of the decision making process and this corresponds well with the DM's unwillingness or inability to frame it.

Since in general it is not possible to verify assumptions about the DM's behavior, it is not practical to make any. But then what are we left with? Is there any room then for a structured supporting methodology? In a quest for the most preferred alternative, has the role of the DM to be reduced just to passively moving from one alternative to another? Is interactive MCDM in fact a form of random search?

Fortunately, we are left with a plentiful supply of riches to deal with the three basic notions of MCDM, namely *efficiency*, criteria, and *trade-offs*. In this book we exploit these notions within the interactive framework to propose Generic Interactive MCDM Support Scheme – $GIS^2$ – applicable to any MCDM problem. $GIS^2$ is just a flexible procedure to help the DM to master the bulk of structured data in complex problems. It is not, however, intended to replace the DM in any aspect of his decision making capacity.

$GIS^2$ follows the interactive decision making principle, where the main role in the scheme is reserved for the DM. With $GIS^2$ the DM navigates *himself* through feasible alternatives with the help of the aforementioned notions of efficiency, criteria, and trade-offs, as his navigating tools. Interactions stop when the DM wishes. The primary role of $GIS^2$ is to support problem structure learning, while decision making necessarily follows on.

$GIS^2$ scheme is by no means a new invention. On the contrary, it is an abstraction of elements common in the existing interactive MCDM methods.

Presenting the $GIS^2$ alone would not merit a book but we propose more than that. First, we propose and exploit in $GIS^2$ "foreseeing" mechanisms to assess the criteria values of alternatives as well as their respective trade-offs, without the necessity to explicitly identify them. Second, we show how to eliminate optimization from MCDM processes and we extend $GIS^2$ to Generic Interactive MCDM Soft Support Scheme – $GIS^3$ – to adopt this idea. In consequence, $GIS^3$ is spanned over two dimensions of softness:

- soft (i.e. interactive) decision processes,
- soft computing (i.e. involving no optimization in decision processes).

**Who should read the book?** In the first place this book will be of interest for all those people *all this fuss is about*, namely MCDM practitioners and lay (in the sense of MCDM methodologies) decision makers. With this readership in mind great efforts have been made to make the book easy to follow.

However, this book is a research monograph and as such it should also be also of interest for research and the academic community, graduate and PhD students included, and anybody with an interest in decision making theory and in MCDM in particular. Also specialists in optimization methods, research and practice orientated, should find this book of interest to understand what their role will be with $GIS^2$ and $GIS^3$ in place.

The book is composed of nine chapters. Chapter 2 to Chapter 4 give a general map of the MCDM field. Chapter 5 to Chapter 9 form the proper research part, with new or recent ideas and results.

For readers who are actual or potential decision makers step-wise (cf. *How to read the book?* paragraph below) reading of the whole material is recommended. Readers already familiar with MCDM would probably wish to confine themselves to Chapter 5 to Chapter 9, but they may also find interest in the concise outline of the interactive MCDM field given in Chapter 2 to Chapter 4.

**How to read the book?** Throughout the book all formal results are given without proofs and the reader is directed to the references.

The sequence of the book can be broken down into three reading tracks. Starting with Chapter 2 each chapter begins with the introductory section "This Chapter is About ..." which presents the contents in a descriptive and informal manner. These eight sections, all with the same title, are the fastest reading track, suitable for those who want to get acquainted with the book at a glance. The second track is more demanding and is recommended for those who are interested in formal aspects of MCDM methodologies but would prefer to leave technical details to subsequent readings. Those readers should simply skip over parts of the text which start with ♣ and end with ♠. The last track is composed of the text in full.

All chapters except Introduction end with *Concluding Remarks* followed by *Annotated References* sections. We believe that grouping references in one place improves readability of the text.

**The outline of the book** The outline of the book is as follows.

In Chapter 2 decision problems to be investigated are formulated, both verbally and formally. Basic notions and definitions are also introduced there.

Chapter 3 presents basic algorithmic tools used in the existing interactive MCDM methods.

Chapter 4 brings a condensed overview of interactive MCDM methodologies and methods.

In Chapter 5 it is shown that the most prominent interactive MCDM methods in the "man" phases of decision making processes can be reduced to manipulations of a few standard and common items. Moreover, in the "machine" phases these methods share the same standard requirements for computations. Because of these observations a form of a standard interface for interactive MCDM methods is propounded.

Chapter 6 presents what we believe is the most relevant, though rudimentary, interactive multiple criteria decision making scheme, namely $GIS^2$, and it is demonstrated how the existing methods fit to this scheme.

Chapter 7 and Chapter 8 present the main idea of the book, namely the idea of using in interactive MCDM problems approximate rather than exact values to save on optimization computing.

Chapter 7 presents how to derive assessments of criteria values of alternatives without resorting directly to solving optimization problems (which otherwise are to be solved to give the exact criteria values).

Chapter 8 shows how to derive assessments of trade-offs of alternatives without resorting directly to solving optimization problems (which otherwise are to be solved to derive exact trade-off values).

In Chapter 9, finally, we discuss and illustrate how to enhance $GIS^2$ methods with the developments of Chapter 7 and Chapter 8 to eventually arrive at a soft version of $GIS^2$, namely $GIS^3$.

## Annotated References

The four phased scheme of decision making has been popularized by Simon (Simon 1977).

Chapter 2

# PRELIMINARIES

*Principieremo col definire un termine di cui è comodo fare uso per scansare lungaggini. Diremo che i componenti di una collettività godono, in una certo posizione, del* MASSIMO DI OFELIMITÀ, *quando è imposible allontanarsi pochissimo da quella posizione giovando, o nuocendo, a tutti i componenti la collettività; ogni piccolissimo spostamento da quella posizione avendo necessariamente per effetto di giovare a parte dei componenti la collettività e di nuocere ad altri.*

[ *We will begin by defining a term which is desirable to use in order to avoid prolixity. We will say that the members of a collectivity enjoy* MAXIMUM OPHELIMITY *in a certain position when it is impossible to find a way of moving from that position very slightly in such a manner that the ophelimity enjoyed by each of the individuals of that collectivity increases or decreases. That is to say, any small displacement in departing from that position necessarily has the effect of increasing the ophelimity which certain individuals enjoy, and decreasing that which others enjoy, of being agreeable to some and disagreeable to others.* ]

**Vilfredo Pareto,**
*Manuale di Economia Politica.*

## 1. This Chapter Is About ...

... comparing alternatives. We compare alternatives with respect to criteria. Since, as a rule, we admit more than one criterion, the matter is not straightforward. But first, let us observe that even with two or more criteria there are situations when in a set of feasible alternatives one alternative is preferred to any other with respect to all criteria. Such an alternative is called *utopian*. Table 2.1 and related Figure 2.1 present the case where only two feasible alternatives are available and one of them (alternative A) is utopian. Figure 2.2. shows the situation where a utopian alternative does not exist (for there is no feasible alternative corresponding to the most preferred criteria values as represented by $\hat{y}$).

*Table 2.1.* Two alternatives, one of which is utopian.

| | *Criterion type* | *Alternative A* | *Alternative B* |
|---|---|---|---|
| Criterion 1 | "better if more" | 75 | 25 |
| Criterion 2 | "better if more" | 75 | 25 |

In reality utopian alternatives happen infrequently. On the other hand, it quite often happens that one of a pair of alternatives is preferred with respect to all criteria. This is called *dominance*. With the notion of dominance we easily arrive at the notion of *efficiency*. We also recall two related notions, namely *weak efficiency* and *proper efficiency*, which we will need in subsequent chapters.

To avoid ambiguities we need a dose of formalism and so we define MCDM problems in terms of sets of feasible alternatives, criteria, and *outcomes*.

Finally, we revisit the idea of trade-off and we define two particular instances of this notion.

In MCDM problems both efficiency and trade-off notions serve to differentiate alternatives.

## 2. Dominance and Efficiency

At the core of MCDM is the notion of efficiency. This is the bridge between informal definition (1.1) and Multiple Criteria Decision Mak-

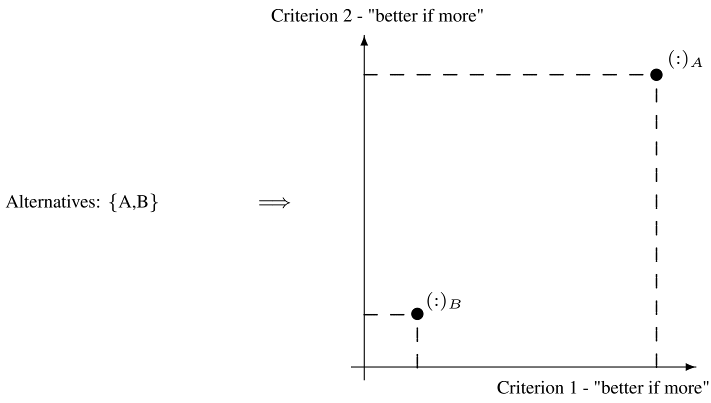


*Figure 2.1.* An example of two alternatives.

ing. The notion emerges immediately and naturally when one evaluates alternatives with respect to more than just one criterion.

If alternatives are compared, for example, with respect to the costs they incur (imagine buying a house), then quite naturally the cheapest seems to be the most preferred. But what if an additional criterion comes into play? In the case of buying a house size is never to be forgotten. A popular belief is that the bigger a house the better (within, of course, certain sensible limits). But what if the two criteria, cost and size, are analyzed jointly? Can the cost be the lowest for the biggest house? If so, this would be a "free lunch", but the reality of real estate markets shows us that bigger the house, the greater the cost. It does not, however, exclude the possibility that we can sometimes discover on a local market a bigger house being cheaper than a smaller one (we ignore any other possible criteria). Such situations are formalized by the notion of dominance.

Given a set of alternatives, a feasible alternative $x$ is called *dominated* if there is another feasible alternative in the set, say alternative $x'$, such that:

- $x'$ is equally or more preferred than $x$ with respect to all criteria, and
- $x'$ is more preferred than $x$ for at least one criterion.

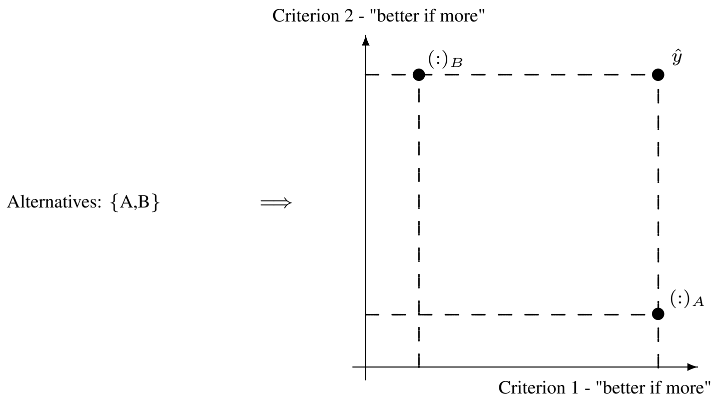


*Figure 2.2.* An example where a utopian alternative does not exist.

If the above holds, the alternative $x'$ is called *dominating*. A pair of alternatives $x$ and $x'$, where $x$ is dominated and $x'$ is dominating, is said to be in *Pareto dominance relation*. Clearly, in a set of more than two alternatives, one alternative can be dominating and at the same time dominated.

Throughout the book we adopt the convention that all criteria are of "better if more" type. We also make the assumption that criteria are represented by numerical values. Then, if a criterion is of "better if less" type, we can always change it to "better if more" type by multiplying all possible values of this criterion by $-1$.

Given a set of feasible alternatives, an alternative which is not dominated by any other alternative of this set is called *efficient*. In other words, an alternative is efficient if there is no other alternative in the set:

- equally or more preferred with respect to all criteria,

and

- more preferred for at least one criterion.

To be efficient an alternative is required much less than to be utopian. Consequently, efficient alternatives are more common than utopian. An utopian alternative is necessarily efficient but not vice versa.

Alternatives which are not efficient are called *nonefficient*.

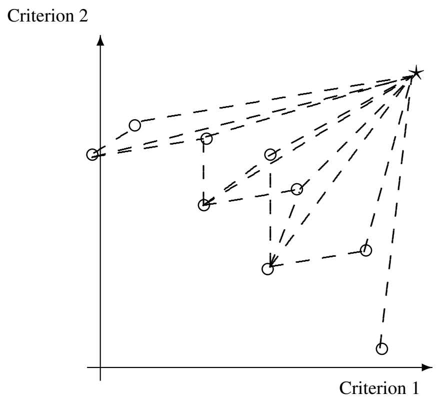


*Figure 2.3.* Pairs of alternatives in Pareto dominance relationships - case I.

With the convention that stars represent utopian alternatives, solid disks represent efficient but not utopian alternatives, and circles represent dominated alternatives, Figure 2.3 and Figure 2.4 show a handful of alternatives represented by values of two criteria. Dashed lines between pairs of alternatives indicate that those alternatives are in Pareto dominance relationship. In Figure 2.3 one alternative is clearly utopian (and therefore it is efficient). That alternative, by definition, is in Pareto dominance relationship with all the remaining alternatives. Figure 2.4 shows the same set of alternatives with the utopian alternative removed. In this case several alternatives are efficient.

If a set of alternatives is given implicitly by a number of conditions (constraints), the number of alternatives can be infinite. It is impossible then to represent graphically all Pareto dominance relationships but we can still do that for selected pairs of alternatives. Figure 2.5 gives an example of a set of alternatives represented by values of two criteria, where the set of criteria values has the shape of a polygon. In this case efficient alternatives are those whose criteria vectors form a part of the polygon border, as marked in the figure by the thick line.

It is common that nonefficient alternatives are neglected in the decision making processes by being clearly unreasonable candidates for the most preferred alternative. In our house buying example, if one can buy more for less, why do otherwise? Why not consume a "free lunch"? But it

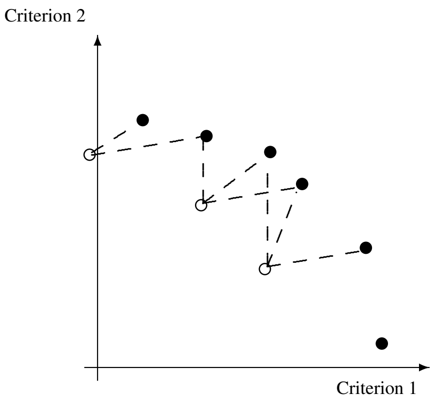


*Figure 2.4.* Pairs of alternatives in Pareto dominance relationships - case II.

is reasonable not to remove nonefficient alternatives from consideration permanently. In a changed decision making environment (e.g. the set of criteria changed) nonefficient (i.e. dominated) alternatives may become efficient. Therefore, it is wise to tag dominated alternatives as only temporarily neglected and keep them for possible future analysis.

With the above remark on the possible non-permanent non-efficiency status of alternatives in mind, we can now safely state that the majority of MCDM methods consist in selecting the most preferred alternative from efficient alternatives.

## 3. Definitions And Problem Settings

In what follows we shall need a dose of formalism to structure decision making problem data.

In the MCDM framework decision making problem (1.1) is formalized as follows:

$$\text{choose an alternative } x \text{ for which vector } f(x),\ x \in X_0 \subseteq \mathcal{X}, \quad \text{is the most preferred,} \tag{2.1}$$

where $\mathcal{X}$ is the set (space) of potential alternatives, $X_0$ is the set of feasible alternatives, $f : \mathcal{X} \to \mathcal{R}^k$ is the criteria map in which $f = (f_1, ..., f_k)$,

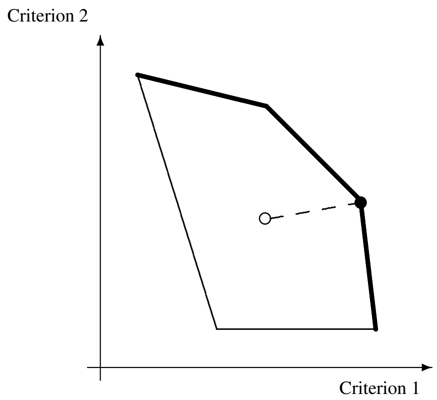


*Figure 2.5.* Pairs of alternatives in Pareto dominance relationships - case III.

and $f_i : \mathcal{X} \rightarrow \mathcal{R}$ are *criteria functions*, $i = 1, ..., k$, $k \geq 2$. As said before, we assume that all criteria are of "better if more" type.

An alternative $x$ for which $f(x)$ is the most preferred vector of criteria values, is the *most preferred* alternative.

The underlying assumption for the main stream of research done in the field of MCDM is that the environment for decision making is certain (as opposed to uncertain). In this book we are concerned exclusively with decision making problems which can be fairly modeled under such an assumption.

♣ If the assumption about the certain nature of the decision making environment is not justified, one has to take appropriate measures to act in an uncertain environment. Below we present, in broad outline, how decision making problems in certain and uncertain environments are formulated and how those two decision frameworks differ.

*Setting A* - certain environment, single criterion.

Given a set of $n$ alternatives, each alternative is evaluated with respect to the criterion and yields a score. Assume that the DM has value function $v$, which to every alternative considered assigns a numerical value $v^j$, $j = 1, ..., n$, representing the DM's preference with respect to the score of that alternative. The most preferred alternative is that showing the highest value of the value function. Table 2.2 shows data for Setting A.

*Table 2.2.* Decision making problem – Setting A.

| | *Alternative* 1 | ... | *Alternative* $n$ |
|---|---|---|---|
| *Criterion* | $\theta^1$ | . | $\theta^n$ |
| *Value* | $v^1$ | . | $v^n$ |

This is the most basic setting of decision making problems. Very often value functions are assumed to be identity mappings, i.e. decisions are made on the base of scores.

*Setting B* - certain environment, multiple criteria.

Given a set of $n$ alternatives, each alternative is evaluated with respect to the set of $k$ criteria and yields a $k$-tuple of scores. Assume that the DM has value function $v$ which to alternative $j$ assigns numerical value $v^j,\ j = 1, ..., n$, representing the DM's preference with respect to $k$-tuple of scores of that alternative. The most preferred alternative is that showing the highest value of value function $v$. Table 2.3 shows data for Setting B.

The existence of a value function of that sort is just a theoretical postulate, with all its practical drawbacks that we discussed in Chapter 1. This book is about how to help the DM to make decisions without resorting to the concept of value function.

Setting B is a problem formulation for a finite number of alternatives. In this book, however, MCDM problems are formulated in the manner which admits a finite and infinite number of alternatives. A good example of an infinite number of alternatives occurs when one selects a percentage of a certain resource to be used or invested.

*Setting C* - uncertain environment, single criterion.

Given a set of $n$ alternatives, each alternative is evaluated with respect to the criterion and yields a score. This is done for every state $i$ of nature which may occur with probability $p_l,\ l = 1, ..., m$. It is assumed that the DM has value function $v$ in the form of expected value of scores: $v^j = \sum_{l=1}^{m} p_l \theta_l^j,\ j = 1, ..., n$. The most preferred alternative is that

*Table 2.3.* Decision making problem – Setting B.

| | *Alternative* 1 | ... | *Alternative* $n$ |
|---|---|---|---|
| *Criterion* 1 | $\theta_1^1$ | $\cdot$ | $\theta_1^n$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| *Criterion* $k$ | $\theta_k^1$ | ... | $\theta_k^n$ |
| *Value* | $v^1$ | ... | $v^n$ |

showing the highest value of the value function. Table 2.4 shows data for Setting C.

*Table 2.4.* Decision making problem – Setting C.

| | *Alternative* 1 | ... | *Alternative* $n$ | *Probability* |
|---|---|---|---|---|
| *State* 1 | $\theta_1^1$ | $\cdot$ | $\theta_1^n$ | $p_1$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| *State* $m$ | $\theta_m^1$ | ... | $\theta_m^n$ | $p_m$ |
| *Expected value* | $v^1$ | ... | $v^n$ | |

*Setting D* - uncertain environment, multiple criteria.

Setting D can be arrived at either from Setting B by admitting uncertainty or from Setting D by admitting multiple criteria.

Given a set of $n$ alternatives, each alternative is evaluated with respect to the set of $k$ criteria and yields $k$-tuple of scores. This is done for every state $i$ of nature which may occur with probability $p_l,\ l = 1, ..., m$. DM's value function $v$ with respect to each criterion is, as in Setting C, the expected value of scores: $v_i^j = \sum_{l=1}^m p_l \theta_{l,i}^j\,,\ i = 1, ..., k,\ j = 1, ..., n.$

Now the DM faces the problem of selecting from among alternatives characterized by $k$-tuples of numbers (expected values). Assume, as in Setting B, that the DM has value function $V$ which to every alternative considered assigns numerical value $V^j,\ j = 1, ..., n$, representing the DM's preference to $k$-tuple of expected values of that alternative. The most preferred alternative is that showing the highest value of value function $V$. Table 2.5 shows data for Setting D.

Research on decision making problems conforming to Setting D is limited. This is due to the involved structure of this framework.

*Table 2.5.* Decision making problem – Setting D.

| | *Alternative* 1 | ... | *Alternative* $n$ | *Probability* |
|---|---|---|---|---|
| *State* 1 | $\theta^1_{1,1}$ ⋮ $\theta^1_{1,k}$ | ⋮ | $\theta^n_{1,1}$ ⋮ $\theta^n_{1,k}$ | $p_1$ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| *State* $m$ | $\theta^1_{m,1}$ ⋮ $\theta^1_{m,k}$ | ⋮ | $\theta^n_{m,1}$ ⋮ $\theta^n_{m,k}$ | $p_m$ |
| *Expected values* | $v^1_1$ ⋮ $v^1_k$ | ⋮ | $v^n_1$ ⋮ $v^n_k$ | |
| *Value* | $V^1$ | ... | $V^n$ | |

Relations between all four formulations are represented graphically in Figure 2.6.

♠

From the algorithmic point of view problem (2.1) is ill defined. In fact, as long as we do not know what "most preferred" means precisely we are not in a position to propose any problem solving method. This information, explicit or implicit, if exists, is in the exclusive possession of the DM. The underlying assumption of MCDM methodologies we are concerned with in this book is that this information cannot be acquired from the DM up front and at once.

In the sequel we frequently exploit the fact that the following problem, called *vector optimization problem*,

$$\textit{vmax}\ f(x),\ \ x \in X_0 \subseteq \mathcal{X}, \tag{2.2}$$

where *vmax* stands for the identification of all efficient alternatives, is almost always well defined. By this we mean that under minor assumptions satisfied in practical applications the solution to (2.2) always exists.

Without any ambiguity alternatives are represented by their criteria values. With this in mind we deal mainly with elements $f(x)$ of set $f(X_0)$ and for the sake of simplicity we use $y$ and $Z$ to denote

$$y = f(x),\ \text{and}\ Z = f(X_0).$$

Clearly, $Z \subseteq \mathcal{R}^k$. We call elements of set $Z$ *outcomes* and space $\mathcal{R}^k$ the *outcome space*. Under this convention, for given feasible alternative $x,\ \ y_i$ is the value of the $i-$th component of outcome $y = f(x)$. Thus, $y_i$ is the value of $i-$th criterion for alternative $x$.

All properties of alternatives considered in subsequent paragraphs and chapters can be defined and operationalized in terms of outcomes. We directly refer to notation $x$ , $X_0$, $f(x)$, $f(X_0)$ only when giving examples of MCDM problems with implicitly (i.e. in the form of constraints) defined feasible alternatives.

Element $\hat{y}$ of $\mathcal{R}^k$, called *utopian,* is calculated as

$$\hat{y}_i = max_{y \in Z}\, y_i,\ i = 1, ..., k.$$

Throughout this book it is assumed that all these maxima exist.

Element $\hat{y}$ need not represent any feasible alternative.

Below we recall the definitions of efficiency, weak efficiency, and proper efficiency of outcomes.

DEFINITION 1 The outcome $\bar{y} \in Z$ is:
- *efficient* if $y_i \geq \bar{y}_i$ , $i = 1, ..., k,\ y \in Z$, implies $y = \bar{y}$,
- *weakly efficient* if there is no $y,\ y \in Z$, such that $y_i > \bar{y}_i$ , $i = 1, ..., k$,

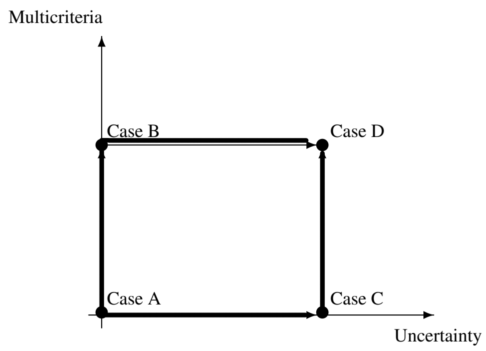


*Figure 2.6.* Four settings of decision making problems.

- *properly efficient* if it is efficient and there exists a finite number $M > 0$ such that for each $i$ we have

$$\frac{y_i - \bar{y}_i}{\bar{y}_j - y_j} \le M$$

for some $j$ such that $y_j < \bar{y}_j$ , whenever $y \in Z$ and $y_i > \bar{y}_i$ . □

Outcomes which are not efficient are called *nonefficient*.

Clearly, from the definition of efficient (nonefficient) alternatives given in Section 2 it follows that an alternative is efficient (nonefficient) if and only if its outcome is efficient (nonefficient).

It is common to call the subset of all efficient outcomes of $Z$ the *Pareto set* and below we shall often use this term.

The notion of efficiency remains at the core of MCDM. For $y \in Z$ the implications:

$$\begin{array}{c} y \text{ is properly efficient } \Rightarrow\ y \text{ is efficient,} \\ y \text{ is efficient } \Rightarrow\ y \text{ is weakly efficient,} \end{array} \tag{2.3}$$

hold, but implications

$$\begin{array}{c} y \text{ is weakly efficient } \Rightarrow\ y \text{ is efficient,} \\ y \text{ is efficient } \Rightarrow\ y \text{ is properly efficient,} \end{array}$$

are, in general, false.

♣ There is a direct relation between the Vilfredo Pareto definition of maximal ophelimity collectively enjoyed, given on the first page of this chapter, and the notion of efficient outcome. Under the assumption that all the criteria are of "better if more" type, and assuming that for each criterion the ophelimity of its value is measured by its value itself, each efficient outcome represents an alternative at which the collective ophelimity, represented by all criteria values jointly, is maximized.
♠

The reason for dealing with the notion of weak efficiency and proper efficiency is purely technical. There exist some algorithmically and computationally convenient methods, presented in Chapter 3, to derive weakly efficient outcomes and properly efficient outcomes (and thus the corresponding alternatives, weakly efficient and properly efficient). The following relations

$$\{\text{all properly efficient outcomes of } Z\} \subseteq \{\text{all efficient outcomes of } Z\}$$

$$\subseteq \{\text{all weakly efficient outcomes of } Z\}, \tag{2.4}$$

which result from implications (2.3), hold. Therefore, it is common to refer to the subset of all weakly efficient outcomes of problem (2.1) (or (2.2)) as an upper approximation of the Pareto set, and the subset of all properly efficient outcomes as a lower approximation of the Pareto set.

Weakly efficient outcomes which are nonefficient occur if two or more outcomes are located on a hyperplane parallel to any of the axes. This is illustrated in Figure 2.7.

Figure 2.8 shows an outcome set where all elements of the boundary marked with the thick line are properly efficient except those three marked with bullets. From this example we intuitively infer that *improperly* (i.e. not properly) efficient outcomes (alternatives) are, in a sense, rare and therefore of little significance in real decision problems.

If set of outcomes $Z$ is polyhedral (the set represented in Figure 2.5 is polyhedral and bounded, thus it is two dimensional polytope, hence polygon), then all efficient outcomes are properly efficient. Then, in this case the notion of proper efficiency is redundant. This notion is also redundant in the case of finite sets $Z$, since in such sets for all efficient outcomes numbers $M$ from Definition 1 of this chapter can be found by enumeration.

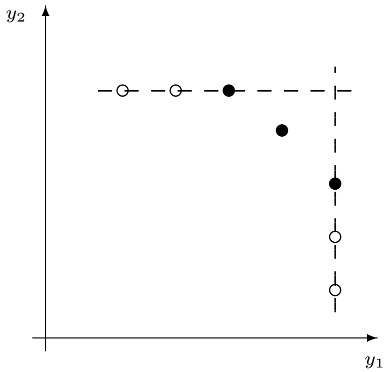


*Figure 2.7.* Weakly efficient and efficient outcomes.

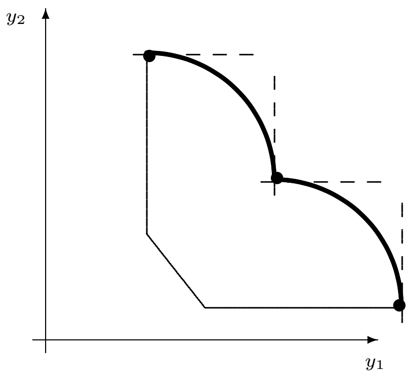


*Figure 2.8.* Improperly efficient outcomes.

♣ Another definition of outcome efficiency, equivalent to that given in Definition 1 of this chapter, can also be given in terms of cones. Set $K$ is a cone if $\lambda K \in K$ for any $\lambda \geq 0$.

Let $R^k_+$ denote the nonnegative orthant $\{y \in \mathcal{R}^k \,|\, y_i \geq 0,\, i = 1, ..., k\}$. The nonnegative orthant is a cone. $R^k_+$ can be interpreted as the set of directions of improvement (recall that all criteria are of "better if more" type), i.e. given an element $y \in \mathcal{R}^k$, any element $y',\ y' \in \mathcal{R}^k,\ y' \neq y,$

such that $y' \in \{y\} + R^k_+$ has all coordinates at least as great as $y$ and at least one coordinate strictly greater than $y$.

The other definition of outcome efficiency states that an outcome $y$ (recall that an outcome is an element of $Z$) is efficient if there is no outcome $y'$, $y' \neq y$, such that $y' \in \{y\} + R^k_+$. In other words, $y \in Z$ is efficient if $(\{y\} + R^k_+) \cap Z = \{y\}$.

The notion of cone is very convenient in MCDM for graphical illustrations of efficiency issues.

First of all, it is easy to check graphically whether an outcome is efficient or not. It is enough to place cone $R^k_+$ at an outcome whose efficiency has to be verified, say outcome $y$, and see if set $\{y\} + R^k_+$ contains any outcome different than $y$. If it does not, $y$ is efficient, otherwise it is nonefficient. In Figure 2.9 bullet marked outcomes are efficient.

If $y$ is nonefficient, then by definition $y$ is *dominated* by some other outcome, say $y'$ ($y'$ *dominates* $y$), such that $y' \in \{y\} + R^k_+$, or equivalently, $y \in \{y'\} - R^k_+$.

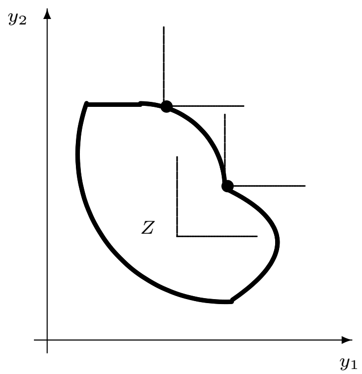


*Figure 2.9.* Checking efficiency with cone $R^k_+$.

With cone $R^k_+$ it is also possible to give another definition of weak efficiency, equivalent to that given in Definition 1 of this chapter. Namely, an outcome $y$ is weakly efficient if there is no outcome $y'$, $y' \neq y$, such that $y' \in \{y\} + int(R^k_+)$, where $int(\cdot)$ denotes the interior of a set. In other words, $y \in Z$ is weakly efficient if $(\{y\} + int(R^k_+)) \cap Z = \{\emptyset\}$.

Hence, to check graphically whether an outcome is weakly efficient or not it is enough to place cone $R_+^k$ at the outcome whose weak efficiency has to be verified, say outcome $y$, and see if set $\{y\} + int(R_+^k)$ contains any outcome. If it does not, $y$ is weakly efficient, otherwise it is not weakly efficient. In Figure 2.10 the bullet marked outcome is weakly efficient.

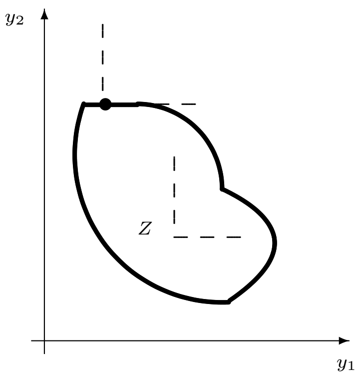


*Figure 2.10.* Checking weak efficiency with interior of cone $R_+^k$.

The definition of efficiency based on cone $R_+^k$ provides for easy generalization of the notion of efficiency and weak efficiency. Let $K$ be any convex cone. We say that outcome $y$ is $K$-efficient if $(\{y\} + K) \cap Z = \{y\}$ and that it is $K$-weakly efficient if $(\{y\} + int(K)) \cap Z = \{\emptyset\}$.

With the generalized definition of efficiency we can also provide a more illustrative, but equivalent, definition of proper efficiency than that given in Definition 1 of this chapter. Namely, an outcome $y$ is properly efficient if there exists a convex cone $K$ such that $R_+^k \setminus \{0\} \subseteq int(K)$ and $y$ is efficient with respect to $K$, i.e. $(\{y\} + K) \cap Z = \{y\}$. In Figure 2.11 the bullet marked outcome is properly efficient.

♠

## 4. Trade-offs

Following Webster's Ninth New Collegiate Dictionary (1987) "trade-off" is understood as: 1. a balancing of factors all of which are not

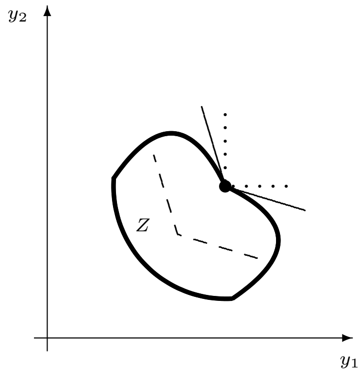


*Figure 2.11.* Checking proper efficiency with a convex cone containing $R^k_+$.

attainable at the same time, 2. a giving up of one thing in return for another.

In MCDM "trade-off" means: a loosing in one outcome component (criterion) to gain the value of another.

Given a pair of outcomes, a trade-off exists only if no outcome dominates the other (cf. Figure 2.12, right drawing). If otherwise, e.g. $y'$ dominates $y$, then $y'$ with respect to $y$ offers only gains, and $y$ with respect to $y'$ offers only losses.

Moreover, it is practical to consider trade-offs only between pairs of efficient outcomes. Indeed, if an outcome $y$ is nonefficient, then there exists an outcome, say $y'$, which dominates $y$. By replacing $y$ by $y'$ we do not deteriorate the value of any component (criterion) value (Figure 2.12, left drawing). Hence, $y'$ offers better trades than $y$. An efficient outcome offers better trades than any outcome dominated by it (cf. Figure 2.12, right drawing).

A trade-off always refers to one selected reference outcome, say $\bar{y}$. But depending on the context, this notion can refer to two different constructs.

The first construct is the so called *point-to-point* trade-off. It is defined for a pair of selected components of a pair of designated efficient outcomes.

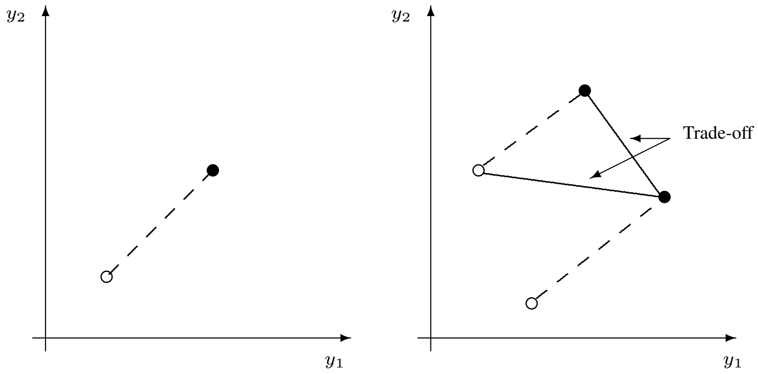


*Figure 2.12.* Pairs of elements for which trade-offs do not exist (dashed lines) or exist (continuous lines).

DEFINITION 2 Point-to-point trade-off $T_{i,j}^{PTP}(\bar{y}, \tilde{y})$, where $\bar{y}$ is a reference efficient outcome and $\tilde{y}$ is an efficient outcome, involving components $i$ and $j$, $i, j = 1, ..., k, i \neq j$, such that

$$\tilde{y}_i - \bar{y}_i \geq 0 \text{ and } \bar{y}_j - \tilde{y}_j > 0, \tag{2.5}$$

is defined as

$$T_{i,j}^{PTP}(\bar{y}, \tilde{y}) = \frac{\tilde{y}_i - \bar{y}_i}{\bar{y}_j - \tilde{y}_j}\,. \tag{2.6}$$

□

For pairs of components $i$ and $j$, $i, j = 1, ..., k$, $i \neq j$, such that the conditions (2.5) do not hold, point-to-point trade-offs are not defined.

For an efficient reference outcome $\bar{y}$ at most $k - 1$ point-to-point trade-offs exist. Indeed, since the point-to-point trade off is defined for a pair of efficient outcomes, say $\bar{y}$ and $\tilde{y}$, for at least one component, say component $l$, the relation $\bar{y}_l - \tilde{y}_l > 0$ must hold for otherwise $\bar{y}$ would be nonefficient. Hence, at most $k - 1$ components fulfill the relation $\tilde{y}_{\cdot} - \bar{y}_{\cdot} \geq 0$ and therefore at most $k - 1$ point-to-point trade-offs exist for an efficient reference outcome $\bar{y}$.

Suppose that one unit of component $i$ has the same value for the DM as one unit of component $j$. If $T_{i,j}^{PTP}(\bar{y}, \tilde{y}) > 1$, then with respect to components $i$ and $j$ reference outcome $\bar{y}$ is more preferred than $\tilde{y}$. And

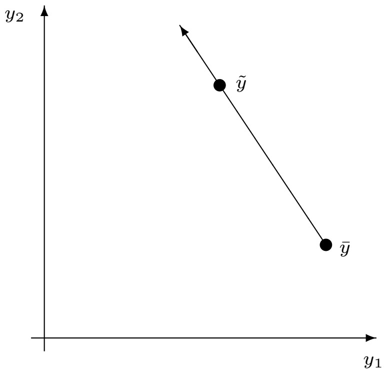


*Figure 2.13.* The double nature of a trade-off: value and direction.

symmetrically, $T^{PTP}_{i,j}(\bar{y}, \tilde{y}) < 1$ indicates that with respect to $i$ and $j$ outcome $\tilde{y}$ is more preferred than $\bar{y}$. However, this simple "which is more preferred" rule works only for a selected pair of components or in the case $k = 2$. In Chapter 6 we shall address the problem of how to make use of trade-off information when more than two components are taken into account. In any case large (note that the meaning of "large" is context-dependent) values of a point-to-point trade-offs $T^{PTP}_{\cdot,\cdot}(\bar{y}, \tilde{y})$ may be an important information for the DM for his outcome evaluations.

