Global Optimization

MOS-SIAM Series on Optimization

This series is published jointly by the Mathematical Optimization Society and the Society for Industrial and Applied Mathematics. It includes research monographs, books on applications, textbooks at all levels, and tutorials. Besides being of high scientific quality, books in the series must advance the understanding and practice of optimization. They must also be written clearly and at an appropriate level for the intended audience.

Series Volumes

Locatelli, Marco and Schoen, Fabio, *Global Optimization: Theory, Algorithms, and Applications*

De Loera, Jesús A., Hemmecke, Raymond, and Köppe, Matthias, *Algebraic and Geometric Ideas in the Theory of Discrete Optimization*

Blekherman, Grigoriy, Parrilo, Pablo A., and Thomas, Rekha R., editors, *Semidefinite Optimization and Convex Algebraic Geometry*

Delfour, M. C., *Introduction to Optimization and Semidifferential Calculus*

Ulbrich, Michael, *Semismooth Newton Methods for Variational Inequalities and Constrained Optimization Problems in Function Spaces*

Biegler, Lorenz T., *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes*

Shapiro, Alexander, Dentcheva, Darinka, and Ruszczyński, Andrzej, *Lectures on Stochastic Programming: Modeling and Theory*

Conn, Andrew R., Scheinberg, Katya, and Vicente, Luis N., *Introduction to Derivative-Free Optimization*

Ferris, Michael C., Mangasarian, Olvi L., and Wright, Stephen J., *Linear Programming with MATLAB*

Attouch, Hedy, Buttazzo, Giuseppe, and Michaille, Gérard, *Variational Analysis in Sobolev and* BV *Spaces: Applications to PDEs and Optimization*

Wallace, Stein W. and Ziemba, William T., editors, *Applications of Stochastic Programming*

Grötschel, Martin, editor, *The Sharpest Cut: The Impact of Manfred Padberg and His Work*

Renegar, James, *A Mathematical View of Interior-Point Methods in Convex Optimization*

Ben-Tal, Aharon and Nemirovski, Arkadi, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*

Conn, Andrew R., Gould, Nicholas I. M., and Toint, Phillippe L., *Trust-Region Methods*

Global Optimization

Theory, Algorithms, and Applications

Marco Locatelli
Università degli Studi di Parma
Parma, Italy

Fabio Schoen
Università degli Studi di Firenze
Firenze, Italy

siam.
Society for Industrial and Applied Mathematics
Philadelphia

Mathematical Optimization Society
Mathematical Optimization Society
Philadelphia

10 9 8 7 6 5 4 3 2 1

Library of Congress Cataloging-in-Publication Data

Locatelli, Marco.
Global optimization : theory, algorithms, and applications / Marco Locatelli, Università degli Studi di Parma, Parma, Italy, Fabio Schoen, Università degli Studi di Firenze, Firenze, Italy.
pages cm. – (MOS-SIAM series on optimization)
Includes bibliographical references and index.
ISBN 978-1-611972-66-5
1. Nonconvex programming. I. Schoen, Fabio. II. Title.
QA402.5.L63 2013
519.7–dc23
2013013213

Contents

Preface

The first systematic overviews on global optimization appeared in 1975–1978 thanks to two fundamental volumes titled *Towards Global Optimization* (Dixon & Szegö, 1975, 1978). At that time, and for many years since, global optimization was considered a quite "exotic" subject, too difficult to be of interest to those looking for nice theories and robust algorithms. Since then, thanks to those pioneering contributions and to the effort of many researchers, things have significantly changed. Today global optimization is considered a top level, dynamic, deep subject. Many of the most important scientific journals (most but not all in the field of mathematical programming and optimization) publish papers on this subject, and the *Journal of Global Optimization*, which has been published since 1991, is completely dedicated to it.

The field has grown so rapidly and interest in the scientific community has spread so widely that it would be quite difficult to introduce in an organic and complete way all of its numerous ramifications. It is so dynamic that, while preparing this book, at some point we had to stop consulting the literature; otherwise the lengthy process of writing would never have been stopped.

We had to make many choices concerning what to include and what to leave out, and thus some subjects have been left out or less extensively discussed. This should not be taken as an indication that they are not relevant. Also, although we tried to be consistent and up-to-date, it is certain that we missed something and perhaps reported some subjects in a way which differs from what the original authors meant. Please take into account that this book reflects our personal view of some topics in global optimization. Any error or omission was made in good faith, and is our responsibility.

It took us more than two years to prepare this material. It has been an exciting period of our lives; trying to explain a subject is the best way we know for learning it. We wish to express our gratitude to all SIAM staff for giving us such an opportunity and for assisting us during the preparation of the manuscript. In particular we are grateful to Philippe Toint, who first proposed and encouraged us to write this book; to Tom Liebling, who acted as editor and helped us with many stimulating suggestions; to the three anonymous referees, who had the patience and competence to read several versions of the book and who led us to introduce many improvements; and to Sara Murphy, who followed us so closely in all these years with great patience and professional support. Of course, we are grateful also to all the people at SIAM who were involved in producing the printed volume.

Finally, some acknowledgments to our families. ML is grateful to Ce, Desi, Noe, and Popy for all the support they gave by simply being around. He is also grateful to his and Cecilia's families and, in particular, to nonne Luisa and Vittoria. FS is grateful to Cristina, Giulio, and Carlotta for their constant support and for being around (but never too much).

Chapter 1

Introduction

1.1 Why we wrote this book and how to read it

This book is devoted to Global Optimization (GO), a wide, relevant, fascinating subject that has become, mostly in recent years, a hot topic in the optimization community and in scientific research in general. Until not so many years ago the subject was quite neglected and most, if not all, research effort was devoted to local optimization. Recently, thanks to the development of new and powerful algorithmic ideas and to the advances in the theoretical analysis and in the evaluation of the complexity of optimization problems in general, the field has seen a tremendous amount of research. Just to give a quantitative idea of such a growth, we notice that the search for the words *global optimization* in `Scopus` returned 1,895 documents in the decade 1990–1999, 10,032 in the decade 2000–2009, and 3,588 from 2009 to 2012. Besides giving an idea of the growing interest in GO, such numbers should also make clear that the few hundred pages of this book are certainly not enough to cover all the possible topics related to the theory, algorithms, and applications arising in this field. On one hand, we felt this was the right moment to write a book on this subject, to collect and organize many research streams in the field, but on the other hand, we immediately realized that an exhaustive listing of all the possible topics was too hard a task to be accomplished within the pages of a single book. We had to make some choices, and thus the book is influenced by our own personal taste, and the selection of topics derives from our own feelings about the research ideas that we consider most relevant within the field. We devoted a lot of effort to organizing the vast material in such a way that the interested reader can find a clear set of directions in understanding the theory behind GO and in choosing the most appropriate method for the solution of relevant optimization problems. With this in mind, we started this book with a chapter on complexity. This is a very natural choice in any book covering practical methods; when confronted with a problem to be solved, the first thing we should do is try to understand the characteristics of the problem and, possibly, its complexity. This analysis might lead to the discovery of a similar problem that admits a relatively efficient solution method. This would guide us toward an exact algorithm or an approximate one, where by "approximate" we mean a method that, although not guaranteed to discover a global optimum, nonetheless provides an exact bound on the quality of the returned solution. On the other hand, as happens in the vast majority of practical problems, the analysis might reveal that the problem is inherently intractable, so that the natural choice will be that of a suitable heuristic method. The chapter on complexity is inevitably quite a technical one; although we think it is extremely relevant for any researcher in the field, we also think that it might be omitted when first reading the book. In fact, many different reading paths might be suggested. As an example, a reader mostly interested in detecting a good solution for a specific, difficult GO problem, but without the need of certifying the quality of the solution itself, might go directly to Chapter 3 in search of a suitable method. The chapter is not conceived as a pure list of methods, but instead is organized in such a way that all GO heuristics are introduced as specializations of a few general schemes. The reader interested in solving a single specific problem will find in that chapter suggestions leading to a good tool. In particular, it will become clear that there is no single method capable of solving all problems and that some of the characteristics of the problem should be considered. As an example, is function evaluation expensive, requiring hours or days of CPU time? Alternatively, is it based on an expensive experiment to be performed, such as a car crash test? If this is the case, some methods are particularly suitable, while others are out of the question. On the other hand, if function evaluation is cheap, local optimization may also be cheap; in this case it is strongly advisable that a GO method exploits the power of local optimization tools. Thus, Chapter 3 guides the reader through an organized tour across different methods, whose use depends on the problem characteristics.

If the reader is not merely interested in detecting a solution but also in certifying its quality, then he should go through Chapters 4 and 5. Chapter 4 deals with the derivation of lower bounds (for minimization problems). Once again, the chapter is structured in such a way that the reader can find suggestions about how to compute lower bounds for different problem classes, ranging from highly structured ones (e.g., those involving quadratic functions) up to mildly structured ones, for which there is a more limited choice of methods to compute lower bounds. Chapter 5 is devoted to branch-and-bound (BB) methods, which represent the usual choice for the exact solution of GO problems. Some of the main ingredients of BB methods are discussed in Chapters 3 and 4, while in Chapter 5 the other relevant ingredients, such as the branching rules, are presented in detail. Again, the reader is guided through different possibilities, depending on the different problem classes. Chapter 5 also deals with the theoretical issue of the convergence (and, in some cases, finiteness) of BB methods.