Point-to-point trade-offs can be associated with a direction, i.e. point-to-point trade-offs $T^{PTP}_{i,j}(\bar{y}, \tilde{y})$, $i, j = 1, ..., k$ , $i \neq j$ , can be read from any point of the ray leaving $\bar{y}$ and passing through $\tilde{y}$, as shown in Figure 2.13.

If outcomes are explicitly given, then point-to-point trade-offs are of little use. For a pair of efficient outcomes point-to-point trade-offs provide no additional information to that represented by components of outcomes (values of criteria). In fact they provide even less information, because not for all pairs of indices point-to-point trade-offs are defined. In that case trade-offs are nothing else than just another form of presenting information on outcome components of a pair of outcomes.

The notion of point-to-point trade-off becomes useful when sets of outcomes are infinite. Given efficient reference outcome $\bar{y}$, $\bar{y} \in Z$, and a pair of indices $i$ and $j$, $i \neq j$, one can pose two questions:

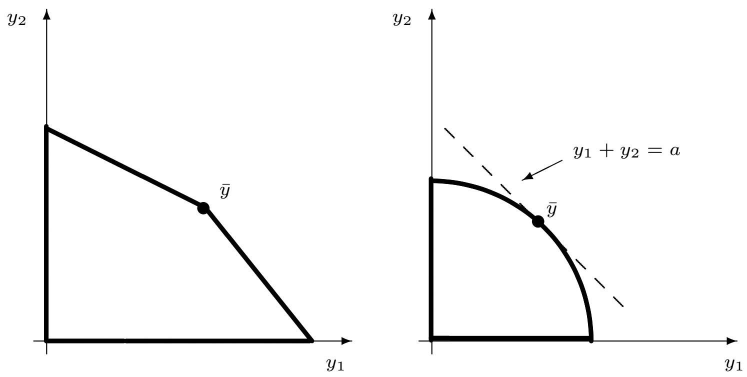


*Figure 2.14.* The maximum of point-to-point trade-offs $T_{\cdot,\cdot}^{PTP}(\bar{y}, y)$ may exist (left drawing) or may not exist (right drawing).

- what is the maximal value of $T_{i,j}^{PTP}(\bar{y}, y)$ over all $y \in Z$, i.e. what is the value of

$$\max_{y \in Z} T_{i,j}^{PTP}(\bar{y}, y),$$

- at which outcome (if at any) this maximum is attained.

If set $Z$ is finite or polyhedral (Figure 2.14, the left drawing), it is guaranteed that $\max_{y \in Z} T_{i,j}^{PTP}(\bar{y}, y)$ always exists. In case $Z$ is polyhedral, all efficient outcomes lying on a line segment starting from $\bar{y}$ have the same point-to-point trade-off values.

It is easy to see that if $Z$ is neither finite nor polyhedral, then $\max_{y \in Z} T_{i,j}^{PTP}(\bar{y}, y)$ may not exist. In Figure 2.14, the right drawing, for efficient reference outcome $\bar{y}$ there is no efficient outcome $y$ such that $\max_{y \in Z} T_{i,j}^{PTP}(\bar{y}, y) = 1$, but clearly $1 - T_{1,2}^{PTP}(\bar{y}, y) < \epsilon$ for some $y \in Z$, where $\epsilon > 0$ is arbitrary small.

A remedy to deal with situations like this above is offered if we replace the maximum with the supremum, where the latter means taking the least upper bound. In the example represented in the right drawing of Figure 2.14, $\sup_{y \in Z} T_{i,j}^{PTP}(\bar{y}, y) = 1$. In general taking just supremum is not very helpful, as illustrated in Figure 2.15 - for efficient reference outcome $\bar{y}$ for no pair of indices $i, j,\ i \neq j$, $\sup_{y \in Z} T_{i,j}^{PTP}(\bar{y}, y)$ is finite. To see this it is enough to project $Z$ on any plane spanned by two out of three axes. For example, in the projection of $Z$ on the plane spanned by axes $y_1$ and $y_2$, for any given number it is always possible to select a pair of values $(y_1, y_2)$ such that $y_2 - \bar{y}_2 > 0,\ \bar{y}_1 - y_1 > 0$, and $\frac{y_2 - \bar{y}_2}{\bar{y}_1 - y_1}$ is greater than this number.

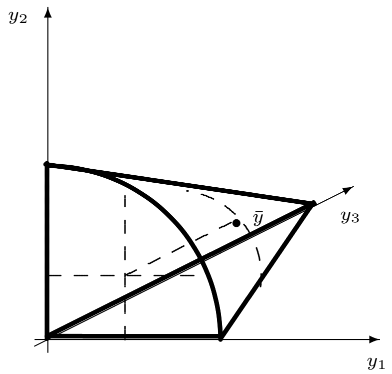


*Figure 2.15.* For outcome $\bar{y}$ and for any pair of indices the supremum of point-to-point trade-offs does not exist.

The problem is, at least partially, solved if we take the supremum not over set $Z$ but over a part of it.

Let $\bar{y} \in Z,\ Z \subseteq \mathcal{R}^k$. For $i = 1, ..., k$, we denote:

$$Z_i^<(\bar{y}) = \{y \in Z \mid y_i < \bar{y}_i\,,\ y_l \geq \bar{y}_l\,,\ l = 1, ..., k,\ l \neq i\}.$$

DEFINITION 3 Global trade–off $T_{i,j}^G(\bar{y})$, where $\bar{y}$ is an efficient reference outcome, involving components $i$ and $j,\ i, j = 1, ..., k,\ i \neq j$, is defined as

$$\sup_{y \in Z_j^<(y)} \frac{y_i - \bar{y}_i}{\bar{y}_j - y_j}\,. \tag{2.7}$$

□

By convention, if $Z_j^<(\bar{y}) = \emptyset$, (i.e. trade-off $T_{i,j}^G$ is not defined) then $T_{i,j}^G(\bar{y}) = -\infty,\ i = 1, ..., k,\ i \neq j.$

♣ The notion of global trade-off applies to both properly and improperly efficient outcomes.

It can be proved that for properly efficient outcomes all global trade-offs are smaller then $+\infty$. Let us recall the special cases of finite or polyhedral sets, where all outcomes are properly efficient.

For improperly efficient outcomes at least one trade-off equals $+\infty$. In Section 3 we observed, however, that in real decision making situations improperly efficient solutions occur infrequently.

In contrast to point-to-point trade-offs, which are defined with respect to two specified outcomes, deriving a global trade-off for an outcome requires calculations that relate to (parts of) outcome set $Z$.

♠

## 5. Concluding Remarks

With the material of this chapter we are well equipped to deal with issues of the next seven chapters. In fact, the notions defined up to now form a firm base for the subsequent presentations and developments and, except for some technical constructs, we shall not introduce any new concepts.

The notion of efficiency is quite intuitive. The notion of trade-off is less so (especially the notion of global trade-off) and the majority of works on MCDM exploit only the first notion without reference to the second. The rationale of using the notion of trade-off (point-to-point or global) in MCDM will be discussed in more detail in Chapter 4 and Chapter 6.

## 6. Annotated References

The definition of efficiency appears in almost every paper on MCDM. Some authors even decline to define this notion in research papers assuming it belongs to a common wisdom. Issues relating to all three related notions of efficiency (weak efficiency, efficiency, proper efficiency) are treated in every MCDM book. One can mention here Haimes et al. (1975), Ignizio (1976), Keeney, Raiffa (1976), Cohon (1978), Hwang et al. (1979), Rietveld (1980), Zeleny (1982), Chankong, Haimes (1983), Guddat et al. (1985), Ignizio (1985), Sawaragi et al. (1985), Yu (1985), Jahn (1986), Steuer (1986), Galas et al. (1987), Tabucanon (1988), Lewandowski, Wierzbicki (1989), Luc (1989), Haimes et al. (1990), Ringuest (1992), Vincke (1992), Kaliszewski (1994), Yoon, Hwang (1995), Skulimowski (1996), Miettinen (1999), and also (a survey paper) Stadler (1979).

The definition of proper efficiency in Definition 1 of this chapter comes from Geoffrion (1968).

The notion of trade-off has been used in the framework of vector optimization (a field with focus on problem (2.2) approached by rigid mathematical analysis) since the early 1950s and then it has penetrated MCDM. It refers to two different but related issues in MCDM. The first issue is that of properties of explicit or implicit value functions (Kuhn, Tucker 1951, Chankong, Haimes 1978, Haimes, Chankong 1979, Sakawa, Yano 1990). Under the assumption of differentiability of an utility function, trade-offs are interpreted as partial derivatives of this function with respect to criteria functions.

The second issue is benefits and costs of moving from one alternative to another measured by values of relative changes in criteria functions (Zionts, Wallenius 1976,1983, Wierzbicki 1990, Halme 1992, Henig, Buchanan 1997, Kaliszewski 1993, 1994b, Kaliszewski, Michalowski 1995, 1997,1999).

The definition of global trade-off is given first in Wierzbicki (1990), and exploited later in Kaliszewski (1993, 1994b). A method of calculating trade-offs which avoids calculating the supremum of a hyperbolic function is proposed in Kaliszewski (1993,1994b). This method is recalled briefly in Chapter 6 and later used in solving an example of a decision making problem.

Chapter 3

# MCDM - BASIC TOOLS

*"Mariners should navigate with caution as the following changes ..."*

**Notices to Mariners**,
*Weekly Edition 12/2005.*

## 1. This Chapter Is About ...

... leveling the ground for Chapter 4. Here we are concerned with the technical problem of efficient outcome derivation.

Clearly, we do not need any elaborate technique to derive efficient outcomes (and filter out nonefficient ones) if alternatives are a few and they are given explicitly, e.g. by a list. In such cases a simple procedure to verify efficiency based on pairwise comparison is sufficient. But when efficient outcomes are numerous or are given implicitly by a set of conditions or constraints, formal methods are needed.

The idea widely used in interactive MCDM is to have efficient outcomes implicitly described in terms of some parameters. This idea is similar to the marine navigation positioning system, in which each point of the globe is uniquely described by its geographical position: the longitude (East, West) and the latitude (South, North). With using such a system a vessel can sail from one navigable point of the globe to another. Given a positioning system several navigation tools can be applied. Such tools must be universal so that all navigable points are accessible from any other.

In the context of interactive MCDM, where all outcomes are "navigable" but only efficient outcomes are of interest, use is made of "positioning systems" called *characterizations*. For a characterization (a positioning system) specific interactive MCDM methods (navigation tools) are available.

Below we present characterizations available for interactive MCDM. These take forms of parametric optimization problems, and to derive a single efficient outcome one has to solve one optimization problem for a selected set of parameters. Parameter selection schemes, which in our maritime analogy correspond to navigation tools, are discussed in the next chapter.

As Chapter 2 presents basic MCDM notions, Chapter 3 presents basic tools for interactive MCDM. Both chapters contain interactive MCDM formalism we exploit in this book.

## 2. Pareto Set Characterizations

A corner stone for every interactive MCDM method is the ability to derive efficient outcomes. Every efficient outcome should be derivable by a computationally effective method but in general it is difficult to satisfy this postulate. However, this can be done for weakly and properly efficient outcomes.

As seen in Section 3 of Chapter 2 (cf. relations (2.4)), a Pareto set can be represented by its lower and upper approximations, i.e. by the set of properly efficient outcomes (lower approximation) and the set of weakly efficient outcomes (upper approximation). So-called characterizations of weakly efficient outcomes and properly efficient outcomes provide for Pareto set approximations in implicit (parametric) forms.

In this section we briefly present three types of characterizations of weakly efficient and properly efficient outcomes exploited in MCDM methods, namely:

- characterizations by weight manipulations,
- characterizations by reference point manipulations,
- characterizations by constraint manipulations.

In this book a characterization means a set of conditions (necessary and sufficient) for an outcome to be, weakly or properly, efficient.

For a characterization to be practical, the conditions should provide a computationally effective method to derive every weakly efficient outcome or every properly efficient outcome. As shown below, such characterizations exist for weakly efficient outcomes and properly efficient outcomes. For efficient outcomes effective characterizations are not available, except for some special cases where the set of properly efficient outcomes coincides with the set of efficient outcomes.

## 2.1 Characterizations By Weight Manipulations

The idea of characterizing weakly efficient outcomes and properly efficient outcomes by weight manipulations consists in reducing vector optimization problem (2.2) to a scalar optimization problem by assigning *numerical weights* to components of outcomes and forming a *surrogate objective function*. The surrogate objective function when maximized (or minimized - depending on the surrogate objective function form) over $Z$ yields at least one (weakly or properly) efficient outcome of the original vector optimization problem. By changing weights and solving resulting optimization problems one derives different (weakly or properly) efficient outcomes.

There are several ways to characterize sets of weakly efficient outcomes and properly efficient outcomes by weight manipulations. It is widely accepted that four characterizations presented in this section have the most plausible theoretical and computational features.

Below we make use of a selected element of outcome space $\mathcal{R}^k$, denoted $y^*$, defined as

$$y_i^* = \hat{y}_i + \epsilon, \ i = 1, ..., k,$$

where $\epsilon$ is any positive number and $\hat{y}$ is the utopian element introduced in Section 1 of Chapter 2 and defined formally in Subsection 2.3 of that chapter. We use $yy'$ to denote the scalar product of two elements $y$ and $y'$ of $\mathcal{R}^k$, i.e.

$$yy' = \sum_{i=1}^{k} y_i y_i' \, .$$

Hence, for example,

$$e^k(y^* - y) = \sum_{i=1}^{k} (y_i^* - y_i),$$

where $e^k$ is the vector of $k$ elements all equal to 1.

Characterization I

*Sufficient condition*
An outcome which solves optimization problem

$$\min_{y \in Z} \max_i \lambda_i((y_i^* - y_i) + \rho e^k(y^* - y)), \tag{3.1}$$

where $\lambda_i > 0,\ i = 1, ..., k$, and $\rho > 0$, is properly efficient. □

♣ *Necessary condition*
Every properly efficient outcome solves optimization problem (3.1) for some $\lambda_i > 0,\ i = 1, ..., k$, and some $\rho > 0$. □
♠

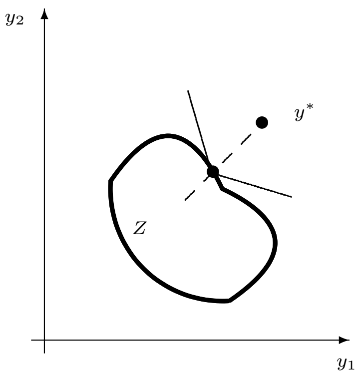


*Figure 3.1.* Deriving properly efficient outcomes with optimization problem (3.1) or (3.2).

Characterization II

*Sufficient condition*
An outcome which solves optimization problem

$$\min_{y \in Z} \max_i \lambda_i(y_i^* - y_i) + \rho e^k(y^* - y), \tag{3.2}$$

where $\lambda_i > 0,\ i = 1, ..., k$, and $\rho > 0$, is properly efficient. □

♣ *Necessary condition*
Every properly efficient outcome solves optimization problem (3.2) for some $\lambda_i > 0,\ i = 1, ..., k$, and some $\rho > 0$. □
♠

Figure 3.1 presents the isogram of the (surrogate) objective function of optimization problem (3.1) or (3.2) for the minimal value of this function over $Z$.

Observe that for $\rho = 0$ problems (3.1) and (3.2) reduce to

$$\min_{y \in Z} \max_i \lambda_i (y_i^* - y_i), \tag{3.3}$$

where $\lambda_i > 0,\ i = 1, ..., k$.

Characterization III

*Sufficient condition*
An outcome which solves optimization problem (3.3) is weakly efficient. □

♣ *Necessary condition*
Every weakly efficient outcome solves optimization problem (3.3) for some $\lambda_i > 0,\ i = 1, ..., k$. □
♠

Figure 3.2 presents the isogram of the (surrogate) objective function of optimization problem (3.3) for the minimal value of this function over $Z$.

We say that set $Z$ is $R_+^k$-convex if $Z - R_+^k$ is a convex set.

Characterization IV

Assume that $Z$ is $R_+^k$-convex.

*Sufficient condition*
An outcome which solves optimization problem

$$\max_{y \in Z} \lambda y, \tag{3.4}$$

where $\lambda_i > 0,\ i = 1, ..., k$, is properly efficient. □

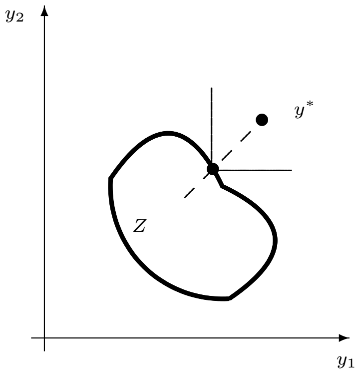


*Figure 3.2.* Deriving weakly efficient outcomes with optimization problem (3.3).

♣ *Necessary condition*
Every properly efficient outcome solves optimization problem (3.4) for some $\lambda_i > 0,\ i = 1, ..., k.$ □
♠

Figure 3.3 presents the isogram of the (surrogate) objective function of optimization problem (3.4) for the maximal value of this function over $Z$.

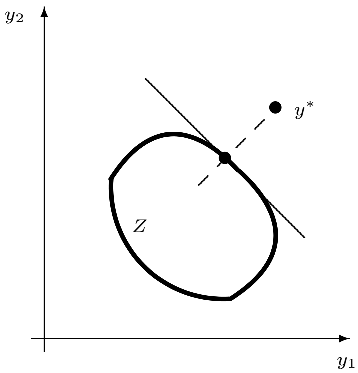


*Figure 3.3.* Deriving properly efficient outcomes with optimization problem (3.4).

♣ One can raise the question of practical solvability of problems (3.1), (3.2), (3.3). However, a simple transformation of each of these problems results in an equivalent formulation. For example, problem (3.1) is clearly equivalent to

$$\max_{y \in Z} t$$

$$t \geq \lambda_i((y_i^* - y_i) + \rho e^k(y^* - y)),\ i = 1, ..., k.$$

To solve the above problem an optimization package is needed capable of handling constraints $y \in Z$ plus $k$ inequality constraints. The form of the surrogate objective function is the simplest possible.

Problems (3.2) and (3.3) can be transformed in the same manner. Thus, provided that set $Z$ is polyhedral, all four problems (3.1), (3.2), (3.3), and (3.4) can be solved by standard linear programming techniques.

♠

♣ By solving problem (3.1) one derives in fact $K^1$-weakly efficient outcomes (which are also $R_+^k$-properly efficient), where

$$K^1 = \{y \in \mathcal{R}^k \mid \lambda_i(y_i + \rho e^k y) \geq 0,\ \lambda_i > 0,\ i = 1, ..., k\},$$

and by solving problem (3.2) one derives $K^2$-weakly efficient outcomes (which are also $R_+^k$-properly efficient), where

$$K^2 = \{y \in \mathcal{R}^k \mid \lambda_i y_i + \rho e^k y \geq 0,\ \lambda_i > 0,\ i = 1, ..., k\},$$

(cf. Chapter 2 Section 2).

For $\rho = 0$ cones $K^1$ and $K^2$ reduce to $K^3$, where $K^3 = R_+^k$. By solving problem (3.3) one derives $K^3$-weakly efficient (i.e. $R_+^k$-weakly efficient) outcomes.

By solving problem (3.4) one derives $K^4$-weakly efficient outcomes (which are also $R_+^k$-properly efficient), where

$$K^4 = \{y \in \mathcal{R}^k \mid \lambda\, y \geq 0,\ \lambda_i > 0,\ i = 1, ..., k\}.$$

♠

♣

REMARK 1 It is worth observing that characterization IV is not a direct consequence of characterization I or characterization II. Passing with $\rho$ to limit $+\infty$, optimization problem (3.1) as well as optimization problem (3.2) reduce to

$$\max_{y \in Z} e^k y,$$

which is only equivalent to one instance of optimization problem (3.4), namely that with all $\lambda_i$ equal, $i = 1, ..., k$.

For optimization problem (3.4) to be the limit of optimization problem (3.1) or (3.2), some additional parameters in (3.1) and (3.2) are required but those are commonly, for the sake of simplicity and without loss of desired properties, neglected.

In general, regarding (3.4) as a special case of (3.1) or (3.2) is only a slight oversimplification with no methodological consequences.

♠

## 2.2 Characterization By Reference Point Manipulations

The idea of characterizing weakly efficient outcomes and properly efficient outcomes by reference point manipulations consists of using a surrogate objective function parameterized by elements of $\mathcal{R}^k$, called *reference points*. A surrogate objective function when minimized over $Z$ yields at least one (weakly or properly) efficient outcome. By changing reference points and solving resulting optimization problems one derives different (weakly or properly) efficient outcomes.

A continuous function $s_{\bar{y}}(y) : \mathcal{R}^k \rightarrow \mathcal{R}$ of parameter $\bar{y}$, where $\bar{y} \in \mathcal{R}^k$, ($\bar{y}$ - a reference point) is called an *achievement function*. It is required that achievement functions posses certain properties, such as being:

- *strictly increasing*,
- *strongly increasing*,
- $\epsilon$-*strongly increasing*,
- *order representing*,
- *order approximating*.

♣ The function $s_{\bar{y}}(y)$ is:

- strictly increasing, if

$$y_i^1 < y_i^2\,,\ i = 1, ..., k,\ \Rightarrow s_{\bar{y}}(y^1) < s_{\bar{y}}(y^2),$$

- strongly increasing, if

$$y_i^1 \le y_i^2\,,\ i = 1, ..., k,\ \text{ and } y_j^1 < y_j^2 \text{ for some } j,$$
$$\Rightarrow s_{\bar{y}}(y^1) < s_{\bar{y}}(y^2),$$

- $\epsilon$-strongly increasing, if for some $\epsilon > 0$

$$y^1 \in y^2 - R_\epsilon^k \backslash \{0\}\ \Rightarrow s_{\bar{y}}(y^1) < s_{\bar{y}}(y^2),$$

where $R_\epsilon^k = \{y \in \mathcal{R}^k \mid dist(y, R_+^k) \le \epsilon\|y\|\}$, (clearly, $R_\epsilon^k$ and $R_+^k \subset R_\epsilon^k$),

- order representing, if it is strictly increasing for any $\bar{y} \in \mathcal{R}^k$ and

$$\{y \in \mathcal{R}^k \mid s_{\bar{y}}(y) > 0\} = \bar{y} + int(R_+^k),$$

- order approximating, if it is strongly increasing for any $\bar{y} \in \mathcal{R}^k$ and

$$\{y \in \mathcal{R}^k \mid s_{\bar{y}}(y) \ge 0\} \subset \bar{y} + R_\epsilon^k\,.$$

♠

For optimization problem

$$\min_{y \in Z} s_{\bar{y}}(y) \tag{3.5}$$

let outcome $\breve{y}$ be its solution, i.e.

$$\breve{y} = \arg(\min_{y \in Z} s_{\bar{y}}(y)).$$

Characterization V

*Sufficient condition*
An outcome which solves optimization problem (3.5) with strictly increasing achievement function $s_{\bar{y}}(y)$ is weakly efficient. □

♣ *Necessary condition*
Every weakly efficient outcome solves optimization problem (3.5) for some order representing achievement function $s_{\bar{y}}(y)$. □
♠

Characterization VI

*Sufficient condition*
An outcome which solves optimization problem (3.5) with $\epsilon-$strongly increasing achievement function $s_{\bar{y}}(y)$ is properly efficient. □

♣ *Necessary condition*
Every properly efficient outcome solves optimization problem (3.5) for some $\epsilon > 0$ and some order approximating achievement function $s_{\bar{y}}(y)$. □

♠

In addition, some achievement functions provide sufficient conditions for efficiency (but not necessary conditions). However, optimization problems (3.5) with those achievement functions are not regarded as computationally efficient.

Various forms of achievement functions exist but achievement functions resulting from objective functions of optimization problems (3.1) and (3.2) via replacing fixed $y^*$ by some reference point $\bar{y}$, i.e. achievement function

$$\max_i \lambda_i((\bar{y}_i - y_i) + \rho e^k(\bar{y} - y)) \tag{3.6}$$

or achievement function

$$\max_i \lambda_i(\bar{y}_i - y_i) + \rho e^k(\bar{y} - y), \tag{3.7}$$

are most often used because with any of them optimization problem (3.5) takes the simplest possible form.

For $\rho > 0$ achievement functions (3.6) and (3.7) are $\epsilon$-strongly increasing and order approximating for some $\epsilon > 0$, and for $\rho = 0$ they reduce to

$$\max_i \lambda_i(\bar{y}_i - y_i), \tag{3.8}$$

and become strictly increasing and order representing.

## 2.3 Characterization By Constraint Manipulations

The idea of characterizing weakly efficient outcomes by constraint manipulations consists in imposing constraints on $k - 1$ arbitrarily selected outcome components. Maximizing the unconstrained outcome component over $Z$ under the imposed constraints yields a weakly efficient outcome. By changing constraints and solving the resulting optimization problems one derives different weakly efficient outcomes.

Characterization VII

*Sufficient condition*
An outcome which solves the following optimization problem

$$\max_{\substack{y \in Z \\ y_l \geq \epsilon_l,\ l=1,\ldots,k,\ l \neq i}} \quad y_i\ , \tag{3.9}$$

where $i \in \{1, ..., k\}$, is weakly efficient. □

♣ *Necessary condition*
Every weakly efficient outcome solves optimization problem (3.9) for some $i \in \{1, ..., k\}$ and some $\epsilon_l,\ l = 1, ..., k,\ l \neq i$. □
♠

## 3. Concluding Remarks

That which of the characterizations presented in this chapter is used in an MCDM method depends on which of three possible ways to express preferences:

- by selecting weights,
- by selecting reference points,
- by selecting constraints on outcome components,

is believed to suit the DM best. Each MCDM method adopts an assumption with respect to this. In the next chapter we make use of the above distinction as a basis for a taxonomy of interactive MCDM methods.

## 4. Annotated References

A formal treatment of characterizations presented in Section 2 of this chapter can be found:

- characterization I: in Choo, Atkins (1983), Wierzbicki (1986), Kaliszewski (1987, 1994a,b),
- characterization II: in Steuer, Choo (1983), Steuer (1986), Wierzbicki (1986), Kaliszewski (1994a,b),
- characterization III: in Bowman (1976), Steuer, Choo (1983), Steuer (1986), Wierzbicki (1986), Kaliszewski (1987, 1994a,b),

- characterization IV: in Geoffrion (1968),
- characterization V and VI: in Wierzbicki (1980, 1986, 1990),
- characterization VII: in Benayoun et al. (1971), Haimes et al. (1971).

Advantages of optimization problems (3.1) and (3.2) over optimization problem (3.4) are discussed in Kaliszewski (2000), cf. also Kaliszewski (1994a,b).

Chapter 4

# MCDM INTERACTIVE METHODS - AN OVERVIEW

*"Man can smile and smile but he is not an investigating animal. He loves the obvious. He shrinks from explanations."*

**Joseph Conrad**,
*The Secret Agent.*

## 1. This Chapter Is About ...

... the presentation of prototypical interactive MCDM methods. These are the most representative methods and any not covered here are variants of these prototypes.

The description of each method is not detailed, and is limited to an outline. There is no introduction of technical details and how they should be implemented. Why have we done this? We want to identify the most important features of these methods. Then we want to trace fundamental differences and similarities. We believe that the reader of this chapter will acquire the knowledge to appreciate the ideas and concepts presented in the next chapter. Step by step we are preparing ourselves for the core of this book. We are getting to the nitty-gritty, and this is dealt with in the following chapters.

## 2. MCDM Interactive Methods

In respect of methodologies, interactive MCDM methods differ by elements: weights, reference points, or constraints, which are selected for manipulations to capture the DM's preferences. This distinction establishes a taxonomy of MCDM methods with three classes: *the weight method class*, *the reference point method class*, and *the constraint method class*. The taxonomy is exhaustive because in fact weights, reference points, and constraints are the only meaningful objects relating to the outcome space and available for manipulations in course of interactive decision making.

As said in Introduction, when solving MCDM problems interactively the search for the most preferred decision consists in a tour between efficient outcomes. At the DM's discretion the tour starts and goes through efficient trial outcomes. At each efficient trial outcome the DM, using his knowledge acquired about the decision problem, expresses his partial preferences with respect to efficient outcomes he evaluates and compares. Then, preferences are used to form a set of "directions" in which the tour can be continued. If the DM does not want to continue, the incumbent trial outcome and the corresponding alternative (or alternatives) become the "most preferred".

In the weight method class the set of efficient outcomes is "mirrored" by the set of weights and DM's preferences are captured by a mechanism of weights.

In the reference point method class DM's preferences are captured by indicating reference points (elements of the outcome space $\mathcal{R}^k$).

In the constraint method class DM's preferences are captured by adding, dropping, tightening, or relaxing constraints on outcome components.

These three classes of methods are not necessarily disjoint. Some methods combine manipulation of constraints with manipulations of reference points.

In this book we do not consider MCDM interactive methods where nonefficient outcomes are derived and evaluated before efficient outcomes are found. The critic of those methods is that the DM wastes to much time evaluating outcomes which are far from the Pareto set.

Below, to level the ground for Chapter 5, we recall briefly the most distinctive features of the weight method class, the reference point method class, and the constraint method class.

## 3. Weight Method Class

In the weight method class DM's preferences are captured by a mechanism of weights (i.e. parameters in optimization problems (3.1), (3.2), (3.3), or (3.4)). A selected vector of weights yields (relates to) a properly efficient or weakly efficient outcome (cf. Figure 3.1, Figure 3.2, or Figure 3.3). The set of admissible weights is systematically searched and reduced, and this amounts to search and reduction of the set of efficient outcomes. Search can be organized in the form of *weight cuts* or *weight zooming*. Reductions of the set of admissible weights constitute a DM preference tracking mechanism and provide for a natural convergence measure and a stopping rule. In fact, convergence can be measured by a volume reduction ratio of the set of admissible weights. Search can be terminated (however, the DM may wish to stop at any moment) if the set of admissible weights is so small that differences between outcomes corresponding to weights from this set become insignificant. Other usual stopping rules such as the limit of elapsed time or the limit of interactions also apply but they are purely technical.

Because of the above properties we attribute a high value to the weight set reduction mechanism. It is worth observing that the weight based convergence measure and the related weight based stopping rule work under no specific assumption about underlying formal models of decision problems.

### 3.1 Weight Cut Methods

The most prominent methods representing this class are the Zionts-Wallenius method and the Dell-Karwan method.

The Zionts-Wallenius method applies to so called linear MCDM problems where sets $X_0$ are defined by linear equalities or inequalities and all criteria functions are linear. In this case sets $Z$ are polyhedral. It is assumed that for each problem considered there exists an *implicit* value function which represents the DM's utility over elements $y \in \mathcal{R}^k$. The DM's implicit value function is then approximated by an implicit linear function

$$\lambda y,$$

where $\lambda$ is an unknown vector of weights with elements $\lambda_i > 0,\ i = 1, ..., k$.

At each iteration of the method a (properly) efficient outcome $y$ is derived by solving problem (3.4), i.e.

$$\max_{y \in Z} \bar{\lambda} y,$$

for some $\bar{\lambda} \in \Lambda$, where $\Lambda$ is the set of admissible vectors of weights. At the beginning $\Lambda = \{\lambda \in \mathcal{R}^k \mid \lambda_i > 0\,,\; i = 1, ..., k\}$. Next, a number of efficient outcomes $y^r, r = 1, ...,$ is derived in the same manner. Pairs of outcomes $(y, y^r)$ are compared by the DM to reveal pairwise preferences. These are used then to shrink the set of admissible vectors of weights $\Lambda$. $\Lambda$ is shrunk by adding weight cuts. At each iteration $\Lambda$ is amended by a constraint formulated in terms of vectors of weights $\lambda$, namely

$$\lambda y > \lambda y^r, \tag{4.1}$$

if the DM prefers $y$ to $y^r$, or

$$\lambda y < \lambda y^r, \tag{4.2}$$

if the DM prefers $y^r$ to $y$. In other words, preferences with respect to $y$ and $y^r$ are expressed by the implicit value function. Any of the above cuts represents the following principle of decisional consistency: all further selections of $\lambda$ to derive outcomes for pairwise comparisons should be consistent with the already revealed DM's (partial) preferences.

When two outcomes $y,\ y^r$ being compared are not significantly different to provide for a clear preference, the DM evaluates the "attractiveness" of the direction $t(y^r - y), t > 0$. If this direction is attractive, it is assumed that the DM gives preference to $y^r$ over $y$, and if otherwise, he gives preference to $y$ over $y^r$. Cut (4.1) or (4.2) is made accordingly.

Evaluation of directions is a distinctive feature of the Zionts-Wallenius method present only in a few other MCDM methods. It should be stressed, however, that evaluation of directions plays in this method only a secondary role, just a technical trick to resolve the problem of possible perceived outcome indifference. In Chapter 6 we bring this form of preference expressing to a level of the principal (but not unique) decision tool.

♣ A set is polyhedral if it can be represented as an intersection of a finite number of halfspaces $a^i y \leq b^i\,,\; i = 1, ..., m\,.$

Polyhedrality of a set entails its convexity. Hence, if $Z$ is polyhedral then, in virtue of Characterization IV (Section 2.1 of Chapter 3), all properly efficient outcomes can be derived by solving optimization problem (3.4). However, the Zionts-Wallenius method makes use of the simplex method and therefore all outcomes derived by this method are vertices of $Z$.

In the Zionts-Wallenius method outcomes $y^r,\ r = 1, ...,$ are efficient vertices *adjacent* to efficient outcome $y$. Two vertices of a set are adjacent if, loosely speaking, they are at two different ends of an edge of this set. There is no need to solve optimization problems from scratch to identify adjacent outcomes $y^r$. They can be found by some simple operations which take advantage of calculations already made to derive $y$ by the simplex method.

The Zionts-Wallenius method offers some convergence properties provided the implicit value function is pseudoconvex. A function $f : \mathcal{R}^k \rightarrow \mathcal{R}$ is pseudoconvex, if it is differentiable at every $x \in \mathcal{R}^k$ and for all $y^1, y^2 \in \mathcal{R}^k$ such that $\nabla f(y^1)(y^2 - y^1) \geq 0$ we have $f(y^2) \geq f(y^1)$.
♠

The Zionts-Wallenius method works properly under the assumption that functions $\lambda y$ (hyperplanes, DMs'implicit value functions) correctly separate preferred outcomes from not preferred ones.

Another approximation of the DM's implicit value function is exploited in the Dell-Karwan method.

In the Dell-Karwan method, which makes no assumption about a specific problem (2.1) formulation, the DM's implicit value function is approximated by an implicit function

$$\max_i \lambda_i(y_i^* - y_i),$$

where $\lambda$ is an unknown vector of weights with elements $\lambda_i > 0,\ i = 1, ..., k$, and $y^*$ is the element defined in Subsection 2.1 of Chapter 3. Similarly to the Zionts-Wallenius method, at each iteration of the Dell-Karwan method pairwise comparisons of $y$ and $y^r$, $r = 1, ...,$ result in weight cuts, namely

$$\max_i \lambda_i(y_i^* - y_i) < \max_i \lambda_i(y_i^* - y_i^r), \tag{4.3}$$

if the DM prefers $y$ to $y^r$, or

$$\max_i \lambda_i(y_i^* - y_i) > \max_i \lambda_i(y_i^* - y_i^r), \tag{4.4}$$

if the DM prefers $y^r$ to $y$. The cuts are used to shrink the set of admissible vectors of weights $\Lambda$. At the beginning $\Lambda = \{\lambda \in \mathcal{R}^k \mid \lambda_i > 0\,,\; i = 1, ..., k\}$.

♣ An extension of the Zionts-Wallenius method and the Dell-Karwan method arises when to approximate the implicit value function one uses an implicit function

$$\max_i \lambda_i((y_i^* - y_i) + \rho e^k(y^* - y)), \tag{4.5}$$

as in optimization problem (3.1),
or

$$\max_i \lambda_i(y_i^* - y_i) + \rho e^k(y^* - y), \tag{4.6}$$

as in optimization problem (3.2), where $\lambda$ is an unknown vector of weights, $\lambda = \{\lambda_1, ..., \lambda_k\},\; \lambda_i > 0\,,\; i = 1, ..., k$.

In Section 5 of Chapter 6 we use functions (4.5) to derive weight cuts when solving a numerical example.
♠

♣ In the Dell-Karwan method no assumption is made on the form of sets $Z$. By virtue of Characterization III (section 1 of Chapter 3) and relations (2.3), no efficient outcome is a priori excluded from being determined by this method. However, when implementing the method care should be taken of outcomes which are weakly efficient but not efficient. This deficiency of the method is eliminated when functions (4.5) or (4.6) are used.

In contrast to the Zionts-Wallenius method, in the Dell-Karwan method, even if $Z$ is polyhedral and the simplex method is used to solve optimization problems, solutions are not confined to vertices of $Z$. Thus, the latter method is more universal than the former. But the latter method is technically more involved. In the Zionts-Wallenius method each pairwise outcome evaluation results in exactly one cut, but in the Dell-Karwan method each pairwise outcome evaluation results, in general, in a series of disjoint cuts. As the result the set of admissible vectors of weights becomes disconnected and therefore difficult to handle. Also, in the Dell-Karwan method to derive each outcome a separate optimization problem has to be solved from scratch.
♠

## 3.2 Tchebycheff Method

The so called Tchebycheff method exploits problem (3.3) to derive (weakly) efficient outcomes. The method consists of the following operations: selecting a number of vectors of weights $\lambda \in \Lambda = \{\lambda \in \mathcal{R}^k \mid \lambda_i > 0,\ i = 1, ..., k\}$, and then, at each iteration:

- solving problem (3.3) for all selected vectors $\lambda$ to derive a number of efficient outcomes,
- selecting by the DM from the outcomes derived the preferred outcome $\tilde{y}$ ,
- selecting a number of vectors $\lambda$ in a neighborhood of vector $\tilde{\lambda}$ corresponding to the preferred outcome $\tilde{y}$.

The above process has an effect of weight "zooming in", i.e. from iteration to iteration a smaller part of the set of admissible vectors of weights $\Lambda$ is investigated.

Similarly as for the Dell-Karwan method, when implementing the Tchebycheff method care should be taken of outcomes which are weakly efficient but not efficient.

## 4. Reference Point Method Class

In the reference point method class the DM expresses preferences by selecting a reference point (an element of the outcome space $\mathcal{R}^k$). A reference point can be an outcome, i.e. an element of $Z$ , or any other element $\mathcal{R}^k$. Then an efficient outcome which "corresponds" to the reference point selected and the achievement function used (cf. Subsection 2.2 of Chapter 3) is derived by solving problem (3.5). The notion of correspondence is intuitively explained in Figure 4.1 (the thin continuous line represents an isogram of achievement function (3.6) or (3.7)). By manipulating reference points the DM is able to derive a subset of the Pareto set and from this subset he selects the most preferred outcome. We call this version of reference point methods the *standard scheme*.

A variant of the standard scheme, the *extended scheme*, requires the DM to select a pair of reference elements of $\mathcal{R}^k$, namely point $y^{res}$, called *reservation point*, and point $y^{asp}$ (such that $y^{asp} \in y^{res} + R^k_+$), called *aspiration point*. It is quite natural (but not necessary) to assume that $y^{res} \in Z$ and $y^{asp} \notin Z$. Given a reservation point and an aspiration point it is possible to construct an achievement function (e.g. achievement

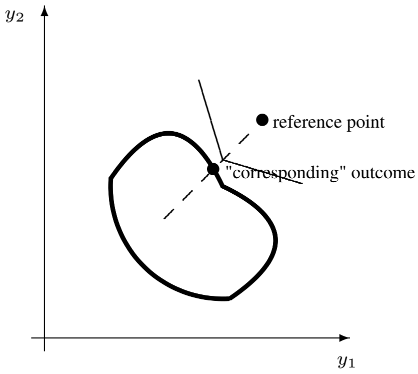


*Figure 4.1.* A reference point and its corresponding (properly) efficient outcome.

function (3.6) or achievement function (3.7), cf. Chapter 5, Section 2) such that an outcome $y$ which maximizes that function over $Z$ is a (weakly or properly) efficient outcome most distant, in a sense, from the reservation point and at the same time closest to the aspiration point. This is schematically illustrated in Figure 4.2 (the thin continuous line represents an isogram of achievement function (3.6) or (3.7)).

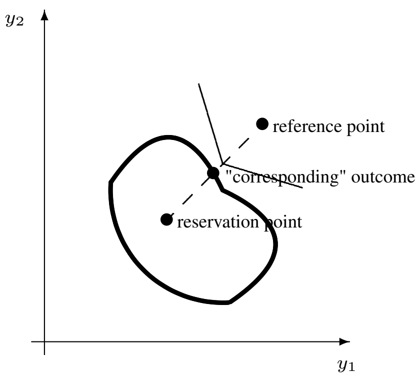


*Figure 4.2.* An aspiration point, a reservation point, and their corresponding (properly) efficient outcome.

## 5. Constraint Method Class

For this class of methods the most representative is perhaps the STEM method. This method requires that at each interaction the DM selects outcome components (criteria) $i,\ i \in I^{>},\ I^{>} \subseteq I,\ I = \{1, ..., k\}$, values of which he wants to be increased at the expense of values of components $i$ , $i \in I^{<},\ I^{<} \subseteq I$, where $I^{>} \cap I^{<} = \emptyset$.

If weakly efficient outcome $\bar{y}$ is derived at an iteration of this method, at the next interaction another weakly efficient outcome is derived by solving the following optimization problem (a variant of problem (3.3)):

$$\min_{y \in Z} \max_{i} \lambda_i (y_i^* - y_i)$$

$$y_i \geq \bar{y}_i \quad \text{for} \quad i \in I^{>} \,,$$

$$y_i \geq \bar{y}_i - b_i \quad \text{for} \quad i \in I^{<} \,,$$

where $\lambda_i > 0$ for $i \in I^{>}$, and $\lambda_i = 0$ for $i \in I^{<}$ , and $b_i$ specifies an admissible decrease of the value of the component $i$ , $i \in I^{<}$ .

## 6. Concluding Remarks

♣ For technical convenience, in numerical implementations of all methods presented above set $\Lambda$ is usually transformed to the following form $\bar{\Lambda} = \{\lambda \in \mathcal{R}^k \mid \lambda_i > 0\,,\ i = 1, ..., k,\ \sum_{i=1}^{k} \lambda_i = 1\}$. This is achieved by dividing each component of $\lambda$ by the sum of all components of that $\lambda$, for all $\lambda \in \Lambda$. There is an obvious one-to-one correspondence between $\Lambda$ and $\bar{\Lambda}$.
♠

It is astonishing but fortunately true that the multitude of interactive MCDM methods fall just to one of three classes which correspond to three types of characterizations presented in Chapter 3. This not only allows simple presenting and smooth "marketing" the field of MCDM to potential users, but has some methodological and technical consequences for the way interactive MCDM methods can be implemented and applied in practice. In particular, the general outline of interactive MCDM methods given in this chapter is sufficient for presenting in Chapter 6 the Generic Decision Supporting Scheme.

Interactive MCDM methods are "soft" in the sense that they deny rigorous formal convergence considerations. The rationale behind this type of methods is an assumption, strongly supported by evidence from practical applications (real, not academic), that there is no way to encapsulate DM preferences into a formal, consistent, and verifiable framework. If so, the convergence issues are to be left to experimental investigations.

## 7. Annotated References

♣ Among methods which proceed by deriving sequences of nonefficient outcomes until an efficient outcome is found one can mention the method of Geoffrion, Dyer, and Feinberg (Geoffrion et al. 1972), Michalowski (Michalowski 1988), Michalowski and Szapiro (Michalowski and Szapiro 1989,1992). As already said, methods of this kind are not considered in this book.

♠

For a more detailed taxonomy of MCDM methods, not exclusively interactive, the reader is referred to the book of Miettinen (Miettinen 1999).

♣ Following the listing of MCDM methods provided in Miettinen (1999) the following assignment of methods to the three distinguished classes seems to be the most appropriate:

- Zionts-Wallenius method (Zionts, Wallenius 1976, 1983), Tchebycheff method (Steuer, Choo 1983, Steuer 1986), Dell-Karwan method (Dell, Karwan 1990),

are members of the weight method class;

- the reference point method (Wierzbicki 1980, 1986, 1990, 1999), STOM method (Nakayama, Sawaragi 1984, Nakayama, Furukawa 1985, Nakayama et al. 1986, Nakayama 1989, 1995), GUESS method (Buchanan 1997), Light Beam Search method (Jaszkiewicz, Słowiński 1994, 1995), Reference Direction Approach method (Korhonen, Laakso 1984, 1985, 1986, Krohonen 1988), Reference Direction method (Narula et al. 1994a, b),

are members of the reference point method class;

- STEM method (Benayoun et al. 1971), ISWT method (Chankong, Haimes 1978, 1983), SPOT method (Sakawa 1982), NIMBUS method (Miettinen, Mäkelä 1995, 1997), are the members of the constraint method class.

It is worth observing that on the methodological level another taxonomy of MCDM methods can be proposed. Namely, methods can be classified with respect the manner in which the DM expresses his partial preferences. He can express his preferences either in the *atomistic* manner in one of two forms: preferred values of components (objective functions) or mutual relations of those (trade-offs), or the *holistic* manner in the form of a pattern (reference) outcome he would like to mimic as closely as possible.

According to the latter taxonomy:

- STEM method, Zionts-Wallenius method, ISWT method, SPOT method, STOM method, Dell-Karwan method, NIMBUS method,

are members of the class where preferences are expressed in the atomistic manner;

- the reference point method, Tchebycheff method, GUESS method, Light Beam Search method, Reference Direction Approach method, Reference Direction method,

are members of the class where preferences are expressed in the holistic manner.

On a detailed level these taxonomies need not to be disjoint. For example, the NIMBUS method (Miettinen, Mäkelä 1995, 1997), a member of the constraint method class, combines some elements of reference point manipulations.
♠

A detailed description of the Zionts-Wallenius method can be found in Zionts, Wallenius (1976,1983) and that of Dell-Karwan in Dell, Karwan (1990). For the definition of a value (or utility) function cf e.g. Edwards (1992), Fishburn (1986). Roy and Wallenius (Roy, Wallenius (1991)) and also Kaliszewski and Zionts (Kaliszewski, Zionts 2004) propose generalizations of the Zionts-Wallenius method to nonlinear problems.

The Tchebycheff methods is proposed in Steuer, Choo (1983) and Steuer (1986).

The reference point method, related to the concept of displaced ideal by Zeleny (Zeleny 1976, 1982)), comes from Wierzbicki and is outlined in a series of papers (Wierzbicki 1980, 1986, 1990, 1999). Applications of this method are reported e.g. in Gal et al. (1999).

The constraint method class contains some of the oldest of MCDM methods, for example the STEM method (Benayoun et al. 1971), which gained in 1970 much popularity. Other readings on methods of this class, to mention just two, are Armann (1989), and Haimes et al. (1971).

# Chapter 5

# UNIVERSAL MCDM INTERFACE

*"... there is so little difference, and the difference means so little."*

**Joseph Conrad**,
*Lord Jim.*

## 1. This Chapter Is About ...

... how any interactive MCDM method from three classes distinguished in Chapter 4 can be framed into a common, *universal interface* (a universal interfacing standard). Taking into account diversity of MCDM methods, the idea is potentially very attractive. Construction of such an interface is founded on the following observations:

- every interactive decision making process can be tiered to the methodological tier and the technical tier,
- the methodological tier comprises evaluations of outcomes (derived by manipulations of weights, reference points, or constraints) and preference expressing,
- all other aspects of interactive decision making are of secondary importance to the course of decision making processes, and as such can be designated "technical"; those aspects form a separate, technical tier; the DM should be concerned with them only at his explicit request but otherwise they should be kept hidden in the background.

In Section 2 of this chapter we show how in the MCDM method classes, outlined in Chapter 4, the methodological tier can be separated from the technical tier. Once those tiers are separated, it becomes apparent that the methodological functionalities of each class of interactive MCDM methods can be realized via one universal interface. In Section 3 we go one step further and we show a way to relax the tight relationship between interactive MCDM methods for solving complex MCDM problems on one side, and optimization methods on the other. We elaborate on that concept further in subsequent chapters, in particular in Chapter 7 and Chapter 8, with a climax in Chapter 9.

## 2. Towards Universal MCDM Interface

We start with the observation that in practice an implementation of any interactive MCDM method should allow for adding, dropping, tightening, or relaxing constraints to the model (2.2), as appropriate. Not only decision making processes are dynamic (the DM can change his preferences in the course of the interactive decision making process) but also modeling of decision making problems can seldom be accomplished in the form of a one-step procedure. One can recall here the four phases of MCDM processes mentioned in Introduction. Therefore, the ability to manipulate constraints should be made a part of general functionality of any MCDM implementation. If so, we should no longer consider constraint manipulation as a specific element of MCDM processes. In consequence, our trichotomous classification of MCDM methods, as proposed in Chapter 4, folds into a dichotomic one. From now on we distinguish only the weight method class and the reference point method class, assuming that the constraint manipulation mechanism is (or at least it should be) implemented irrespective of whether this is requested by the implemented method itself, or just to ensure modeling flexibility.

Below we discuss similarities and dissimilarities of the weight method class and the reference point method class with respect to methodology and technicalities. We attempt to identify as much common elements of these two classes as possible to eventually encapsulate them into a universal MCDM interface.

The idea of a universal interface is built around the assumption that in the weight method class outcomes are derived by solving optimization problem (3.1), or optimization problem (3.2), or optimization problem

(3.3) (which is a special case of optimization problem (3.1) and optimization problem (3.2)).

## 2.1 Weight Methods Versus Reference Point Methods

### 2.1.1 Weight Versus Reference Point Methods - Methodological Paradigms

As observed in Chapter 4, on the methodological level the weight method class and the reference point method class represent two entirely different decision making paradigms. In the weight method class preferences are expressed in the form of designated weights selected from the set of admissible weights. In the reference point method class preferences are expressed in the form of designated elements selected from the outcome space.

### 2.1.2 Weight Versus Reference Point Methods - Technical Requirements

In the weight method class selecting at each iteration a vector $\lambda$ amounts in fact to selecting a half line. This half line starts from $y^*$ and contains the apexes of isograms of the objective function of optimization problem (3.1) or optimization problem (3.2) (or optimization problem (3.3) if $\rho = 0$) (cf. Figure 3.1 and Figure 3.2). Hence, the half line has the form

$$\{y^{apex} \mid y^{apex} = y^* - t\tau,\, t \geq 0\},$$

for some direction (vector) $\tau = (\tau_1, ..., \tau_k)$. In course of an interactive decision making process one gets a "fan" of half lines, all starting at $y^*$ (Figure 5.1).

Also in the standard scheme of the reference point method class selecting at each iteration reference point $y^{ref}$ amounts in fact to selecting a half line. This half line starts from $y^{ref}$ and contains the apexes of isograms (recall that, following the best practice, we assume that the achievement function has the form (3.6) or (3.7)) of the objective function of optimization problem (3.1) or optimization problem (3.2) (or optimization problem (3.3) if $\rho = 0$) (cf. Figure 3.1 and Figure 3.2). Hence, the half line has the form

$$\{y^{apex} \mid y^{apex} = y^{ref} - t\tau,\, t \geq 0\},$$

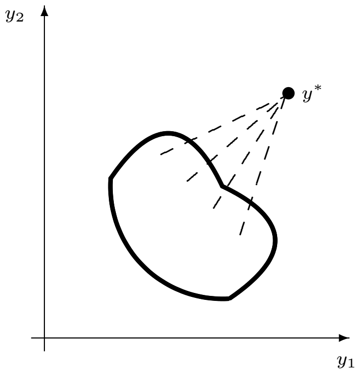


*Figure 5.1.* A fan-type decision making process.

where $t \leq 0$, if $y^{ref} \in Z$, $t \geq 0$, if $y^{ref} \notin Z$, for some direction (vector) $\tau = (\tau_1, ..., \tau_k)$. In the course of an interactive decision making process one gets a "forest" of half lines (Figure 5.2), each starting at some $y^{ref}$.

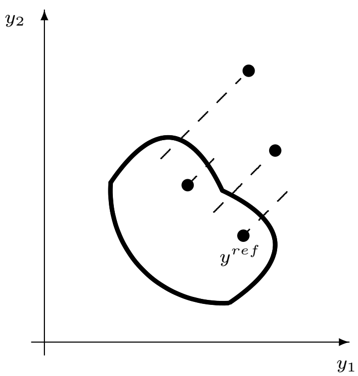


*Figure 5.2.* A forest-type decision making process.

In the extended scheme of the reference point method class the DM selects at each iteration reservation point $y^{res}$ and aspiration point $y^{asp}$, $y^{asp} \in y^{res} + R^k_+$, which amounts in fact to selecting a half line. This half line starts from $y^{asp}$ and contains the apexes of isograms of (recall that we assume that the achievement function has the form (3.6) or (3.7)) of the objective function of optimization problem (3.1) or optimization problem (3.2) (or optimization problem (3.3) if $\rho = 0$) (cf. Figure 3.1 and Figure 3.2). Hence, the half line has the form

$$\{y^{apex} \,|\, y^{apex} = y^{asp} - t\tau,\, t \geq 0\,\},$$

where $t \leq 0$ if $y^{asp} \in Z$, and $t \leq 0$ if $y^{asp} \notin Z$, for some direction (vector) $\tau = (\tau_1, ..., \tau_k)$. In the course of an interactive decision making process one gets a "maze" of half-lines (Figure 5.3), each starting at some $y^{asp}$.

♣ In the weight method class direction $\tau$ which defines the half line

$$\{y^{apex} = y^{apex} - t\tau,\; t \geq 0\},$$

containing the apexes of isograms of the objective function of optimization problem (3.1) or optimization problem (3.2), can be found as follows.

Consider first optimization problem (3.1). Given vector of weights $\lambda,\ \lambda_i > 0,\ i = 1, ..., k$, and parameter $\rho > 0$, apex $y^{apex}$ of an isogram of function

$$\max_i \lambda_i((y_i^* - y_i) + \rho e^k(y^* - y))$$

satisfies

$$t = \lambda_1((y_1^* - y_1^{apex}) + \rho e^k(y^* - y^{apex}))$$

$$\dots$$

$$= \lambda_i((y_i^* - y_i^{apex}) + \rho e^k(y^* - y^{apex}))$$

$$\dots$$

$$= \lambda_k((y_k^* - y_k^{apex}) + \rho e^k(y^* - y^{apex})),$$

where $t$ is a constant.

Without loss of generality we can assume $t = 1$. To find the corresponding element $y^{apex^1}$ we have to solve the set of equations

$$\lambda_i((y_i^* - y_i^{apex^1}) + \rho e^k(y^* - y^{apex^1})) = 1,\ i = 1, ..., k. \qquad (5.1)$$

With $y^{apex^1}$ found and with $t = 1$ we have

$$y^{apex^1} = y^* - \tau,$$

hence

$$\tau = y^* - y^{apex^1}. \tag{5.2}$$

Observe that for $\rho = 0$,

$$y_i^* - y_i^{apex^1} = \frac{1}{\lambda_i},\ i = 1, ..., k,$$

hence

$$\tau_i = \frac{1}{\lambda_i},\ i = 1, ..., k.$$

On the other hand, any $\bar{y} \in Z$ is the apex of the isogram of

$$\max_i \bar{\lambda}_i((y_i^* - y_i) + \rho e^k(y^* - y)) = 1,$$

where

$$\bar{\lambda}_i = ((y_i^* - \bar{y}_i) + \rho e^k(y^* - \bar{y}))^{-1},\ i = 1, ..., k.$$

Indeed, in this case $\bar{y}$ clearly solves (5.1). By (5.2)

$$\tau = y^* - \bar{y}. \tag{5.3}$$

It is also worth observing that $\tau$ can have nonpositive components, i.e. the condition $\tau_i > 0,\ i = 1, ..., k$, is not in general valid, as shown by the example below. In other words, the relation $y^{apex} \in y^* - R_+^k$ is not in general valid.

EXAMPLE 1 Let $y^* = (-0.050,\ 1.300)$, $\rho = 0.660$, $\lambda = (0.800,\ 0.200)$. For $t = 1$ we get $y^{apex} = (0.546,\ -2.092)$, thus $\tau = (-0.596,\ 3.392)$. □

Consider now the objective function of optimization problem (3.2). Given vector of weights $\lambda$, $\lambda_i > 0$, $i = 1, ..., k$, and parameter $\rho > 0$, apex $y^{apex}$ of an isogram of the function

$$\max_i \lambda_i(y_i^* - y_i) + \rho e^k(y^* - y)$$

satisfies

$$\tilde{t} = \lambda_1(y_1^* - y_1^{apex}) + \rho e^k(y^* - y^{apex})$$

$$...$$

$$= \lambda_i(y_i^* - y_i^{apex}) + \rho e^k(y^* - y^{apex})$$

$$...$$

$$= \lambda_k(y_k^* - y_k^{apex}) + \rho e^k(y^* - y^{apex}),$$

where $\tilde{t}$ is a constant, or equivalently

$$t = \lambda_1(y_1^* - y_1^{apex}) = ... = \lambda_k(y_k^* - y_k^{apex}),$$

where $t$ - a constant.

Without loss of generality we can assume $t = 1$. Hence

$$y_i^{apex^1} = y_i^* - \frac{1}{\lambda_i} = y_i^* - \tau_i, \ i = 1, ..., k, \tag{5.4}$$

where

$$\tau_i = \lambda_i^{-1}, \ i = 1, ..., k. \tag{5.5}$$

Formula (5.5) clearly defines $\tau$ also for optimization problem (3.3).

It is worth observing that in the case of optimization problem (3.2) and optimization problem (3.3), $\tau$ does not depend on $\rho$, and the condition $\tau > 0$ holds. In other words, the relation $y^{apex} \in y^* - R_+^k$ is valid.

In the standard scheme of the reference point method class directions $\tau$ can be found in the same manner as in the case of the weight method class.

In the extended scheme of the reference point method class directions $\tau$ are defined simply by $\tau_l = |y_l^{res} - y_l^{asp}|, \ l = 1, ..., k$.

♠

The *decision controls* of the three different decision processes types are summarized in Table 5.1.

From Table 5.1 we see that though the weight method class and the reference point method class represent different decision making paradigms, they are technically very similar. Indeed, in each method of these

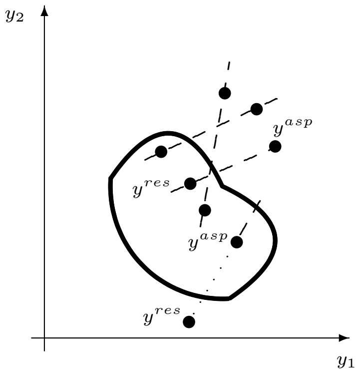


*Figure 5.3.* A maze-type decision making process.

two classes, to proceed to the next iteration, i.e. to derive an efficient outcome for evaluation, two decision controls are required to be specified: either two elements of $\mathcal{R}^k$ (as in the extended scheme of the reference point method class), or a direction in $\mathcal{R}^k$ and an element of $\mathcal{R}^k$ (as in the weight method class and in the standard scheme of the reference point method class). In technical terms, this reduces to specifying a direction and an element in either case, because two elements of a space uniquely define a direction in this space.

The method classes differ by presence or absence of fixed decision controls, and by elements selected to serve as decision controls. Methods in the reference point method class can be regarded as more flexible than methods in the weight method class since in the former no decision control is a priori fixed. In practice however, as opposed to the theory, too much flexibility is not always an asset.

Taking one step further, we can make the weight method class and the extended scheme of the reference point method class technically equivalent under the following restriction of the latter. Assume that $y^{asp} = y^*$, i.e. aspiration points are fixed in all iterations, and the DM can change only reservation points. It is easy to observe that such a restriction does not constrain possible selections of directions $\tau$, and therefore does not constrain possible selections of efficient outcomes. If the above assumption is justified, then with $y^{asp} = y^*$ methods of either

*Table 5.1.* Decision controls.

| | | decision controls | |
|---|---|---|---|
| | | fixed | to be selected |
| weight method class | | $y^*$ | $\tau$ |
| reference point method class | *standard scheme* | - | $\tau,\ y^{ref}$ |
| reference point method class | *extended scheme* | - | $y^{res},\ y^{asp}$ |

class realize fan-type decision making processes (Figure 5.1). Such a justification can probably be provided only on the base of accumulated practical experience, and such experience seems to be still lacking.

Reducing MCDM processes to fan-type has a methodological advantage. Namely, in a fan-type process the DM expresses in fact his preferences in terms of proportions of concessions necessary to depart from utopian element $\hat{y}$ (in fact, from element $y^*$, but this distinction is purely technical) to attain an efficient outcome. The notion of *direction of concessions* naturally arises here. Since the DM, as a rule, attributes more importance to potential losses than to potential gains, fan-type processes may be an appropriate framework to express his preferences more carefully. Moreover, as shown by formula (5.1) and formula (5.4), there is one-to-one correspondence between vectors $\tau$ and vectors $\lambda$, so the preference capturing mechanism in the weight method class can be easily translated to a preference capturing mechanism using the notion of direction of concessions.

## 3. Universal Interface

We call the Universal Interface any MCDM interface which allows the DM to manipulate all the decision controls listed in Table 5.1.

A simple implementation of the Universal Interface can have the form of an application with the screen representing Table 5.1 and active fields in this screen for the DM to input values of selected decision controls.

At each iteration, to process decision controls and possibly some additional parameters specified by the DM, the Universal Interface has to communicate with an optimization engine capable of solving instances of problem (2.1) ((2.2)) (cf. Figure 5.4). The optimization engine should handle the original problem data and at each iteration perform all the nec-

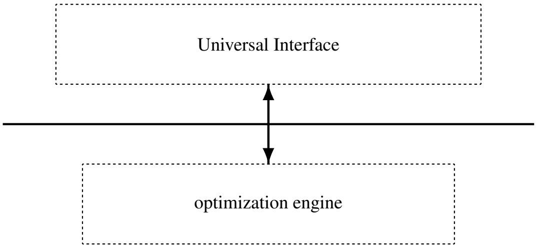


*Figure 5.4.* Universal Interface and an optimization engine.

essary computing. The DM should be relieved of the task of optimization engine selection, as this can be done by a specialist. Potentially, there can be many optimization engines offered by different software providers or vendors.

Data exchange between the Universal Interface and optimization engines should be organized via a standardized message format and message content. At present stand-alone optimization engines for interactive MCDM with standardized message formats and message contents are not available. Up to now MCDM method implementations are, as a rule, all-inclusive and contain proprietary built-in optimization engines. But with a standard for message content established there should be a substantial incentive for software providers to fill this gap. Such a standard is established in the previous section in the form of Table 5.1.

## 4. Concluding Remarks

In this chapter the most distinctive features of interactive MCDM methods are abstracted and organized into one common framework - the Universal Interface. The Universal Interface does not impose any restriction on the DM in manipulating weights or reference points to express his partial preferences as he searches for the most preferred outcome. The Universal Interface just separates the decision controls from the technical aspects of decision making processes, the latter now placed in the secondary, technical tier.

There are clear advantages of viewing interactive MCDM methods from the perspective of the Universal Interface. With the Universal Interface in place presenting interactive MCDM principles and merits becomes a much simpler task than it was before. Moreover, interactive MCDM issues can be addressed separately by two groups: in the methodological tier by actual DMs and decision theory researchers, and in the technical tier by research and technical persons from areas of decision theory, computer science, mathematics, software developing, and computing service providing.

## 5. Annotated References

A detailed discussion about a general framework and stages of decision making processes can be found, for example, in Simon (1977).

Here we follow steps taken in Gardiner, Steuer (1994a,b), where a rudimentary, unified interactive decision making algorithm is shown to be a common framework for a wide selection of MCDM methods. Though this work is influential on the MCDM research community in that it tracks down common elements in seemingly diverse methods, its impact on practical applicability of MCDM methods seems to be limited. This should be attributed to the fact that the perspective of a unified MCDM framework provided by Gardiner and Steuer, appealing to MCDM research community, remains still too intricate for lay DMs. In this chapter we pursuit the same direction as Gardiner and Steuer, but we convey the idea of unification and standardization in the field of interactive MCDM much further.

The idea of MCDM class prototyping is discussed in Kaliszewski (2004).

Asymmetries in DMs attitudes to potential losses and potential gains are investigated in Kahneman, Tversky (1979).

Chapter 6

# GENERIC INTERACTIVE MCDM SUPPORT SCHEME

*"The onlookers see most of the game."*

**A proverb**.

## 1. This Chapter Is About ...

... taking one more step. As shown in the previous chapter, any interactive MCDM method can be operated from the Universal Interface. In this chapter we frame two MCDM method classes distinguished in Chapter 5, namely the weight method class and the reference point method class, into a generic interactive MCDM support scheme. We illustrate the concept of the Universal Interface and the concept of generic support scheme by a numerical example.

We restrict ourselves to methodological issues of the proposed scheme, leaving the discussion of its versatility and its practical use to Chapter 9.

## 2. Generic Interactive MCDM Support Scheme

Below, in Section 3, we propose a generic interactive MCDM support scheme. For this aim we have to recall how DM's preferences can be expressed and then encapsulated to guide interactive decision making.

The algorithmic description of the scheme we provide is very simple and rudimentary.

## 2.1 Preference Expressing In MCDM

Repetitive "man-machine" interactions are the underlying methodological principle in searching for preferred outcomes in any interactive MCDM method. As a rule, the search is guided by the DM's preferences with respect to (preferred) values of outcome components (values of criteria). Preferences can be expressed explicitly, via constraints on outcome components (which in fact amounts to the temporary or permanent exclusion of subsets of efficient outcomes from being considered preferred) - any MCDM method should allow constraint manipulations (cf. Section 2 of Chapter 5), or implicitly, via weights or reference points (as in the weight method class or the reference point method class).

The majority of MCDM methods use explicit or implicit preference expressing with respect to absolute preferences, i.e. preferences with respect to values of outcome components. An exception to that general picture is the already mentioned Zionts-Wallenius method. This method makes use of absolute preferences and relative preferences, i.e. preferences with respect to relative changes in outcome component values (measured by the "attractiveness" of a direction, as explained in Chapter 4, Subsection 3.1,

Both absolute and relative preferences serve the ultimate goal, which is to arrive at the most preferred outcome showing the most satisfactory compromise of outcome component values.

### 2.1.1 Expressing Absolute Preferences

The primary form of expressing preferences in interactive MCDM methods is to indicate, explicitly (via constraints) or implicitly (via weights or reference points), desired changes of some, possibly all, components in the incumbent outcome. All interactive MCDM methods implement this form.

### 2.1.2 Expressing Relative Preferences

Global trade-offs carry information on potential relative changes in outcome components if a given efficient outcome is replaced by another outcome. Following the definition of global trade-off, this information pertains to the maximal possible effect of such changes.

Global trade-offs are clearly value carriers. High values of global trade-offs for a given outcome reveals the possibility of existence of other outcomes, which could be more desirable (one can "trade more for less"). By the same token, low values of global trade-offs for pairs of outcome components can be expected in the case of the most preferred outcome.

As no rule can be given what is to be regarded as "high" or "low" global trade-off values, relative preferences are (in the same way as absolute preferences) entirely dependent on the DM perception of the decision making problem.

Global trade-offs are general constructs applicable to any particular form of problem (2.2). Therefore global trade-offs seem to be a suitable tool to capture DM's relative preferences.

At this point we do not have any tool to ensure derivation of efficient outcomes with global trade-offs which match the DM relative preferences. As it is now, the DM can only accept global trade-offs of an outcome or give that outcome no preference. But tools of that kind are presented in Chapter 8.

♣ It is often conjectured that when expressing his partial preferences with respect to values of outcome components (objective functions) the DM is consistent with his implicit value function. This conjecture can be extended to encompass global trade-offs.

Filtering out outcomes with high values of global trade-offs can be interpreted as a partial and implicit specification of the DM's value function. Given a value function, there exists a number such that efficient outcomes with all global trade-offs greater than that number cannot be the most preferred. This justification comes from the generally accepted assumption that value functions are componentwise increasing functions.

To illustrate, let us consider the case as shown in Figure 6.1. Assume two efficient outcomes $y^1$ and $y^2$, and value function $v$ are given. Denote $\alpha^1$ the maximal global trade-off for $y^1$, $\alpha^2$ the maximal global trade-off for $y^2$. Suppose that $\alpha^2 > \alpha^1$ but $v(y^1) > v(y^2)$. Thus, setting global trade-off upper bound equal to $\alpha^2$ and eliminating all outcomes with global trade-offs above this bound shrinks the Pareto set but does not eliminate the most preferred outcome.

Certainly, with more than two dimensions of the outcome space such a simple interpretation is not available. However, still outcomes with

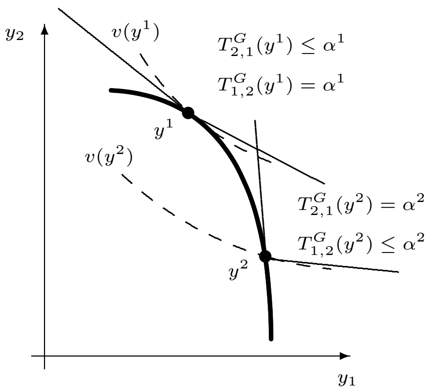


*Figure 6.1.* Global trade-offs and indifference curves of a value function.

excessively high global trade-offs can be, at the DM's discretion, eliminated from potential candidates for the most preferred outcome.
♠

### 2.1.3 Expressing Absolute And Relative Preferences

In MCDM literature it is common to assume that searching for the most preferred outcome is guided by absolute preferences. At the same time, research into behavioral aspects of MCDM demonstrates that people often simplify their choices with respect to values and emphasize relative changes as preference carriers.

♣ To frame both forms of preference expressing one can refer to the notion of *psychological stability*. An outcome $y$ is said to be *psychologically stable* if it has *satisfactory* values of components and has *acceptable* global trade-offs. This is a purely qualitative and descriptive definition giving no prescription what the satisfactory values of outcome components and outcome global trade-offs should be. Moreover, since those values depend on scaling of output components, DM's satisfaction is purely subjective.
♠

### 2.1.4 Balancing Between Absolute And Relative Preference Expressing

One possible and quite extreme scenario of preference expressing occurs when the DM specifies his preference with respect to values of outcome components but neglects global trade-offs. The majority of interactive MCDM methods follow this scenario. Another extreme scenario occurs when the DM expresses preferences only with respect to global trade-offs. In addition to these two extreme scenarios two mixed scenarios are obvious options:

- a subset of outcomes with preferred values of components is determined first, and next, to provide for so called *isolation effect*, the DM resorts to global trade-off (i.e. relative) preference expressing and evaluation,
- a subset of outcomes with preferred values of global trade-offs is determined first, and next, to provide for the isolation effect, outcomes from this subset with preferred values of components are identified.

In addition, if there is no reason for a priori exclusion of some outcomes, any combination of the above extreme scenarios of preference expressing should be available to the DM.

### 2.1.5 Firm And Non-firm Preferences

When expressing preferences in the form of weights, it is reasonable to ask the DM if he regards his preferences with respect to a pair of compared outcomes as *firm*. Firm preferences are those in which the DM is very confident and perceives them as stable, i.e. there is limited chance that he could change them at the later stages of the decision making process. Otherwise, preferences are designated as *non-firm*.

Weight cuts, as discussed in Chapter 4, should be formed only with respect to firm preferences.

## 3. Generic Interactive MCDM Support Scheme - $GIS^2$

Here we present a generic interactive MCDM support scheme. To ensure generality of this scheme we admit both forms of preference expressing, absolute and relative.

*Generic Interactive MCDM Support Scheme* – $GIS^2$ – is composed of the following five steps.

$GIS^2$

0. Derive a number of efficient outcomes and for each of them calculate global trade-offs. Select arbitrarily one outcome and denote it as *incumbent*. Denote the other outcomes as *candidates*.
1. Ask the DM to evaluate each candidate against the incumbent. For each evaluation ask the DM if his preference is firm or non-firm. Ask the DM to select the new incumbent.
2. Ask the DM if he wants to continue the search. If no, terminate with the incumbent as the most preferred outcome.
3. Ask the DM to select objects to manipulate: weights or reference points. If the DM selects weights to manipulate, for each evaluation of Step 1 resulting in a firm preference form a weight cut. Ask the DM to express his preferences using the objects selected.
4. Derive a number of efficient outcomes satisfying preferences expressed by the DM in Step 3, denote them as candidates, and for each candidate calculate global trade-offs. Go to Step 1.

$GIS^2$ encompasses the essence of interactive MCDM. Every existing MCDM method can be operationalized within this scheme.

♣ Below we discuss the steps of $GIS^2$ in more detail. To derive a candidate in Step 0 and in Step 4 any of the characterizations of properly efficient outcomes presented in Chapter 3 can be used. The choice of characterization depends on what objects the DM prefers to manipulate: weights (as in the characterizations by weight manipulations) or reference points (as in the characterizations by reference point manipulations). If the DM has no preference, the rule of thumb applies. For each characterization simple heuristic rules exist for selecting the very first outcome. For characterizations by weight manipulations a commonly accepted rule is to use weights which are all equal. For characterizations by reference point manipulations a commonly accepted rule is to use point $\hat{y}$ as a reference point.

Providing in Step 0 global trade-off values can be implemented in two ways. The first possibility is to provide the DM with efficient outcomes and let him to select pairs of indices $i, j = 1, ..., k,\ i \neq j$, for which global trade-offs are to be calculated. The second possibility is that for each efficient outcome global trade-offs for all pairs of indices are calculated beforehand.

In Step 1 the DM has to indicate a preferred outcome from a number of candidates and the incumbent. If the DM is unable to do this he should at least indicate one candidate outcome more preferred than the incumbent and such an outcome becomes the new incumbent. If the DM is not able to indicate even one candidate outcome more preferred than the incumbent, then the incumbent remains unchanged.

Step 2 contains the only stopping rule which is methodologically consistent with the interactive MCDM principle. The process stops as soon as the DM is satisfied with the current incumbent. The absence of any assumption about the structure of the DM preferences allows for no formal considerations of the decision process convergence. Only if the DM consistently adheres to expressing preferences in the form of weights $\lambda$, and subsequent sets of admissible weights reflect this consistency, convergence measures based on a volume reduction ratio of the set of admissible weights $\Lambda$ (as described in Chapter 4, Section 3) are viable.

There are two obvious technical indicators of the extension of the decision process, namely the number of interactions and elapsed time. Each of them can be used to set a technical stopping rule.

In Step 3 the DM expresses his implicit preferences in one of the forms:

- in the form of weights, by selecting (implicitly or explicitly) a vector from the set of admissible weights,

- in the form of a reference point, which next has to be "mimicked" as closely as possible by an efficient outcome to be derived in Step 4.

He can also express his explicit preferences in the form of constraints on outcome components.

Step 4 replicates actions of Step 0, but in Step 4 the derivation of new candidates is guided by the DM preferences expressed in Step 3.

♠

In Section 5 of this chapter we provide a simple operationalization of $GIS^2$ and we use it in a decision process structured along $GIS^2$ lines to solve a practical example introduced in the next section .

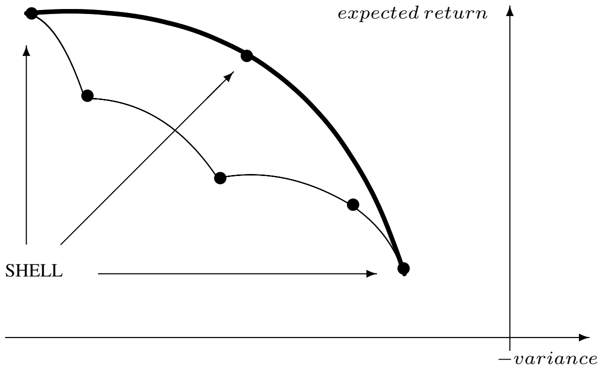


*Figure 6.2.* A shell of the portfolio selection problem.

## 4. Practical Example

A spectacular application of the proposed approach is the famous Markowitz mean-variance portfolio selection problem, a widely publicized example of optimization applications in finance. In Section 5 of this chapter we solve an instance of this problem by a combination of weight and reference point manipulations guided by DM absolute and relative preferences, as specified by $GIS^2$.

In the Markowitz problem the most preferred portfolio is selected from available stocks to maximize portfolio expected return (a measure of gain) and minimize portfolio variance (a measure of risk). The problem comprises one linear and one quadratic criterion over a continuum of possible combinations of available stocks. In this case the Pareto set has the form as indicated in Figure 6.2 by the thick line (to maintain consistently throughout this book all criteria in the form "the more the better", the negative of variance is maximized).

Selected stocks are to consume all the available capital, usually normalized to one, so stock participation in the portfolio is represented by fractions which sum to one. We consider the case where borrowing (short sale) for some stock is allowed (so the participation of an individual stock in the portfolio can be negative). Here the original problem is modified by the requirement that stocks which cannot be sold short are to participate in the portfolio up to or above certain specified threshold levels.

The formulation of the problem is as follows.

| | |
|---|---|
| $\max -\sum_{i=1}^{n}\sum_{j=1}^{n}\beta_{ij}x_i x_j$ | (maximize negative of portfolio variance) |
| $\max \sum_{i=1}^{n} e_i x_i$ | (maximize portfolio expected return) |
| $\sum_{i=1}^{n} x_i = 1,$ | ("all the capital" constraint) |
| $m_j\delta_j \leq x_j \leq M\delta_j\ ,$ | ("no small trades" constraints for all $j$ |
| $\delta_j = 0$ or $1,$ | such that stock $j$ cannot be sold short) |

where $\beta_{ij}$ denotes the covariance matrix coefficient for stock $i$ and stock $j$, $e_i$ and $m_i$ denote the expected return and the participation threshold for stock $i$, respectively, $M$ is a sufficiently large number.