Finally, Chapter 6 is devoted to problems and applications. It is divided into two parts. The first part discusses benchmark test problems. This might be of interest to those willing to develop a new algorithm and compare its performance with respect to standard GO tools. It is mainly a part devoted to researchers in the field. Of course, this part might be skipped if the interest is in using an existing method and not developing a new one. The second part describes some applications of GO. We had a broad range of possible applications among which to choose. We decided to give a brief account of a few problems that we studied in recent years: finding the conformation of a molecule (or of an atomic cluster), finding the optimal packing of disks (or spheres, or other shapes) in a container, and planning an optimal trajectory for interplanetary space missions. These problems are very relevant in their own field of application and very difficult from a computational point of view. Moreover, they provide a very good set of testbeds for some of the algorithms introduced in this book. While the rest of the book is strongly influenced by the theories and methods that we considered the most interesting ones, Chapter 6 does not suggest that these are the most relevant applications of GO. It is a set of personal stories, in which we just wanted to let the reader know that by using some general ideas described in the book, coupled with some special purpose tricks tailored to the problem at hand, many difficult and, in our opinion, extremely fascinating problems can be tackled successfully.

1.2 Basic terminology and outline of the book

Given a function f, called *objective function*, and a set $S \subseteq \mathbb{R}^n$, called *feasible region*,

$$\min_{x \in S} f(x) \tag{1.1}$$

is called a GO problem. Throughout the book, unless otherwise stated, we will only deal with global minimization problems, in view of the trivial conversion of any maximization problem into a minimization one. GO deals with the theoretical and algorithmic aspects involved in the detection of *global minima* of the problem, i.e., points $x^* \in S$ such that

$$f(x^*) \leq f(x) \quad \forall\, x \in S.$$

Throughout the book we will always assume, unless otherwise stated, that f is a continuous function. Moreover, S is usually assumed to be a nonempty compact set, so that the Weierstrass theorem guarantees the existence of at least one global minimum. The introduction of the concept of *local minimum*, i.e., any point $\bar{x} \in S$ such that for some $\varepsilon > 0$

$$f(\bar{x}) \leq f(x) \quad \forall\, x \in S \;:\; \|x - \bar{x}\|_2 \leq \varepsilon,$$

allows us to define the scope of GO more precisely. The general problem (1.1) includes many subclasses that usually should *not* be included in the field of GO, namely all those problems, like the convex ones, for which any local minimum of f over S is also a global minimum. The scope of GO should be restricted to problems that fulfill the further requirement that f is a multimodal function over S, having local minima over S which are not global ones.

As an introductory example, we present a problem that is very simple to describe, but at the same time very hard to solve: the disk packing problem, where n rigid disks with equal radius r have to be placed within a given region $D \subset \mathbb{R}^2$ in such a way that they do not overlap and their common radius is maximized. The problem can be easily transformed into an equivalent one that consists in displacing n points so that their minimum pairwise distance is maximized. If D is the unit square, the disk packing problem can be formulated as a GO one with the feasible set

$$S = \left\{ (x_1, y_1, \ldots, x_n, y_n, r) \;:\; \begin{array}{l} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \geq r, \ \forall i, j = 1,n, \ i < j \\ x_i, y_i \in [r, 1-r], \ \forall\, i = 1,n \end{array} \right\}$$

and the objective function

$$f(x_1, y_1, \ldots, x_n, y_n, r) = -r.$$

Its equivalent formulation as a dispersion problem is characterized by the feasible set

$$S = \{(x_1, y_1, \ldots, x_n, y_n) : \ x_i, y_i \in [0,1], \ i = 1, \ldots, n\},$$

and the objective function is

$$f(x_1, y_1, \dots, x_n, y_n) = - \min_{i,j=1,\dots,n,\ i<j} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}.$$

In spite of the simplicity of its description, this problem has a huge number of local minima that are not global ones. The study of this problem led to some surprising results. In particular, for $D = [0,1]^2$ and $n = k^2$, and $k \in \{2, \dots, 6\}$, the regular $k \times k$ grid has been geometrically proved to be the global minimum of the problem; for $k = 7$ it was computationally shown that the regular grid is not globally optimal. In fact, in Figure 1.1 a solution is reported in which the radius r is approximately equal to 0.07169268, which is strictly larger than the radius $1/14$ associated to the regular grid. We refer to Section 6.2.3 for more details about this problem.

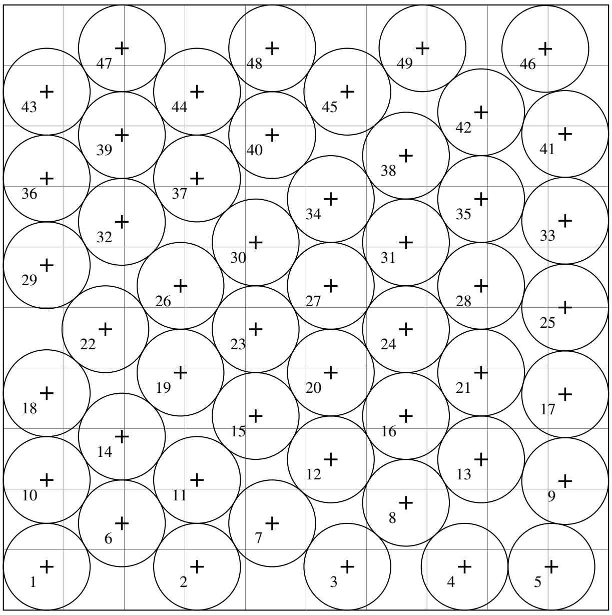


Figure 1.1. *Putative optimal packing of* 49 *disks in the unit square*

As previously remarked, we could not cover in detail all the topics related to GO. Some of them, such as global optimality conditions (for which we can refer to the survey (J.-B. Hiriart-Urruty, 1995) or to papers such as (J.-B. Hiriart-Urruty, 1998; Jeyakumar, Rubinov, & Wu, 2007; Jeyakumar & Li, 2011; Rubinov & Wu, 2009)) are not discussed in this book. Some other topics are only mentioned without deeply investigating them. These include, e.g., mixed integer nonlinear programming (MINLP), a subject which is strictly related to GO and has attracted a lot of research in the recent years (see, e.g., (Burer & Letchford, 2012; Lee & Leyffer, 2011) for recent reviews about MINLP). After mentioning what we have *not* done, in what follows we state what we have done and give an outline of the book, briefly discussing the contents of each chapter.

Chapter 2 deals with the theoretical issue of complexity. The difficulty of GO problems becomes immediately evident after the simple observation that any binary problem can be reformulated as a GO one. However, in this chapter we show that the class of GO problems is very wide and includes many subclasses, differing from each other in view of the properties of the objective function and/or of the feasible region. Consequently, they have different difficulty levels, ranging from subclasses for which solutions can be obtained in polynomial time, up to subclasses for which even the detection of an approximate solution with an arbitrary low precision is a hard task, and going through other intermediate subclasses, namely those for which a fully polynomial time approximation scheme (FPTAS) is available, those for which a polynomial time approximation scheme (PTAS) is available, and those for which approximate solutions can be obtained in polynomial time only if the required precision is not too high. Those readers more interested in the practical aspects of GO may probably skip this chapter without compromising their understanding of the following ones. However, though mostly of theoretical interest, Chapter 2 gives some indications of the suitability of applying exact methods to GO subclasses: the higher the difficulty level of a subclass is, the lower is the dimension of the problem instances within the subclass for which we can expect to efficiently detect a globally optimal solution by some exact method, i.e., a method that also certifies the global optimality of the solution.

The complexity issues discussed in Chapter 2 show that heuristic methods, i.e., methods which do not give any guarantee about the quality of the returned solution, are always an option (sometimes the only reasonable one) for GO problems. Chapter 3 is dedicated to the description of heuristic approaches for GO problems. A distinction is made between GO problems for which function evaluations and, possibly, local searches are relatively cheap, and GO problems where even a single function evaluation is rather costly. For the former class of GO problems, we have followed a top-down approach: instead of discussing for each existing heuristic approach its different variants presented in the literature (whose number is often very large), we have first tried to identify the common aspects of the existing heuristic approaches. To this end we define a meta-heuristic whose individual steps are shared by the existing heuristics, and later on we discuss the different ways to implement such steps in the different heuristics. For each heuristic we have introduced the basic principles and given some bibliographic hints for the reader interested in the details. In the latter class of GO problems, the high cost of a single function evaluation changes the perspective when defining a solution method: in this case much computational effort is dedicated to the choice of the points where the observation of function values should be placed. This is done by building models of the function based on previous observations. Due to the high cost of function evaluations, both building the models and exploring them to detect the next point at which to place the next observation may be (and, in fact, usually are) expensive tasks. The final section of this chapter is dedicated to the theoretical issue of convergence of the heuristics if no stopping rule is adopted. The intuitive result that for poorly structured problems convergence can be guaranteed only if the set of observed points is dense within the feasible region is formalized in this section.