Because of binary variables the problem is nonconvex, which necessitates using one of optimization problems (3.1) or (3.2) to derive properly efficient outcomes (the original mean-variance problem is $R_+^k$-convex and in that case solving optimization problem (3.4) is advised as simplest). Here we make use of optimization problem (3.1).

Data is taken from the well-known "three-stock" example by Markowitz and we amend it with arbitrarily selected numbers for thresholds $m_i$. In the Markowitz example there are three stocks denoted as ATT, GMC, USX, characterized by the following covariance matrix and the expected returns over one investment period:

| | $ATT$ | $GMC$ | $USX$ | |
|---|---|---|---|---|
| $ATT$ | 0.01080754 | 0.01240721 | 0.01307513 | |
| $GMC$ | 0.01240721 | 0.05839170 | 0.05542639 | covariance matrix |
| $USX$ | 0.01307513 | 0.05542639 | 0.09422681 | |
| | 0.0890833 | 0.213667 | 0.234583 | expected returns |
| | none | none | 0.3 | thresholds |

We arbitrarily assume that the value of an individual stock selected for inclusion into the portfolio cannot be greater than (constraint of that sort mimics market regulations on short selling) ten times the value of the available capital, therefore we set $M = 10$.

Let $y\_v = -\sum_{i=1}^{n}\sum_{j=1}^{n}\beta_{ij}x_i x_j$ and $y_e = \sum_{i=1}^{n} e_i x_i$.

To illustrate working with $GIS^2$ we assume that the DM follows an arbitrary but rational (in a vague sense) decision making scenario.

The usual practice is that the Markowitz model is solved usually once: the DM sets the minimal level of expected return he is willing to accept (which reduces the first criterion to a constraint) and a portfolio which maximizes negative of variance for that level of expected return is found. But such a procedure severely limits the DM in his decision power.

Here we solve the problem by an interactive decision making process following a hypothetic scenario which simulates preferences and behavior of the DM. We assume that the DM is willing to search for a portfolio which satisfactorily compromises expected return $y_e$ with variance $-y_{-v}$ .

The initial set of admissible vectors of weights is $\bar{\Lambda} = \{\lambda \,|\, \lambda_{-v} \geq 0,\ \lambda_e \geq 0,\ \lambda_{-v} + \lambda_e = 1\}$.

Optimization problem (3.1) for the portfolio mean-variance problem specified above takes the form

$$\begin{gathered} \min \max\{\lambda_{-v}((1+\rho)(y^*_{-v} + \textstyle\sum_{i=1}^3 \sum_{j=1}^3 \beta_{ij} x_i x_j) \\ +\rho(y^*_e - \textstyle\sum_{i=1}^3 e_i x_i)), \\ \lambda_e((1+\rho)(y^*_e - \textstyle\sum_{i=1}^3 e_i x) + \rho(y^*_{-v} + \textstyle\sum_{i=1}^3 \sum_{j=1}^3 \beta_{ij} x_i x_j))\}, \\ \textstyle\sum_{i=1}^3 x_i = 1, \\ m_3\delta_3 \leq x_3 \leq M\delta_3\,, \\ \delta_3 = 0 \text{ or } 1. \end{gathered} \tag{6.1}$$

To be solvable by a standard optimization algorithm, (6.1) is transformed to (cf. Subsection 2.1 of Chapter 3)

$$\begin{gathered} \min t \\ t \geq \lambda_{-v}((1+\rho)(y^*_{-v} + \textstyle\sum_{i=1}^3 \sum_{j=1}^3 \beta_{ij} x_i x_j) + \rho(y^*_e - \textstyle\sum_{i=1}^3 e_i x_i)), \\ t \geq \lambda_e((1+\rho)(y^*_e - \textstyle\sum_{i=1}^3 e_i x) + \rho(y^*_{-v} + \textstyle\sum_{i=1}^3 \sum_{j=1}^n \beta_{ij} x_i x_j)), \\ \textstyle\sum_{i=1}^3 x_i = 1, \\ m_3\delta_3 \leq x_3 \leq M\delta_3\,, \\ \delta_3 = 0 \text{ or } 1. \end{gathered} \tag{6.2}$$

Since the problem is unbounded in $y_e$, we arbitrarily add a constraint $y_e \leq 2$, hence $\hat{y}_e = 2$. On the other hand, $y_{-v}$ attains over the feasible set its maximum equal to $-0.011$, so we put $\hat{y}_{-v} = -0.011$.

By selecting $\epsilon = 0.001$ we get $y^* = (\hat{y}_{-v} + \epsilon, \hat{y}_e + \epsilon) = (-0.010, 2.001)$.

We arbitrarily set parameter $\rho$ to $0.01$.

Global trade-offs are calculated by the method described at the beginning of the next section.

To simulate the DM's behavior it is necessary to assume that we know his value function. We arbitrarily assume that the DM's value function is

$$f^{DM}(y) = -30y_{-v}^2 + y_e^2 .$$

Over the feasible set this function attains its maximum (to find this value an optimization problem is solved) equal to $f^{DM_{max}} = 1.379$ at $y^{DM_{max}} = (y_{-v}^{DM_{max}}, y_e^{DM_{max}}) = (-0.058,\ 1.216)$.

However, it has to be stressed here again that in practice the DM's value function is neither known nor it can be easily revealed. Even the conjecture of its existence in an implicit form in the DM's mind is often disputed and is certainly not proved in general. We use the function $f^{DM}(y)$ purely to set a decision framework for the numerical example.

## 5. $GIS^2$ At Work

Below we solve the problem introduced in the previous section following the steps of $GIS^2$. Though objects available for manipulations are, as specified by $GIS^2$, weights and reference points, to stress that $GIS^2$ falls into the framework of the Universal Interface, as described in Chapter 5, at each iteration references are made to vectors $\tau$.

♣ We can calculate global trade-offs by a more refined method than calculating the value of (2.7) directly.

Let $Z$ be closed and bounded, and let $\bar{y} \in Z$ be an outcome for which global trade-off $T_{ji}^G(\bar{y})$ exists. It is proved that:

- either there exists the maximal value $\bar{\rho}$ of parameter $\rho$, $\rho > 0$, for which $\bar{y}$ solves the problem

$$\max_{y \in Z_i^{\leq}(\bar{y})} (y_i + \rho y_j),$$

where $Z_i^{\leq}(\bar{y}) = \{y \in Z \mid y_i \leq \bar{y}_i,\ y_l \geq \bar{y}_l,\ l = 1, ..., k,\ l \neq i\}$, (cf. Section 4 of Chapter 2 for the definition of the related construct $Z_i^{<}(y)$), and

$$T_{ji}^{G}(\bar{y}) = (\bar{\rho})^{-1},$$

- or $\bar{y}$ solves the above problem for every positive $\rho$ and

$$T_{ji}^{G}(\bar{y}) = 0.$$

We use this method of global trade-off calculation in all the numerical examples of the book.

♠

For a given vector of weights $\lambda,\ \lambda_i > 0,\ i = 1, ..., k$, let $y(\lambda)$ denote a (properly) efficient outcome derived be solving optimization problem (3.1) with that $\lambda$.

EXAMPLE 1 In the version of $GIS^2$ presented below at each iteration only one candidate is derived.

*Initialization*

0. $\bar{\Lambda} = \{\lambda \mid \lambda_i > 0,\ i = 1, 2,\ \lambda_1 + \lambda_2 = 1\}$.

To start the decision making process two vectors of weights

$$\lambda^{inc} = (0.5, 0.5),$$
$$\lambda^{c} = (0.98, 0.02),$$

where superscript *inc* stays for "incumbent", and superscript *c* for "candidate", are selected from $\bar{\Lambda}$ by the rule of thumb. To simplify notation, incumbent $y(\lambda^{inc})$ and candidates $y(\lambda^{c})$ are not distinguished by iteration number.

$GIS^2$ sends down to the optimization engine:

- vector $\tau^{inc}$ and element $y^*$,
- vector $\tau^{c}$ and element $y^*$.

Since optimization problem (3.1) is used, vectors $\tau$ are defined indirectly by vectors $\lambda$ (cf. formula (5.1) and (5.2)). So $GIS^2$ sends down in fact vector $\lambda^{inc}$ and parameter $\rho$, and vector $\lambda^{c}$ and parameter $\rho$, respectively (with optimization problem (3.2) or optimization problem (3.3) there is an explicit correspondence between vectors $\tau$ and $\lambda$, cf. Chapter 5).

♣ Nowhere in the book the claim is made that in the framework of the weight method class the DM expresses his preference by specifying vectors $\tau$. However, it is shown (Subsection 2.1 of Chapter 5) that there is a one-to-one correspondence between vectors $\tau$ and pairs $(\lambda, \rho)$.
♠

The optimization engine returns:

- (properly) efficient outcome $y(\lambda^{inc}) = (-0.499, 1.511)$, $f^{DM}(y(\lambda^{inc})) = -5.192$,
- (properly) efficient outcome $y(\lambda^{c}) = (-0.019, 1.138)$, $f^{DM}(y(\lambda^{c})) = 1.285$.

Next, $GIS^2$ sends down to the optimization engine:

- two pairs of indices $(-v, e)$ and $(e, -v)$, and (properly) efficient outcome $y(\lambda^{inc})$,
- two pairs of indices $(-v, e)$ and $(e, -v)$, and (properly) efficient outcome $y(\lambda^{c})$.

The optimization engine returns:

- global trade-off $T^G_{-v,e}(y(\lambda^{inc})) = 2.28$,
- global trade-off $T^G_{e,-v}(y(\lambda^{inc})) = 0.44$,
- global trade-off $T^G_{-v,e}(y(\lambda^{c})) = 0.30$,
- global trade-off $T^G_{e,-v}(y(\lambda^{c})) = 3.33$.

*Iteration 1*

1. Suppose that (consistently with the values of the assumed DM's implicit value function) the DM prefers $y(\lambda^{c})$ to $y(\lambda^{inc})$ and this preference is firm.

The reason for such a preference can be that outcome $y(\lambda^{inc})$ shows unacceptable high risk as compared to $y(\lambda^{c})$ and this risk is not outweighed by the significant expected return of $y(\lambda^{inc})$.

So $y(\lambda^{inc}) = (-0.019, 1.138)$, $f^{DM}(y(\lambda^{c})) = 1.285$.

2. Suppose the DM wants to continue because outcome $y(\lambda^{inc})$ has a potential for improvement of gain at the expense of risk, as indicated by the value of global trade-off $T^G_{e,-v}(y(\lambda^c))$ equal to $3.33$.

3. Suppose that the DM selects weights to manipulate. The firm preference expressed in step 1 gives rise to a weight cut resulting from function (4.5). In consequence, one gets two disjoint sets of conditions for $\lambda$, of which only one is consistent, namely the set:

$$\begin{aligned} \lambda_{-v} &> 0, \\ \lambda_e &< 0.572\, \lambda_{-v}\,. \end{aligned} \tag{6.3}$$

The above conditions update the initial (i.e. formed of the non-negativity and the 'sum to one' constraints) set of admissible weights $\bar{\Lambda}$.

Suppose that the DM selects from the updated set of admissible weights vector $\lambda^c = (0.75, 0.25)$.

4. A new candidate is derived.

$GIS^2$ sends down to the optimization engine:

- vector $\tau^c$ (in fact vector $\lambda^c$ and parameter $\rho$) and element $y^*$.

The optimization engine returns:

- (properly) efficient outcome $y(\lambda^c) = (-0.217, 1.362)$, $f^{DM}(y(\lambda^c)) = 0.439$.

Next, $GIS^2$ sends down to the optimization engine:

- two pairs of indices $(-v, e)$ and $(e, -v)$, and (properly) efficient outcome $y(\lambda^c)$.

The optimization engine returns:

- global trade-off $T^G_{-v,e}(y(\lambda^c)) = 1.49$.
- global trade-off $T^G_{e,-v}(y(\lambda^c)) = 0.71$.

*Iteration 2*

1. Suppose that (consistently with the values of the assumed DM's implicit value function) the DM prefers $y(\lambda^{inc}) = (-0.019, 1.138)$ to $y(\lambda^c) = (-0.217, 1.362)$ but this preference is not firm.

The reason for such a preference can be that outcome $y(\lambda^c)$ shows unacceptable high risk as compared to $y(\lambda^{inc})$ and this risk is not outweighed by the significant gain of $y(\lambda^c)$.

So $y(\lambda^{inc}) = (-0.019, 1.138)$, $f^{DM}(y(\lambda^{inc})) = 1.285$.

2. Suppose the DM wants to continue because outcome $y(\lambda^{inc})$ has a potential for improvement of gain at the expense of risk, as indicated by the value of global trade-off $T^G_{e,-v}(y(\lambda^{inc}))$ equal 3.33.

3. Suppose that the DM selects weights to manipulate. Since the preference expressed in step 1 of the current iteration is not firm, no update of the set of admissible weights is available.

Suppose that the DM selects from the set of admissible weights vector $\lambda^c = (0.90, 0.10)$.

4. A new candidate is derived.

$GIS^2$ sends down to the optimization engine:

- vector $\tau^c$ (in fact, vector $\lambda^c$ and parameter $\rho$) and element $y^*$.

The optimization engine returns:

- global efficient outcome $y(\lambda^c) = (-0.086, 1.251)$, $f^{DM}(y(\lambda^c)) = 1.345$.

Next, $GIS^2$ sends down to the optimization engine:

- two pairs of indices $(-v, e)$ and $(e, -v)$, and (properly) efficient outcome $y(\lambda^c)$.

The optimization engine returns:

- global trade-off $T^G_{-v,e}(y(\lambda^c)) = 0.89$,
- global trade-off $T^G_{e,-v}(y(\lambda^c)) = 1.12$.

*Iteration 3*

1. Suppose that (consistently with the values of the assumed DM's implicit value function) the DM prefers $y(\lambda^c) = (-0.086, 1.251)$ to $y(\lambda^{inc}) = (-0.019, 1.138)$ but this preference is not firm.

The reason for such a preference can be that outcome $y(\lambda^c)$ satisfactorily compromises risk with gain and shows little potential for improvement of gain at the expense of risk, as indicated by the value of global trade-off $T^G_{e,-v}(y(\lambda^{inc}))$ equal to 1.12.

So $y(\lambda^{inc}) = (-0.086, 1.251)$, $f^{DM}(y(\lambda^c)) = 1.345$.

2. Suppose the DM wants to continue, despite the fact that $y(\lambda^{inc})$ shows little potential for improvement of gain at the expense of risk, as indicated by the value of global trade-off $T^G_{e,-v}(y(\lambda^{inc}))$ equal to 1.12.

3. Suppose that the DM selects reference points to manipulate. Suppose that as a reference point the DM selects $y^{ref} = (-0.070, 1.300)$.

4. A new candidate is derived. For that purpose vector $\lambda^c = (0.90, 0.10)$ is selected from the set of admissible weights by the rule of thumb. This is exactly the same vector of weights used to derive the current incumbent $y(\lambda^{inc})$ because such a weight combination yields up to now the most satisfactory compromise of risk and gain.

$GIS^2$ sends down to the optimization engine:

- vector $\tau^c$ (in fact vector $\lambda^c$ and parameter $\rho$) and element $y^{ref}$.

The optimization engine returns:

- (properly) efficient outcome $y(\lambda^c) = (-0.076, 1.240)$, $f^{DM}(y(\lambda^c)) = 1.364$.

Next, $GIS^2$ sends down to the optimization engine:

- two pairs of indices $(-v, e)$ and $(e, -v)$, and (properly) efficient outcome $y(\lambda^c)$.

The optimization engine returns:

- global trade-off $T^G_{-v,e}(y(\lambda^c)) = 0.85$,
- global trade-off $T^G_{e,-v}(y(\lambda^c)) = 1.18$.

*Iteration 4*

1. Suppose that (consistently with the values of the assumed DM's implicit value function) the DM prefers $y(\lambda^c) = (-0.076, 1.240)$ to $y(\lambda^{inc}) = (-0.086, 1.251)$ but this preference is not firm.

The reason for such a preference can be that though outcome $y(\lambda^{inc})$ satisfactorily compromises risk with gain and shows little potential for gain improvement at the expense of risk, as indicated by the value of global trade-off $T^G_{e,-v}(y(\lambda^{inc}))$ equal to $1.12$, the overall DM evaluation of $y(\lambda^c)$ is higher than of $y(\lambda^{inc})$.

So $y(\lambda^{inc}) = (-0.076, 1.240)$, $f^{DM}(y(\lambda^{inc})) = 1.364$.

2. Suppose the DM does not want to continue. The process terminates.

The decision process is summarized in Table 6.1. □

In general, one can expect that by proceeding along $GIS^2$ lines quite satisfactory outcomes (alternatives) can be arrived at. But no particular conclusion can be drawn from the fact that in the numerical example solved above the decision making process terminates at an outcome (portfolio) for which the assumed value function takes a value so close to the maximal value of that function over the feasible set.

*Table 6.1.* Successive iteration data for the example portfolio selection problem of Chapter 6.

| It. | Preference via $\lambda$, $y$ | Outcome | Value fun. | Trade-off $T^G_{-v,e}$ | $T^G_{e,-v}$ |
|---|---|---|---|---|---|
| 0 | $(0.50, 0.50)$, $y^*$ | $(-0.499,\ 1.511)$ | -5.192 | 2.28 | 0.44 |
| | $(0.98, 0.02)$, $y^*$ | $(-0.019,\ 1.138)$ | 1.285 | 0.30 | 3.33 |
| 1 | *Incumbent* | $(-0.019,\ 1.138)$ | 1.285 | | |
| | $(0.75, 0.25)$, $y^*$ | $(-0.217,\ 1.362)$ | 0.439 | 1.49 | 0.71 |
| 2 | *Incumbent* | $(-0.019,\ 1.138)$ | 1.285 | | |
| | $(0.90, 0.10)$, $y^*$ | $(-0.086,\ 1.251)$ | 1.345 | 0.89 | 1.12 |
| 3 | *Incumbent* | $(-0.086,\ 1.251)$ | 1.345 | | |
| | $(0.90, 0.10)$, $y^{ref}$ | $(-0.076,\ 1.240)$ | 1.364 | 0.85 | 1.18 |
| 4 | *Incumbent* | $(-0.076,\ 1.240)$ | 1.364 | | |

## 6. Concluding Remarks

As observed before, the scheme proposed above exploits no new ideas. In its present form it is composed of a simple loop of two actions, namely of outcome derivation and outcome evaluation, where evaluations are made on the basis of two types of information: outcome component values and outcome global trade-offs. Outcome evaluations guide subsequent outcome derivation via either weight manipulations or reference point manipulations. All the remaining elements, by no means insignificant in specific decision making problem settings, can be designated as technical, hidden from the DM in the background, and brought out at his explicit request only.

The scheme is the best we can do with no assumptions made about the DM preferences and behavior, and about the decision making problem structure. We are not going to make any assumptions about the DM for, as stated before, such assumptions are not verifiable. We certainly can make assumptions about the problem structure to get specialized versions of $GIS^2$. This, however, would bring us back to various MCDM methods of which $GIS^2$ is an abstraction. Therefore, in this book we confine ourselves to presentation and extension (in Chapter 9) of $GIS^2$ in general terms.

To make the Universal Interface and $GIS^2$ popular tools for MCDM and to provide for their widespread use in the realm of decision making we have to overcome one more obstacle, which is the need to use optimization when solving MCDM complex problems. We tackle this problem in the next two chapters. We aim at releasing interactive MCDM from the "optimization grip", which is a paramount challenge. Except specifically simply structured MCDM problems with sets of feasible alternatives given in the form of lists, interactive MCDM methods rely heavily on use of optimization problems and optimization techniques. We challenge this reliance in the next two chapters, and eventually, in Chapter 9, we present a soft computing form of $GIS^2$.

## 7. Annotated References

The idea behind $GIS^2$ can be traced back to numerous papers, especially those published in late 90's of the former century and recently, where authors express awareness of insurmountable problems related to practical verification of assumptions made about the DM behavior and his consistency when he makes his decisions. In particular, we can point to Buchanan (1997) as a representative example of this stream of research.

Supporting schemes with relative preference expressing based on trade-offs are proposed in Kaliszewski et al. (1997), Kaliszewski, Michalowski (1999), Kaliszewski (2000), and Kaliszewski, Zionts (2004). The concept of the isolation effect based on relative preference expressing is applied to stock portfolio selection in Jog, Kaliszewski, Michalowski (1999). Optimization independent MCDM support is advocated in Kaliszewski (2004).

The mean-variance portfolio problem is originally proposed in Markowitz (1959) to become a classical problem in mathematical finance literature (cf. e.g. Elton, Gruber (1995), Zenios (1993)). From Markowitz (1959) we take data for the example described in Section 4 and solved in Section 5 of this chapter.

To calculate global trade-offs in Section 5 of this chapter we exploit the method described in Kaliszewski (1993,1994b) based on parametric programming instead of finding the maximum of the formula (2.7) directly.

Chapter 7

# BOUNDS ON OUTCOME COMPONENTS

*" 'Now,' said Rabbit, 'this is a Search, and I've Organized it -' "*

**A. A. Milne**,
*The House at Pooh Corner.*

## 1. This Chapter Is About ...

... our response to the challenge of creating simple and low computing-intensive methods of interactive MCDM. In this and the next two chapters we show how bounds on outcome components can be used in interactive MCDM, and how to economize on computing costs. All this and no trivia.

Rarely decision making problems are solved with exact data. Numbers to be dealt with are usually approximations or round-offs. Therefore exact computing makes little sense. Taking this one step further, we can say that *soft computing* schemes are advisable to cope with the interactive MCDM paradigm.

So, in response to this challenge, below we present a way of calculating lower and upper bounds on efficient outcome components for a very small computing cost. If sufficiently tight, those bounds provide valid assessments of efficient outcomes. The bounds can be used in any interactive MCDM method instead outcome component true values.

This development bridges the interactive MCDM and soft computing.

Moreover, as shown below, this development relieves interactive MCDM methods of direct dependence on optimization techniques and software. In consequence, bounds on outcome components can be calculated in the same, uniform manner, no matter what type of models (linear, nonconvex, discrete) problem (2.1) belongs to. This is possible because, as shown below, there exists one common formula for lower and upper bounds, respectively, independent of model type.

In the next section we reveal the main idea of this chapter in a more detailed way. In Section 3 we present static bounds on Pareto sets and in Section 4 we present Pareto set approximations based on finite subsets of Pareto sets. In Section 5 we present formulas for calculating lower and upper bounds on components of efficient outcomes. We illustrate how those bounds can work in practice by solving, in Section 6 of this chapter, a numerical example.

## 2. Bounds On Efficient Outcome Components

As observed in Introduction, MCDM problems are nowadays usually solved interactively via sequences of "man-machine" interactions. Each interaction is composed of the "man" phase - the DM expresses his partial preferences, and the "machine" phase - the underlying model of the problem is used to derive an efficient outcome which fits those preferences.

The whole process would be obviously more effective if efficient outcomes conforming to the DM's partial preferences could be assessed prior to their derivation. To this aim let us assume that DM's partial preferences are encapsulated in and expressed by weights or at least that weights are just a technical tool to distinguish and then to identify efficient outcomes. We develop assessments of efficient outcomes in the form of bounds on efficient outcome components with weights as parameters. As shown below, this idea is fully operational.

Calculating bounds on outcome components is an alternative to deriving outcomes explicitly, and this offers significant computation savings. Such an approach can also reduce technical complexity of MCDM methods. We postpone a detailed discussion of these issues to Chapter 9.

The idea behind parametric bounds is straightforward. It is based on the assumption that a certain number of efficient outcomes is derived up-front (i.e. before starting an interactive MCDM process). As well known (cf. Chapter 3), efficient outcomes can be derived by maximizing (or minimizing) a certain function over the outcome set. Then, as shown below, the function, and an up-front derived outcome which maximizes (or minimizes) this function, yield parametric (with weights as parameters) lower and upper bounds on components of other efficient outcomes.

## 3. Bounds On Pareto Set

A prerequisite of any interactive MCDM process is to enclose the set of all efficient outcomes of $Z$, i.e. the Pareto set, by a box, i.e. to derive $\bar{L},\ \bar{U}$ such that

$$\bar{L}_i \leq y_i \leq \bar{U}_i\,,\ i = 1, ..., k,\ y \text{ - any efficient outcome of } Z.$$

For any practical decision problem such bounds clearly exist. If problem (2.1) does not provide for such bounds, it should be modified accordingly to ensure boundedness of the Pareto set.

Tight upper bounds $\bar{U}_i$ are given in a straightforward manner by

$$\bar{U}_i = \hat{y}_i\,,\ i = 1.....k,$$

where $\hat{y}$ is the utopian element defined in Chapter 2.

Deriving tight lower bounds requires laborious computations. It is common then to use $y^N$ as an approximation of $\bar{L}$ , where $y^N$ is given by

$$y_i^N = \min_{j \in \{1,...,k\}} \tilde{y}_i^j\,,\ i = 1.....k,$$

and

$$\tilde{y}^j = arg\ \max_{y \in Z} y_j\,,\ j = 1, ..., k.$$

However, it should be observed that the relation

$$y_i^N \leq y_i\,,\ y \text{ - any efficient outcome of } Z,$$

is not, in general, valid.

Bounds $\bar{L}$ and $\bar{U}$ are static. In particular, these bounds do not change as an interactive decision making process progresses and more outcomes become explicitly identified.

## 4. Pareto Set Approximations

Any finite subset $S,\ S \subseteq Z$, of weakly efficient outcomes derived by solving optimization problem (3.1), (3.2), or (3.4), for a number of vectors $\lambda, \lambda_i > 0, i = 1, ..., k$, is called *shell*. Hence, $S = \{y(\lambda^s)\},\ s = 1, ..., |S|$ and $y(\lambda^s)$ is a solution of the corresponding optimization problem with $\lambda = \lambda^s$. Shell are *discrete representations* of Pareto sets.

An outcome which is not derived explicitly (i.e. it is not an *explicit* outcome) but is only designated by selecting vector $\lambda$ is called an *implicit* outcome.

In the next section we use shells to calculate parametric bounds on implicit outcomes, with weights $\lambda$ as parameters.

What we are aiming at is as follows. Suppose a vector of weights $\lambda$ is given. Let $y(\lambda)$ denote an implicit efficient outcome of $Z$, which would be derived if optimization problem (3.1), (3.2), (3.3), or (3.4) were solved with that $\lambda$. Let $L(y(\lambda))$ and $U(y(\lambda))$ be vectors of lower and upper bounds on components of $y(\lambda)$, respectively. These bounds form an *assessment* $[y(\lambda)]$ of $y(\lambda)$,

$$[y(\lambda)] = \{L(y(\lambda)), U(y(\lambda))\}.$$

To simplify notation we put $L(y(\lambda)) = L(\lambda)$ and $U(y(\lambda)) = U(\lambda)$.

The set of assessments calculated with a shell of $Z$ for all possible $\lambda$ forms a Pareto set *parametric approximation*.

♣ Every shell of $Z$ defines by definition a *continuous approximation* of the Pareto set. The approximation depends on which optimization problem is used to derive elements of the shell. To see this recall (cf. Section 3 of Chapter 2) that any $\bar{y} \in Z$ which solves problem (3.1), (3.2), (3.3), or (3.4) for some $\lambda, \lambda_i > 0, i = 1, ..., k$, and some $\rho > 0$, as appropriate, satisfies the following condition

$$(\{\bar{y}\} + int(K^i)) \cap Z = \{\emptyset\},$$

where $K^i,\ i \in \{1, 2, 3, 4\}$, is one of the cones defined in Subsection 2.1 of Chapter 3. Hence, $\bar{y}$ is $K^i$-weakly efficient.

Thus, each element $\bar{y}$ of a shell of $Z$ (i.e. $K^i$-weakly efficient element) defines two *dead regions*, namely

*upper dead region*

$$D^U(\bar{y}) = int(\{\bar{y}\} + K^i) \cap \{y \,|\, \bar{L} \leq y \leq \bar{U}\},$$

and

*lower dead region*

$$D^L(\bar{y}) = int(\{\bar{y}\} - K^i) \cap \{y \,|\, \bar{L} \leq y \leq \bar{U}\},$$

where $i \in \{1, 2, 3, 4\}$, depending on which problem (3.1), (3.2), (3.3), or (3.4) is used to derive elements of the shell.

Observe that $K^1$ is a function of $\rho$, $K^2$ is a function of $\rho$ and $\lambda$, $K^3$ $(K^3 = R^k_+)$ is constant, and $K^4$ is a function of $\lambda$ (cf. Subsection 2.1 of Chapter 3).

Clearly, given $K^i$, no $K^i$-weakly efficient outcome belongs to the upper dead region defined by $\bar{y}$, for if otherwise, $\bar{y}$ would not be $K^i$-weakly efficient.

Similarly, given $K^i$, no $K^i$-weakly efficient outcome belongs to the lower dead region defined by $\bar{y}$, for if otherwise such an outcome would not be $K^i$-weakly efficient.

For selected $K^i$, $i \in \{1, 2, 3, 4\}$, a continuous approximation of the Pareto set is the union of shell elements and the space left of the box $\{y \in \mathcal{R}^k \mid \bar{L} \leq y \leq \bar{U}\}$ after removing from it the union of upper dead regions and the union of lower dead regions, i.e. the set

$$S \cup (\{y \in \mathcal{R}^k \mid \bar{L} \leq y \leq \bar{U}\} \setminus \{\cup_{y \in S} D^U(y) \bigcup \cup_{y \in S} D^L(y)\}).$$

This continuous approximation contains all $K^i$-weakly efficient outcomes of $Z$. In virtue of the obvious relations

$$R^k_+ \subseteq K^i, \; i = 1, ..., 4,$$

the following relation holds

$$\{\text{set of } K^i\text{-weakly efficient outcomes}\}$$

$$\subseteq \{\text{set of } R^k_+\text{-weakly efficient outcomes}\}.$$

Thus, this continuous approximation of the Pareto set may not contain some $R^k_+$-weakly efficient outcomes, and in consequence, it may not contain $R^k_+$-efficient (i.e. efficient) outcomes.

This is illustrated schematically in Figure 7.1. The left drawing shows a Pareto set continuous approximation resulting from a shell composed of two properly efficient outcomes derived by solving problem (3.1) or (3.2) (circles mark weakly efficient outcomes belonging to the continuous approximation, small black discs mark elements belonging to the union of dead regions, and big black discs mark elements of the shell). The right drawing shows a Pareto set continuous approximation resulting from a shell composed of three weakly efficient outcomes derived by solving problem (3.3).

Continuous approximations of Pareto sets with cones $K^4$ are used for specific purposes in Section 8 of this chapter.

Shells and the related Pareto set continuous approximations yield, as shown below, lower and upper bounds on outcomes.

♠

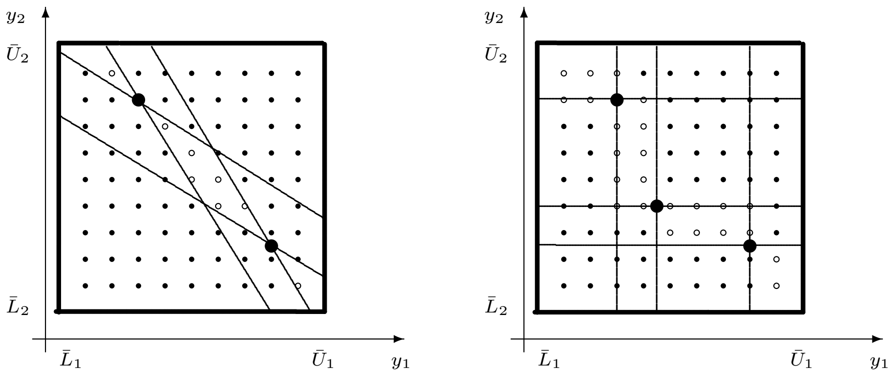


*Figure 7.1.* Continuous approximations of the Pareto set depending on whether shell elements are derived by optimization problem (3.1), (3.2), or (3.3).

## 5. Parametric Bounds On Efficient Outcomes

In this section we present formulas for bounds on components of efficient outcomes. The bounds are dynamic, i.e. they become stronger with increasing number of outcomes included in the shell they are related to. These bounds are also parametric with weights $\lambda$ used in optimization problems (3.1), (3.2), (3.3), or (3.4) as parameters. Computational costs to calculate such bounds are negligible. Here again (cf. Chapter 3) vector of weights $\lambda$ plays the role of a tool to "navigate" over the set of efficient outcomes.

Formulas we show may, at the first glance, look complicated but in fact they consist of no more than operations of adding and taking maxima over finite sets of numbers.

## 5.1 Lower Bounds

Below we show formulas to calculate lower bounds on outcome components. For a given vector of weights $\lambda,\ \lambda_i > 0,\ 1, ...k$, let $y(\lambda)$ be an implicit efficient outcome $y(\lambda)$, which would be derived if optimization problem (3.1), (3.2), (3.3), or (3.4) were solved with that $\lambda$.

Suppose that shell $S$ of problem (2.2) is given. For calculating lower bounds it makes no difference which optimization problem was used to derive shell elements. In particular, shell elements can be derived by solving optimization problem (3.1) or (3.2) with different values of parameter $\rho$.

When optimization problem (3.1) is used to derive (properly) efficient outcomes we have the following lower bounding formula

$$y(\lambda)_i \geq L(S,\lambda)_i$$

$$= \max\{y_i^* - (\lambda_i(1+\rho))^{-1} \max_{y \in S}[\max_j \lambda_j((y_j^* - y_j) \tag{7.1}$$

$$+\rho e^k(y^* - y))] + \tfrac{\rho}{1+\rho} \textstyle\sum_{j \neq i}^{k}(y_j^* - U(\lambda)_j), \bar{L}_i\},\ i = 1, ..., k,$$

where $U(\lambda)_j$ are such that $y(\lambda)_j \leq U(\lambda)_j\ ,\ j = 1, ..., k,\ j \neq i$. One possible selection of $U(\lambda)_j$ is $\bar{U}_j\ ,\ j = 1, ..., k,\ j \neq i$.

When optimization problem (3.2) is used to derive (properly) efficient outcomes we have

$$y(\lambda)_i \geq L(S,\lambda)_i$$

$$= \max\{y_i^* - (\lambda_i + \rho)^{-1} \max_{y \in S}[\max_j \lambda_j(y_j^* - y_j) + \rho e^k(y^* - y)]$$

$$+\tfrac{\rho}{\lambda_i+\rho} \textstyle\sum_{j \neq i}^{k}(y_j^* - U(\lambda)_j), \bar{L}_i\},\ i = 1, ..., k, \tag{7.2}$$

where $U(\lambda)_j\ ,\ j = 1, ..., k,\ j \neq i$, are defined as above.

When optimization problem (3.3) is used to derive (weakly) efficient outcomes we obtain the lower bounding formula by putting $\rho = 0$ in either of the above two bounding formulas, which results in

$$y(\lambda)_i \geq L(S,\lambda)_i$$

$$= \max\{y_i^* - \lambda_i^{-1} \max_{y \in S}[\max_j \lambda_l(y_j^* - y_j)], \bar{L}_i\},\ i = 1, ..., k. \tag{7.3}$$

Finally, when optimization problem (3.4) is used to derive (properly) efficient outcomes we have

$$y(\lambda)_i \geq L(S,\lambda)_i$$

$$= \max\{\max_{y\in S}(y_i + \lambda_i^{-1}\textstyle\sum_{j\neq i}^{k}\lambda_j(y_j - U(\lambda)_j)), \bar{L}_i\},\ i = 1,...,k, \quad (7.4)$$

where $U(\lambda)_j,\ j = 1,...,k,\ j \neq i$, are defined as above.

♣ To show how the above lower bounds are derived, let us consider Pareto set characterizations by weight manipulations presented in Section 2 of Chapter 2, namely Characterization I, Characterization II, and Characterization IV. We start with Characterization I.

As stated in Chapter 3, a properly efficient outcome of $Z$ can be explicitly identified by solving optimization problem (3.1), i.e.

$$\min_{y\in Z}\max_i \lambda_i((y_i^* - y_i) + \rho e^k(y^* - y)) \text{ for some } \lambda_i > 0,\ i = 1,...,k,$$

and some $\rho > 0$.

Suppose that (an implicit) $y(\lambda)$ solves (3.1) for some $\lambda, \lambda_i > 0,\ i = 1,...,k$, and some $\rho > 0$ . Then, for each $y \in Z$,

$$\min_{y\in Z}\max_i \lambda_i((y_i^* - y_i) + \rho e^k(y^* - y))$$

$$= \max_i \lambda_i((y_i^* - y(\lambda)_i) + \rho e^k(y^* - y(\lambda)))$$

$$\leq \max_i \lambda_i((y_i^* - y_i) + \rho e^k(y^* - y)).$$

This entails that for each $y \in Z$ and for each $i = 1,...,k$,

$$\lambda_i((y_i^* - y(\lambda)_i) + \rho e^k(y^* - y(\lambda)))$$

$$\leq \max_j \lambda_j((y_j^* - y_j) + \rho e^k(y^* - y)).$$

Thus, for each $y \in Z$ and for each $i = 1,...,k$,

$$y(\lambda)_i \geq y_i^* - (\lambda_i(1+\rho))^{-1}(\max_j \lambda_j((y_j^* - y_j) + \rho e^k(y^* - y)))$$

$$+\tfrac{\rho}{1+\rho}\textstyle\sum_{j\neq i}^{k}(y_j^* - y(\lambda)_j).$$

With outcomes of shell $S$, $|S| \geq 1$, upper bounds $U(\lambda)_j$, $j = 1,...,k,\ j \neq i$, and lower bounds $\bar{L}_i$, $i = 1,...,k$, we obtain formula (7.1).

By a similar argument we obtain formula (7.2) and formula (7.4).

Since optimization problem (3.4) with all weights equal to one is the limit of problem (3.1) or problem (3.2) for $\rho \to +\infty$ (cf. Remark 1 of Chapter 3), passing in formulas (7.1) or (7.2) with $\rho$ to the limit $+\infty$ we should obtain formula (7.4) with all weights equal to one. It is easy to check that this indeed holds.

♠

The idea of lower bounds (7.1) can be explained as shown in Figure 7.2. Properly efficient outcomes of $S = \{y(\lambda^s)\},\ s = 1, ..., |S|$, are used to derive lower bounds on components of outcome $y(\lambda)$, which solves optimization problem (3.1) for that $\rho$ and that $\lambda$. The same argument applies to bounds (7.2) and (up to the role of parameter $\rho$) to bounds (7.3) (cf. Figure 7.3), and to bounds (7.4) (cf. Figure 7.4).

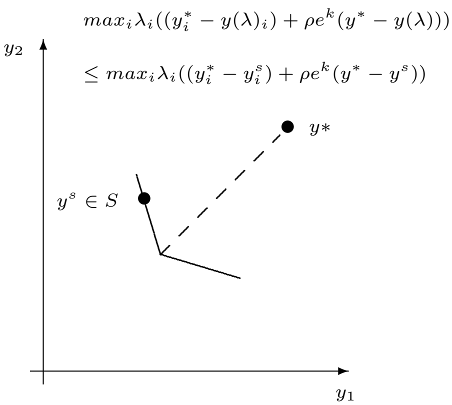


*Figure 7.2.* A graphical interpretation of lower bounds when optimization problem (3.1) is used to derive (properly) efficient outcomes.

## 5.2 Upper Bounds

Below we show formulas to calculate upper bounds on outcome components. For a given vector of weights $\lambda,\ \lambda_i > 0,\ i = 1, ..., k$, let $y(\lambda)$ be, as in the previous subsection, an implicit outcome $y(\lambda)$, which would be derived if optimization problem (3.1), or (3.2), (3.3), or (3.4) were solved with that $\lambda$.

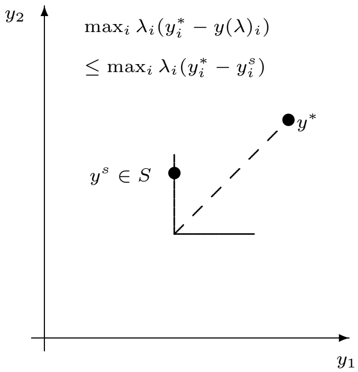


*Figure 7.3.* A graphical interpretation of lower bounds when optimization problem (3.3) is used to derive (weakly) efficient outcomes.

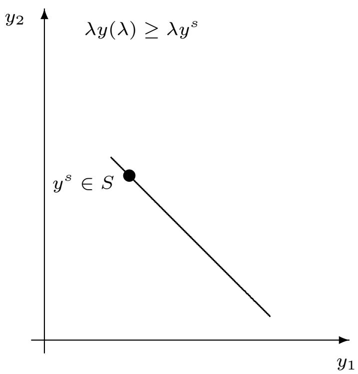


*Figure 7.4.* A graphical interpretation of lower bounds when optimization problem (3.4) is used to derive (properly) efficient outcomes.

Suppose that shell $S$ of problem (2.2) is given. For calculating upper bounds, in contrast to lower bounds, shells have to be composed of outcomes which are derived by solving the same (up to $\lambda$) optimization problem which would be used to derive explicitly an outcome for which upper bounds are calculated. Therefore, we work below with three types of shells:

- $S^\rho$, where $\rho > 0$ - shells which elements are derived by solving optimization problem (3.1) for various $\lambda$ and that $\rho$,
- $S^0$ - shells which elements are derived by solving optimization problem (3.3) for various $\lambda$,
- $S^{+\infty}$ - shells which elements are derived by solving optimization problem (3.4) for various $\lambda$.

Shells $S^\rho$ and $S^{+\infty}$ are composed of properly efficient outcomes, whereas shells $S^0$ are composed of weakly efficient outcomes.

Given vector $\lambda$, $\lambda_i > 0,\ i = 1, ..., k$, and parameter $\rho > 0$, let $\bar{y}$ denote a properly efficient outcome which solves optimization problem (3.1) for that $\lambda$ and that $\rho$. To stress the association between $\lambda$ and $\bar{y}$ we denote $\lambda = \lambda(\bar{y})$.

REMARK 1 It is easy to show that $\bar{y}$ solves optimization problem (3.1) also for the same $\rho$ and for $\bar{\lambda}(\bar{y})$, where

$$\bar{\lambda}(\bar{y})_i = ((y_i^* - \bar{y}_i) + \rho e^k(y^* - \bar{y}))^{-1},\ i = 1, ..., k. \tag{7.5}$$

For $\bar{\lambda}(\bar{y})$ we clearly have, by the definition of $y^*$,

$$\bar{\lambda}(\bar{y})_i > 0,\ i = 1, ..., k,$$

and

$$\bar{\lambda}(\bar{y})_i((y_i^* - \bar{y}_i) + \rho e^k(y^* - \bar{y})) = 1,\ \ i = 1, ..., k.$$

Vectors of weights $\bar{\lambda}(\bar{y})$ are used in upper bound formulas given below.

To show that outcome $\bar{y}$ solves optimization problem (3.1) with $\bar{\lambda}(\bar{y})$ defined by (7.5) observe that since it solves optimization problem (3.1) for some $\lambda > 0$ and $\rho > 0$, we have

$$\min_{y \in Z} \max_i \lambda_i((y_i^* - y_i) + \rho e^k(y^* - y))$$

$$= \max_i \lambda_i((y_i^* - \bar{y}_i) + \rho e^k(y^* - \bar{y})).$$

Thus, for any $y \in Z$ there exists index $l,\ l \in \{1, ..., k\}$, such that

$$\max_i \lambda_i((y_i^* - y_i) + \rho e^k(y^* - y))$$

$$= \lambda_l((y_l^* - y_l) + \rho e^k(y^* - y))$$

$$\geq \lambda_l((y_l^* - \bar{y}_l) + \rho e^k(y^* - \bar{y})).$$

The last inequality remains valid if $\lambda$ is replaced by $\bar{\lambda}(\bar{y})$. Moreover, since $\bar{\lambda}(\bar{y})_i((y_i^* - \bar{y}_i) + \rho e^k(y^* - \bar{y})) = 1$ for all $i = 1, ..., k$, then

$$\bar{\lambda}(\bar{y})_l((y_l^* - y_l) + \rho e^k(y^* - y))$$

$$\geq \bar{\lambda}(\bar{y})_l((y_l^* - \bar{y}_l) + \rho e^k(y^* - \bar{y})) = 1,$$

and

$$\max_i \bar{\lambda}(\bar{y})_i((y_i^* - y_i) + \rho e^k(y^* - y))$$

$$\geq 1 = \max_i \bar{\lambda}(\bar{y})_i((y_i^* - \bar{y}_i) + \rho e^k(y^* - \bar{y})).$$

Since the above relation holds for every $y \in Z$, then

$$\min_{y \in Z} \max_i \bar{\lambda}(\bar{y})_i((y_i^* - y_i) + \rho e^k(y^* - y))$$

$$= \max_i \bar{\lambda}(\bar{y})_i((y_i^* - \bar{y}_i) + \rho e^k(y^* - \bar{y})),$$

hence $\bar{y}$ solves (3.1) with $\bar{\lambda}(\bar{y})$. □

Suppose that $\rho > 0$ and shell $S^\rho$, $|S^\rho| \geq 1$, derived by solving optimization problem (3.1) with that $\rho$ is given. When optimization problem (3.1) is used to derive (properly) efficient outcomes we have the following upper bounding formula

$$y(\lambda)_i \leq U(S^\rho, \lambda)_i$$

$$= \min\{\min_{\bar{y} \in S^\rho}[\min_{l \in I(\lambda)}\{y_l^* + \tfrac{\rho}{1+\rho}\textstyle\sum_{j \neq l}^k y_j^* - \bar{\lambda}(\bar{y})_l^{-1}(1+\rho)^{-1}$$

$$-\tfrac{\rho}{1+\rho}\textstyle\sum_{j \neq l}^k L(\lambda)_j\}]\}, \bar{U}_i\},\ i = 1, ..., k, \tag{7.6}$$

where $I(\lambda)$ is a subset of indices $\{1, 2, ..., k\}$ such that if $l \in I(\lambda)$ then $t^l = \min\{t^1, ..., t^k\}$, where

$$t^i = ((\tau_i + \rho e^k \tau)\bar{\lambda}(\bar{y})_i)^{-1},$$

$\tau$ is defined by formula (5.2), and $L(\lambda)_j$ are such that $y(\lambda)_j \geq L(\lambda)_j$, $j = 1, ..., k,\ j \neq l$. One possible selection of $L(\lambda)_j$ is $\bar{L}_j,\ j = 1, ..., k,\ j \neq l$.

Unfortunately, problem (3.2) does not provide for an upper bounding formula in a closed-end form.

Suppose that shell $S^0$, $|S^0| \geq 1$, is given. When optimization problem (3.3) is used to derive (weakly) efficient outcomes we obtain the upper bounding formula by putting $\rho = 0$ into formula (7.6), which results in

$$y(\lambda)_i \leq U(S^0, \lambda)_i$$

$$= \min\{\min_{\bar{y} \in S^0}\{\min_{l \in I(\lambda)}(y_l^* - \bar{\lambda}(\bar{y})_l^{-1}\}, \bar{U}_i\}, \tag{7.7}$$

$$i = 1, ..., k\,.$$

Finally, suppose that shell $S^{+\infty}$, $|S^{+\infty}| \geq 1$, is given. When optimization problem (3.4) is used to derive (properly) efficient outcomes we have the following upper bounding formula

$$y(\lambda)_i \leq U(S^{+\infty}, \lambda)_i$$

$$= \min\{\min_{\bar{y} \in S^{+\infty}}(\bar{y}_i + \lambda(\bar{y})_i^{-1} \textstyle\sum_{j \neq i}^k \lambda(\bar{y})_j(\bar{y}_j - L(\lambda)_j)), \bar{U}_i\}, \tag{7.8}$$

$$i = 1, ..., k,$$

where $L(\lambda)_j,\ j = 1, ..., k,\ j \neq i$, are defined as above.

As seen from formula (7.8), in this case bounds on outcome components do not depend on $\lambda$ but only on bounds on other components. Therefore, in the case of $L(\lambda)_j = \bar{L}_j,\ j = 1, ..., k,\ j \neq i$, bounds (7.8) become static with respect to weights $\lambda_i,\ i = 1, ..., k$, and $U(S^{+\infty}, \lambda) = U(S^{+\infty})$.

♣ To show how the above upper bounds are derived let us consider one of the Pareto set characterizations by weight manipulations presented in Section 2 of Chapter 2, namely Characterization I.

Let for some $\rho > 0,\ y^s \in S^\rho$ and let $\bar{\lambda}(y^s)$ be defined by formula (7.5). Suppose that (an implicit) $y(\lambda)$ solves optimization problem (3.1) for that $\rho$ and some vector $\lambda,\ \lambda_i > 0,\ i = 1, ..., k$. Then, by Remark 1 of this chapter, we have

$$\min_{y \in Z} \max_i \bar{\lambda}(y^s)_i((y_i^* - y_i) + \rho e^k(y^* - y))$$

$$= \max_i \bar{\lambda}(y^s)_i((y_i^* - y_i^s) + \rho e^k(y^* - y^s)) = 1$$

$$\leq \max_i \bar{\lambda}(y^s)_i((y_i^* - y(\lambda)_i) + \rho e^k(y^* - y(\lambda)).$$

Suppose that

$$\begin{aligned}&\max_i \bar{\lambda}(y^s)_i((y_i^* - y(\lambda)_i) + \rho e^k(y^* - y(\lambda)))\\ &= \bar{\lambda}(y^s)_l((y_l^* - y(\lambda)_l) + \rho e^k(y^* - y(\lambda)))\end{aligned} \tag{7.9}$$

for some $l \in \{1, ..., k\}$. From this we get

$$1 \leq \max_i \bar{\lambda}(y^s)_i((y_i^* - y(\lambda)_i) + \rho e^k(y^* - y(\lambda)))$$

$$= \bar{\lambda}(y^s)_l((y_l^* - y(\lambda)_l) + \rho e^k(y^* - y(\lambda))).$$

Finally,

$$1 = \bar{\lambda}(y^s)_l((y_l^* - y_l^s) + \rho e^k(y^* - y^s))$$

$$\leq \bar{\lambda}(y^s)_l((y_l^* - y(\lambda)_l) + \rho e^k(y^* - y(\lambda))).$$

and

$$y(\lambda)_l \leq y_l^* + \tfrac{\rho}{1+\rho} \textstyle\sum_{j \neq l}^k y_j^* - \bar{\lambda}(y^s)_l^{-1}(1+\rho)^{-1} - \tfrac{\rho}{1+\rho} \textstyle\sum_{j \neq l}^k y(\lambda)_j \, .$$

Now we have to determine indices $l$ for which (7.9) holds. Observe first that isograms of the objective function of optimization problem (3.1), i.e.

$$\max_i \lambda_i((y_i^* - y_i) + \rho e^k(y^* - y))$$

are displaced cones $K^1$, where $K^1 = \{y \in \mathcal{R}^k \mid y_i + \rho e^k y \geq 0, \; i = 1, ..., k\}$ ($K^1$ is independent of $\lambda$).

Outcome $y(\lambda)$ which solves the optimization problem (3.1) can be somewhere on the borders of displaced cones $K^1$, which have their apexes on the half line $y = y^* - t\tau$, where $t \geq 0$, and $\tau_i, \; i = 1, ..., k$, are defined by $\lambda, \; \rho$, and formula (5.2) (cf. Figure 7.5).

On the other hand, outcome $y(\lambda)$ cannot be in the area above the isogram

$$1 = \max_i \bar{\lambda}(y^s)_i((y_i^* - y_i^s) + \rho e^k(y^* - y^s)) = \max_i \bar{\lambda}(y^s)_i((y_i^* - y_i)$$

$$+\rho e^k(y^* - y)),$$

for otherwise $y^s$ would not be a solution of optimization problem (3.1) with $\lambda = \bar{\lambda}(y^s)$ (cf. Figure 7.5). Hence, (7.9) holds for indices $l \in$

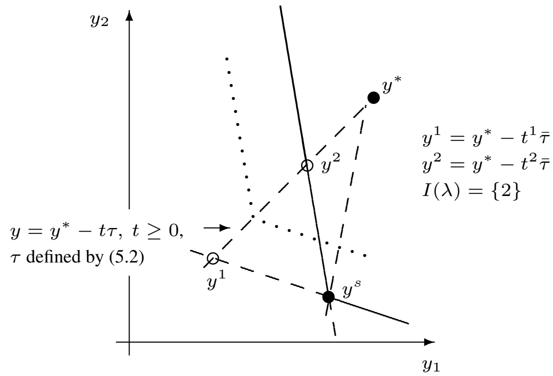


*Figure 7.5.* Determining indices satisfying relation (7.8).

$\{1, ..., k\}$, for which $t_l$ is minimal in $\{t^1, ..., t^k\}$, where $t^i,\ i = 1, ..., k$, are such that

$$y = y^* - t^i\tau,$$

$$\bar{\lambda}(y^s)_i((y_i^* - y_i) + \rho e^k(y^* - y)) = 1. \tag{7.10}$$

In other words, indices $l$ for which $t^l$ are minimal in $\{t^1, ..., t^k\}$ define the intercept of the half line $y = y^* - t\tau$, where $t \geq 0$, with the isogram of $\max_i \bar{\lambda}(y^s)_i((y_i^* - y_i) + \rho e^k(y^* - y)) = 1$ (in Figure 7.5 the intercept is denoted $y^2$).

Solving (7.10) for $t^i$ we get

$$t^i = ((\tau_i + \rho e^k\tau)\bar{\lambda}(y^s)_i)^{-1},\ i = 1, ..., k.$$

With outcomes of shell $S^\rho$, lower bounds $L(\lambda)_j$ , $j = 1, ..., k,\ j \neq i$, and upper bounds $\bar{U}_i$ , $i = 1, ..., k$, we obtain formula (7.6).

By a similar argument applied to optimization problem (3.4) we obtain formula (7.8).

Since optimization problem (3.4) with all weights equal to one is the limit of optimization problem (3.1) for $\rho \to +\infty$ (cf. Remark 1 of Chapter 3), passing in formula (7.6) with $\rho$ to the limit $+\infty$ we should obtain formula (7.8) with all weights equal to one. It is easy to check that this indeed holds.

♠

The idea of upper bounds (7.6) can be explained as shown in Figure 7.6. Properly efficient outcomes of $S^\rho = \{y(\lambda^s)\},\ s = 1, ..., |S^\rho|$, derived by solving optimization problem (3.1) with that $\rho$, are used to derive upper bounds on components of outcome $y(\lambda)$, which solves optimization problem (3.1) for that $\rho$ and that $\lambda$. The same argument applies (up to the role of parameter $\rho$) to bounds (7.7) (cf. Figure 7.7) and bounds (7.8) (cf. Figure 7.8).

The following argument explains why for $\rho = 0$ in some cases it is possible to get $L(S^0, \lambda)_i = U(S^0, \lambda)_i$ for some $\lambda$ and some $i$. Let two elements of $S^0$, $y^1$ and $y^2$, and a half line corresponding to $\lambda$ (cf. formula (5.1) and formula (5.2)) be as shown in Figure 7.9. In this case we clearly get $L(S^0, \lambda)_2 = U(S^0, \lambda)_2$ . Indeed, $y(\lambda)_2$ cannot be greater than $y^1_2 = y^2_2$ , for otherwise this would contradict weak efficiency of $y^1$. But at the same time $y(\lambda)_2$ cannot be lower than $y^1_2 = y^2_2$ for otherwise it would contradict weak efficiency of $y(\lambda)$. Hence, the above equality of bounds must hold.

♣

REMARK 2 When deriving upper bounds (7.6) and (7.7) on components of (properly or weakly, respectively) efficient outcomes we make use of the fact that the relations $\bar{\lambda}(\bar{y})_i > 0,\ i = 1, ..., k$, hold for any $y \in Z$. It is easy to observe that upper bound formulas remain valid for any $y^{ref}$ replacing $y^*$, provided that relations

$$(y^{ref}_i - y_i) + \rho e^k(y^{ref} - y) > 0,\ i = 1, ..., k, \qquad (7.11)$$

hold for any $y \in Z$. However, for given $y^{ref}$ and given $\rho \geq 0$ relations (7.11) need not, in general, to hold (cf. Figure 7.10).

To resolve this we can proceed as follows. Observe that for all reference points different from $y^{ref}$ but lying on the half line

$$\{y^{apex} \mid y^{apex} = y^{ref} - t\tau,\ \ t \text{ unconstrained in sign}\},$$

where $\tau$ is given by formula (5.2), the achievement function (3.6) yields the same efficient outcomes.

Thus, to derive the same efficient outcomes as with $y^{ref}$ but to have relation (7.11) satisfied, it suffices to select an appropriate element $y^{ref'}$

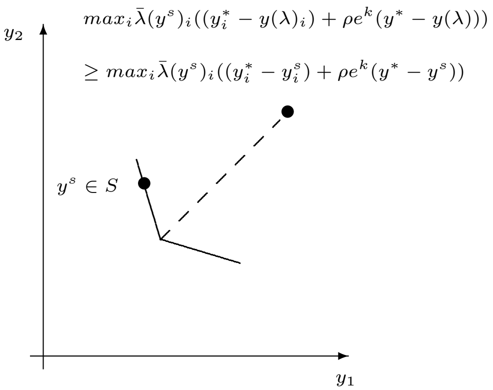


*Figure 7.6.* A graphical interpretation of upper bounds when optimization problem (3.1) is used to derive (properly) efficient outcomes.

such that it lies on the half line. This is illustrated in Figure 7.10 for the case $y^{ref} \notin Z$.

The relations (7.11) are clearly satisfied if $y^{ref'}$ is selected such that the conditions

$$(y_i^{ref} - \bar{U}_i) + \rho e^k(y^{ref} - \bar{U}) > 0, \ i = 1, ..., k, \tag{7.12}$$

hold. Hence, the above relation can be used an indicator for a proper selection of $y^{ref'}$.

We account for this observation when solving a practical problem in Section 5 of Chapter 9 . □

♠

## 6. Parametric Bound Dynamics

It follows from the form of the derived bounds that tightness of the lower (upper) bounds for component $i$ of a (properly or weakly, respectively) efficient outcome depends on $\lambda$ (except bound (7.8)), cardinality of the shell, and in case of bounds (7.1), (7.2), (7.4) ((7.6), (7.8)), on the tightness of upper (lower) bounds on components $j = 1, ..., k, \ i \neq j$.

We can always improve (precisely: not worsen) lower (upper) bounds by including more outcomes to the shell or by tightening complementary

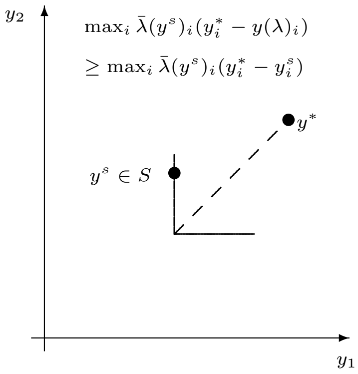


*Figure 7.7.* A graphical interpretation of upper bounds when optimization problem (3.3) is used to derive (properly) efficient outcomes.

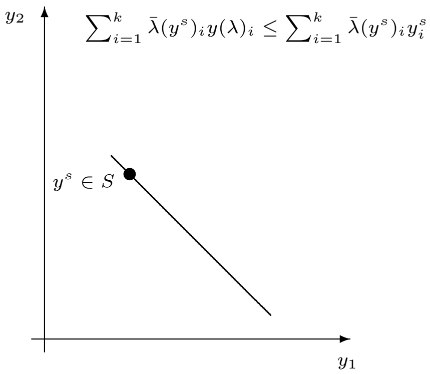


*Figure 7.8.* A graphical interpretation of upper bounds when optimization problem (3.4) is used to derive (properly) efficient outcomes.

upper (lower) bounds. Each of these actions is reflected in what we call *bound external dynamics* and *bound internal dynamics*, respectively.

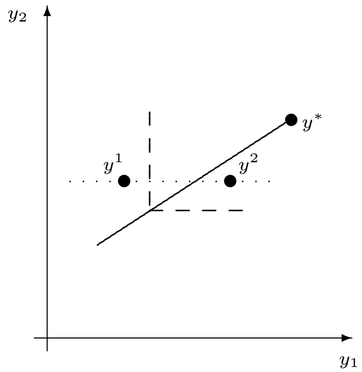


*Figure 7.9.* An explanation for the case where lower and upper bounds are equal.

## 7. Numerical Example

To get an idea whether bounds we derived above are practical and worth "the trouble", we perform now a small but illustrative numerical experiment.

EXAMPLE 1 We assume $Z$ is a sphere, $Z = \{y \in \mathcal{R}^3 \mid y_1^2 + y_2^2 + y_3^2 \leq 49\}$. Clearly, the set of efficient outcomes for $Z$ is $\{y \in \mathcal{R}^3 \mid y_1^2 + y_2^2 + y_3^2 = 49,\ y_i \geq 0,\ i = 1,2,3\}$, and $\bar{U}_i = \hat{y}_i = 7,\ \bar{L}_i = 0,\ i = 1,2,3$. To get an insight on quality of bounds we derived let us consider, quite arbitrarily, all outcomes which are properly efficient and have positive integer coefficients $y_1$ and $y_2$. In this way we get a fair dispersion of selected properly efficient outcomes over $Z$. There are 30 such outcomes where components are given by the following formula

$$y_1 \text{ and } y_2 \text{ - integer,}$$

$$y_i \geq 1,\ i = 1,2,$$

$$y_1^2 + y_2^2 \leq 49, \quad y_3 = \sqrt{49 - y_1^2 - y_2^2}\,.$$

We denote these outcomes $y^s,\ s = 1, ..., 30$.

Below we test bound tightness in two extremal cases, namely for the case $\rho = +\infty$ and $\rho = 0$. In Chapter 9 a numerical example is solved where bounds are calculated for some positive but finite $\rho$.

To calculate lower bounds we use also outcomes $\tilde{y}^1 = (7, 0, 0)$, $\tilde{y}^2 = (0, 7, 0)$, $\tilde{y}^3 = (0, 0, 7)$, (not properly efficient), which have to be derived anyway to get $\hat{y}$. Thus, for calculating lower bounds for each $y^s$ there are 32 outcomes available to serve as a specific shell.

## Case $\rho = +\infty$

We consider bounds (7.4) and (7.8) which correspond to the case when optimization problem (3.4) is used to derive shell elements.

In our example, at each properly efficient outcome there is a unique tangent hyperplane with positive coefficients. Since $Z$ is a sphere, vector $\lambda^s,\ \lambda_i^s > 0,\ i = 1, ..., k$, for which outcome $y(\lambda^s),\ s = 1, ..., 30$, solves (uniquely) optimization problem (3.4) is given by $y^s - 0$. Therefore, all shells composed of elements $y^s$ can be regarded $S^{+\infty}$ type.

### Testing Bound Tightness

Here we calculate bounds for all outcomes $y^s,\ s = 1, ..., 30$, and verify the quality of such bounds against the actual $y^s$. We assume that for each outcome $y^s$ all remaining 29 outcomes are available to serve for that $y^s$ as shell elements.

Recall (cf. bounding formula (7.8) that in this case upper bounds do not depend on $\lambda$, so $U(S^{+\infty}, \lambda) = U(S^{+\infty})$.

To calculate $L(S^{+\infty}, \lambda)_i$ with formula (7.4) we need $U(\lambda)_j\,,\ j = 1, ..., k,\ j \neq i$. Analogously, to calculate $U(S^{+\infty})_i$ with formula (7.8) we need $L(\lambda)_j\,,\ j = 1, ..., k,\ j \neq i$. To this aim, for each outcome $y^s$ (and the corresponding $\lambda^s$) we assume two settings of upper bounds $U(\lambda)_i\,,\ i = 1, 2, 3$, namely

- setting 1: $U(\lambda^s)_i = \bar{U}_i = \hat{y}_i\,,\ i = 1, 2, 3$, (the most loose upper bounds),

- setting 2: $U(\lambda^s)_i = y_i^s\,,\ i = 1, 2, 3$, (the most tight upper bounds).

Symmetrically, for each outcome $y^s$ (and the corresponding $\lambda^s$) we assume two settings of lower bounds $L(\lambda)_i,\ i = 1, 2, 3$, namely

- setting 1: $L(\lambda^s)_i = \bar{L}_i = 0\,,\ i = 1, 2, 3$, (the most loose lower bounds),

- setting 2: $L(\lambda^s)_i = y_i^s$ , $i = 1, 2, 3$, (the most tight lower bounds).

The results of calculations of bounds (7.4) are given in Table 7.1. Column 1 contains outcome numbers. Columns 2,3,4 contain components of $y(\lambda^s),\ s = 1, ..., 30$, columns 5,6,7 contain lower bounds for the upper bound setting 1, and columns 8,9,10 contain lower bounds for upper bound setting 2.

The results of calculations of bounds (7.8) are given in Table 7.2. Column 1 contains outcome numbers. Columns 2,3,4 contain components of $y(\lambda^s),\ s = 1, ..., 30$, columns 5,6,7 contain upper bounds for lower bound setting 2. For lower bound setting 1 and for all 30 outcomes upper bounds are equal to $\bar{U}_1 = \bar{U}_2 = \bar{U}_3 = 7$, i.e. bounds are not tighter than upper bounds provided by $\hat{y}$. Therefore, they are not included in Table 7.2.

**Testing Bound External Dynamics**

To get some more insight into the mechanics of bounds it is advisable to continue our experiment. This time we want to investigate bound external dynamics, i.e. the effect of adding elements to shells. To this aim we calculate bounds for arbitrary selected outcome no. 30 (outcome numbering as in Table 7.1 and Table 7.2). We start with shell $S^{+\infty} = \{[15]\}$, where $[i]$ denotes outcome numbered $i$ (i.e. $[15] = (3, 3, 5.57)$), and add to the shell at each iteration one outcome $y^s$ at time in a random order.

The results are given in Table 7.3. The columns of the table contain: column 1 - iteration number, column 2 - outcome number, columns 3,4,5 - components of outcome indicated in column 2, columns 6,7,8 contain lower bounds and columns 9,10,11 upper bounds on $[30]$ after the successive outcome is added to the shell.

This time we use the lower bound and upper bound settings 2 only.

After 8 iterations bound values stabilize and the successive adding of outcomes $[i],\ i \neq 30$, to the shell has no effect on them.

Clearly, bound external dynamics for a specified outcome strongly depends on the order in which outcomes are added to the shell.

**Testing Bound Internal Dynamics**

It is shown earlier in this section (by calculating bounds with bound setting 1 and bound setting 2) that bounds (7.4) and (7.8) exhibit internal dynamics. However, we not observe, at least in the example, bound

*Table 7.1.* Testing bound (7.4) tightness in Example 1 of Chapter 7 ($\rho = +\infty$).

| 1<br>No. | 2<br>$y_1$ | 3<br>$y_2$ | 4<br>$y_3$ | 5<br>$L_1$ | 6<br>$L_2$ | 7<br>$L_3$ | 8<br>$L_1$ | 9<br>$L_2$ | 10<br>$L_3$ |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 6.86 | 0.00 | 0.00 | 5.03 | 0.48 | 0.48 | 6.78 |
| 2 | 1 | 2 | 6.63 | 0.00 | 0.00 | 4.14 | 0.48 | 1.74 | 6.55 |
| 3 | 1 | 3 | 6.24 | 0.00 | 0.00 | 3.28 | 0.47 | 2.82 | 6.16 |
| 4 | 1 | 4 | 5.66 | 0.00 | 0.47 | 2.38 | 0.46 | 3.87 | 5.56 |
| 5 | 1 | 5 | 4.80 | 0.00 | 1.58 | 1.34 | 0.45 | 4.89 | 4.68 |
| 6 | 1 | 6 | 3.46 | 0.00 | 2.86 | 0.00 | 0.39 | 5.90 | 3.29 |
| 7 | 2 | 1 | 6.63 | 0.00 | 0.00 | 4.14 | 1.74 | 0.48 | 6.55 |
| 8 | 2 | 2 | 6.40 | 0.00 | 0.00 | 3.20 | 1.74 | 1.74 | 6.32 |
| 9 | 2 | 3 | 6.00 | 0.24 | 0.00 | 0.00 | 1.72 | 2.81 | 5.91 |
| 10 | 2 | 4 | 5.39 | 0.00 | 0.00 | 1.20 | 1.73 | 3.87 | 5.29 |
| 11 | 2 | 5 | 4.47 | 0.00 | 0.63 | 0.00 | 1.72 | 4.89 | 4.35 |
| 12 | 2 | 6 | 3.00 | 0.00 | 2.23 | 0.00 | 1.70 | 5.90 | 2.80 |
| 13 | 3 | 1 | 6.24 | 0.00 | 0.00 | 3.28 | 2.82 | 0.47 | 6.16 |
| 14 | 3 | 2 | 6.00 | 0.00 | 0.00 | 2.24 | 2.82 | 1.73 | 5.91 |
| 15 | 3 | 3 | 5.57 | 0.00 | 0.00 | 1.15 | 2.80 | 2.80 | 5.46 |
| 16 | 3 | 4 | 4.90 | 0.00 | 0.00 | 0.00 | 2.79 | 3.85 | 4.77 |
| 17 | 3 | 5 | 3.87 | 0.00 | 0.04 | 0.00 | 2.77 | 4.86 | 3.70 |
| 18 | 3 | 6 | 2.00 | 0.00 | 2.17 | 0.00 | 2.67 | 5.83 | 1.50 |
| 19 | 4 | 1 | 5.66 | 0.47 | 0.00 | 2.38 | 3.87 | 0.46 | 5.56 |
| 20 | 4 | 2 | 5.39 | 0.00 | 0.00 | 1.20 | 3.87 | 1.73 | 5.29 |
| 21 | 4 | 3 | 4.90 | 0.00 | 0.00 | 0.00 | 3.85 | 2.79 | 4.77 |
| 22 | 4 | 4 | 4.12 | 0.00 | 0.00 | 0.00 | 3.80 | 3.80 | 3.93 |
| 23 | 4 | 5 | 2.83 | 0.00 | 0.04 | 0.00 | 3.75 | 4.80 | 2.47 |
| 24 | 5 | 1 | 4.80 | 1.58 | 0.00 | 1.34 | 4.89 | 0.45 | 4.68 |
| 25 | 5 | 2 | 4.47 | 0.63 | 0.00 | 0.00 | 4.89 | 1.72 | 4.35 |
| 26 | 5 | 3 | 3.87 | 0.04 | 0.00 | 0.00 | 4.86 | 2.77 | 3.70 |
| 27 | 5 | 4 | 2.83 | 0.04 | 0.00 | 0.00 | 4.80 | 3.75 | 2.47 |
| 28 | 6 | 1 | 3.46 | 2.86 | 0.00 | 0.00 | 5.90 | 0.39 | 3.29 |
| 29 | 6 | 2 | 3.00 | 0.18 | 0.00 | 0.00 | 5.00 | 3.00 | 3.87 |
| 30 | 6 | 3 | 2.00 | 2.17 | 0.00 | 0.00 | 5.83 | 2.67 | 1.50 |

cycling, i.e. $U(S^{+\infty}, \lambda)$ yields $L(S^{+\infty}, \lambda)$, which yields $U^1(S^{+\infty}, \lambda)$, $U^1(S^{+\infty}, \lambda) < U(S^{+\infty}, \lambda)$, which yields $L^1(S^{+\infty}, \lambda)$, $L^1(S^{+\infty}, \lambda) > L(S^{+\infty}, \lambda)$, and so on.

*Table 7.2.* Testing bound (7.8) tightness in Example 1 of Chapter 7 ($\rho = +\infty$).

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|
| No. | $y_1$ | $y_2$ | $y_3$ | $U_1$ | $U_2$ | $U_3$ |
| 1 | 1 | 1 | 6.86 | 1.26 | 1.26 | 6.93 |
| 2 | 1 | 2 | 6.63 | 1.26 | 2.19 | 6.71 |
| 3 | 1 | 3 | 6.24 | 1.27 | 3.17 | 6.33 |
| 4 | 1 | 4 | 5.66 | 1.27 | 4.13 | 5.76 |
| 5 | 1 | 5 | 4.80 | 1.28 | 5.11 | 4.92 |
| 6 | 1 | 6 | 3.46 | 1.30 | 6.10 | 3.67 |
| 7 | 2 | 1 | 6.63 | 2.19 | 1.26 | 6.71 |
| 8 | 2 | 2 | 6.40 | 2.19 | 2.19 | 6.48 |
| 9 | 2 | 3 | 6.00 | 2.20 | 3.17 | 6.08 |
| 10 | 2 | 4 | 5.39 | 2.21 | 4.13 | 5.48 |
| 11 | 2 | 5 | 4.47 | 2.23 | 5.11 | 4.59 |
| 12 | 2 | 6 | 3.00 | 2.33 | 6.10 | 3.18 |
| 13 | 3 | 1 | 6.24 | 3.17 | 1.27 | 6.33 |
| 14 | 3 | 2 | 6.00 | 3.17 | 2.20 | 6.08 |
| 15 | 3 | 3 | 5.57 | 3.18 | 3.18 | 5.67 |
| 16 | 3 | 4 | 4.90 | 3.20 | 4.15 | 5.01 |
| 17 | 3 | 5 | 3.87 | 3.26 | 5.14 | 4.02 |
| 18 | 3 | 6 | 2.00 | 3.34 | 6.17 | 2.33 |
| 19 | 4 | 1 | 5.66 | 4.13 | 1.27 | 5.76 |
| 20 | 4 | 2 | 5.39 | 4.13 | 2.21 | 5.48 |
| 21 | 4 | 3 | 4.90 | 4.15 | 3.20 | 5.01 |
| 22 | 4 | 4 | 4.12 | 4.20 | 4.20 | 4.29 |
| 23 | 4 | 5 | 2.83 | 4.20 | 5.21 | 3.10 |
| 24 | 5 | 1 | 4.80 | 5.11 | 1.28 | 4.92 |
| 25 | 5 | 2 | 4.47 | 5.11 | 2.23 | 4.59 |
| 26 | 5 | 3 | 3.87 | 5.14 | 3.26 | 4.02 |
| 27 | 5 | 4 | 2.83 | 5.21 | 4.20 | 3.10 |
| 28 | 6 | 1 | 3.46 | 6.10 | 1.30 | 3.67 |
| 29 | 6 | 2 | 3.00 | 6.10 | 2.33 | 3.18 |
| 30 | 6 | 3 | 2.00 | 6.07 | 3.20 | 2.16 |

## Case $\rho = 0$

We continue the example and calculate bounds (7.3) and (7.7) for the same data as before.

*Table 7.3.* Testing bound (7.4) and bound (7.8) external dynamics for outcome no. 30 in Example 1 of Chapter 7 ($\rho = +\infty$).

| 1 Iteration | 2 No. | 3 $y_1$ | 4 $y_2$ | 5 $y_3$ | 6 $L_1$ | 7 $L_2$ | 8 $L_3$ | 9 $U_1$ | 10 $U_2$ | 11 $U_3$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 15 | 3 | 3 | 5.57 | 4.19 | 0.00 | 0.00 | 7.00 | 7.00 | 4.11 |
| 2 | 19 | 4 | 1 | 5.66 | 4.22 | 0.00 | 0.00 | 7.00 | 7.00 | 3.82 |
| 3 | 11 | 2 | 5 | 4.47 | 4.22 | 0.00 | 0.00 | 7.00 | 5.10 | 3.82 |
| 4 | 22 | 4 | 4 | 4.12 | 5.21 | 0.77 | 0.00 | 7.00 | 4.19 | 3.15 |
| 5 | 24 | 5 | 1 | 4.80 | 5.21 | 0.83 | 0.00 | 6.77 | 4.19 | 2.82 |
| 6 | 18 | 3 | 6 | 2.00 | 5.21 | 0.83 | 0.00 | 6.77 | 4.19 | 2.82 |
| 7 | 26 | 5 | 3 | 3.87 | 5.62 | 2.25 | 0.87 | 6.36 | 3.60 | 2.49 |
| 8 | 29 | 6 | 2 | 3.00 | 5.83 | 2.67 | 1.50 | 6.07 | 3.20 | 2.17 |
| 9 | 10 | 2 | 4 | 5.39 | 5.83 | 2.67 | 1.50 | 6.07 | 3.20 | 2.17 |
| . | . | . | . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . | . | . | . |
| 29 | 6 | 1 | 6 | 3.46 | 5.83 | 2.67 | 1.50 | 6.07 | 3.20 | 2.17 |
| 30 | 30 | 6 | 3 | 2.00 | 6.00 | 3.00 | 2.00 | 6.00 | 3.00 | 2.00 |

As shown in Remark 1 of this chapter, vectors of weights $\lambda^s,\ \lambda_i^s > 0,\ i = 1, ..., k$, for which outcome $y(\lambda^s),\ s = 1, ..., 30$, solves optimization problem (3.3) is given by $(y_i^* - y_i^s)^{-1}$ (cf. formula (7.5)). Therefore, all shells composed of elements $y^s,\ s = 1, ..., 30$, can be regarded as $S^0$ type shells.

We arbitrarily assume $y^* = (8, 8, 8)$.

**Testing Bound Tightness**

The results of calculations of bounds (7.3) and bounds (7.7) are given in Table 7.4. Column 1 contains outcome numbers. Columns 2,3,4 contain components of $y^s,\ s = 1, ..., 30$, columns 5,6,7 contain lower bounds, columns 8,9,10 contain upper bounds.

It is interesting to note that bounds (7.3) and bounds (7.7) are uniformly tighter than bounds (7.4) and bounds (7.8) calculated for lower and upper bound setting 1.