Computing lower bounds for GO problems is a relevant task both in order to provide a measure of the quality of the solution returned by a heuristic approach and as part of exact GO methods. Chapter 4 is dedicated to this issue. The ability of returning lower bounds for the optimal value of the GO problem (1.1) is strictly related to the ability of defining polynomially solvable (usually convex) relaxations of the problem. The notion of best possible convex underestimator of a function over some region, or convex envelope, is extensively discussed in the chapter. Next, different techniques to derive convex relaxations and, thus, lower bounds are discussed. These include the reformulation-linearization technique (RLT), the α-BB approaches, bounds based on the knowledge of the Lipschitz constant, dual Lagrangian bounds, bounds based on interval arithmetic, and McCormick relaxations. Relaxations for specific GO subclasses are also discussed. In particular, attention is devoted to problems involving quadratic functions, but other subclasses, such as polynomial, difference-of-convex (DC), and difference-of-monotonic problems, have also been taken into consideration.

In spite of the already-mentioned difficulty of GO problems, exact methods can be sometimes applied for solving instances of moderately large dimension belonging to some (usually highly structured) GO subclasses. Except for some problems with rather special properties, most of the exact methods for GO problems are BB methods. This is the subject of Chapter 5. Following an approach similar to the one followed for the heuristic techniques, we first present a rather general BB method, and later on we discuss the individual steps of the method and the different ways these have been implemented in the literature. In fact, one of these steps, the lower bound computation, is already extensively discussed in Chapter 4. A large section is devoted to the discussion of different branching strategies. Conditions under which the method converges to a globally optimal solution or returns such a solution (or, at least one “close enough” to it) after a finite time are discussed. Also some operations such as domain reduction strategies, which are not strictly necessary for proving convergence or finiteness results but often have a strong impact on the practical performance of the method, are illustrated in this chapter.

Chapter 6 is dedicated to a discussion about GO test problems and applications. In both cases the discussion is certainly not exhaustive. Our intention was not to list all known test functions, but rather to give some hints for a correct choice of the set of test functions. Also, as paper reviewers, we have often observed that test problems are not always carefully selected. Sometimes test problems which may still make sense to assess the validity of GO problems like, e.g., those for expensive functions, are used to validate, e.g., gradient-based methods even if some trivial gradient-based methods are already extremely efficient in solving these problems. As an example of this, we selected a large set of test problems from the literature and applied to them the most natural and simple method based on local searches, the Multistart method, showing how many of the problems are easily solved by this approach and do not represent a valid choice, at least for gradient-based GO methods. Although we have not discussed in detail specific test problems, we have given references to the existing literature and for some classes of test problems we have briefly discussed the challenging aspects of the class. For applications, once again we have not given a complete list of them, a task which is presumably very hard to accomplish given the large number of publications in this field. Instead, after providing some pointers to the existing literature, driven by our personal experiences, we have discussed in some detail four applications, namely molecular conformation, distance geometry, packing, and space trajectory planning problems.

The book also contains Appendix A, where we briefly review some basic notions and results about convexity, and Appendix B, where we list some frequently used symbols.

Finally, we would like to refer the reader to the book’s companion web site, http://www.siam.org/books/mo15, wherein we offer extensions and additional material.

Chapter 2

Complexity

2.1 Introduction

Before discussing more practical issues about GO, in this chapter we deal with a question which is mostly theoretical: how difficult is it to solve GO problems? In fact, the question can be easily answered. Any binary optimization problem

$$\begin{aligned} \min \quad & \mathbf{c}^T \mathbf{x} \\ & \mathbf{A}\mathbf{x} \leq \mathbf{b}, \\ & \mathbf{x} \in \{0,1\}^n, \end{aligned}$$

for which it is well known that the detection of an optimal (and, in some cases, also approximate) solution is a hard task (see, e.g., Papadimitriou & Steiglitz, 1998), can be converted into a GO one by replacing the binary constraints

$$x_i \in \{0,1\}, \quad i = 1, \ldots, n,$$

with the quadratic equations

$$x_i(1 - x_i) = 0, \quad i = 1, \ldots, n.$$

Thus, solving GO problems is at least as difficult as solving binary optimization problems. However, we can get into more details and answer the following other question: does the structure help in reducing the difficulty of GO problems? Or, stated in another way, are there “simple” or, at least, “not too difficult” subclasses of GO problems? The answer is yes, and in what follows we will discuss some of these subclasses and their complexity. As we will see, all the subclasses are characterized by objective functions and/or feasible regions of relatively simple form.

2.2 Preliminary notions

In the following discussion we will take as understood some basic notions about complexity, such as the definitions of the classes $\mathcal{P}$ and $\mathcal{NP}$, or those of $\mathcal{NP}$-complete and

$\mathcal{NP}$-hard problems (we refer the reader to existing textbooks about the subject, such as Arora & Barak, 2009; Garey & Johnson, 1979; Papadimitriou & Steiglitz, 1998). However, in what follows, some other useful preliminary notions will be recalled.

2.2.1 Models of computation

For most of the subclasses we are going to discuss, the *bit model of computation* will be suitable. In the bit model the input must be encoded as a string of symbols. For instance, for the class of *quadratic programming* (QP) problems

$$\begin{aligned} \min \quad & \tfrac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x}+\mathbf{c}^T\mathbf{x} \\ & \mathbf{A}\mathbf{x} \geq \mathbf{b}, \end{aligned}$$

where a quadratic function is minimized over a polyhedral region, the input is made up by the quadruplet

$$\mathbf{Q} \in \mathbb{Q}^{n\times n}, \ \ \mathbf{c} \in \mathbb{Q}^{n}, \ \ \mathbf{A} \in \mathbb{Q}^{m\times n}, \ \ \mathbf{b} \in \mathbb{Q}^{m},$$

where $\mathbb{Q}$ denotes the set of rational numbers. The following are some standard definitions.

Definition 2.1. *A class of optimization problems is a collection $\mathcal{I}$ of instances. Each instance $I \in \mathcal{I}$ is represented by a pair (S_I, f_I), where S_I is the* feasible region *of the instance, and f_I is the* objective function *of the instance. An instance is solved if a point $\mathbf{x}_I^* \in S_I$ is detected which minimizes f_I over S_I, or it is otherwise established that no such point exists. The point $\mathbf{x}_I^*$ is called the* optimal solution *of the instance, while $f_I(\mathbf{x}_I^*)$ is called an* optimal value *of the instance.*

Note that the above definition is for minimization problems. Of course, a completely analogous definition can be given for maximization problems. For the class of QP problems all instances can be obtained by varying the quadruplet $(\mathbf{Q}, \mathbf{c}, \mathbf{A}, \mathbf{b})$.

Definition 2.2. *An algorithm $\mathcal{A}$ is said to solve a class $\mathcal{I}$ of optimization problems if, when receiving the input instance $I \in \mathcal{I}$, it returns an optimal solution of I or, otherwise, establishes that no such solution exists. The number of elementary operations (sum, product, and so on) required by $\mathcal{A}$ in order to solve an instance I is denoted by $\#op_{\mathcal{A}}(I)$.*

Definition 2.3. *For a given class of optimization problems $\mathcal{I}$, the dimension of an instance $I \in \mathcal{I}$ is denoted by $dim(I)$ and is the length of a binary encoding of the input data defining the instance.*

For the class of QP problems the dimension of an instance is the length of the binary encoding of the quadruplet $(\mathbf{Q}, \mathbf{c}, \mathbf{A}, \mathbf{b})$ defining the instance.

Definition 2.4. *The (worst-case) complexity of an algorithm in the bit model of computation is defined by the following function of the dimension:*

$$t_{\mathcal{A}}(k) = \max_{I\in\mathcal{I}\,:\,dim(I)=k} \#op_{\mathcal{A}}(I);$$

i.e., for a given dimension k, $t_{\mathcal{A}}(k)$ is the maximum number of elementary operations required by $\mathcal{A}$ in order to solve instances whose dimension is k.

The bit model has a limited application and can usually be employed only with problems involving polynomial or ratio functions with rational coefficients. For more general problems the *black box model of computation* can be used. In such a model the input is made up by some subroutines which allow, e.g., the computation of the values of the functions defining an instance of the problem, and the related subgradients, at a given point (it is assumed that exact real arithmetic operations can be performed). Each call of a subroutine has a unit cost, and the complexity of a solution algorithm for the problem is measured in terms of the dimension, the number of subroutine calls, and the required precision ε (maximum allowed difference between the function value at the point returned by the algorithm and the global minimum value). The black box model is the one usually employed to study the complexity of convex programming problems, i.e., problems where both the objective function and the feasible set are convex (see, e.g., (Ben-Tal & Nemirovski, 2001; Nemirovski & Yudin, 1983)). Under mild assumptions, convex programming problems are solvable in polynomial time. For this reason such problems are classified as "easy" and are used to define relaxations of more difficult nonconvex problems (see Chapter 4).

Both models of computation might also include some degree of randomness through, e.g., some uniformly generated random bits, thus giving rise to so-called *randomized algorithms*. For a randomized algorithm, the objective function value of the solution returned by the algorithm itself is a random variable whose *expected* value is computed in order to evaluate the quality of the algorithm.

2.2.2 Decision and optimization problems

Traditional complexity theory deals with decision problems rather than optimization problems. In a decision problem a further input datum ξ (required to be rational in the bit model of computation) has to be considered. Then, the decision problem is defined as follows.

Definition 2.5. *Let an objective function f, a feasible region X, and a value ξ be given. The* decision problem *asks whether there exists a point $\bar{\mathbf{x}} \in X$ such that $f(\bar{\mathbf{x}}) \leq \xi$.*

Note that the output of a decision problem is either a *yes* or a *no*. If we are able to solve an optimization problem, we are obviously also able to solve the corresponding decision problem with the same objective function and feasible region for any ξ. On the other hand, if a lower and an upper bound for the optimal value are available, a binary search allows solving, at least within a given accuracy, an optimization problem through the solution of the corresponding decision problem for different ξ values. Although, for the sake of precision, when dealing with complexity issues we should consider decision problems, in view of the strict relation between such problems and the optimization problems, in what follows we will always refer to the latter.

2.2.3 Approximation problems

Once $\mathcal{NP}$-hardness of an optimization problem has been established, we might wonder how difficult it is to detect *approximate* solutions with a theoretical guarantee of a bounded distance from the optimal value. Different notions of *approximate solutions* exist. Here we report the following two definitions for a given GO problem $\min_{\mathbf{x}\in X} f(\mathbf{x})$ with minimum and maximum values over X denoted by f_* and f^*, respectively.

Definition 2.6. *Given $\varepsilon \in [0,1]$, a point $\bar{\mathbf{x}} \in X$ is said to be an ε-approximate solution for the problem if*

$$f(\bar{\mathbf{x}}) - f_* \le \varepsilon(f^* - f_*).$$

Definition 2.7. *Given $\varepsilon \ge 0$, a point $\bar{\mathbf{x}} \in X$ is said to be an ε-approximate solution for the problem if*

$$f(\bar{\mathbf{x}}) - f_* \le \varepsilon |f_*|.$$

Definition 2.6 is based on the range of the function values over the feasible region. It has the advantage of being invariant both with respect to translations and with respect to scaling of the objective function. Unfortunately, it becomes meaningless in all those cases for which f is unbounded from above over X. Definition 2.7 is only invariant with respect to scalings of the objective function. Furthermore, when $f_* = 0$ the ε-approximation problem reduces to the optimization problem. However, such a definition is still meaningful when f is unbounded from above over X. Unless otherwise stated, throughout this chapter we will always refer to Definition 2.6.

Through ε-approximation problems we are able to recognize different levels of difficulty for $\mathcal{NP}$-hard problems. At the first, simplest, level we have all problems admitting a fully polynomial time approximation scheme, which is defined below.

Definition 2.8. *A* fully polynomial time approximation scheme (FPTAS) *for a problem is a class of algorithms $\{\mathcal{A}_\varepsilon\}$, $\varepsilon > 0$, such that for each fixed $\varepsilon > 0$, the algorithm $\mathcal{A}_\varepsilon$ returns an ε-approximate solution of the problem (with respect to one of the given definitions of approximate solution) in a number of elementary operations which is polynomial with respect to the problem dimension* and with respect to the inverse of the required precision $\frac{1}{\varepsilon}$ *in the bit model. (In the black box model the same definition applies with the number of elementary operations replaced by the number of subroutine calls.)*

If polynomiality with respect to $\frac{1}{\varepsilon}$ cannot be achieved, then we are led to the second level, that of problems admitting a polynomial time approximation scheme.

Definition 2.9. *A* polynomial time approximation scheme (PTAS) *for a problem is a class of algorithms $\{\mathcal{A}_\varepsilon\}$, $\varepsilon > 0$, such that for each fixed $\varepsilon > 0$, the algorithm $\mathcal{A}_\varepsilon$ returns an ε-approximate solution of the problem (with respect to one of the given definitions of approximate solution) in a number of elementary operations which is polynomial with respect to the problem dimension in the bit model. (Again, in the black box model the same definition applies with the number of elementary operations replaced by the number of subroutine calls.)*

The third level includes problems for which the ε-approximation problem is solvable in polynomial time as soon as ε is large enough, giving rise to the class $\mathcal{APX}$, which is an abbreviation of "approximable."

Definition 2.10. *The class $\mathcal{APX}$ is composed of problems for which there exists a sufficiently large constant value ε such that the ε-approximation problem is solvable in polynomial time.*

Finally, the fourth level includes all problems that do not belong to $\mathcal{APX}$, i.e., all problems for which ε-approximation is $\mathcal{NP}$-hard for every fixed ε ($\varepsilon < 1$, if Definition 2.6 is employed).

In the following sections we will discuss GO problem classes of increasing difficulty, from those solvable in polynomial time (Section 2.3) up to those for which even detecting an approximate solution is a difficult task (Section 2.6), going through problem classes admitting a FPTAS (Section 2.4) and a PTAS (Section 2.5). In each section we will mention results for different problem classes, but details will be given only for one such class. In Section 2.7 we will give an overview of complexity results for problems with a quadratic objective function and different feasible regions. Before proceeding, we also point out that good surveys on the complexity of GO problems can be found in (de Klerk, 2008) and (Vavasis, 1995).

2.3 "Simple" GO problems

While detecting the global minimum of a multimodal function is, in general, a difficult task, in some cases the special structure of the objective function and of the feasible region makes the task relatively simple. For some classes of GO problems one might exploit prior knowledge about the position of global minima. A typical example is the minimization of a *quasiconcave function* (see Definition A.27) over a polytope. The following result can be proven.

Theorem 2.11. *The minimum of a quasiconcave function f over a compact convex set $X \subset \mathbb{R}^n$ is attained at an extreme point of X.*

Proof. In view of Carathéodory's theorem (see, e.g., (Rockafellar, 1970)), each point $\mathbf{x} \in X$ can be obtained as the convex combination of at most $n+1$ extreme points $\mathbf{v}^j$, $j = 1,\dots,n+1$, of X. It is readily seen that by definition of the quasiconcave function,

$$f(\mathbf{x}) \geq \min\{f(\mathbf{v}^j) \ : \ j = 1,\dots,n+1\}.$$

Therefore, for each $\mathbf{x} \in X$ which is *not* an extreme point, there always exists at least an extreme point of X whose function value is not larger than $f(\mathbf{x})$. □

For a polyhedral region the extreme points are the vertices of the region, and the above result shows that the search for a global minimum over a polytope can be restricted to such vertices. Therefore, if the feasible region X is a polytope with a number of vertices polynomial with respect to n, then a simple enumeration of the function values at the vertices of X allows us to detect the global minimum after a polynomial number of function evaluations. This happens, e.g., when the feasible region is an n-dimensional simplex with vertices $\mathbf{v}^1,\dots,\mathbf{v}^{n+1}$:

$$X = \left\{ \sum_{j=1}^{n+1} \lambda_j \mathbf{v}^j \ : \ \sum_{j=1}^{n+1} \lambda_j = 1, \ \lambda_j \geq 0, \ j = 1,\dots,n+1 \right\}.$$

In this case $n+1$ function evaluations are enough to detect a global minimum. In some other cases the feasible region, though of simple form, has an exponential number of vertices with respect to n. This happens, e.g., when $X = \prod_{i=1}^n [a_i, b_i]$ is an n-dimensional box (with 2^n vertices). In this case we cannot always guarantee that the global minimum can be detected in polynomial time even for concave quadratic objective functions (see Section 2.6). However, further restrictions on the objective function can lead to the efficient detection of a global minimum. For instance, if

$$f(\mathbf{x}) = \sum_{i=1}^{n} f_i(x_i),$$

i.e., f is a separable function and each one-dimensional function f_i is assumed to be quasiconcave, then the coordinates x_i^*, $i = 1, \ldots, n$, of a global minimum point can be computed as

$$x_i^* \in \arg\min\{f_i(x_i) \;:\; x_i \in \{a_i, b_i\}\};$$

i.e., a global minimum is detected by $2n$ evaluations of the one-dimensional functions.