*Table 7.4.* Testing bound (7.3) and bound (7.7) tightness in Example 1 of Chapter 7 ($\rho = 0$).

| 1<br>No. | 2<br>$y_1$ | 3<br>$y_2$ | 4<br>$y_3$ | 5<br>$L_1$ | 6<br>$L_2$ | 7<br>$L_3$ | 8<br>$U_1$ | 9<br>$U_2$ | 10<br>$U_3$ |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 6.86 | 0.00 | 0.00 | 6.69 | 2.00 | 2.00 | 7.00 |
| 2 | 1 | 2 | 6.63 | 0.00 | 1.00 | 6.41 | 2.00 | 3.00 | 6.86 |
| 3 | 1 | 3 | 6.24 | 0.02 | 2.30 | 6.00 | 2.00 | 4.00 | 6.63 |
| 4 | 1 | 4 | 5.66 | 0.19 | 3.54 | 5.39 | 2.00 | 5.00 | 6.00 |
| 5 | 1 | 5 | 4.80 | 0.29 | 4.70 | 4.47 | 2.00 | 6.00 | 5.39 |
| 6 | 1 | 6 | 3.46 | 0.28 | 5.80 | 3.00 | 2.00 | 7.00 | 4.12 |
| 7 | 2 | 1 | 6.63 | 1.00 | 0.00 | 6.41 | 3.00 | 2.00 | 6.86 |
| 8 | 2 | 2 | 6.40 | 1.00 | 1.00 | 6.14 | 3.00 | 3.00 | 6.63 |
| 9 | 2 | 3 | 6.00 | 1.00 | 2.17 | 5.67 | 3.00 | 4.00 | 6.24 |
| 10 | 2 | 4 | 5.39 | 1.00 | 3.33 | 4.95 | 3.00 | 5.00 | 5.66 |
| 11 | 2 | 5 | 4.47 | 1.00 | 4.50 | 3.88 | 3.00 | 6.00 | 4.80 |
| 12 | 2 | 6 | 3.00 | 1.00 | 5.67 | 2.17 | 3.00 | 7.00 | 3.46 |
| 13 | 3 | 1 | 6.24 | 2.30 | 0.02 | 6.00 | 4.00 | 2.00 | 6.63 |
| 14 | 3 | 2 | 6.00 | 2.17 | 1.00 | 5.67 | 4.00 | 3.00 | 6.24 |
| 15 | 3 | 3 | 5.57 | 2.00 | 2.00 | 5.08 | 4.00 | 4.00 | 6.00 |
| 16 | 3 | 4 | 4.90 | 2.00 | 3.20 | 4.28 | 4.00 | 5.00 | 5.39 |
| 17 | 3 | 5 | 3.87 | 2.00 | 4.40 | 3.05 | 4.00 | 6.00 | 4.47 |
| 18 | 3 | 6 | 2.00 | 2.00 | 5.60 | 0.80 | 4.00 | 7.00 | 3.00 |
| 19 | 4 | 1 | 5.66 | 3.54 | 0.19 | 5.39 | 5.00 | 2.00 | 6.00 |
| 20 | 4 | 2 | 5.39 | 3.33 | 1.00 | 4.95 | 5.00 | 3.00 | 5.66 |
| 21 | 4 | 3 | 4.90 | 3.20 | 2.00 | 4.28 | 5.00 | 4.00 | 5.39 |
| 22 | 4 | 4 | 4.12 | 3.00 | 3.00 | 3.15 | 5.00 | 5.00 | 4.90 |
| 23 | 4 | 5 | 2.83 | 3.00 | 4.25 | 1.54 | 5.00 | 6.00 | 3.87 |
| 24 | 5 | 1 | 4.80 | 4.70 | 0.29 | 4.47 | 6.00 | 2.00 | 5.39 |
| 25 | 5 | 2 | 4.47 | 4.50 | 1.00 | 3.88 | 6.00 | 3.00 | 4.80 |
| 26 | 5 | 3 | 3.87 | 4.40 | 2.00 | 3.05 | 6.00 | 4.00 | 4.47 |
| 27 | 5 | 4 | 2.83 | 4.25 | 3.00 | 1.54 | 6.00 | 5.00 | 3.87 |
| 28 | 6 | 1 | 3.46 | 5.80 | 0.28 | 3.00 | 7.00 | 2.00 | 4.12 |
| 29 | 6 | 2 | 3.00 | 5.67 | 1.00 | 2.17 | 7.00 | 3.00 | 3.46 |
| 30 | 6 | 3 | 2.00 | 5.60 | 2.00 | 0.80 | 7.00 | 4.00 | 3.00 |

**Testing Bound External Dynamics**

Testing external dynamics of bound (7.3) and bound (7.7) is performed in the same way as for bounds (7.4) and (7.8). We start with shell

$S^0 = \{[15]\}$ and add to the shell at each iteration one outcome $y^s$ at time in the same order as when testing external dynamics for bound (7.4) and bound (7.8).

The results of testing are given in Table 7.5. The columns of the table contain: column 1 - iteration number, column 2 - outcome number, columns 3,4,5 - components of outcome indicated in column 2, columns 6,7,8 contain lower bounds and columns 9,10,11 upper bounds on $[30]$ after the successive outcome is added to the shell.

*Table 7.5.* Testing bound (7.3) and bound (7.7) external dynamics for outcome no. 30 in Example 1 of Chapter 7 ($\rho = 0$).

| 1<br>Iteration | 2<br>No. | 3<br>$y_1$ | 4<br>$y_2$ | 5<br>$y_3$ | 6<br>$L_1$ | 7<br>$L_2$ | 8<br>$L_3$ | 9<br>$U_1$ | 10<br>$U_2$ | 11<br>$U_3$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 15 | 3 | 3 | 5.57 | 3.00 | 0.00 | 0.00 | 7.00 | 7.00 | 5.77 |
| 2 | 19 | 4 | 1 | 5.66 | 4.00 | 0.00 | 0.00 | 7.00 | 7.00 | 5.66 |
| 3 | 11 | 2 | 5 | 4.47 | 4.00 | 0.00 | 0.00 | 7.00 | 7.00 | 5.66 |
| 4 | 22 | 4 | 4 | 4.12 | 4.00 | 0.00 | 0.00 | 7.00 | 7.00 | 5.66 |
| 5 | 24 | 5 | 1 | 4.80 | 5.00 | 0.50 | 0.00 | 7.00 | 7.00 | 5.66 |
| 6 | 18 | 3 | 6 | 2.00 | 5.00 | 0.50 | 0.00 | 7.00 | 6.00 | 5.66 |
| 7 | 26 | 5 | 3 | 3.87 | 5.00 | 0.50 | 0.00 | 7.00 | 6.00 | 3.87 |
| 8 | 29 | 6 | 2 | 3.00 | 5.60 | 2.00 | 0.80 | 7.00 | 6.00 | 3.00 |
| 9 | 10 | 2 | 4 | 5.39 | 5.60 | 2.00 | 0.80 | 7.00 | 6.00 | 3.00 |
| 10 | 13 | 3 | 1 | 6.24 | 5.60 | 2.00 | 0.80 | 7.00 | 6.00 | 3.00 |
| 11 | 1 | 1 | 1 | 6.86 | 5.60 | 2.00 | 0.80 | 7.00 | 6.00 | 3.00 |
| 12 | 28 | 6 | 1 | 3.46 | 5.60 | 2.00 | 0.80 | 7.00 | 6.00 | 3.00 |
| 13 | 16 | 3 | 4 | 4.90 | 5.60 | 2.00 | 0.80 | 7.00 | 6.00 | 3.00 |
| 14 | 12 | 2 | 6 | 3.00 | 5.60 | 2.00 | 0.80 | 7.00 | 6.00 | 3.00 |
| 15 | 2 | 1 | 2 | 6.63 | 5.60 | 2.00 | 0.80 | 7.00 | 6.00 | 3.00 |
| 16 | 8 | 2 | 2 | 6.40 | 5.60 | 2.00 | 0.80 | 7.00 | 6.00 | 3.00 |
| 17 | 25 | 5 | 2 | 4.47 | 5.60 | 2.00 | 0.80 | 7.00 | 6.00 | 3.00 |
| 18 | 3 | 1 | 3 | 6.24 | 5.60 | 2.00 | 0.80 | 7.00 | 6.00 | 3.00 |
| 19 | 27 | 5 | 4 | 2.83 | 5.60 | 2.00 | 0.80 | 7.00 | 4.00 | 3.00 |
| 20 | 9 | 2 | 3 | 6.00 | 5.60 | 2.00 | 0.80 | 7.00 | 4.00 | 3.00 |
| . | . | . | . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . | . | . | . |
| 29 | 6 | 1 | 6 | 3.46 | 5.60 | 2.00 | 0.80 | 7.00 | 4.00 | 3.00 |
| 30 | 30 | 6 | 3 | 2.00 | 6.00 | 3.00 | 2.00 | 6.00 | 3.00 | 2.00 |

After 19 iterations bound values stabilize and the successive adding of outcomes $[i],\ i \neq 30$, to the shell has no effect on them.

Clearly, also in the case $\rho = 0$ bound external dynamics strongly depends on the order in which outcomes are added to the shell.

**Testing Bound Internal Dynamics**

Bounds (7.3) and (7.6) are not interrelated, hence in this case bound internal dynamics does not occur. □

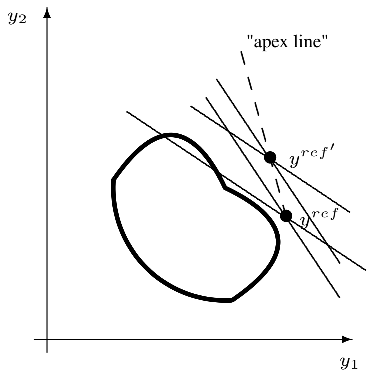


*Figure 7.10.* An illustration to Remark 2 of Chapter 7.

## 8. Parametric Bounds When $k = 2$ And $Z$ Is $R^k_+$-convex

Assuming that optimization problem (3.3) is used to derive (weakly) efficient outcomes, a more straightforward method to calculate parametric bounds can be proposed when $k = 2$ and $Z$ is $R^k_+$-convex.

If $Z$ is $R^k_+$-convex, then under some minor assumptions which we discuss below, deriving efficient outcomes by solving optimization problem (3.3) is equivalent to finding the intercept of half line $y = y^* - \tau t,\ t \geq 0$, with $Z$, where $\tau_i = \lambda_i^{-1},\ i = 1, ..., k$, (cf. Figure 7.11 and formula (5.5)).

Suppose that shell $S^0$ of $Z$ is derived by solving optimization problem (3.4) for $n$ vectors $\lambda^s,\ \lambda_i^s > 0,\ i = 1, ..., k,\ s = 1, ..., n.$ In other words, $S^0 = \{y(\lambda^s)\},\ s = 1, ..., n$, where $y(\lambda^s)$ is a solution of

$$\max_{y \in Z} \lambda^s y.$$

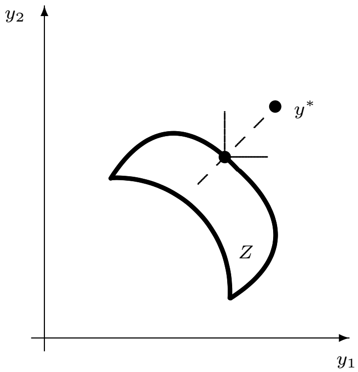


*Figure 7.11.* An interpretation of solving optimization problem (3.3) when the Pareto set is $R^k_+$-convex.

Hence, $S^0 = S^{+\infty}$.

Elements of $S^0$ can be ordered, increasingly or decreasingly, with respect to the value of, for example, the first component. Pairs of elements of $S^{+\infty}$ which correspond to two successive values of this component are called *neighbor elements*. Neighbor elements define $n-1$ *cutting lines*

$$y = \alpha y' + (1-\alpha)y'',$$

where $\alpha \in \mathcal{R}$ is any number and $y',\ y''$ are neighbor elements of $S^0$.

A lower bound $L(S^0, \lambda)$ for $y(\lambda)$ is given by the intercept of half line $y = y^* - \tau t,\ t \geq 0$, with that cutting line for which $t$ has the largest value (cf. Figure 7.12, a lower bound marked by square).

Indeed, since $Z$ is $R^k_+$-convex, no efficient outcome can be found "below" (in the sense as $y^*$ lies "above") any line segment connecting neighbor elements of the shell.

Elements of $S^0$ define $n$ *supporting lines*

$$\lambda^s y = \lambda^s y(\lambda^s),\ y(\lambda^s) \in S^0.$$

An upper bound $U(S^0, \lambda)$ for $y(\lambda)$ is given by the intercept of half line $y = y^* - \tau t,\ t \geq 0$, with that supporting line for which $t$ has the largest value (cf. Figure 7.13, an upper bound marked by star).

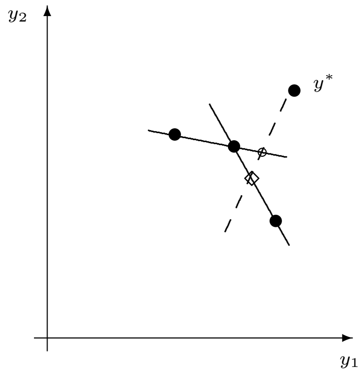


*Figure 7.12.* A simplified method to calculate lower bounds when the Pareto set is $R^k_+$-convex.

Indeed, no outcome can be found "above" (in the sense as $y^*$ lies "above") any line $\lambda^s y(\lambda^s) = \lambda^s y,\ y(\lambda^s) \in S^0$, because $y(\lambda^s)$ solves $\max_{y \in Z} \lambda^s y$.

These bounds are explicit, i.e. the bound on $y(\lambda)_i$ does not depend on values of other components of $y(\lambda)$. This contrast with implicit bounds (7.1), (7.2), (7.4), (7.6), and (7.8).

In the considered case one gets an assessment of $y(\lambda)$ in the form

$$[y(\lambda)] = L(S^0, \lambda) + t(U(S^0, \lambda) - L(S^0, \lambda)),$$

for $0 \leq t \leq 1$. Hence, "the region of uncertainty" of the assessment is a segment on half line $y = y^* - \tau t,\ t \geq 0$ (cf. Figure 7.14).

Some precautions are to be taken for half line $y = y^* - \tau t,\ t \geq 0$, not to miss $Z$, and this amounts to putting some bounds on values $\lambda_1$ and $\lambda_2$ depending on the value of parameter $\epsilon$ used to define $y^*$ (cf. Chapter 3, Subsection 2.1).

♣ Let

$$\check{y}_1 = \max_{y \in Z;\, y_2 = \hat{y}_2} y_1$$

and

$$\check{y}_2 = \max_{y \in Z;\, y_1 = \hat{y}_1} y_2.$$

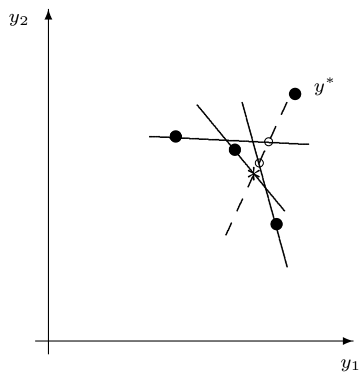


*Figure 7.13.* A simplified method to calculate upper bounds when the Pareto set is $R^k_+$-convex.

It is easy to show that under assumptions adopted in this section (namely $k = 2$, and $Z$ is $R^k_+$-convex) half line $y = y^* - \tau t,\ t \geq 0$, intercepts $Z$ only if (cf. Figure 7.15) the following conditions hold

$$\frac{\epsilon}{y_1^* - \check{y}_1} \leq \frac{\lambda_1}{\lambda_2} \leq \frac{y_2^* - \check{y}_2}{\epsilon},$$

where, we recall, $y^* = \hat{y} + \epsilon,\ \epsilon > 0$ (cf. Chapter 3, Subsection 2.1).
♠

## 9. Controlling Parametric Bound Tightness

The simplest "ad hoc" procedure to decide if a shell provides for sufficient bound tightness is as follows. For a selected element of the shell lower and upper bounds are calculated using all the remaining elements of the shell, and next bounds are compared with true values of components of that element. This procedure is repeated for all elements from the shell. In fact, this is exactly the procedure we apply to test bound tightness in Example 1 of this chapter. If bounds are not sufficiently tight (sufficient tightness is a context-dependent notion), the shell should be amended to become a more accurate discrete approximation of the Pareto set of the corresponding MCDM problem.

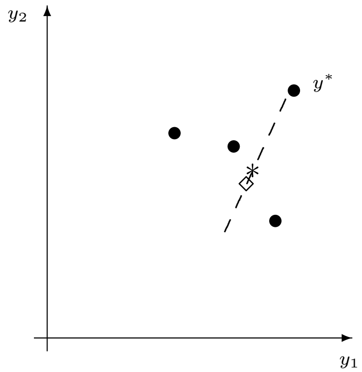


*Figure 7.14.* When the Pareto set is $R^k_+$-convex "the region of uncertainty" for bounds reduces to a line segment.

But this procedure provides information on bound tightness only at elements of a shell, and this information can be taken as only indicative for bound tightness for other efficient outcomes.

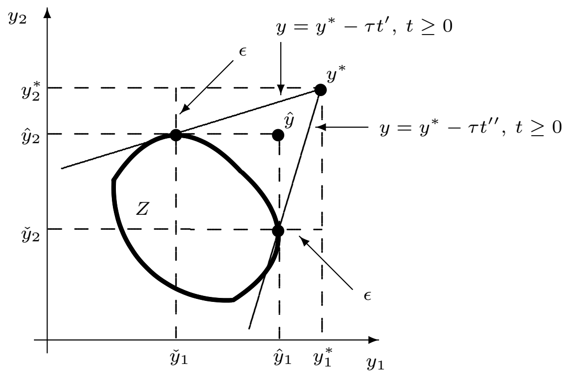


*Figure 7.15.* An illustration of admissible relations between $\lambda_1$ and $\lambda_2$ to guarantee that line $y = y^* - \tau t,\ t \geq 0$, intercepts $Z$.

♣ A more systematic procedure to control bound tightness can be proposed. Given a shell, each efficient outcome $\bar{y}$ of the shell defines two regions:

$$(\{\bar{y}\} + R_+^k) \cap \{y \,|\, \bar{L} \le y \le \bar{U}\},$$

$$(\{\bar{y}\} - R_+^k) \cap \{y \,|\, \bar{L} \le y \le \bar{U}\}.$$

The regions are the upper dead region and the lower dead region, respectively, taken with their respective borders (i.e. they are closures of respective dead regions) (cf. Section 4 of this chapter).

Clearly, no efficient outcome except $\bar{y}$ belongs to any of these two regions, for if otherwise such an outcome would dominate $\bar{y}$ (contradicting efficiency of $\bar{y}$) or would be dominated by $\bar{y}$ (contradicting its efficiency status).

Each efficient outcome $\bar{y}$ of a shell defines $2^k - 2$ *live regions*:

$$(\{\bar{y}\} + R_\pm^k) \cap \{y \,|\, \bar{L} \le y \le \bar{U}\},$$

where $R_\pm^k$ is any of sets $\{y \,|\, \bar{y}_i \sim y_i \,,\ i = 1, ..., k\}$, $\sim$ is either $\le$ or $\ge$ and $R_\pm^k \ne R_+^k$ , $R_\pm^k \ne -R_+^k$ .

Outcome $\bar{y}$ is called *root* of the live regions it defines. *Live restricted region* is the largest box (rectangular prism, in two dimensional space - the largest rectangle) in a live region, which do not contain in its interior any element of the shell. Figure 7.16 shows a live region and a live restricted region for outcome $\bar{y}$ (the root) (bullets represent shell elements).

The concept of live restricted regions provides for a way to control bound tightness. In general, bound tightness can be controlled indirectly by controlling the size (measured e.g. by $\max_i \max_{y,\,y'} (y_i - y'_i)$ over a live restricted region) of the maximal live restricted region. If for a given shell bounds are not tight enough (as detected for example by the "ad hoc" procedure described above), new elements should be included in the shell to decrease the size of the maximal live restricted region. Controlling the maximal size of live restricted regions keeps discrete representations of Pareto sets provided by shells uniform and clearly such a heuristics improves bound tightness.

Let $\epsilon_i$ define the required bound tightness with respect to index $i$, i.e.

$$U_i(S^0, \lambda) - L_i(S^0, \lambda) \le \epsilon_i \tag{7.13}$$

for any $y(\lambda) \in Z$, where $S^0$ is a shell.

For any live restricted region we can calculate $A_i = \max_{y,\,y'} |y_i - \bar{y}_i|$, where $\bar{y}$ is the root of this region, $\bar{y} \in S$. Then, it can be shown that for

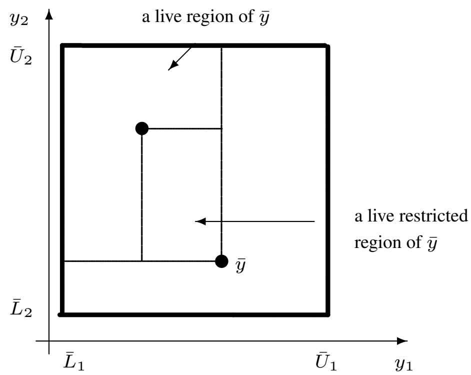


*Figure 7.16.* Root $\bar{y}$, one of its live regions, and one of its live restricted regions.

each efficient outcome $y(\lambda)$ which belongs to that live restricted region the following relation clearly holds

$$U_i(S, \lambda) - L_i(S, \lambda) \leq A_i \,.$$

Condition (7.13) is satisfied if

$$A_i = \max_{y, \bar{y}} |y_i - \bar{y}_i| \leq \epsilon_i \,, \tag{7.14}$$

for all live restricted regions. It is equivalent to say that condition (7.13) is satisfied if for each outcome $\bar{y}$ from the shell there is another outcome $\tilde{y}$ in the shell such that $|\tilde{y}_i - \bar{y}_i| \leq \epsilon_i$ .

In practice complex problems, as a rule, are formulated implicitly in terms of decisions, constraints and criteria. Hence, to ensure the required bound tightness, requirements have to be imposed explicitly on admissible decisions rather than on outcomes. This can be achieved with the help of the notion of lipschitzian functions.

A function $f : X \rightarrow \mathcal{R}$ is said to be lipschitzian if $|f(x) - f(x')| \leq \tilde{L}||x - x'||$ for any $x, x' \in X$, where $\tilde{L}$ is a constant.

Suppose that criterion function $f_i$ is lipschitzian over the set of feasible alternatives $X_0$ (cf. Chapter 2). We have

$$|f_i(x) - f_i(x')| = |y_i - y_i'| \leq \tilde{L}|x - x'| \tag{7.15}$$

for any $x, x' \in X_0$ .

To satisfy (condition 7.14) and hence also condition (7.13), it is enough to require that for each outcome $\bar{y} = f(\bar{x})$ belonging to the shell it contains another outcome $y = f(x)$ such that

$$\tilde{L}|x - \bar{x}| \leq \epsilon_i \, . \tag{7.16}$$

If for some outcome $\bar{y}$ of the shell condition (7.16) is not fulfilled, then an efficient alternative $x$ of $X_0$ which satisfies this condition has to be determined and the corresponding outcome included into the shell.

Analogous requirements apply to all the remaining indices $j = 1, ..., k$, $j \neq i$.

♠

## 10. Concluding Remarks

The ability to calculate bounds on efficient outcomes eliminates from the interactive MCDM methods the following paradox. There is a clear, but seemingly generally accepted, asymmetry between the DM and the model with respect to the accuracy of the information provided. Though it is accepted that the DM provides his preferences incomplete and in vague terms, the existing interactive methods require the model to provide exact values. With imprecise information (partial preferences) provided by the DM one would rather expect, and accept, a margin of imprecision in values provided by the model. If the size of the margin is kept under control, then imprecise information provided by the model, in principle, should neither distort the decision making processes nor impair the DM in his ability to arrive at the most preferred decision.

The paradox we mention above has a grave consequence for the practical application of interactive MCDM methods. Namely, exact values are provided by solving optimization problems. Optimization is not a barrier, technical and perceptional, only for those who have acquired at least some knowledge in this field. Such persons certainly constitute a negligible minority of all potential decision makers. Thus, optimization is certainly a barrier for widespread acceptability and popularity of MCDM methods.

The bounds presented in this chapter are of particular interest in decision making problems with large-scale underlying problems (2.2). With such problems explicit derivation of successive outcomes (decisions) may be costly and time consuming. Therefore, the number of trial outcomes derived during the decision process should be kept to a minimum. The bounds we have derived can help the DM to direct the search for successive outcomes more carefully (the DM can be provided with unlimited number of implicit outcomes for evaluations) and at negligible computation cost, as compared to optimization calculations. On the other hand, determining a shell prior to starting the decision process is just a technical matter. We elaborate this issue in detail in Chapter 9.

To summarize, bounds on outcome components presented in this book provide a breakthrough to prohibitive size and solution time bottlenecks in interactive MCDM by allowing the DM to interact with a soft representation of the underlying mathematical model rather than with the exact model itself.

The development presented in this chapter corresponds well with other recent efforts to provide the DM with low-cost tools to navigate in complex problems through a maze of efficient outcomes and respective decisions. One such tool is presented in the next chapter.

Consistently with the assumption made in Section 2 of Chapter 5, that the Universal Interface exploits optimization problem (3.1), (3.2), or (3.3), in the next chapters we make no use of bounds (7.4), (7.8). Those bounds are included into this chapter solely for the sake of completeness.

## 11. Annotated References

The observation that $y^N$ is not a valid lower bound on efficient outcomes of $Z$ comes from Weistroffer (1985).

Derivation of tight lower bounds $\bar{L}$ was studied in Korhonen et al. (1997).

The idea of parametric lower bounds on efficient outcome components can be traced down to Geromel, Ferreira (1991). In this work it is shown how to calculate lower parametric bounds with respect to the shell composed of $k$ efficient outcomes which are solutions of optimization problems $\max_{y \in Z} y_i$, $i = 1, ..., k$. The problem of parametric upper bounds is not considered there.

This chapter is based on the following works: Kaliszewski (2001a,b, 2002b, 2003a,b, 2004).

Problems of constructing Pareto set representations of required properties, pertaining to the problem of bound tightness discussed in this chapter, are considered in Steuer (1986) and, with respect to lipschitzian criteria functions, in Evtushenko, Potapov (1986).

For $k = 2$ the required bound tightness can be achieved by the methods developed for convex curve approximations, cf. Burkard et al. (1987), Fruhwirth et al. (1988), Ruhe, Fruhwirth (1989), Yang, Goh (1997).

Chapter 8

# BOUNDS ON GLOBAL TRADE-OFFS

*"So I want you, Pooh, to search by the Six Pine Trees first, and then work your way towards Owl's House ... "*

**A. A. Milne**,
*The House at Pooh Corner.*

## 1. This Chapter Is About ...

... how we can further characterize, and differentiate, efficient outcomes by using low computing-intensive tools. We re-examine the notion of global trade-off that we first discussed in Chapter 2 and then exploited in Chapter 6. Formal findings show that exploiting this notion in the framework of interactive MCDM can be made computationally inexpensive.

A lack of simple and low computing-intensive methods to calculate global trade-offs has been a barrier for their widespread use in MCDM. Below we show how this problem can be overcome.

The present chapter is the second in turn where formulas to calculate bounds on quantities characterizing outcomes are presented. Here we present formulas for bounds on outcome global trade-offs. In contrast to bounds on outcome components (cf. Chapter 7), bounds on outcome global trade-offs are static, i.e. they are not sensitive to new information acquired during interactive MCDM processes. They are, however, parametric with weights as parameters. Similarly as in the case of bounds on outcome components, computation cost to calculate bounds on outcome global trade-offs is negligible.

A global trade-off is an information specific to a designated efficient outcome. In order to calculate a global trade-off one has to derive an efficient outcome first by solving an optimization problem. This, combined with the fact that in general to calculate one global trade-off another optimization problem has to be solved, explains why using of the global trade-off information has been seldom advocated in the MCDM framework. The development presented in this chapter is believed to change this situation.

## 2. Bounds On Global Trade-offs

Computation cost of calculating exact values of global trade-offs, i.e. calculating the value of formula (2.6), is at least of the same order of magnitude as solving an optimization problem to derive an efficient outcome. Recall that there are $n \times (n-1)$ potential global trade-offs at an outcome. Hence, advantages of exploiting global trade-offs in decision making processes are easily outweighed by the disadvantage of increased computation effort. Fortunately, we can propose a way to circumvent this barrier.

In Section 3 of Chapter 6, in the $\clubsuit - -\spadesuit$ "environment", we show how to calculate the exact values of global trade-offs by a more refined method than calculating the value of global trade-off directly (i.e. by formula 2.6). However, even with this refinement, the computational effort to calculate all global trade-offs for an efficient outcome, except some small problems and perhaps some special cases, is still prohibitive.

It happens that in contrast to true values of global trade-offs, upper bounds on global trade-offs are given "free" and a priori to solving optimization problems used in characterizations of properly efficient and weakly efficient outcomes, presented in Chapter 3. Sufficient parts of these characterizations, this time accounting for global trade-offs, take the following extended forms.

Characterization I′

*Sufficient condition*

An outcome $\bar{y}$ which solves optimization problem (3.1), i.e. the problem

$$\min_{y \in Z} \max_{i} \lambda_i((y_i^* - y_i) + \rho e^k (y^* - y)),$$

where $\lambda_i > 0,\ i = 1, ..., k$, and $\rho > 0$, is properly efficient. Moreover,

$$T_{i,j}^G(\bar{y}) \leq (1+\rho)\rho^{-1} \tag{8.1}$$

for each $i, j \in \{1, ..., k\},\ i \neq j$. □

REMARK 1 It follows from Characterization I′ that if all $T_{i,j}^G(\bar{y}) \leq 1$, $i, j = 1, ..., k,\ i \neq j$, then $\bar{y}$ is a solution of optimization problem (3.1) for some vector $\lambda$ and any $\rho$. But if $T_{i,j}^G(\bar{y}) > 1$ for some $i, j,\ i \neq j$, then taking $\rho > 1/(T_{i,j}^G(\bar{y}) - 1)$ excludes $\bar{y}$ from solutions of optimization problem (3.1) solved with that $\rho$, no matter what $\lambda$ is (cf. Example 1 of this chapter). □

Characterization II′

*Sufficient condition*

An outcome $\bar{y}$ which solves optimization problem (3.2), i.e. the problem

$$\min_{y \in Z} \max_i \lambda_i(y_i^* - y_i) + \rho e^k(y^* - y),$$

where $\lambda_i > 0,\ i = 1, ..., k$, and $\rho > 0$, is properly efficient. Moreover,

$$T_{i,j}^G(\bar{y}) \leq (\lambda_i + \rho)\rho^{-1} \tag{8.2}$$

for each $i, j \in \{1, ..., k\},\ i \neq j$. □

REMARK 2 It follows from Characterization II′ that if all $T_{i,j}^G(\bar{y}) \leq 1$, $i, j = 1, ..., k,\ i \neq j$, then $\bar{y}$ is a solution of optimization problem (3.2) for some vector $\lambda$ and any $\rho$. But if $T_{i,j}^G(\bar{y}) > 1$ for some $i, j,\ i \neq j$, then selecting $\lambda$ and taking $\rho > \lambda_i/(T_{i,j}^G(\bar{y}) - 1)$ excludes $\bar{y}$ from solutions of optimization problem (3.2) solved with that $\rho$ and that $\lambda$. □

Recall that $Z$ is $R_+^k$-convex if $Z - R_+^k$ is convex.

Characterization IV′

Assume that $Z$ is $R_+^k$-convex.

*Sufficient condition*

An outcome $\bar{y}$ which solves optimization problem (3.4), i.e. problem

$$\max_{y \in Z} \lambda y,$$

where $\lambda_i > 0,\ i = 1, ..., k$, is properly efficient.

Moreover,

$$T^G_{i,j}(\bar{y}) \leq \frac{\lambda_j}{\lambda_i} \tag{8.3}$$

for each $i, j \in \{1, ..., k\},\ i \neq j$. If for some $\lambda > 0$, $\lambda y$ is a unique (up to scalar multiplication) linear function such that

$$\max_{y \in Z} \lambda y = \lambda \bar{y},$$

then, provided that $T^G_{i,j}(\bar{y}) \neq -\infty$ (i.e. global trade-off $T^G_{i,j}(\bar{y})$ is defined),

$$T^G_{i,j}(\bar{y}) = \frac{\lambda_j}{\lambda_i} \tag{8.4}$$

for all $i, j = 1, ..., k,\ \ i \neq j$.

REMARK 3 It follows from Characterization IV′ that taking $\lambda_j/\lambda_i > T^G_{i,j}(\bar{y})$ excludes $\bar{y}$ from solutions of optimization problem (3.4) solved with those $\lambda_i$ and $\lambda_j$. □

REMARK 4 Depending on the optimization problem solved, for given $\rho$ bounds on global trade-offs are either fixed (cf. (8.1)), or parametric with respect to (components of) vector $\lambda$ (cf. (8.2), (8.3), (8.4)). These observations have some consequences for the development presented in Chapter 9. □

## 3. Numerical Examples

EXAMPLE 1 Let $Z$ be given as in Figure 8.1. We have

$$\begin{array}{lll}
\text{for } y = \{a\}, & T^G_{1,2}(y) = 4, & T^G_{2,1}(y) = -\infty, \\
\text{for } y = (a,b), & T^G_{1,2}(y) = 4, & T^G_{2,1}(y) = 0.25, \\
\text{for } y = \{b\}, & T^G_{1,2}(y) = 1, & T^G_{2,1}(y) = 0.25, \\
\text{for } y = (b,c), & T^G_{1,2}(y) = 1, & T^G_{2,1}(y) = 1, \\
\text{for } y = \{c\}, & T^G_{1,2}(y) = 0.125, & T^G_{2,1}(y) = 1, \\
\text{for } y = (c,d), & T^G_{1,2}(y) = 0.125, & T^G_{2,1}(y) = 8, \\
\text{for } y = \{d\}, & T^G_{1,2}(y) = -\infty, & T^G_{2,1}(y) = 8,
\end{array}$$

where $(y', y'')$ represents the open line segment between $y'$ and $y''$, i.e. $(y', y'') = \{y \mid y = \alpha y' + (1-\alpha) y'',\ 0 < \alpha < 1\}$.

Suppose we want to derive some properly efficient outcomes but only those for which global trade-offs are bounded as follows

$$T^G_{1,2}(y) \le 2 \quad \text{and} \quad T^G_{2,1}(y) \le 4\,.$$

To this aim we can make use, for example, of Characterization II'. We set

$$\frac{\lambda_2+\rho_2}{\rho_1} \le 2 \quad \text{and} \quad \frac{\lambda_1+\rho_1}{\rho_2} \le 4\,,$$

which gives

$$\rho_1 \le \frac{\lambda_1+4\lambda_2}{14} \quad \text{and} \quad \rho_2 \le \frac{\lambda_2+2\lambda_1}{14}\,.$$

For example, for $\lambda_1 = \lambda_2 = 0.5$ we have $\rho_1 = \frac{5}{14}\,,\ \rho_2 = \frac{3}{14}\,$. According to Characterization II$'$ and as illustrated in Figure 8.1, taking $\rho_1$ and $\rho_2$ satisfying the above relations guarantees that no outcome from the segments $(a,b)\cup\{a\}$ and $(c,d)\cup\{d\}$, and all outcomes from the segment $(b,c)\cup\{b\}\cup\{c\}$ can be derived by solving

$$\min_{y\in Z} \max((\lambda_1+\rho_1)(y^*_1-y_1)+\rho_2(y^*_2-y_2)),\ (\lambda_2+\rho_2)(y^*_2-y_2)$$
$$+\rho_1(y^*_1-y_1)),$$

where $\lambda_1 > 0,\ \lambda_2 > 0.$ □

EXAMPLE 2 Let us revisit the numerical Example 1 of Chapter 7. As already observed, in the problem considered there, at each properly efficient outcome $y$ of $Z$ there is a unique tangent hyperplane with positive coefficients. Thus, to derive global trade-offs we can make use of formula (8.4), which provides exact global trade-offs. Since $Z$ is a part of a sphere, vector $\lambda^s$ , $\lambda^s_i > 0,\ i = 1,...,k,\ s = 1,...,30$, for which $y(\lambda)^s$ solves (uniquely) optimization problem (3.4) is given by $y^s - 0$.

Global trade-offs for vectors $\lambda^s$ for all 30 properly efficient outcomes $y(\lambda)^s$ considered in the example are given in Table 8.1. From formula (8.4) it follows that $T^G_{i,j}(\cdot) = (T^G_{i,j})^{-1}(\cdot)$ for $i,j = 1,2,3,\ i \ne j$, whenever $T^G_{i,j}(\cdot) \ne -\infty$ (i.e. global trade-off $T^G_{i,j}(\cdot)$ is defined). □

♣ Sufficient parts of Characterization I' and Characterization II' can be further modified to provide for more flexibility in bounding trade-offs. Let $I = \{1,...,k\}$.

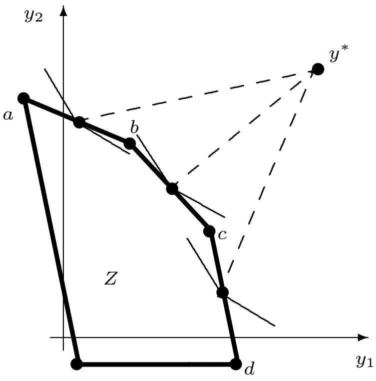


*Figure 8.1.* Selective outcome derivation in Example 1 of Chapter 8.

Characterization I$''$

*Sufficient condition*

An outcome $\bar{y}$ which solves the optimization problem

$$\min_{y \in Z} \max_{i} \lambda_i((y_i^* - y_i) + \sum_{t \in I} \rho_t(y_t^* - y_t)), \tag{8.5}$$

where $\lambda_i > 0,\ i \in I$, and $\rho_t > 0, t \in I$, is properly efficient. Moreover,

$$T_{t,i}^G(\bar{y}) \leq (1 + \rho_i)\rho_t^{-1}$$

for each $i, t \in I,\ i \neq t$. □

Characterization II$''$

*Sufficient condition*

An outcome $\bar{y}$ which solves the optimization problem

$$\min_{y \in Z} \max_{i} \lambda_i(y_i^* - y_i) + \sum_{t \in I} \rho_t(y_t^* - y_t), \tag{8.6}$$

where $\lambda_i > 0,\ i \in I$, and $\rho_t > 0, t \in I$, is properly efficient. Moreover,

$$T_{t,i}^G(\bar{y}) \leq (\lambda_i + \rho_i)\rho_t^{-1}$$

for each $i, t \in \{1, ..., k\},\ i \neq t$. □

Let $I_1 \subseteq I,\ |I_1| \geq 2,\ I_2 = I \setminus I_1$ .