While the above subclasses of GO problems are quite simple, there are other subclasses of GO problems for which the ability to solve them in polynomial time is less obvious. Such subclasses usually have a *hidden convexity* property. In other words, although nonconvex, such problems admit a convex reformulation. The best known example is the so-called *trust region* (TR) problem, where a quadratic function is minimized over an n-dimensional sphere, i.e., for some $\rho > 0$,

$$\begin{array}{ll} \min & \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x} \\ & \|\mathbf{x}\|_2^2 \leq \rho. \end{array} \tag{2.1}$$

The name "TR problem" comes from the fact that such a problem arises as a subproblem which has to be solved at each iteration of a TR method for local optimization (see, e.g., (Fletcher, 2000; Conn, Gould, & Toint, 2000)). In (Vavasis & Zippel, 1990) it was proven that the decision version of this problem is solvable in polynomial time in the bit model (of course, if $\mathbf{Q}, \mathbf{c}$ have rational entries, and ρ and the threshold value ξ are rational numbers). The TR problem can be reformulated as a *semidefinite program* (see (Fortin & Wolkowicz, 2004)), so that all complexity results about semidefinite programming can be applied to it. Following (Beck & Teboulle, 2009), we derive a semidefinite programming reformulation for a class of problems which generalizes the TR one. The class is defined as

$$\begin{array}{ll} \min & \frac{\mathbf{x}^T\mathbf{A}_1\mathbf{x}+2\mathbf{b}_1^T\mathbf{x}+c_1}{\mathbf{x}^T\mathbf{A}_2\mathbf{x}+2\mathbf{b}_2^T\mathbf{x}+c_2} \\ & \|\mathbf{L}\mathbf{x}\|^2 \leq \rho, \end{array} \tag{2.2}$$

where $\rho > 0$, $\mathbf{L}$ is an $r \times n$ matrix ($r \leq n$) with full row rank, and $\|\cdot\|$ is some norm. Notice that the subclass with

- $\mathbf{A}_2 = \mathbf{O}$, $\mathbf{b}_2 = \mathbf{0}$, $c_2 = 1$;
- $r = n$ and $\mathbf{L} = \mathbf{I}_n$ (identity matrix of order n); and
- $\|\cdot\|$ is the Euclidean norm;

is exactly the TR class. The following assumption is made:

$$\exists\, \eta \geq 0 \;:\; \mathbf{M} = \begin{pmatrix} \mathbf{A}_2 & \mathbf{b}_2 \\ \mathbf{b}_2^T & c_2 \end{pmatrix} + \eta \begin{pmatrix} \mathbf{L}^T\mathbf{L} & 0 \\ 0 & -\rho \end{pmatrix} \succ \mathbf{O}. \tag{2.3}$$

Such a condition guarantees that the quadratic function at the denominator is always strictly positive over the feasible region. Indeed, the condition implies

$$\begin{bmatrix} \mathbf{x}^T & 1 \end{bmatrix} \left[\begin{pmatrix} \mathbf{A}_2 & \mathbf{b}_2 \\ \mathbf{b}_2^T & c_2 \end{pmatrix} + \eta \begin{pmatrix} \mathbf{L}^T\mathbf{L} & 0 \\ 0 & -\rho \end{pmatrix} \right] \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} > 0,$$

i.e.,

$$\mathbf{x}^T\mathbf{A}_2\mathbf{x} + 2\mathbf{b}_2^T\mathbf{x} + c_2 + \eta(\|\mathbf{L}\mathbf{x}\|^2 - \rho) > 0.$$

Then,

$$\|\mathbf{L}\mathbf{x}\|^2 \leq \rho, \;\; \eta \geq 0 \;\; \Rightarrow \;\; \mathbf{x}^T\mathbf{A}_2\mathbf{x} + 2\mathbf{b}_2^T\mathbf{x} + c_2 > 0.$$

In order to simplify the discussion, the semidefinite reformulation is derived only for the case $r = n$ (i.e., $\mathbf{L}$ is a nonsingular matrix of order n). For the more technical proofs of the case $r < n$, we refer the reader to (Beck & Teboulle, 2009). Note that for $r = n$ the feasible region is a compact set, so that the minimum is certainly attained. First, problem (2.2) is reformulated as a nonconvex homogeneous problem. If, in (2.2), the change of variable $\mathbf{x} = \mathbf{y}/s$, $s > 0$, is made, then the problem can be rewritten as

$$\begin{aligned} \min \quad & \tfrac{\mathbf{y}^T\mathbf{A}_1\mathbf{y} + 2s\mathbf{b}_1^T\mathbf{y} + c_1 s^2}{\mathbf{y}^T\mathbf{A}_2\mathbf{y} + 2s\mathbf{b}_2^T\mathbf{y} + c_2 s^2} \\ & \|\mathbf{L}\mathbf{y}\|^2 - \rho s^2 \leq 0, \\ & s > 0, \end{aligned}$$

and, consequently, also as

$$\begin{aligned} \min \quad & \mathbf{y}^T\mathbf{A}_1\mathbf{y} + 2s\mathbf{b}_1^T\mathbf{y} + c_1 s^2 \\ & \mathbf{y}^T\mathbf{A}_2\mathbf{y} + 2s\mathbf{b}_2^T\mathbf{y} + c_2 s^2 = 1, \\ & \|\mathbf{L}\mathbf{y}\|^2 - \rho s^2 \leq 0, \\ & s > 0. \end{aligned} \tag{2.4}$$

Notice that the constraint $s > 0$ can be substituted by $s \geq 0$ by observing that there is no feasible solution for (2.4) with $s = 0$ if $\mathbf{L}$ is nonsingular. The dual of problem (2.4) is

$$\begin{aligned} \max \quad & \alpha \\ & \beta \geq 0, \\ & \begin{pmatrix} \mathbf{A}_1 & \mathbf{b}_1 \\ \mathbf{b}_1^T & c_1 \end{pmatrix} \succeq \alpha \begin{pmatrix} \mathbf{A}_2 & \mathbf{b}_2 \\ \mathbf{b}_2^T & c_2 \end{pmatrix} - \beta \begin{pmatrix} \mathbf{L}^T\mathbf{L} & 0 \\ 0 & -\rho \end{pmatrix}. \end{aligned} \tag{2.5}$$

A result proven in (Polyak, 1998) ensures that strong duality holds, so that the optimal value α^* of (2.5) is equal to the optimal value of (2.2). In (Beck & Teboulle, 2009) it is also shown that once α^* is known, an optimal solution of (2.2) can be recovered by solving the following problem:

$$\begin{aligned} \min \quad & \mathbf{x}^T(\mathbf{A}_1 - \alpha^*\mathbf{A}_2)\mathbf{x} + 2(\mathbf{b}_1 - \alpha^*\mathbf{b}_2)^T\mathbf{x} + c_1 - \alpha^* c_2 \\ & \|\mathbf{Lx}\|^2 \leq \rho, \end{aligned}$$

which is a *generalized TR problem* (a TR problem where the norm is not required to be Euclidean), for which efficient approaches exist (see, e.g., (Beck, Ben-Tal, & Teboulle, 2006; Sima, Van Huffel, & Golub, 2004)). Now, let us consider the procedure based on the iterative solutions of the generalized TR problems

$$\begin{aligned} \mathbf{x}^{k+1} \in \arg\min \quad & \mathbf{x}^T(\mathbf{A}_1 - \alpha_k\mathbf{A}_2)\mathbf{x} + 2(\mathbf{b}_1 - \alpha_k\mathbf{b}_2)^T\mathbf{x} + c_1 - \alpha_k c_2, \\ & \|\mathbf{Lx}\|^2 \leq \rho, \end{aligned} \tag{2.6}$$

where

$$\alpha_k = \frac{\mathbf{x}^{kT}\mathbf{A}_1\mathbf{x}^k + 2\mathbf{b}_1^T\mathbf{x}^k + c_1}{\mathbf{x}^{kT}\mathbf{A}_2\mathbf{x}^k + 2\mathbf{b}_2^T\mathbf{x}^k + c_2}.$$

Let us denote by $\mathbf{x}^*$ an optimal solution of the problem (2.2) and by

$$\begin{aligned} & h(\mathbf{x}) = \tfrac{h_1(\mathbf{x})}{h_2(\mathbf{x})} \quad \text{with} \\ & h_1(\mathbf{x}) = \mathbf{x}^T\mathbf{A}_1\mathbf{x} + 2\mathbf{b}_1^T\mathbf{x} + c_1, \\ & h_2(\mathbf{x}) = \mathbf{x}^T\mathbf{A}_2\mathbf{x} + 2\mathbf{b}_2^T\mathbf{x} + c_2, \end{aligned}$$

the objective function of the same problem. We provide an upper bound for the number of iterations of such a procedure after which it returns a point $\bar{\mathbf{x}}$ such that

$$h(\bar{\mathbf{x}}) - h(\mathbf{x}^*) \leq \varepsilon. \tag{2.7}$$

We need the following lemma.

Lemma 2.12. *For each* $\mathbf{x}$ *such that* $\|\mathbf{Lx}\|^2 \leq \rho$*,*

$$h_2(\mathbf{x}) \geq (1 + \|\mathbf{x}\|)\delta \geq \delta,$$

where $\delta > 0$ *is the minimum eigenvalue of the matrix* $\mathbf{M}$ *defined in* (2.3).

Proof. By the definition of $\mathbf{M}$ we have

$$\mathbf{M} = \begin{pmatrix} \mathbf{A}_2 & \mathbf{b}_2 \\ \mathbf{b}_2^T & c_2 \end{pmatrix} + \eta \begin{pmatrix} \mathbf{L}^T\mathbf{L} & 0 \\ 0 & -\rho \end{pmatrix} \succeq \delta\mathbf{I}_{n+1}.$$

Then, if $\mathbf{z} = (\mathbf{x}^T \quad 1)$, we have

$$\mathbf{z}^T\mathbf{M}\mathbf{z} = h_2(\mathbf{x}) + \eta(\|\mathbf{Lx}\|^2 - \rho) \geq \delta(\|\mathbf{x}\|^2 + 1),$$

from which the result immediately follows. □

Next, we prove the following theorem about the reduction of the function values from one iteration to the next.

Theorem 2.13. *Let* $\{\mathbf{x}_k\}$ *be the sequence generated by* (2.6)*. There exists* $\gamma \in (0,1)$ *such that*

$$(h(\mathbf{x}_{k+1}) - h(\mathbf{x}^*)) \leq \gamma(h(\mathbf{x}_k) - h(\mathbf{x}^*)). \tag{2.8}$$

Proof. We first notice that

$$\|\mathbf{L}\mathbf{x}_k\| \leq \rho \;\; \Rightarrow \;\; \|\mathbf{x}_k\| \leq U = \frac{\rho}{\lambda_1}, \tag{2.9}$$

where $\lambda_1 > 0$ is the minimum eigenvalue of $\mathbf{L}\mathbf{L}^T$. Next, we notice that we can rewrite (2.6) as

$$\begin{aligned} \mathbf{x}^{k+1} \in \arg\min \quad & h_2(\mathbf{x})(h(\mathbf{x}) - h(\mathbf{x}_k)), \\ & \|\mathbf{L}\mathbf{x}\|^2 \leq \rho. \end{aligned}$$

Then,

$$h_2(\mathbf{x}_{k+1})(h(\mathbf{x}_{k+1}) - h(\mathbf{x}_k)) \leq h_2(\mathbf{x}^*)(h(\mathbf{x}^*) - h(\mathbf{x}_k)),$$

so that

$$h(\mathbf{x}_{k+1}) - h(\mathbf{x}^*) \leq \left(1 - \frac{h_2(\mathbf{x}^*)}{h_2(\mathbf{x}_{k+1})}\right)(h(\mathbf{x}_k) - h(\mathbf{x}^*)).$$

In view of (2.9), we have

$$h_2(\mathbf{x}_{k+1}) \leq \mu_1 U^2 + 2\|\mathbf{b}_2\|U + c_2,$$

where μ_1 is the maximum eigenvalue of $\mathbf{A}_2$. Moreover, in view of Lemma 2.12,

$$h_2(\mathbf{x}^*) \geq \delta.$$

Then, the result of the theorem follows by taking

$$\gamma = 1 - \frac{\delta}{\mu_1 U^2 + 2\|\mathbf{b}_2\|U + c_2} \in (0,1). \qquad \square$$

It follows from this theorem that

$$(h(\mathbf{x}_k) - h(\mathbf{x}^*)) \leq \gamma^k(h(\mathbf{x}_0) - h(\mathbf{x}^*))$$

(note that we can always take $\mathbf{x}_0 = \mathbf{0}$). Then, the number of iterations after which (2.7) is satisfied is not larger than

$$\frac{1}{\log\left(\frac{1}{\gamma}\right)}\left(\log\left(\frac{1}{\varepsilon}\right) + \log(h(\mathbf{x}_0) - h(\mathbf{x}^*))\right).$$

In fact, in (Beck & Teboulle, 2009) a stronger result is proven, showing that (2.7) can be satisfied in at most

$$D_1 + D_2\sqrt{\log\left(\frac{1}{\varepsilon}\right)}$$

iterations for some positive constants D_1, D_2.

As a final remark, we notice that other classes of problems having the property of hidden convexity exist. For instance, in (Ye & Zhang, 2003) it is shown that, under suitable conditions, the problem of minimizing a quadratic function over a feasible region defined by two quadratic inequalities admits a semidefinite programming reformulation.

2.4 GO problems admitting a FPTAS

GO problems admitting a FPTAS lie at the border between "easy" and "difficult" problems. In the field of combinatorial optimization, the best-known problem admitting a FPTAS is the knapsack problem. An analogous result holds for a continuous optimization problem. The *separable quadratic knapsack problem*

$$\begin{array}{ll} \min & \sum_{i=1}^{n} d_i x_i^2 + c_i x_i \\ & \sum_{i=1}^{n} a_i x_i = b, \\ & \ell_i \leq x_i \leq u_i, \qquad i = 1,\ldots,n, \end{array}$$

was proven to be $\mathcal{NP}$-hard in (Sahni, 1974). This is true even if the objective function is concave (i.e., $d_i \leq 0$, $i = 1,\ldots,n$). Indeed, consider the *subset problem*, where given n positive integers $a_1,\ldots,a_n$ and an integer b, we want to establish whether a subset $I \subseteq \{1,\ldots,n\}$ exists such that

$$\sum_{i\in I} a_i = b.$$

The problem is known to be $\mathcal{NP}$-complete. It can easily be seen that such a problem admits a *yes* answer if and only if the following concave separable quadratic knapsack problem has optimal value equal to 0:

$$\begin{array}{ll} \min & \sum_{i=1}^{n} x_i(1 - x_i) \\ & \sum_{i=1}^{n} a_i x_i = b, \\ & 0 \leq x_i \leq 1, \qquad i = 1,\ldots,n. \end{array}$$

However, on the positive side, a FPTAS for quadratic knapsack problems based on dynamic programming has been derived in (Vavasis, 1992). Moreover, for the special case where the feasible region is the unit simplex Δ_n (see Definition A.3), an approach has been proposed in (Bomze & Locatelli, 2012) based on a parametrization of the problem, whose computational cost is $O(n\log(n))$; i.e., this special case is solvable in polynomial time.

The class of quadratic programming (QP) problems

$$\begin{array}{ll} \min & \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x} \\ & \mathbf{A}\mathbf{x} \leq \mathbf{b} \end{array}$$

is solvable in polynomial time if the Hessian matrix $\mathbf{Q}$ is positive semidefinite (then, the problem is a convex one), but has been proven to be $\mathcal{NP}$-hard even if the Hessian of the quadratic function has a single negative eigenvalue (Pardalos & Vavasis, 1991). However, when the feasible region is a polytope, in (Vavasis, 1992) an algorithm has been proposed for detecting an ε-approximate solution with computational cost

$$O\left(cost(CP)\left(\frac{n(n+1)}{\sqrt{\varepsilon}}\right)^s\right),$$

where $cost(CP)$ is the cost of solving a convex QP problem with the same size as the nonconvex one, and s is the number of negative eigenvalues for the Hessian of the quadratic function. This turns out to be a FPTAS when $\mathbf{Q}$ has a *fixed* number of negative eigenvalues.

A quite special quadratic objective function with a single negative eigenvalue is the product of two affine functions. This gives rise to the class of 2*LMP problems* where the product of two affine functions over a polyhedral region is minimized:

$$\begin{aligned} \min \quad & (\mathbf{c}^T\mathbf{x}+c_0)(\mathbf{d}^T\mathbf{x}+d_0) \\ & \mathbf{A}\mathbf{x} \leq \mathbf{b}. \end{aligned} \tag{2.10}$$

It is required that both affine functions are nonnegative over the feasible region. Class 2LMP has been proven to be $\mathcal{NP}$-hard in (Matsui, 1996). However, different FPTAS for such a class have been derived. Note that since the polyhedral region is not required to be bounded, we cannot apply the result proven in (Vavasis, 1992), and the objective function might be unbounded from above over the feasible region. Therefore, here and for the rest of the section we will refer to Definition 2.7 of the ε-approximation problem. The first FPTAS for such class has been given in (Kern & Woeginger, 2007). Later on, two different FPTAS for the same class, independently derived in (Depetrini & Locatelli, 2009) and (Goyal, Genc-Kaya, & Ravi, 2011), improved the dependency on $\frac{1}{\varepsilon}$ with respect to the FPTAS proposed in (Kern & Woeginger, 2007).

Following (Depetrini & Locatelli, 2011), we derive a FPTAS for a class of problems that includes 2LMP as a special case. The proposed FPTAS, when reduced to 2LMP, is not the one with the best-known complexity with respect to $\frac{1}{\varepsilon}$, but on the other hand it has a wider applicability. Before proceeding, we also remark that a further extension to a larger class of problems has been independently proposed in (Locatelli, 2013; Mittel & Schulz, 2012).