Characterization I‴

*Sufficient condition*

An outcome $\bar{y}$ which solves the optimization problem

$$\min_{y \in Z} \max[\max_{i \in I_1} \lambda_i((y_i^* - y_i) + \sum_{t \in I_1} \rho_t(y_t^* - y_t)),\ \max_{i \in I_2} \lambda_i(y_i^* - y_i)], \quad (8.7)$$

where $\lambda_i > 0,\ i \in I$, and $\rho_t > 0,\ t \in I_1$ , is weakly efficient. Moreover,

$$T_{t,i}^G(\bar{y}) \leq (1 + \rho_i)\rho_t^{-1}$$

for each $i \in I_1$ , $t \in I_1$ , $t \neq i$. □

Characterization II‴

*Sufficient condition*

An outcome $\bar{y}$ which solves the optimization problem

$$\min_{y \in Z} \max[\max_{i \in I_1} \lambda_i(y_i^* - y_i) + \sum_{t \in I_1} \rho_t(y_t^* - y_t),\ \max_{i \in I_2} \lambda_i(y_i^* - y_i)], \quad (8.8)$$

where $\lambda_i > 0,\ i \in I$, and $\rho_t > 0,\ t \in I_1$ , is weakly efficient. Moreover,

$$T_{t,i}^G(\bar{y}) \leq (\lambda_i + \rho_i)\rho_t^{-1},$$

for each $i \in I_1$ , $t \in I_1$ , $t \neq i$. □

♠

# 4. Concluding Remarks

With Chapter 7 and Chapter 8 we are in possession of tools which provide, for a selected vector $\lambda$:

- bounds on efficient outcome components,

and

- bounds on efficient outcome trade-offs,

for an implicit outcome $y(\lambda)$, i.e. a (weakly or properly) efficient outcome which would be derived if a certain optimization problem were solved with that $\lambda$. Having this, we claim that we have responded to the

*Table 8.1.* Global trade-offs for Example 1 of Chapter 7.

| 1<br>No. | 2<br>$y_1$ | 3<br>$y_2$ | 4<br>$y_3$ | 5<br>$T^G_{1,2}(y)$ | 6<br>$T^G_{1,3}(y)$ | 7<br>$T^G_{2,3}(y)$ |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 6.86 | 1.00 | 1.00 | 6.86 |
| 2 | 1 | 2 | 6.63 | 2.00 | 6.63 | 3.32 |
| 3 | 1 | 3 | 6.24 | 3.00 | 6.24 | 2.08 |
| 4 | 1 | 4 | 5.66 | 4.00 | 5.66 | 1.42 |
| 5 | 1 | 5 | 4.80 | 5.00 | 4.80 | 0.96 |
| 6 | 1 | 6 | 3.46 | 6.00 | 3.46 | 0.58 |
| 7 | 2 | 1 | 6.63 | 0.50 | 3.32 | 6.63 |
| 8 | 2 | 2 | 6.40 | 1.00 | 3.20 | 3.20 |
| 9 | 2 | 3 | 6.00 | 1.50 | 3.00 | 2.00 |
| 10 | 2 | 4 | 5.39 | 2.00 | 2.70 | 1.35 |
| 11 | 2 | 5 | 4.47 | 2.50 | 2.24 | 0.89 |
| 12 | 2 | 6 | 3.00 | 3.00 | 1.50 | 0.50 |
| 13 | 3 | 1 | 6.24 | 0.33 | 2.08 | 6.24 |
| 14 | 3 | 2 | 6.00 | 0.67 | 2.00 | 3.00 |
| 15 | 3 | 3 | 5.57 | 1.00 | 1.86 | 1.86 |
| 16 | 3 | 4 | 4.90 | 1.33 | 1.63 | 1.23 |
| 17 | 3 | 5 | 3.87 | 1.67 | 1.29 | 0.77 |
| 18 | 3 | 6 | 2.00 | 2.00 | 0.67 | 0.33 |
| 19 | 4 | 1 | 5.66 | 0.25 | 1.42 | 5.66 |
| 20 | 4 | 2 | 5.39 | 0.50 | 1.35 | 2.70 |
| 21 | 4 | 3 | 4.90 | 0.75 | 1.23 | 1.63 |
| 22 | 4 | 4 | 4.12 | 1.00 | 1.03 | 1.03 |
| 23 | 4 | 5 | 2.83 | 1.25 | 0.71 | 0.57 |
| 24 | 5 | 1 | 4.80 | 0.20 | 0.96 | 4.80 |
| 25 | 5 | 2 | 4.47 | 0.40 | 0.89 | 2.24 |
| 26 | 5 | 3 | 3.87 | 0.60 | 0.77 | 1.29 |
| 27 | 5 | 4 | 2.83 | 0.80 | 0.57 | 0.71 |
| 28 | 6 | 1 | 3.46 | 0.17 | 0.58 | 3.46 |
| 29 | 6 | 2 | 3.00 | 0.33 | 0.50 | 1.50 |
| 30 | 6 | 3 | 2.00 | 0.50 | 0.33 | 0.67 |

call formulated in Preface for simple, low computing-intensive MCDM methods. We return to this topic and discuss it in detail in the next chapter.

Consistently with the assumption made in Section 2 of Chapter 5, that the Universal Interface exploits optimization problem (3.1), (3.2), or (3.3), in the next chapters we make no use of bounds (8.3). Those bounds are included into this chapter solely for the sake of completeness.

## 5. Annotated References

Bounds on global trade-offs for outcomes derived by solving optimization problem (3.1) are established first in Wierzbicki (1990), and for outcomes derived by solving optimization problem (3.2) in Kaliszewski (1994b).

More flexible global trade-off bounding schemes via making use of variations of optimization problem (3.1) or (3.2) are proposed in Kaliszewski and Michalowski (1995, 1997) (cf. characterizations I', II', IV', I'', II'', I''', II''' in this chapter).

The idea to use such schemes to differentiate efficient outcomes as a decision supporting tool is put forward and elaborated in Kaliszewski, Michalowski (1995, 1997, 1999), Kaliszewski, Michalowski, Kersten (1997), Kaliszewski (2000), Kaliszewski, Zionts (2004), and further extended in Koktener-Karasakal, Michalowski (2003).

The practical importance and consequences of deriving efficient outcomes with a priori bounds on global trade-offs is demonstrated, for example, when solving a water quality management problem modeled as an MCDM problem, as reported in Makowski et al. (1996).

Chapter 9

# SOFT COMPUTING FOR COMPLEX MCDM PROBLEMS

*"Diversity of opinion about a work of art shows that work is new, complex, and vital."*

**Oscar Wilde**,
*The Picture of Dorian Gray.*

## 1. This Chapter Is About ...

... building on the material presented in all previous chapters, but especially in Chapters 5 to Chapter 8. We demonstrate practical applicability of two sets of quantities elaborated in Chapters 7 to 8, namely:

- bounds (lower and upper) on efficient outcome components,
- bounds (upper) on efficient outcome trade-offs.

We also point to potential impact these concepts can have on interactive MCDM methods. Finally, we discuss issues related to implementation of these two concepts into existing interactive MCDM methods, and illustrate viability of these concepts by solving numerical examples.

## 2. Solving Complex MCDM Problems With Soft Computing

When dealing with complex problems existing interactive MCDM methods rely on *hard* (exact) computing. This and the fact that derivation of efficient outcomes and calculation of global trade-offs requires optimization makes these methods high computing-intensive.

With bounds on efficient outcome components and efficient outcome global trade-offs presented in Chapters 7 and 8 we are in position to switch the interactive MCDM framework from hard to *soft* (i.e. inexact, approximate) *computing*. By this we can economize on computing costs.

Soft computing can be incorporated into interactive MCDM methods in two places.

The first place are DM evaluations of the exact values of outcome components, which can be replaced by evaluation of bounds on outcome components. Such a replacement is fully justified in practical decision making problems in which effects are measured up to some meaningful units. Below, following the convention introduced in Chapter 7, we refer to a pair of parametric bounds $L(\lambda)$ and $U(\lambda)$ calculated for an implicit efficient outcome $y(\lambda)$ as its assessment, $[y(\lambda)] = \{L(\lambda),\ U(\lambda)\}$, in that sense that

$$L(\lambda) \leq y(\lambda) \leq U(\lambda), \tag{9.1}$$

(cf. Chapter 7, Section 4).

Fairness of such assessments, i.e. tightness of bounds (9.1) is discussed and formulas to calculate $L(\lambda)$ and $U(\lambda)$ are presented in Chapter 7.

The second place where soft computing can be incorporated into interactive MCDM are evaluations of (the exact values of) global trade-offs. In such evaluations global trade-offs can be represented by evaluations of their corresponding upper bounds. If $y(\lambda)$ is an implicit efficient outcome and $T_{i,j}^G(y(\lambda))$ one of its global trade-offs, then

$$T_{i,j}^G(y(\lambda)) \leq W_{i,j}(\lambda), \tag{9.2}$$

where $W_{i,j}(\lambda)$ is a parametric upper bound on $T_{i,j}^G(y(\lambda))$. Formulas to calculate such bounds are presented in Chapter 8.

Below we discuss in more detail issues related to incorporating soft computing into interactive MCDM methods in general, and in particular into $GIS^2$, which, as claimed in Chapter 6, can be regarded as a generic representative of interactive MCDM methods.

# 3. Applications Of Parametric Bounds To Interactive MCDM Methods

Bounds on efficient outcome components (9.1) ((7.1) - (7.3), (7.6) - (7.7)) and bounds on global trade-offs (9.2) ((8.1) - (8.2)) can be incorporated easily into any instance of a wide spectrum of existing interactive MCDM methods, which use problems (3.1) - (3.3) to identify outcomes (alternatives). Existence of simple tools to assess outcome components and to assess outcome global trade-offs without the necessity of explicit outcome derivation can have a significant impact on popular usage of interactive MCDM methods.

Consistently with the assumption made in Section 2 of Chapter 5, that the Universal Interface exploits optimization problem (3.1), (3.2), or (3.3), here we do not consider optimization problem (3.4), bounds on efficient outcome components (7.4), (7.8), and bounds on global trade-offs (8.3).

There are three essential issues related to the practical applicability of parametric bounds of type (9.1) and (9.2). They are:

- the issue of optimization and cost effectiveness,
- the issue of fitness,
- the issue of versatility.

In the next three consecutive subsections we discuss these issues with respect to bounds on outcome components.

There is no need to discuss these issues in detail with respect to bounds on global trade-offs. Bounds of type (9.2) are easily calculable a priori once vector $\lambda$ and parameter $\rho$ are specified. In fact, these bounds are given almost at no computation cost (cf. Chapter 8). Hence, global trade-off bound calculations are indeed cost effective. For the same reason global trade-off bound calculations fit any interactive MCDM method.

## 3.1 Optimization and Cost Effectiveness

As said before, when dealing with complex problems existing interactive MCDM methods use optimization to derive outcomes. This, we believe, is the main obstacle for widespread use of these methods for solving practical decision making problems. Using optimization methods requires some skill. Lay use of optimization methods is hampered by the constant concern: *are the strings of digits I receive from the optimization package meaningful*? Support from an analyst or facilitator can be great help, but certainly far from all DMs like to reveal their extent (or lack) of analytical abilities and make their preferences known to the public, even to sworn to silence consultants.

Moreover, it is necessary to have access to external computer facilities, when a desk top computer does not have sufficient capacity. This is essential requirement for medium- and large-scale MCDM problems. An on-line connection can be costly, and subject to bottlenecks. An off-line connection may be time-delayed and therefore frustrating.

The standard action of any interactive MCDM method is to derive at each iteration at least one efficient outcome for the DM to evaluate. An alternative, soft computing approach would be to calculate bounds (9.1) (outcome assessment) on outcome components instead of deriving outcomes explicitly, which, given a shell, is a much easier task.

If sufficiently tight, bounds could be used then to decide whether the corresponding implicit efficient outcome, if explicitly derived, could be of interest to the DM. If this outcome could be of no interest to the DM, it does not need to be explicitly derived, which results in saving both time and optimization computing. If this outcome could be of interest to the DM, the DM has still two options. The first option is to have the outcome explicitly derived and then evaluated. The second option for the DM is to evaluate bounds on components of this outcome instead of exact values of components, which again results in saving both time and optimization computing. In the extreme scenario explicit derivation of outcomes is eliminated, as illustrated in Figure 9.1.

To get some insight into potential savings on computations when using outcome assessments instead of just outcomes, let us consider the following numerical example.

EXAMPLE 1 Let $Z = \{y \in R^2 \mid \sum_{i=1}^{2} y_i = 1,\ y_i \geq 0,\ i = 1, 2\}$.

We have $\hat{y} = (1.0, 1.0)$, $\bar{L}_i = 0,\ \bar{U}_i = 1,\ i = 1, 2$.

We arbitrarily set $y_i^* = \hat{y}_i + \epsilon,\ i = 1, 2$, where $\epsilon = 0.1$, hence $y_i^* = 1.1,\ i = 1, 2$.

For calculating upper and lower bounds we assume $L(\lambda)_i = \bar{L}_i$ and $U(\lambda)_i = \bar{U}_i,\ i = 1, 2$ for all $\lambda$.

Suppose shell $S^0$ is given, $S^0 = \{(0.0, 1.0), (1.0, 0.0), (0.75, 0.25), (0.25, 0.75), (0.5, 0.5)\}$ and bounds on outcome components are cal-

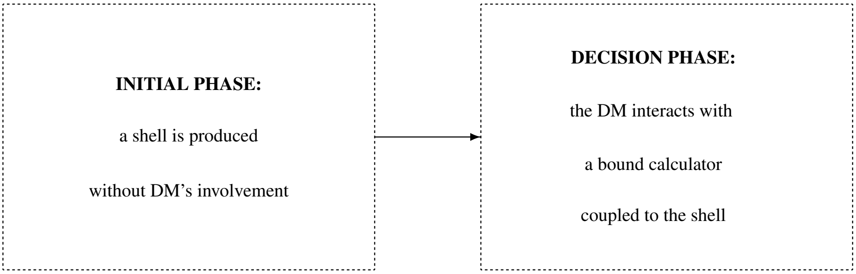


*Figure 9.1.* Phases of decision processes respective to the role of shell.

culated by formula (7.3) and (7.7). These formulas are related to optimization problem (3.3. Thus, vectors $\bar{\lambda}$ for which shell elements solve optimization problem (3.1) are given by formula (7.5) with $\rho = 0$.

♣ Observe that if to calculate bounds on outcome components one applies the method proposed in Section 8 of Chapter 7 (this method applies for $Z$ is $R^k_+$-convex), then, because of the shape of $Z$, in this particular case one gets the exact component values.
♠

Below we follow the steps of $GIS^2$ modified by use of bounds on outcome components. For the sake of clarity of presentation, in this example we drop any reference to trade-offs.

*Initialization*

0. $\bar{\Lambda} = \{\lambda \mid \lambda_i > 0,\ i = 1, 2,\ \lambda_1 + \lambda_2 = 1\}$.

To start the decision making process two vectors of weights

$$\lambda^{inc} = (0.67,\ 0.33),$$
$$\lambda^{c} = (0.33,\ 0.67),$$

are selected from $\bar{\Lambda}$ by the rule of thumb. To simplify notation, incumbent and candidates are not distinguished by the iteration number.

$GIS^2$ sends down to the bound calculator:

- vector $\tau^{inc}$ and element $y^*$,
- vector $\tau^c$ and element $y^*$,

where vectors $\tau$ are defined by formula (5.5).

The bound calculator returns:

- assessment of implicit (weakly) efficient outcome $y(\lambda^{inc})$,
- assessment of implicit (weakly) efficient outcome $y(\lambda^c)$,

see Table 9.1 (cf. Figure 9.2, where rectangles defined by the corresponding bounds represent the assessments of outcomes $y(\lambda^{inc})$ and $y(\lambda^c)$).

*Table 9.1.* Example 1 of Chapter 9, Initialization: bounds on $y(\lambda)$ for $\lambda^{inc}$ and $\lambda^c$.

| | $\lambda_1$ | $\lambda_2$ | $L(\lambda)_1$ | $U(\lambda)_1$ | $L(\lambda)_2$ | $U(\lambda)_2$ |
|---|---|---|---|---|---|---|
| $y(\lambda^{inc})$ | 0.67 | 0.33 | 0.68 | 0.75 | 0.25 | 0.50 |
| $y(\lambda^c)$ | 0.33 | 0.67 | 0.25 | 0.50 | 0.68 | 0.75 |

*Iteration 1*

1. Suppose that on the base of examining bounds the DM prefers the outcome corresponding to $\lambda^c$ (i.e. implicit $y(\lambda^c)$) to the outcome corresponding to $\lambda^{inc}$ (i.e. implicit $y(\lambda^{inc})$). Hence, there is no need to derive explicitly $y(\lambda^{inc})$ and $y(\lambda^c)$ for evaluation.

So implicit $y(\lambda)$ corresponding to $\lambda = (0.33,\ 0.67)$ becomes the (implicit) incumbent.

2. Suppose the DM wants to continue.

3. Suppose that DM, on the base of examining bounds for the incumbent, decides that his preferred outcome should satisfy additional constraints, namely $y_1 \leq 0.40,\ y_2 \geq 0.50$. This results in a modified problem with the following data update:

$$\hat{y}_1 = 0.4,\ y_1^* = 0.5,$$

$$\bar{U}_1 = U(\lambda) = 0.40,\ \bar{L}_2 = L(\lambda) = 0.50 \text{ for all } \lambda \in \bar{L}.$$

4. An assessment of a new implicit candidate is derived. For that purpose vector $\lambda^c = (0.67,\ 0.33)$ is selected form the set of admissible weights $\bar{\Lambda}$. This is the same vector selected in Initialization but since the problem is modified, this vector refers now (probably but not necessarily) to a different outcome. Because of problem modification we expect also to get a different assessment for $y(\lambda^{inc})$.

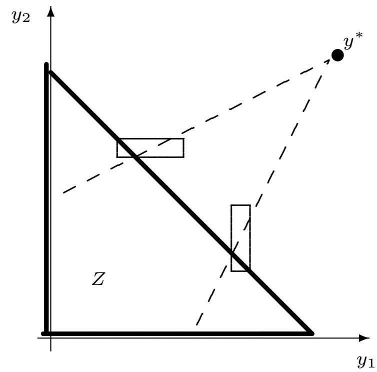


*Figure 9.2.* Example 1 of Chapter 9, Initialization: bounds on $y(\lambda)$ for $\lambda^{inc}$ and $\lambda^c$.

$GIS^2$ sends down to the bound calculator:

- vector $\tau^c$ and element $y^*$,
- vector $\tau^{inc}$ and element $y^*$.

The bound calculator returns:

- assessment of implicit (weakly) efficient outcome $y(\lambda^c)$,
- assessment of implicit (weakly) efficient outcome $y(\lambda^{inc})$,

see Table 9.2 (cf. Figure 9.3).

From Table 9.2 we see that bound tightness, as compared to bounds in Table 9.1, deteriorated. This is the consequence of the shrunk shell: with the constraint $y_1 \leq 0.4$ outcomes $(1.0, 0.0)$, $(0.75, 0.25)$, $(0.5, 0.5)$ are no longer valid elements of $S^0$. Depending on the required bound tightness, one can either include more efficient outcomes (feasible to the modified problem) into the shell, if bounds are not sufficiently tight,

*Table 9.2.* Example 1 of Chapter 9, Iteration 1: bounds on $y(\lambda)$ for $\lambda^{inc}$ and $\lambda^c$.

| i | $\lambda_1^i$ | $\lambda_2^i$ | $L(\lambda^i)_1$ | $U(\lambda^i)_1$ | $L(\lambda^i)_2$ | $U(\lambda^i)_2$ |
|---|---|---|---|---|---|---|
| $y(\lambda^{inc})$ | 0.33 | 0.67 | 0.00 | 0.25 | 0.85 | 1.00 |
| $y(\lambda^c)$ | 0.67 | 0.33 | 0.25 | 0.40 | 0.60 | 0.75 |

or proceed with the shell unchanged, if otherwise. In this example we proceed with the latter scenario.

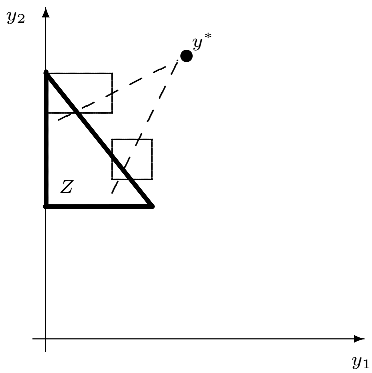


*Figure 9.3.* Example 1 of Chapter 9, Iteration 2: bounds on $y(\lambda)$ for $\lambda^{inc}$ and $\lambda^c$.

*Iteration 2*

1. Suppose that on the base of examining bounds the DM prefers outcome corresponding to $\lambda^{inc}$ (i.e. implicit $y(\lambda^{inc})$) to outcome corresponding to $\lambda^c$ (i.e. implicit $y(\lambda^c)$). Hence, there is still no need to derive explicitly $y(\lambda^c)$ and $y(\lambda^{inc})$ for evaluation.

So implicit $y(\lambda)$ corresponding to $\lambda = (0.33,\ 0.67)$ remains the (implicit) incumbent.

2. Suppose the DM wants to continue.

3. Suppose that DM, on the base of examining bounds for the incumbent, points out to vector of weights $\lambda^c = (0.50, 0.50)$ as an interesting option.

4. An assessment of a new implicit candidate corresponding to $\lambda^c = (0.50, 0.50)$ is derived.

$GIS^2$ sends down to the bound calculator:

- vector $\tau^c$ and element $y^*$.

The bound calculator returns:

- assessment of implicit (weakly) efficient outcome $y(\lambda^c)$,

see Table 9.3 (cf. Figure 9.4).

*Table 9.3.* Example 1 of Chapter 9, Iteration 2: bounds on $y(\lambda)$ for $\lambda^{inc}$ and $\lambda^c$.

| i | $\lambda_1^i$ | $\lambda_2^i$ | $L(\lambda^i)_1$ | $U(\lambda^i)_1$ | $L(\lambda^i)_2$ | $U(\lambda^i)_2$ |
|---|---|---|---|---|---|---|
| $y(\lambda^{inc})$ | 0.33 | 0.67 | 0.00 | 0.25 | 0.85 | 1.00 |
| $y(\lambda^c)$ | 0.50 | 0.50 | 0.15 | 0.25 | 0.75 | 1.00 |

*Iteration 3*

1. Suppose that on the base of examining bounds the DM prefers outcome corresponding to $\lambda^c$ (i.e. implicit $y(\lambda^c)$) to outcome corresponding to $\lambda^{inc}$ (i.e. implicit $y(\lambda^{inc})$). Hence, there is still no need to derive explicitly $y(\lambda^{inc})$.

So implicit $y(\lambda)$ corresponding to $\lambda = (0.50, 0.50)$, becomes the (implicit) incumbent.

2. Suppose the DM does not want to continue. The (weakly) efficient outcome $y(\lambda^{inc})$ is derived explicitly by solving problem (3.3) which yields $y(\lambda^{inc}) = (0.20, 0.80)$. □

♣ One can think of exploiting shells which are built during standard courses of interactive decision making since at each iteration at least one efficient outcome is derived. When such a shell consists of a sufficient

number of outcomes and lower and upper bounds become sufficiently tight, (clearly, bounds become more tight as the number of outcomes in the shell increases), the DM can switch from evaluating efficient outcomes to evaluating efficient outcome assessments. As already observed, the sufficient tightness of bounds is a context-dependent notion. But one can hardly expect that during standard courses of interactive decision making shells of sufficient cardinality can be built. It has been found by experience that the number of iterations required to reach an efficient outcome, which for the DM is the most preferred, is usually small.
♠

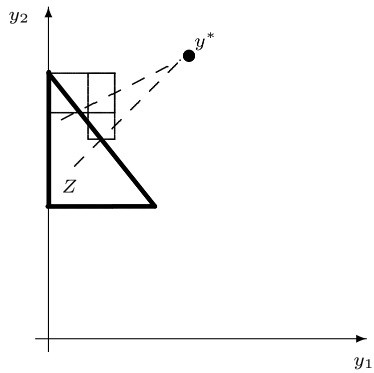


*Figure 9.4.* Example 1 of Chapter 9, Iteration 3: bounds on $y(\lambda)$ for $\lambda^{inc}$ and $\lambda^c$.

♣ In some instances it is possible to have weight cuts formed (cf. Chapter 4) even for implicit outcomes. Suppose that for two implicit outcomes $y(\lambda^1)$, $y(\lambda^2)$ their assessments $\{L(\lambda^1), U(\lambda^1)\}$ and $\{L(\lambda^2), U(\lambda^2)\}$ are calculated by formula (7.2) and formula (7.6). Suppose that on the base of the assessments the DM prefers implicit outcome $y(\lambda^1)$ to implicit $y(\lambda^2)$. It is reasonable to assume that the DM prefers $U(\lambda^1)$ to $L(\lambda^2)$ irrespective of whether $U(\lambda^1) \in Z$ or $L(\lambda^2) \in Z$. It is possible then to form a weight cut with respect to $U(\lambda^1)$ and $L(\lambda^2)$. For example, consider weight cuts resulting from function (4.5). If the DM prefers

$U(\lambda^1)$ to $L(\lambda^2)$, then the weight cut takes the form

$$\begin{aligned} &\max_i \lambda_i((y_i^* - U(\lambda^1)_i) + \rho e^k(y^* - U(\lambda^1)) \\ &< \max_i \lambda_i((y_i^* - L(\lambda^2) + e^k(y^* - L(\lambda^2))\,. \end{aligned} \tag{9.3}$$

However, this cut is not trivial only if $U(\lambda^1)$ does not dominate $L(\lambda^2)$.

We exploit cuts arising from outcome bounds when solving a numerical example in Section 6 of this chapter.

♠

## 3.2 Fitness

The second issue related to the practical applicability of parametric bounds on outcome components to interactive MCDM methods is the issue of fitness. This is how the concept of parametric bounds fits these methods and how such bounds can be incorporated into them. A framing of interactive MCDM method classes proposed in Section 2 of Chapter 5 forms a basis for discussion of possible ways to implement these bounds.

Once shell $S^0$ or $S^\rho$ is established, lower and upper parametric bounds on components of (properly or weakly) efficient outcomes are fully defined by weight vectors $\lambda$, parameter $\rho$, and element $y^*$ (cf. formulas (7.1), (7.3), (7.6), (7.7).

In other words, as seen from formulas (5.2) and (5.1), this is equivalent to the statement that these bounds are in fact defined by half lines starting from $y^*$ along some vector $\tau$ (cf. Figure 7.2, Figure 7.3, Figure 7.6, and Figure 7.7).

In Remark 2 of Chapter 7 it is shown how the bounds can be adapted to the reference point method class with reference point $y^{ref}$ replacing $y^*$.

As shown in Chapter 5 (cf. Table 5.1) vector $\tau$ and elements $y^*$ or $y^{ref}$ are the only objects used within the Universal Interface (for elements $y^{res}$ and $y^{asp}$ define vectors $\tau$).

Thus, parametric bounds fit any method which falls into the Universal Interface framework.

## 3.3 Versatility

The development of parametric bounds, together with the idea of the Universal Interface, can give a new stimulus to applications of interactive MCDM since it permits the relocation technicalities related to decision making to the background of decision making processes.

With this development the need for solving optimization problems can be confined to establishing shells and this can be left to external computing service providers. Such providers would cope with original underlying models, do all necessary optimization computations, and provide shells which guarantee a priori assumed bound tightness.

It is left then to the "machine" phase of the interactive decision making process to cope solely with simple formulas (a simple bound calculator is needed instead of an optimization engine) which can even be implemented in a spreadsheet (cf. Figure 9.1).

## 3.4 Practical Example

Let us exemplify the arguments given up to now in Section 2 and Section 3 of this chapter by solving the Markowitz mean-variance problem with soft computing.

To find the most preferred portfolio reflecting the DM's risk versus gain profile in the interactive manner with hard computing (i.e. with explicit efficient outcome derivation) several efficient portfolios have to be derived explicitly. This involves optimization, i.e solving optimization problem (3.1) or (3.2). This also requires that the DM has an access, direct or indirect, to a solver capable of solving problems with several hundreds of variables (the number of stocks listed on large exchanges reaches well over thousand), and to a significant computing power.

With soft computing, if a shell for a specific instance of the Markowitz problem is known, the DM is able to locate the most preferred portfolio $y(\lambda)$ within the range

$$L(\lambda) \leq y(\lambda) \leq U(\lambda), \tag{9.4}$$

where $L(\lambda)$ and $U(\lambda)$ are calculated as shown in Chapter 7.

It is clear that the effort to derive a shell for a practical size of the Markowitz problem may be well beyond capacities of even some financial institutions, not to mention individual investors. But shells could be derived and published e.g. by exchanges or brokerage houses as a service to investors. This would result in a more appropriate allocation of tasks: investors would concentrate on portfolio building, whereas optimization would be left to optimization specialists and computing service providers.

In the Section 6 of this chapter we solve with soft computing the same instance of the Markowitz mean-variance problem solved already in Section 5 of Chapter 6.

## 4. Soft Computing with $GIS^2$

The best way to corroborate the soundness of our arguments with respect to the issue of optimization and cost effectiveness, fitness, and versatility of parametric bounds, is to show how those issues can be interpreted and implemented along lines of $GIS^2$ – Generic Interactive MCDM Support System - presented in Chapter 6. We attempt to achieve that in this and the next section.

Below we modify the steps of the original $GIS^2$ to get its soft version. Within this soft version deriving implicit outcomes or calculating trade-offs is understood as calculating bounds on outcome components and bounds on global trade-offs. This modification results in *Generic Interactive MCDM Soft Support Scheme* - $GIS^3$.

$GIS^3$ mirrors the same pattern of five steps of which $GIS^2$ is composed.

$GIS^3$

0. Derive a number of **implicit** efficient outcomes and for each of them calculate global trade-offs. Select arbitrarily one outcome and denote it as *incumbent*. Denote the other outcomes as *candidates*.
1. Ask the DM to evaluate each candidate against the incumbent. For each evaluation ask the DM if his preference is firm or non-firm. Ask the DM to select the new incumbent.
2. Ask the DM if he wants to continue the search. If no, terminate with the incumbent as the most preferred **implicit** outcome.
3. Ask the DM to select objects to manipulate: weights or reference points. If the DM selects weights to manipulate, for each evaluation of Step 1 resulting in a firm preference form a weight cut. Ask the DM to express his preferences using the objects selected.
4. Derive a number of **implicit** efficient outcomes satisfying preferences expressed by the DM in Step 3, denote them as

candidates, and for each candidate calculate global trade-offs. Go to Step 1.

## 5. $GIS^3$ At Work

We illustrate working with $GIS^3$ on the example presented in Section 4 of Chapter 6. The formulation of the problem remains unchanged.

Though objects available for manipulations are, as specified by $GIS^3$, weights and reference points, to stress that $GIS^3$ falls into the framework of the Universal Interface, as described in Chapter 5, at each iteration references are made to vectors $\tau$.

Here we put $\bar{U}_{-v} = \hat{y}_{-v}$ and $\bar{U}_e = \hat{y}_e$, where as before $\hat{y}_{-v} = -0.011$, $\hat{y}_e = 2$. For $\hat{y}_e$ the minimal value of $y_{-v}$ is $-5.218$, so we put $\bar{L}_{-v} = -5.218$. For $\hat{y}_{-v}$ the minimal value of $y_e$ is equal to $1.085$, so we put $\bar{L}_e = 1.085$.

*Table 9.4.* The shell of the example portfolio selection problem of Chapter 9.

| s | $\lambda_1^s$ | $\lambda_2^s$ | $y_1^s$ | $y_2^s$ | $T^G_{-v,e}$ | $T^G_{e,-v}$ |
|---|---|---|---|---|---|---|
| 1 | 0.05 | 0.95 | -1.884 | 1.920 | 4.46 | 0.22 |
| 2 | 0.15 | 0.85 | -1.317 | 1.782 | 3.75 | 0.27 |
| 3 | 0.25 | 0.75 | -0.980 | 1.685 | 3.23 | 0.31 |
| 4 | 0.35 | 0.45 | -0.747 | 1.608 | 2.80 | 0.36 |
| 5 | 0.45 | 0.55 | -0.572 | 1.542 | 2.44 | 0.41 |
| 6 | 0.55 | 0.45 | -0.433 | 1.481 | 2.12 | 0.47 |
| 7 | 0.65 | 0.35 | -0.317 | 1.422 | 1.81 | 0.55 |
| 8 | 0.75 | 0.25 | -0.217 | 1.362 | 1.49 | 0.67 |
| 9 | 0.85 | 0.15 | -0.128 | 1.293 | 1.12 | 0.89 |
| 10 | 0.95 | 0.05 | -0.044 | 1.195 | 0.58 | 1.72 |

For calculating upper and lower bounds we assume $L(\lambda)_i = \bar{L}_i$ and $U(\lambda)_i = \bar{U}_i,\ i = e, -v$, for all $\lambda$.

With the assessment $[y(\lambda)] = \{L(\lambda), U(\lambda)\}$ of an efficient outcome $y(\lambda)$ we get an assessment of the assumed DM's value function $[f^{DM}(y(\lambda)] = \{f^{DM}(L(\lambda)), f^{DM}(U(\lambda))\}$.

We use formula (7.1) and (7.6) to calculate outcome (portfolio) bounds, but in contrast to computations for $GIS^2$ (Section 5 of Chapter 6) this time parameter $\rho$ can be varied to set a priori bounds on global trade-offs of outcomes to be derived, explicitly or implicitly. As shown by formula (7.1) and (7.6), bounds on outcome components also depend on $\rho$.

EXAMPLE 2 Assume that the following shell of the problem considered is given: $S^\rho = \{y(\lambda^s)\}, s = 1, ..., 10$, where vectors $\lambda^s$, (properly) efficient outcomes $y(\lambda^s)$, and the corresponding global trade-offs are listed in Table 9.4. All elements of the shell are derived with $\rho = 0.01$.

Since $y^*$ is an eligible reference point, it is treated below as such.

In the version of $GIS^3$ presented below at each iteration only one candidate is selected.

*Initialization*

0. $\bar{\Lambda} = \{\lambda \mid \lambda_i > 0,\ i = -v, e,\ \lambda_v + \lambda_e = 1\}$.

To start the decision making process two vectors of weights

$$\lambda^{inc} = (0.50, 0.50),$$
$$\lambda^{c} = (0.98, 0.02),$$

are selected from $\bar{\Lambda}$ by the rule of thumb. As in computations with $GIS^2$ in Section 5 of Chapter 6, to simplify notation the incumbent $y(\lambda^{inc})$ and candidates $y(\lambda^c)$ are not distinguished by the iteration number. At this point parameter $\rho$ is arbitrarily set to $0.01$ .

$GIS^3$ sends down to the bound calculator:

- vector $\tau^{inc}$, element $y^*$, and parameter $\rho = 0.01$,
- vector $\tau^{c}$, element $y^*$, and parameter $\rho = 0.01$,

Similarly as in $GIS^2$, since optimization problem (3.1) is used, vectors $\tau$ are defined indirectly by vectors $\lambda$ (cf. formula (5.2)), so $GIS^3$ sends down in fact vector $\lambda^{inc}$ and vector $\lambda^c$, respectively, for the corresponding vectors $\tau$ to be calculated (with optimization problem (3.2) or optimization problem (3.3) there is an explicit correspondence between vectors $\tau$ and $\lambda$, cf. Chapter 5).

Changing a reference point (e.g. $y^*$) to another reference point can violate relation (7.11) (cf. Remark 2 of Chapter 7). Therefore, at each iteration the reference point selected is sent down to the bound calculator for checking whether it satisfies (7.11), and if not, $y^{ref}$ has to be modified to $y^{ref'}$, as described in Remark 2 of Chapter 7.

The bound calculator returns:

- assessment of implicit (properly) efficient outcome $y(\lambda^{inc})$
$[y(\lambda^{inc})] = \{(-0.534,\ 1.447),\ (-0.429,\ 1.588)\}$,
together with assessment of the DM's value function
$[f^{DM}(y)(\lambda^{inc}))] = \{-6.380,\ -3.002\}$,

- assessment of implicit (properly) efficient outcome $y(\lambda^{c})$
$[y(\lambda^{c})] = \{(-0.052,\ 1.085),\ (-0.011,\ 1.246)\}$,
together with assessment of the DM's value function
$[f^{DM}(y(\lambda^{c}))] = \{1.096,\ 1.549\}$.

With $\rho = 0.01$ trade-offs $T^G_{-v,e}(y(\lambda^c))$ and $T^G_{e,-v}(y(\lambda^c))$ have an upper bound equal to 101, (cf. Characterization I', Remark 1, and Remark 4 of Chapter 8).

*Iteration 1*

1. Suppose that (consistently with the assessment of the assumed DM's implicit value function) the DM prefers implicit outcome $y(\lambda^c)$ to implicit outcome $y(\lambda^{inc})$ and this preference is firm.

An explanation of that preference can be that the risk range for implicit outcome $y(\lambda^{inc})$ is unacceptable, in contrast to favorable risk range of implicit outcome $y(\lambda^c)$.

The high value of bound on global trade-offs for both $y(\lambda^{inc})$ and $y(\lambda^c)$ provides no base for preference manifestation.

So $\lambda^{inc} = (0.98, 0.02)$, $[y(\lambda^{inc})] = \{(-0.052,\ 1.085),\ (-0.010,\ 1.246)\}$, $[f^{DM}(y(\lambda^{inc}))] = \{1.096,\ 1.549\}$.

2. Suppose the DM wants to continue.

3. Suppose that the DM selects weights to manipulate. The firm preference expressed in step 1 gives rise to a weight cut resulting from function (4.5). Observe that the upper assessment $\{-0.011,\ 1.246\}$ of $y(\lambda^c)$ does not dominate lower assessment $\{-0.544,\ 1.477\}$ of $y(\lambda^{inc})$ (cf. Subsection 3.1 of this chapter). In consequence, one gets two disjoint sets of conditions on $\lambda$, only one of which is consistent, namely the set:

$$\begin{gathered} \lambda_{-v} \geq 0, \\ \lambda_e < 0,701\lambda_{-v}\,. \end{gathered} \tag{9.5}$$

The above conditions update the initial (i.e. formed of the nonnegativity condition and the "sum to one" constraint) set of admissible weights $\bar{\Lambda}$.

Suppose that the DM selects from the updated set of admissible weights vector $\lambda^c = (0.75,\, 0.25)$.

4. A new candidate is derived.

$GIS^3$ sends down to the bound calculator:

- vector $\tau^c$ (in fact, vector $\lambda^c$), element $y^*$, and parameter $\rho = 0.01$ .

The the bound calculator returns:
- assessment of implicit (properly) efficient outcome $y(\lambda^c)$
$[y(\lambda^c)] = \{(-0.224,\, 1.360),\, (-0.214,\, 1.471)\}$,
together with assessment of the DM's value function
$[f^{DM}(y(\lambda^c))] = \{0.348,\, 0.785\}$.

With $\rho = 0.01$ trade-offs $T^G_{-v,e}(y(\lambda^c))$ and $T^G_{e,-v}(y(\lambda^c))$ have an upper bound equal to 101.