2.4.1 A FPTAS for linear fractional-multiplicative programming problems

The class we are going to consider is that of *linear fractional-multiplicative programming* (LFMP) problems

$$\begin{aligned} \min \quad & \Lambda_{i=1}^{p} \tfrac{\mathbf{c}^{i\,T}\mathbf{x}+c_{0i}}{\mathbf{d}^{i\,T}\mathbf{x}+d_{0i}} \\ & \mathbf{x} \in P = \{\mathbf{x} \in \mathbb{R}^n \ : \ \mathbf{A}\mathbf{x} \geq \mathbf{b}, \ \mathbf{x} \geq \mathbf{0}\}, \end{aligned} \tag{2.11}$$

where $\Lambda \in \{\sum, \prod\}$. The class LFMP includes some well-known classes of problems in the field of global optimization such as the class of *linear (sum-of-ratios) fractional* *programming* (LFP) if $\Lambda = \sum$ (see, e.g., (Schaible & Shi, 2003a) for a survey), or *linear multiplicative programming* (LMP) when $\Lambda = \prod$ and

$$\mathbf{d}^{i^T}\mathbf{x} + d_{0i} \equiv 1 \quad \forall\, i = 1, \ldots, p$$

(see, e.g., (Konno & Kuno, 1995a)), whose special case $p = 2$ is the 2LMP problem. All data defining the objective function and the feasible region are assumed to be integers. Moreover, it is assumed that all the affine functions are strictly positive over the feasible region, i.e.,

$$\mathbf{c}^{i^T}\mathbf{x} + c_{0i} > 0,\ \mathbf{d}^{i^T}\mathbf{x} + d_{0i} > 0 \quad \forall\, \mathbf{x} \in P,\ \ \forall\, i \in \{1, \ldots, p\}, \tag{2.12}$$

although the results can be extended to the case of nonnegative (and, in some cases, even negative) numerator functions (see (Depetrini & Locatelli, 2011)). It is also assumed that each single ratio attains its minimum value ℓ_i over P. In such a case the value must be attained at a vertex of P (see, e.g., (Schaible & Ibaraki, 1983)) and, in view of (2.12), $\ell_i > 0$. Each value ℓ_i can be efficiently computed (see, e.g., (Dinkelbach, 1967)).[1] Notice that by taking any vertex in P and evaluating the objective function at it in order to define an upper bound U' for the problem (2.11), we can impose the following upper bounds for the values of the ratio functions *at any optimal solution of* (2.11):

$$\frac{\mathbf{c}^{i^T}\mathbf{x} + c_{0i}}{\mathbf{d}^{i^T}\mathbf{x} + d_{0i}} \le u_i = \begin{cases} \dfrac{U'}{\prod_{j=1,\ j\neq i}^{p} \ell_j} & \text{if } \Lambda = \prod, \\ U' & \text{if } \Lambda = \sum. \end{cases} \tag{2.13}$$

First we give a parametric reformulation of problem (2.11) as a box-constrained p-dimensional problem, and then we derive a FPTAS for the problem *when p is fixed.*

A parametric reformulation

Denote by $\boldsymbol{\ell} = (\ell_1 \ldots \ell_p)$ and $\mathbf{u} = (u_1 \ldots u_p)$ the p-dimensional vectors whose components are, respectively, the lower and upper bounds for the ratio functions. Let $\boldsymbol{\gamma} = (\gamma_1 \ldots \gamma_p) \in [\boldsymbol{\ell}, \mathbf{u}]$ and define

$$P(\boldsymbol{\gamma}) = P \cap \{\mathbf{x} \in \mathbb{R}^n\ :\ \mathbf{c}^{i^T}\mathbf{x} + c_{0i} = \gamma_i(\mathbf{d}^{i^T}\mathbf{x} + d_{0i})\},\ \ i = 1, \ldots, p.$$

$P(\boldsymbol{\gamma})$ is a subset of the level set of all the points in P where the objective function value is equal to $\Lambda_{i=1}^{p}\gamma_i$. Then define the p-dimensional function h_1 as follows:

$$h_1(\boldsymbol{\gamma}) = \begin{cases} \Lambda_{i=1}^{p}\gamma_i & \text{if } P(\boldsymbol{\gamma}) \neq \emptyset, \\ +\infty & \text{otherwise.} \end{cases} \tag{2.14}$$

Notice that problem (2.11) turns out to be equivalent to the following problem:

$$\min_{\boldsymbol{\gamma} \in [\boldsymbol{\ell}, \mathbf{u}]} h_1(\boldsymbol{\gamma}).$$

[1] In fact, we point out that for the following development $\ell_i > 0$ could also be a lower bound for the minimum value of the ith single ratio, rather than being the minimum value itself.

Now, substitute the equality constraints with inequality constraints and define the region

$$P'(\boldsymbol{\gamma}) = P \cap \{\mathbf{x} \in \mathbb{R}^n \ : \ {\mathbf{c}^i}^T\mathbf{x} + c_{0i} \le \gamma_i({\mathbf{d}^i}^T\mathbf{x} + d_{0i})\}, \ \ i = 1,\dots,p.$$

It is obviously true that $P'(\boldsymbol{\gamma}) \supseteq P(\boldsymbol{\gamma})$ for all $\boldsymbol{\gamma} \in [\boldsymbol{\ell},\mathbf{u}]$. A further p-dimensional function h_2 is defined as h_1 with $P(\boldsymbol{\gamma})$ substituted by $P'(\boldsymbol{\gamma})$. In view of the relation between $P(\boldsymbol{\gamma})$ and $P'(\boldsymbol{\gamma})$, for all $\boldsymbol{\gamma} \in [\boldsymbol{\ell},\mathbf{u}]$ we have that $h_2(\boldsymbol{\gamma}) \le h_1(\boldsymbol{\gamma})$. Now let

$$\alpha(\Lambda) = \begin{cases} p & \text{if } \Lambda = \prod, \\ 1 & \text{if } \Lambda = \sum. \end{cases}$$

We can prove the following proposition.

Proposition 2.14. *For each* $\bar{\boldsymbol{\gamma}} \in [\boldsymbol{\ell},\mathbf{u}]$ *and any* $\delta \in (0,1]$ *we have that*

(a)

$$\delta^{\alpha(\Lambda)}\Lambda_{i=1}^p \bar{\gamma}_i \le \delta^{\alpha(\Lambda)}h_2(\bar{\boldsymbol{\gamma}}) \le h_2(\delta\bar{\boldsymbol{\gamma}}) \le h_1(\delta\bar{\boldsymbol{\gamma}});$$

(b) *for any* $\eta \in (0,1)$*, if* $\delta \ge (1-\eta)^{1/\alpha(\Lambda)}$*, then*

$$h_1(\boldsymbol{\gamma}) \ge (1-\eta)h_2(\bar{\boldsymbol{\gamma}}) \quad \forall\, \boldsymbol{\gamma} \in [\delta\bar{\boldsymbol{\gamma}},\bar{\boldsymbol{\gamma}}];$$

(c) *there exists* $\tilde{\boldsymbol{\gamma}} \in [\boldsymbol{\ell},\mathbf{u}]$ *such that* $h_1(\tilde{\boldsymbol{\gamma}}) \le h_2(\bar{\boldsymbol{\gamma}})$.

Proof.

(a) This follows immediately from $h_2(\boldsymbol{\gamma}) \le h_1(\boldsymbol{\gamma})$ for any $\boldsymbol{\gamma} \in [\boldsymbol{\ell},\mathbf{u}]$ and noticing that, in view of (2.12), $P'(\bar{\boldsymbol{\gamma}}) \supseteq P'(\delta\bar{\boldsymbol{\gamma}})$ if $\delta \in (0,1]$ and $\bar{\boldsymbol{\gamma}} \in [\boldsymbol{\ell},\mathbf{u}]$.

(b) It is enough to observe that in view of the first point proven above, a lower bound for h_1 over the set $[\delta\bar{\boldsymbol{\gamma}},\bar{\boldsymbol{\gamma}}]$ is $\delta h_2(\bar{\boldsymbol{\gamma}})$ if $\Lambda = \sum$, and $\delta^p h_2(\bar{\boldsymbol{\gamma}})$ if $\Lambda = \prod$.

(c) The result is trivial if $h_2(\bar{\boldsymbol{\gamma}}) = \infty$. If $h_2(\bar{\boldsymbol{\gamma}}) < \infty$, let $\mathbf{x}^* \in P'(\bar{\boldsymbol{\gamma}})$. We define $\tilde{\boldsymbol{\gamma}}$ as follows:

$$\tilde{\gamma}_i = \frac{{\mathbf{c}^i}^T\mathbf{x}^* + c_{0i}}{{\mathbf{d}^i}^T\mathbf{x}^* + d_{0i}} \le \bar{\gamma}_i \quad \forall\, i \in \{1,\dots,p\}.$$

Then, $\mathbf{x}^*$ also belongs to $P(\tilde{\boldsymbol{\gamma}})$ and, consequently, we have that

$$\Lambda_{i=1}^p \tilde{\gamma}_i = h_2(\tilde{\boldsymbol{\gamma}}) = h_1(\tilde{\boldsymbol{\gamma}}) \le h_2(\bar{\boldsymbol{\gamma}}) = \Lambda_{i=1}^p \bar{\gamma}_i. \qquad \square$$

In what follows we will introduce an algorithm for the solution of problem (2.11), and the results proved in the proposition above will be used to show that the algorithm is able to return an ε-approximate solution of the problem.

An ε-approximation algorithm

Here we present an ε-approximation algorithm for LFMP. The algorithm is based on the evaluation of the function h_2 at a *nonuniform* grid over the p-dimensional box $[\boldsymbol{\ell}, \mathbf{u}]$.

Algorithm `AppLFMP`

Initialization Let

$$\mathcal{F} = \{\mathbf{u}\}, \quad \mathcal{V} = \emptyset, \quad U = f(\mathbf{u}).$$

Set

$$\bar{\delta} = (1-\eta)^{\frac{1}{\alpha(\Lambda)}}, \tag{2.15}$$

where

$$\eta = \frac{\varepsilon}{1+\varepsilon}. \tag{2.16}$$

Step 1. Select a point $\bar{\boldsymbol{y}} \in \mathcal{F}$. Set $\mathcal{V} = \mathcal{V} \cup \{\bar{\boldsymbol{y}}\}$. Consider the 2^p points

$$(\xi_1 \bar{y}_1 \ldots \xi_p \bar{y}_p),$$

where $\xi_i \in \{\bar{\delta}, 1\}$ for all i, and discard all such points for which $\xi_i \bar{y}_i < \ell_i$ for at least an index i. Let $\mathcal{G}$ be the set of the remaining points.