*Iteration 2*

1. Suppose that (consistently with the assessment of the assumed DM's implicit value function) the DM prefers implicit outcome $y(\lambda^{inc})$ to implicit outcome $y(\lambda^c)$ and this preference is not firm.

An explanation of that preference can be that the risk range for implicit outcome $y(\lambda^c)$ is unacceptable, in contrast to favorable risk range of implicit outcome $y(\lambda^{inc})$.

The high value of bound on global trade-offs for both $y(\lambda^{inc})$ and $y(\lambda^c)$ provides no base for preference manifestation.

So $\lambda^{inc} = (0.98, 0.02)$, $[y(\lambda^{inc})] = \{(-0.052,\, -0.011),\, (1.085,\, 1.246)\}$, $[f^{DM}(y(\lambda^{inc}))] = \{1.096,\, 1.549\}$.

2. Suppose the DM wants to continue.

3. Suppose that the DM selects weights to manipulate. Since the preference expressed in step 1 of the current iteration is not firm, no update of the set of admissible weights is available.

Suppose that the DM selects from the updated set of admissible weights vector $\lambda^c = (0.90\,, 0.10)$.

4. A new candidate is derived.

$GIS^3$ sends down to the bound calculator:

- vector $\tau^c$ (in fact, vector $\lambda^c$), element $y^*$, and parameter $\rho = 0.01$.

The the bound calculator returns:

- assessment of implicit (properly) efficient outcome $y(\lambda^c)$
$[y(\lambda^c)] = \{(-0.100,\ 1.195),\ (-0.043,\ 1.343)\}$,
together with assessment of the DM's value function
$[f^{DM}(y(\lambda^c))] = \{1.130,\ 1.749\}$.

With $\rho = 0.01$ trade-offs $T^G_{-v,e}(y(\lambda^c))$ and $T^G_{e,-v}(y(\lambda^c))$ have an upper bound equal to 101.

*Iteration 3*

1. Suppose that (consistently with the assessment of the assumed implicit value function) the DM prefers assessment of implicit outcome $y(\lambda^c)$, namely $[y(\lambda^c)] = \{(-0.100,\ 1.195),\ (-0.043,\ 1.343)\}$ to assessment of implicit outcome $y(\lambda^{inc})$, namely $[y(\lambda^{inc})] = \{(-0.052,\ 1.085),$ $(-0.011,\ 1.246)\}$ but this preference is not firm.

An explanation for that preference can be that on the basis of its assessment outcome $y(\lambda^c)$ seems to satisfactory compromise risk with gain, as compared to $[y(\lambda^{inc})]$.

The high value of bound on global trade-offs for both $y(\lambda^{inc})$ and $y(\lambda^c)$ provides no base for preference manifestation.

So $\lambda^{inc} = (0.90, 0.10)$, $[y(\lambda^{inc})] = \{(-0.100,\ 1.195),\ (-0.043$ $1.343)\}$, $[f^{DM}(y(\lambda^{inc}))] = \{1.130,\ 1.749\}$.

2. Suppose the DM wants to continue.

3. Suppose that the DM selects reference points to manipulate. Suppose that as a reference point the DM selects $y^{ref} = (-0.4, 1.200)$.

Suppose further that the DM is not satisfied with any outcome $y$ if trade-off $T^G_{-v,e}(y) > 2.5$. Hence, from now on derived outcomes should satisfy $T^G_{-v,e}(y) \leq 2.5$. This implies condition on $\rho$, namely $2.5 = (1+\rho)/\rho$, hence $\rho = 0.66$ (cf. Chapter 8, Characterization I').

The constraint on trade-off $T^G_{-v,e}(y)$ eliminates from the shell the first four elements (see Table 9.4), since clearly with $\rho = 0.66$ optimization problem (3.1) delivers only outcomes for which all trade-offs $\leq 2.5$. Thus, they cannot be used further to calculate upper bounds and the shell contains now elements [5] to [10] of the original shell. Though bounds calculated with the shrunk shell may deteriorate, in this example no new elements are added to the shell.

4. A new candidate is derived. For that purpose vector $\lambda^c = (0.90,\ 0.10)$ is selected. This is exactly the same vector of weights used to derive the current incumbent $y(\lambda^{inc})$ because such a combination of weights yields up to now the most satisfactory compromise of risk and gain.

Observe that since $y^*$ is replaced by $y^{ref}$ preferences encapsulated in the current set of admissible weights are not valid any more. Hence, the set of admissible weights has now to be reset to its initial form $\bar{\Lambda} = \{\lambda \,|\, \lambda_{-v} > 0,\ \lambda_e > 0,\ \lambda_{-v} + \lambda_e = 1\}$, as at the beginning of the decision making process.

$GIS^3$ sends down to the bound calculator:

- the vector $\tau^c$ (in fact, vector $\lambda^c$), reference point $y^{ref}$, and parameter $\rho = 0.66$.

With modified $\rho$ and reference point $y^{ref}$ relations (7.11) do not hold (in fact, they do not hold for any element $[5]$ to $[10]$). Hence, the bound calculator has to modify $y^{ref}$ as shown in Remark 2 of Chapter 7. Solving equations (5.1) one gets $y^{apex} = (-1.262,\ 0.338)$ and $\tau = y^{apex} - y^{ref} = (0.862,\ 0.862)$. Now $y^{ref'} = y^{ref} - t\tau$ for some $t$. All relations (7.11) are satisfied e.g. for $t = -0.8$, which results in $y^{ref'} = (0.290,\ 1.890)$, as implied by conditions (7.12) all satisfied with that $y^{ref'}$.

With the shell shrunk bound $\bar{U}_e = U(\lambda)_e$ is updated to $1.542$ and bound $\bar{L}_{-v} = U(\lambda)_{-v}$ to $-0.572$ (cf. Table 9.4).

The bound calculator returns:

- assessment of implicit (properly) efficient outcome $y(\lambda^c)$

$[y(\lambda^c)] = \{(-0.051,\ 1.085),\ (-0.011,\ 1.200)\}$,

together with assessment of the DM's value function

$[f^{DM}(y(\lambda^c))] = \{1.100,\ 1.437\}$.

With $\rho = 0.66$ global trade-offs $T^G_{-v,e}(y(\lambda^c))$ and $T^G_{e,-v}(y(\lambda^c))$ have an upper bound equal to $2.5$.

*Iteration 4*

1. Suppose that (consistently with the assessment of the assumed DM's implicit value function) the DM prefers assessment of implicit outcome $y(\lambda^{inc})$, namely $[y(\lambda^{inc})] = \{(-0.100,\ 1.195),\ (-0.043,\ 1.343)\}$ to assessment of implicit outcome $y(\lambda^c)$, namely $[y(\lambda^c)] = \{(-0.051,\ 1.085),\ (-0.011,\ 1.200)\}$.

An explanation for that preference can be that though implicit outcome $y(\lambda^c)$ satisfactory compromises risk with gain and shows no significant potential for improvement of gain at the expense of risk, as indicated by the bound on global trade-off $T^G_{e,-v}(y(\lambda^{inc}))$, the overall evaluation of $y(\lambda^{inc})$ is higher than of $y(\lambda^c)$.

So $\lambda^{inc} = (0.90,\ 0.10)$, $[y(\lambda^{inc})] = \{(-0.100,\ 1.195),\ (-0.043,\ 1.343)\}$, $[f^{DM}(y(\lambda^{inc}))] = \{1.130,\ 1.749\}$.

2. Suppose the DM does not want to continue. The process terminates.

The decision process is summarized in Table 9.5. □

REMARK 1 In the instance of $GIS^3$ exploited in this section we resort, quite arbitrarily, to making use of the efficient outcome assessments only. However, if the DM wishes at some iteration to know the exact values of components of an efficient outcome, he should be satisfied with his request. Suppose that in the above example, in step 4 of Iteration 3, the DM wishes, prior to stopping, that implicit efficient outcome $y(\lambda^c) = (0.90,\ 0.10)$ is derived explicitly. Then he gets $y(\lambda^c) = (-0.096,\ 1.262)$ and $f^{DM}(y(\lambda^c)) = 1.318$.

In calculations of Section 5 of Chapter 6 the same vector $\lambda^c$ is selected and $y(\lambda^c) = (-0.086,\ 1.251)$ is explicitly derived by solving optimization problem (3.1). Here the same problem with the same vector $\lambda^c$ is solved but $y^*$ is replaced by $y^{ref'}$, and therefore the derived outcomes do not coincide. □

## 6. Concluding Remarks

Researchers nowadays tend to be very modest when estimating the extent to which mathematical methods and models can be substitute for human beings when it comes to decision making. A role which is complementary to the DM rather than competitive is attributed to such

*Table 9.5.* Successive iteration data for the example portfolio selection problem of Chapter 9.

| It. | Preference via $\lambda$, $y$ | Implicit outomes | Value fun. | Bound on trade-offs |
|---|---|---|---|---|
| 0 | $(0.50, 0.50),\ y^*$ | $\{(-0.534, 1.447),$ $(-0.429, 1.588)\}$ | $\{-6.380, -3.002\}$ | 101 |
| | $(0.98, 0.02),\ y^*$ | $\{(-0.052, 1.085),$ $(-0.010, 1.246)\}$ | $\{1.096, 1.549\}$ | 101 |
| 1 | *Incumbent* | $\{(-0.052, 1.085),$ $(-0.010, 1.246)\}$ | $\{1.096, 1.549\}$ | |
| | $(0.75, 0.25),\ y^*$ | $\{(-0.224, 1.360),$ $(-0.214, 1.471)\}$ | $\{0.348, 0.785\}$ | 101 |
| 2 | *Incumbent* | $\{(-0.052, 1.085),$ $(-0.010, 1.246)\}$ | $\{1.096, 1.549\}$ | |
| | $(0.90, 0.10),\ y^*$ | $\{(-0.100, 1.195),$ $(-0.043, 1.343)\}$ | $\{1.130, 1.749\}$ | 101 |
| 3 | *Incumbent* | $\{(-0.100, 1.195),$ $(-0.043, 1.343)\}$ | $\{1.130, 1.749\}$ | |
| | $(0.90, 0.10), y^{ref}$ | $\{(-0.300, 1.182),$ $(-0.183, 1.290)\}$ | $\{-1.303, 0.660\}$ | 2.5 |
| 4 | *Incumbent* | $\{(-0.100, 1.195),$ $(-0.043, 1.343)\}$ | $\{1.130, 1.749\}$ | |

methods and models, and this role is mainly to handle bulky data and perform laborious data processing.

Moreover, it is more and more often admitted that decision making tools, to be practical, must be based on soft computing tools.

The idea presented in this book to exploit parametric bounds on efficient outcome components and efficient outcome global trade-offs is in fact an attempt to soften the complexity of existing MCDM methods, but without compromising on any aspect of MCDM methodologies developed over decades.

The parametric bounds bridge over the gap and enable to combine, on one side the standard approach to MCDM, in which outcomes are derived in the course of decision making process, and on the other side the approach in which, for purpose of decision making, Pareto sets are replaced by their (derived beforehand) discrete representations. With the parametric bounds in place and shells provided, the process of selecting the most preferred outcome (alternative) can be carried over such representations with any, in principle, of the available MCDM methodologies. For the reasons already explained before, the resulting combined approach seems to be the most appropriate for complex, medium- and large-scale MCDM problems.

It is important to stress once again that, with the exception of the initial phase, where a shell for the problem considered has to be derived by solving optimization problems, in $GIS^2$ and $GIS^3$ all calculations are elementary and comprise of adding, subtracting, and selecting maximal or minimal numbers from finite sets.

## 7. Annotated References

This chapter cumulates ideas and developments contained in a series of papers:

- for parametric bounds on outcome components: Kaliszewski (2001a, b, 2002b, 2003b, 2004),
- for parametric bounds on global trade-offs: Kaliszewski (1994b, 2000, 2002a), Kaliszewski, Michalowski (1995, 1997), Kaliszewski, Michalowski, Kersten (1997), Kaliszewski, Zionts (2004),
- for generic interactive support schemes: Kaliszewski, Michalowski, Kersten (1997), Kaliszewski, Michalowski (1999), Kaliszewski (2002b, 2003a, 2004), Kaliszewski, Zionts (2004).

The idea of replacing Pareto sets with their discrete representations of given accuracy is elaborated in Evtuschenko, Potapov (1986).

# References


Armann R., (1989), Solving multiobjective programming problems by discrete representation. *Optimization*, 20, 483-492.

Benayoun R., de Montgolfier J., Laritchev O., (1971), Linear programming with multiple objective functions: Step Method (STEM). *Mathematical Programming*, 1, 366-375.

Bowman V.J. Jr., (1976), On the relationship of the Tchebycheff norm and the efficient frontier of multiple-criteria objectives. In: *Multiple Criteria Decision Making*, (Thirez H., Zionts S., eds.), Lecture Notes in Economics and Mathematical Systems, 130, Springer Verlag, Berlin, 76-85.

Buchanan J.T., (1997), A naïve approach for solving MCDM problems: the GUES method. *Journal of Operational Research Society*, 48, 202-206.

Burkard R.E., Hamacher H.W., Rothe G., (1987), Approximation of convex functions and applications in mathematical programming. *Report 89, Institut für Mathematik*, Technische Universität Graz.

Chankong V., Haimes Y.Y., (1978), The interactive surrogate worth trade-off (ISWT) method for muliobjective decision-making. In: *Multiple Criteria Problem Solving,* (Zionts S., ed.), Lecture Notes in Economics and Mathematical Systems, 155, Springer-Verlag, Berlin, 42-67.

Chankong V., Haimes Y.Y., (1983), *Multiobjective Decision Making Theory and Methodology*. Elsevier Science Publishing Co., New York.

Choo E.U., Atkins D.R., (1983), Proper efficiency in nonconvex programming. *Mathematics of Operations Research*, 8, 467-470.

Cohon J.L., (1978), *Multiobjective Programming and Planning*. Academic Press, New York.

Dell R.F., Karwan M.H., (1990), An interactive MCDM weight space reduction method utilizing a Tchebycheff utility function. *Naval Research Logistics*, 37, 263-277.

Edwards W., (ed.), (1992), *Utility Theories: Measurements and Applications*. Kluwer, Boston.

Elton E.J., Gruber M.J., (1995), *Modern Portfolio Theory and Investment Analysis*. John Wiley & Sons, New York.

Evtushenko Yu. G., Potapov M.A., (1986), *Methods of Numerical Solution of Multicriterion Problems*. Soviet Mathematical Doklady, 34, 420-423.

Fishburn P.C., (1986), *Utility Theory for Decision Making*. John Wiley & Sons, New York.

Fruhwirth B., Burkard R.E., Rothe G., (1988), Approximation of convex curves with applications to the bicriterial minimum cost flow problem. *Report 119, Institut für Mathematik*, Technische Universität Graz.

Gal T., Stewart Th., Hanne Th., (eds.), (1999), *Multicriteria Decision Making - Advances in MCDM: Models, Algorithms, Theory and Applications*. Kluwer Academic Publishers.

Galas Z., Nykowski I., Żółkiewski Z., (1987), *Multicriteria Programming* (in Polish). Państwowe Wydawnictwo Ekonomiczne, Warszawa.

Gardiner L.R., Steuer R.E., (1994a), Unified interactive multiple objective programming. *European Journal of Operational Research*, 74, 391-406.

Gardiner L.R., Steuer R.E., (1994b), Unified interactive multiple objective programming: an open architecture for accomodating new procedures. *Journal of the Operational Research Society*, 45, 1456-1466.

Geoffrion A.M., (1968), Proper efficiency and the theory of vector maximization. *Journal of Mathematical Analysis and its Applications*, 22, 618-630.

Geoffrion A.M., Dyer J.S., Feinberg A., (1972), An interactive approach to muliti-criterion optimization with an application to the operation of an academic department. *Management Science*, 19, 357-368.

Geromel J.C., Ferreira P.A.V., (1991), An upper bound on properly efficient solutions in multiobjective optimisation. *Operation Research Letters*, 10, 83-86.

Guddat J., Guerra Vasquez F., Tammer K., Wendler K., (1985), *Multiobjective and Stochastic Optimization Based on Parametric Optimization*. Akademie-Verlag, Berlin.

Haimes Y.Y., Chankong V., (1979), Khun-Tucker multipliers as trade-offs in multiobjective decision-making analysis. *Automatica*, 15, 59-72.

Haimes Y.Y., Hall W.A., Freedman H.T., (1975), *Multiobjective Optimization in Water Resources Systems*. Elsevier Scientific Publishing Company, Amsterdam.

Haimes Y.Y., Lasdon L.S., Wismer D.A., (1971), On bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man and Cybernetics*, 1, 296-297.

Haimes Y.Y., Tarvainen K., Shima T., Thadathil J., (1990), *Hierarchical Multiobjective Analysis of Large-Scale Systems*. Hemisphere Publishing Corporation, New York.

Halme M., (1992), Local characterizations of efficient solutions in interactive multiple objective programming. *Acta Academiae Oeconomicae Helsingiesis*, Series A, 84.

Henig M.I., Buchanan J., (1997), Tradeoff directions in multiobjective optimization problems. *Mathematical Programming*, 78, 357-374.

Hwang C.L., Masud A.S.M., Paidy S.R., Yoon K., (1979), *Multiple Objective Decision Making - Methods and Applications*. Lecture Notes in Economics and Mathematical Systems, 164, Springer Verlag, Berlin.

Ignizio J.P., (1976), *Goal Programming and Extensions*. Lexington Books, D.C. Heath and Company.

Ignizio J.P., (1985), *Introduction to Linear Goal Programming*. Sage Publications, Sage University Press, Beverly Hills.

Jahn J., (1986), *Mathematical Vector Optimization in Partially Ordered Linear Spaces*, Peter Lang, Frankfurt am Main.

Jaszkiewicz A., Słowiński R., (1994), The light-beam search over a nondominated surface of a multiple-objective problem. In: *Multiple Criteria Decision Making – Proceedings of Tenth International Conference: Expand and Enrich the Domains of Thinking and Application*, (Thzeng G.H., Wand H.F., Wen U.P., Yu P.L, eds.), Springer-Verlag, Berlin, 87-99.

Jaszkiewicz A., Słowiński R., (1995), The light-beam search - outranking based interactive procedure for multiple-objective mathematical programming. In: *Advances in Multicriteria Analysis* (Pardalos P.M., Siskos Y., Zopoundis C., eds.), Kluwer Academic Publishers, Dordrecht, 129-146.

Jog V., Kaliszewski I., Michalowski W., (1999), Using trade-off information in the attributes' investing. *Journal of Multi-criteria Decision Analysis*, 189-199.

Kahneman D., Tversky A., (1979), Prospect theory: an analysis of decision under risk. Econometrica, 47, 263-291.

Kaliszewski I., (1987), A modified weighted Tchebycheff metric for multiple objective programming. *Computers and Operations Research*, 14, 315-323.

Kaliszewski I., (1993), Calculating trade-offs by two-step parametric programming. *Central European Journal of Operational Research and Economics*, 2, 291-305.

Kaliszewski I., (1994a), A theorem on nonconvex functions and its application to vector optimization. *European Journal of Operations Research*, 80, 439-449.

Kaliszewski I., (1994b), *Quantitative Pareto Analysis by Cone Separation Technique*. Kluwer Academic Publishers, Boston.

Kaliszewski I., (2000), Using trade-off information in decision making algorithms. *Computers & Operations Research*, 27, 161–182.

Kaliszewski I., (2001a), Eliciting information on efficient outcomes prior to explicitly identyfying them. *Scientific Transactions of the Department of Mechanics*, Koszalin University of Technology, Koszalin, 157-164.

Kaliszewski I., (2001b), Multiple criteria decision making without optimization. *Systems Science*, 27, 5-13.

Kaliszewski I., (2002a), Trade-offs - a lost dimension in multiple criteria decision making. In: *Multi-Objective Programming and Goal Programming, Recent Developments*, (Trzaskalik T., Michnik J., eds.), Physica-Verlag, 115-126.

Kaliszewski I., (2002b), Multiple criteria decision making with deductive approximate representation of efficient outcomes. Control & Cybernetics, 31, 949-964.

Kaliszewski I., (2003a), Weighting versus reference point multiple criteria decision making methods - analogies and differences. *Journal of Telecommunication and Information Technologies*, 3, 9-15.

Kaliszewski I., (2003b), Dynamic parametric bounds on efficient outcomes in multiple criteria decision making. *European Journal of Operational Research*, 147, 94-107.

Kaliszewski I., (2004), Out of the mist – towards decision-maker-friendly multiple criteria decision making support. *European Journal of Operational Research*, 158, 293-307.

Kaliszewski I., Michalowski W., (1995), Generation of outcomes with selectively bounded tradeoffs. *Foundations of Computing and Decision Sciences*, 20, 113-122.

Kaliszewski I., Michalowski W., (1997), Efficient solutions and bounds on tradeoffs. *Journal of Optimization Theory and Applications*, 94, 381-394.

Kaliszewski I., Michalowski W., (1999), Searching for psychologically stable solutions of multiple criteria decision problems. *European Journal of Operational Research*, 118, 549-562.

Kaliszewski I., Michalowski W., Kersten G.E., (1997), A hybrid interactive technique for MCDM problems. In: *Essays in Decision Making*, (Karwan M., Spronk J., Wallenius J., eds.), Springer Verlag, Heidelberg, 48-59.

Kaliszewski I., Zionts S., (2004), Generalization of the Zionts-Wallenius multicriteria decision making algorithm. *Control & Cybernetics*, 3, 477-500.

Keeney R.L., Raiffa H., (1976), *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. John Wiley & Sons, New York.

Koktener-Karasakal E., Michalowski W., (2003), Incorporating wealth information into a multiple criteria decision making model. *European Journal of Operational Research*, 150, 204-219.

Korhonen P., (1988), A visual reference direction approach to solving discrete multiple criteria problems. *European Journal of Operations Research*, 34, 152-159.

Korhonen P., Laakso J., (1984), A visual interactive method for solving the multiple-criteria problem. In: *Interactive Decision Analysis*, (Grauer M., Wierzbicki A.P., eds.), Lecture Notes in Economics and Mathematical Systems, 229, Springer-Verlag, Berlin, 146-153.

Korhonen P., Laakso J., (1985), On developing a visual interactive multiple criteria method - an outline. In: *Decision Making with Multiple Objectives*, (Haimes Y.Y., Chankong V., eds.), Lecture Notes in Economics and Mathematical Systems, 242, Springer-Verlag, Berlin, 272-281.

Korhonen P., Laakso J., (1986), A visual interactive method for solving the multiple criteria problem. *European Journal of Operational Research*, 24, 227-287.

Korhonen P., Salo S., Steuer R., (1997), A heuristic for estimating nadir criterion values in multiple objective linear programming. *Operations Research*, 45, 751-757.

Kuhn H.W., Tucker A., (1951), Nonlinear programming. *Proceedings of the Second Berkeley Symposium on Mathematics, Statistics, and Probability*, University of California Press, Berkeley.

Lewandowski A., Wierzbicki A.P., (eds.), (1989), *Aspiration Based Decision Support Systems, Theory, Software, Applications*. Lecture Notes in Economics and Mathematical Systems, 331, Springer Verlag, Berlin.

Luc D.T., (1989), *Theory of Vector Optimization*. Lecture Notes in Economics and Mathematical Systems, 319, Springer Verlag, Berlin.

Makowski M., Somlyody L., Watkins D., (1996), *Multiple criteria analysis for water quality management in the Nitra basin*. Journal of the American Water Resources Association, 32, 937-942.

Markowitz H.M., (1959), *Portfolio Selection, Efficient Diversification of Investments*. John Wiley & Sons, New York.

Michalowski W., (1988), Use of the displaced worst compromise in interactive multiobjective programming. *IEEE Transactions on Systems, Man, and Cybernetics*, 18, 472-477.

Michalowski W., Szapiro T., (1989), A procedure for worst outcomes displacement in multiple criteria decision making. *Computers and Operations Research*, 16, 195-206.

Michalowski W., Szapiro T., (1992), A bi-reference procedure for interactive multiple criteria programming. *Operations Research*, 40, 247-258.

Miettinen K.M., (1999), *Nonlinear multiobjective optimization*. Kluwer Academic Publishers, Dordrecht.

Miettinen K.M., Mäkelä M.M., (1995), Interactive bundle-based method for nondifferentiable multiobjective optimization: NIMBUS. *Optimization*, 34, 231-246.

Miettinen K.,M., Mäkelä M.M., (1997), Interactive method NIMBUS for non-differentiable multiobjective optimization problems. In: *Multicriteria Analysis* (Clímaco J., ed.), Springer-Verlag, 310-319.

Nakayama H., (1989), Sensitivity and trade-off analysis in multiple objective programming. In: *Methodology and Software for Interactive Decision Support*, (Lewandowski A., Stanchev I., eds.), Lecture Notes in Economics and Mathematical Systems, 337, Springer-Verlag, Berlin, 86-93.

Nakayama H., (1995), Aspiration level approach to interactive multi-objective programming and its applications. In: *Advances in Multicriteria Analysis* (Pardalos P.M., Siskos Y., Zopoundis C., eds.), Kluwer Academic Publishers, Dordrecht, 147-174.

Nakayama H., Furukawa K., (1985), Satisficing trade-off method with an application to multiobjective structural design. *Large Scale Systems*, 8, 47-57.

Nakayama H., Sawaragi Y., (1984), Satisfacing trade-off method for multiobjective programming. In: *Interactive Decision Analysis* (Grauer M., Wierzbicki A.P., eds.), Lecture Notes in Economics and Mathematical Systems, 229, Springer-Verlag, Berlin, 113-122.

Nakayama H., Nomura J., Sawada K., Nakajima R., (1986), An application of satisficing trade-off method to a blending problem of industrial materials. In: *Large-Scale Modelling and Interactive Decision Analysis* (Fandel G., Grauer M., Kurzhanski A., Wierzbicki A.P., eds.), Lecture Notes in Economics and Mathematical Systems, 273, Springer-Verlag, Berlin, 303-313.

Narula S.C., Kirilov L., Vassilev V., (1994a), An interactive algorithm for solving multiple objective nonlinear programming problems. In: *Multiple Criteria Decision Making - Proceedings of the Tenth International Conference: Expand and Enrich the Domains of Thinking and Application* (Thzeng G.H., Wand H.F., Wen U.P., Yu P.L., eds.), Springer-Verlag, Berlin, 119-127.

Narula S.C., Kirilov L., Vassilev V., (1994b), Reference direction approach for solving multiple objective nonlinear programming problems. *IEEE Transactions on Systems, Man, and Cybernetics*, 24, 804-806.

Rietveld P., (1980), *Objective Decision Methods and Regional Planning*. North-Holland Publishing Company.

Rniguest J.L., (1992), *Multiobjective Optimization: Behavioral and Computational Considerations*. Kluver Academic Publishers, Boston.

Roy A., Wallenius J., (1991), Nonlinear and unconstrained multiple-objective optimization: algorithm, computation, and application. *Naval Research Logistics*, 38, 623-635.

Ruhe G., Fruhwirth B., (1989), $\epsilon$-optimality for bicriteria programs and its application to minimum cost flows. *Report 140, Institut für Mathematik*, Technische Universität Graz.

Sakawa M., (1982), Interactive mulitiobjective decision making by the Sequential Proxy Optimization Technique: SPOT. *European Journal of Operational Research*, 9, 386-396.

Sakawa M., Yano H., (1990), Trade-off rates in the hyperplane method for multiple-objective optimization. *European Journal of Operations Research*, 44, 105-118.

Sawaragi Y., Nakayama H., Tanino T., (1985), *Theory of Multiobjective Optimization*. Academic Press, New York.

Simon H.A., (1977), *The New Science of Management Decision*. Prentice-Hall, New Jersey.

Skulimowski A.M.J., (1996), *Decision Support Systems Based on Reference Sets*. AGH Publishers, Kraków.

Stadler W., (1979), A survey of multicriteria optimization or the vector optimization problem, Part I (1776-1960). *Journal of Optimization Theory and Applications*, 29, 1-52.

Steuer R.E. (1986), *Multiple Criteria Optimization: Theory, Computation and Application*. John Wiley & Sons, New York.

Steuer R.E., Choo E.U., (1983), An interactive weighted Tchebycheff procedure for multiple objective programming. *Mathematical Programming*, 26, 326-344.

Tabucanon M.T., (1988), *Multiple Criteria Decision Making in Industry*. Elsevier Science Publishers B.V., Amsterdam.

Vincke P., (1992), *Multicriteria Decision-Aid*. John Wiley & Sons, Inc., Chichester.

Weistroffer H.R., (1985), Careful usage of pesimistic values is needed in multiple objectives optimization. *Operations Research Letters*, 4, 23-25.

Wierzbicki A.P., (1977), Basic properties of scalarization functionals for multiobjective optimization. *Mathematische Operationsforschung und Statistik, ser. Optimization*, 8, 55-60.

Wierzbicki A.P., (1980), The use of reference objectives in multiobjective optimization. In: *Multiple Criteria Decision Making; Theory and Applications*, (Fandel G., Gal T., eds.), Lecture Notes in Economics and Mathematical Systems, 177, Springer Verlag, Berlin, 468-486.

Wierzbicki A.P., (1986), On the completeness and constructiveness of parametric characterizations to vector optimization problems. *OR Spectrum*, 8, 73-87.

Wierzbicki A.P., (1990), Multiple criteria solutions in noncooperative game theory, Part III: Theoretical Foundations. *Discussion Paper 288*, Kyoto Institute of Economic Research, Kyoto University, Kyoto.

Wierzbicki A.P., (1999), Reference point approaches. In: *Multicriteria Decision Making - Advances in MCDM: Models, Algorithms, Theory and Applications* (Gal T., Stewart Th., Hanne Th., eds.), Kluwer Academic Publishers.

Yang X.Q., Goh C.J., (1997), A method for curve approximation. *European Journal of Operational Research*, 97, 205-212.

Yoon K.P., Hwang C.-L., (1995), *Multiple Attribute Decision Making: An Introduction*. Sage Publications, Inc., Thousand Oaks.

Yu P.L., (1985), *Multiple Criteria Decision Making: Concepts, Techniques and Extensions*. Plenum Press, New York.

Zeleny M., (1976), The theory of displaced ideal. In: *Multiple Criteria Decision Making Kyoto 1975*, (Zeleny M., ed.), Lecture Notes in Economics and Mathematical Systems, 123, Springer Verlag, Berlin, 153-206.

Zeleny M., (1982), *Multiple Criteria Decision Making*. McGraw-Hill, Inc.

Zenios S., (1993), *Financial Optimization*. Cambridge University Press, Cambridge.

Zionts S., Wallenius J., (1976), An interactive programming method for solving the multiple criteria problem. *Management Science*, 22, 652-663.

Zionts S., Walenius J., (1983), An interactive multiple objective linear programming method for a class of underlying nonlinear value functions. *Management Science*, 29, 519-529.

# Sources of Quotations


Conrad Joseph, *Lord Jim*. Penguin Books, 1994.

Conrad Joseph, Author's Note to *The Secret Agent*. Wordsworth Editions Limited, 1993.

Milne A.A., *The House at Pooh Corner*. Bilingual edition by Prószyński i S-ka, Warsaw, 2001.

*Notices to Mariners*. Weekly Edition 12/2005, ENGLAND, South Coast, Falmouth to Plymouth, The United Kingdom Hydrographic Office.

Pareto Vilfredo, *Manuale di Economia Politica*. Societa Editrice Libraria, Milano, Italy, 1906, translated into English by Ann S. Schwier as *Manual of Political Economy*, Macmillan, New York, 1971.

Thoreau Henry David, *A Week on the Concord and Merrimac Rivers*. The New American Library of World Literature, Inc., 1961.

Wilde Oscar, *The Picture of Dorian Gray*. Barnes & Noble Classics Series, 2004.

# Index

***Early Titles in the***
## INTERNATIONAL SERIES IN OPERATIONS RESEARCH & MANAGEMENT SCIENCE
**Frederick S. Hillier, Series Editor,** *Stanford University*

Saigal/ *A MODERN APPROACH TO LINEAR PROGRAMMING*
Nagurney/ *PROJECTED DYNAMICAL SYSTEMS & VARIATIONAL INEQUALITIES WITH APPLICATIONS*
Padberg & Rijal/ *LOCATION, SCHEDULING, DESIGN AND INTEGER PROGRAMMING*
Vanderbei/ *LINEAR PROGRAMMING*
Jaiswal/ *MILITARY OPERATIONS RESEARCH*
Gal & Greenberg/ *ADVANCES IN SENSITIVITY ANALYSIS & PARAMETRIC PROGRAMMING*
Prabhu/ *FOUNDATIONS OF QUEUEING THEORY*
Fang, Rajasekera & Tsao/ *ENTROPY OPTIMIZATION & MATHEMATICAL PROGRAMMING*
Yu/ *OR IN THE AIRLINE INDUSTRY*
Ho & Tang/ *PRODUCT VARIETY MANAGEMENT*
El-Taha & Stidham/ *SAMPLE-PATH ANALYSIS OF QUEUEING SYSTEMS*
Miettinen/ *NONLINEAR MULTIOBJECTIVE OPTIMIZATION*
Chao & Huntington/ *DESIGNING COMPETITIVE ELECTRICITY MARKETS*
Weglarz/ *PROJECT SCHEDULING: RECENT TRENDS & RESULTS*
Sahin & Polatoglu/ *QUALITY, WARRANTY AND PREVENTIVE MAINTENANCE*
Tavares/ *ADVANCES MODELS FOR PROJECT MANAGEMENT*
Tayur, Ganeshan & Magazine/ *QUANTITATIVE MODELS FOR SUPPLY CHAIN MANAGEMENT*
Weyant, J./ *ENERGY AND ENVIRONMENTAL POLICY MODELING*
Shanthikumar, J.G. & Sumita, U./ *APPLIED PROBABILITY AND STOCHASTIC PROCESSES*
Liu, B. & Esogbue, A.O./ *DECISION CRITERIA AND OPTIMAL INVENTORY PROCESSES*
Gal, T., Stewart, T.J., Hanne, T. / *MULTICRITERIA DECISION MAKING: Advances in MCDM Models, Algorithms, Theory, and Applications*
Fox, B.L. / *STRATEGIES FOR QUASI-MONTE CARLO*
Hall, R.W. / *HANDBOOK OF TRANSPORTATION SCIENCE*
Grassman, W.K. / *COMPUTATIONAL PROBABILITY*
Pomerol, J-C. & Barba-Romero, S. / *MULTICRITERION DECISION IN MANAGEMENT*
Axsäter, S. / *INVENTORY CONTROL*
Wolkowicz, H., Saigal, R., & Vandenberghe, L. / *HANDBOOK OF SEMI-DEFINITE PROGRAMMING: Theory, Algorithms, and Applications*
Hobbs, B.F. & Meier, P. / *ENERGY DECISIONS AND THE ENVIRONMENT: A Guide to the Use of Multicriteria Methods*
Dar-El, E. / *HUMAN LEARNING: From Learning Curves to Learning Organizations*
Armstrong, J.S. / *PRINCIPLES OF FORECASTING: A Handbook for Researchers and Practitioners*
Balsamo, S., Personé, V., & Onvural, R./ *ANALYSIS OF QUEUEING NETWORKS WITH BLOCKING*
Bouyssou, D. et al. / *EVALUATION AND DECISION MODELS: A Critical Perspective*
Hanne, T. / *INTELLIGENT STRATEGIES FOR META MULTIPLE CRITERIA DECISION MAKING*
Saaty, T. & Vargas, L. / *MODELS, METHODS, CONCEPTS and APPLICATIONS OF THE ANALYTIC HIERARCHY PROCESS*
Chatterjee, K. & Samuelson, W. / *GAME THEORY AND BUSINESS APPLICATIONS*
Hobbs, B. et al. / *THE NEXT GENERATION OF ELECTRIC POWER UNIT COMMITMENT MODELS*
Vanderbei, R.J. / *LINEAR PROGRAMMING: Foundations and Extensions, 2nd Ed.*
Kimms, A. / *MATHEMATICAL PROGRAMMING AND FINANCIAL OBJECTIVES FOR SCHEDULING PROJECTS*
Baptiste, P., Le Pape, C. & Nuijten, W. / *CONSTRAINT-BASED SCHEDULING*
Feinberg, E. & Shwartz, A. / *HANDBOOK OF MARKOV DECISION PROCESSES: Methods and Applications*
Ramík, J. & Vlach, M. / *GENERALIZED CONCAVITY IN FUZZY OPTIMIZATION AND DECISION ANALYSIS*
Song, J. & Yao, D. / *SUPPLY CHAIN STRUCTURES: Coordination, Information and Optimization*
Kozan, E. & Ohuchi, A. / *OPERATIONS RESEARCH/ MANAGEMENT SCIENCE AT WORK*
Bouyssou et al. / *AIDING DECISIONS WITH MULTIPLE CRITERIA: Essays in Honor of Bernard Roy*

***Early Titles in the***
**INTERNATIONAL SERIES IN**
**OPERATIONS RESEARCH & MANAGEMENT SCIENCE**
*(Continued)*

Cox, Louis Anthony, Jr. / *RISK ANALYSIS: Foundations, Models and Methods*
Dror, M., L'Ecuyer, P. & Szidarovszky, F. / *MODELING UNCERTAINTY: An Examination of Stochastic Theory, Methods, and Applications*
Dokuchaev, N. / *DYNAMIC PORTFOLIO STRATEGIES: Quantitative Methods and Empirical Rules for Incomplete Information*
Sarker, R., Mohammadian, M. & Yao, X. / *EVOLUTIONARY OPTIMIZATION*
Demeulemeester, R. & Herroelen, W. / *PROJECT SCHEDULING: A Research Handbook*
Gazis, D.C. / *TRAFFIC THEORY*
Zhu/ *QUANTITATIVE MODELS FOR PERFORMANCE EVALUATION AND BENCHMARKING*
Ehrgott & Gandibleux/ *MULTIPLE CRITERIA OPTIMIZATION: State of the Art Annotated Bibliographical Surveys*
Bienstock/ *Potential Function Methods for Approx. Solving Linear Programming Problems*
Matsatsinis & Siskos/ *INTELLIGENT SUPPORT SYSTEMS FOR MARKETING DECISIONS*
Alpern & Gal/ *THE THEORY OF SEARCH GAMES AND RENDEZVOUS*
Hall/*HANDBOOK OF TRANSPORTATION SCIENCE - 2nd Ed.*
Glover & Kochenberger/ *HANDBOOK OF METAHEURISTICS*
Graves & Ringuest/ *MODELS AND METHODS FOR PROJECT SELECTION: Concepts from Management Science, Finance and Information Technology*
Hassin & Haviv/ *TO QUEUE OR NOT TO QUEUE: Equilibrium Behavior in Queueing Systems*
Gershwin et al/ *ANALYSIS & MODELING OF MANUFACTURING SYSTEMS*

**** A list of the more recent publications in the series is at the front of the book ****