Step 2. Evaluate h_2 at all the points belonging to $\mathcal{G} \setminus \mathcal{F}$ and set

$$U = \min\left\{U, \min_{\boldsymbol{y} \in \mathcal{G} \setminus \mathcal{F}} h_2(\boldsymbol{y})\right\}.$$

Step 3. Set $\mathcal{F} = (\mathcal{F} \cup \mathcal{G}) \setminus \mathcal{V}$.

Step 4. If $\mathcal{F} = \emptyset$, then STOP; otherwise go back to Step 1.

We prove the following theorem.

Theorem 2.15. *Algorithm* `AppLFMP` *is an ε-approximation algorithm for the problem LFMP.*

Proof. We notice that Algorithm `AppLFMP` evaluates the function h_2 at the following points:

$$\left(\bar{\delta}^{k_1} u_1 \ldots \bar{\delta}^{k_p} u_p\right),$$

where the k_i's are integer values ranging between 0 and

$$\bar{k}_i = \max\{k_i \ : \ \bar{\delta}^{k_i} u_i \geq \ell_i\}. \tag{2.17}$$

These points form a nonuniform grid over the box $[\boldsymbol{\ell}, \mathbf{u}]$. For each $\boldsymbol{y} \in [\boldsymbol{\ell}, \mathbf{u}]$, there exists an integer vector $(k_1 \ldots k_p)$ with $0 \leq k_i \leq \bar{k}_i$ for all i, such that

$$\boldsymbol{y} \in \prod_{i=1}^{p} [\bar{\delta}^{k_i+1} u_i, \bar{\delta}^{k_i} u_i].$$

Therefore, in view of the definition of h_1 and h_2, Proposition 2.14(b), and the definition (2.15) of $\bar{\delta}$, we have that

$$h_1(\boldsymbol{\gamma}) \geq h_2(\boldsymbol{\gamma}) \geq (1-\eta)h_2\left(\bar{\delta}^{k_1}u_1,\ldots,\bar{\delta}^{k_p}u_p\right).$$

Then, we can conclude that

$$\min_{\boldsymbol{\gamma}\in[\boldsymbol{\ell},\mathbf{u}]} h_1(\boldsymbol{\gamma}) \geq \min_{\boldsymbol{\gamma}\in[\boldsymbol{\ell},\mathbf{u}]} h_2(\boldsymbol{\gamma}) \geq (1-\eta) \min_{k_i\in\{0,\ldots,\bar{k}_i\}\ \forall\ i} h_2\left(\bar{\delta}^{k_1}u_1,\ldots,\bar{\delta}^{k_p}u_p\right).$$

In view of Proposition 2.14(c), a solution $\tilde{\boldsymbol{\gamma}} \in [\boldsymbol{\ell},\mathbf{u}]$ such that

$$\min_{k_i\in\{0,\ldots,\bar{k}_i\}\ \forall i} h_2\left(\bar{\delta}^{k_1}u_1,\ldots,\bar{\delta}^{k_p}u_p\right) \geq h_1(\tilde{\boldsymbol{\gamma}})$$

can be easily detected once all the points

$$\left(\bar{\delta}^{k_1}u_1\ldots\bar{\delta}^{k_p}u_p\right), \quad k_i \in \{0,\ldots,\bar{k}_i\}, \quad i=1,\ldots,p,$$

are known. Therefore,

$$h_1(\tilde{\boldsymbol{\gamma}}) \leq \frac{1}{1-\eta}\min_{\boldsymbol{\gamma}\in[\boldsymbol{\ell},\mathbf{u}]} h_1(\boldsymbol{\gamma}) = (1+\varepsilon)\min_{\boldsymbol{\gamma}\in[\boldsymbol{\ell},\mathbf{u}]} h_1(\boldsymbol{\gamma}),$$

where the last equality follows from the definition (2.16) of η. □

Complexity of the algorithm

The number of points at which h_2 is evaluated is

$$\prod_{i=1}^{p}(\bar{k}_i+1).$$

Then, recalling the definition (2.17) of $\bar{k}_i$ we can prove the following lemma.

Lemma 2.16. *The number of points at which the function f is evaluated by* AppLFMP *is not larger than*

$$\prod_{i=1}^{p}\left\lceil 1+\frac{\alpha(\Lambda)\log\left(\frac{u_i}{l_i}\right)}{\eta}\right\rceil. \tag{2.18}$$

A well-known result from linear algebra (see, e.g., (Nemhauser & Wolsey, 1988, page 123)) states that, given θ, the largest of all the input data in the definition of a polyhedron P, the jth coordinate of any vertex $\mathbf{x}^0$ of P has value p_j/q for integers p_j and q such that

$$0 \leq p_j \leq (n\theta)^n,\ j = 1,\dots,n, \quad 1 \leq q \leq (n\theta)^n.$$

Noticing that the ℓ_i and u_i values are derived from evaluations of the affine functions at vertices of P, such a result can be exploited to derive lower bounds for the ℓ_i values and upper bounds for the u_i values, ending up with the following limitation from above for the $\log(\frac{u_i}{\ell_i})$ values.

Lemma 2.17. *We have that*

$$\log\left(\frac{u_i}{l_i}\right) \leq 2p\log(2) + 2\alpha(\Lambda)(2n+1)[\log(n)+\log(\theta)].$$

Proof. For each i, let $\mathbf{x}_i^*$ be the vertex of P at which the positive minimum value ℓ_i of the ith ratio is attained over P, i.e.,

$$\ell_i = \frac{\mathbf{c}^{iT}\mathbf{x}_i^* + c_{0i}}{\mathbf{d}^{iT}\mathbf{x}_i^* + d_{0i}} > 0.$$

In view of the above-mentioned result from linear algebra and of (2.12), we must have that

$$\mathbf{c}^{iT}\mathbf{x}_i^* + c_{0i} \geq \frac{1}{n^n\theta^n}, \quad \mathbf{d}^{iT}\mathbf{x}_i^* + d_{0i} \leq n^{n+1}\theta^{n+1} + \theta \leq 2n^{n+1}\theta^{n+1}.$$

Therefore, we can conclude that

$$\ell_i \geq \frac{1}{2n^{2n+1}\theta^{2n+1}}.$$

Next, we search for an upper bound for the u_i values. Taking into account the definition (2.13), we can exploit again the result from linear algebra to derive through simple but tedious computations the following upper bounds for the u_i values:

$$u_i \leq 2^{2p-1}(n)^{(2\alpha(\Lambda)-1)(2n+1)}(\theta)^{(2\alpha(\Lambda)-1)(2n+1)}.$$

Therefore, combining the upper bound for u_i and the lower bound for ℓ_i, we are finally led to

$$\log\left(\frac{u_i}{l_i}\right) \leq 2p\log(2) + 2\alpha(\Lambda)(2n+1)[\log(n)+\log(\theta)]. \qquad \square$$

The result of this lemma, combined with (2.18), the definition (2.16) of η, and the fact that each evaluation of f requires the solution of an LP, finally leads to the following theorem.

Theorem 2.18. *For a fixed p value, the number of operations required by* `AppLFMP` *in order to detect an ε-approximate solution for* (2.11) *is bounded from above by*

$$O\left(cost(LP)\frac{\alpha(\Lambda)^p(4n+2)^p[\log(n)+\log(\theta)]^p}{\varepsilon^p}\right),$$

where $cost(LP)$ is the (polynomial) cost for solving the linear programming problem needed to evaluate h_2 at some point.

In view of the above theorem we can conclude that for *fixed p*, the ε-approximation algorithm AppLFMP is a FPTAS (fully polynomial time approximation scheme) for the problem LFMP. On the other hand, the same theorem shows that we have an exponential increase of the computational time for AppLFMP as p increases, which has often been observed in practical algorithms for LMP or LFP also.

2.5 GO problems admitting a PTAS

One step forward in terms of difficulty with respect to the problems discussed in the previous section are those admitting a PTAS but not a FPTAS. In this section we mention some of them and discuss one in detail, the *standard quadratic programming* problem (StQP). StQP problems are defined as follows:

$$\begin{aligned} \min \quad & \mathbf{x}^T\mathbf{Q}\mathbf{x} \\ & \mathbf{x} \in \Delta_n, \end{aligned} \tag{2.19}$$

where $\mathbf{Q}$ is a symmetric square matrix of order n and Δ_n is the n-dimensional unit simplex. Note that omitting linear terms in the definition of the objective function in (2.19) is not a real restriction. Indeed, for any $\mathbf{x} \in \Delta_n$

$$\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x} = \mathbf{x}^T\left[\mathbf{Q} + 1/2(\mathbf{c}\mathbf{e}^T + \mathbf{e}\mathbf{c}^T)\right]\mathbf{x}.$$

While simple in some special cases (e.g., when $\mathbf{Q}$ is positive or negative semidefinite or, as already mentioned in Section 2.4, when $\mathbf{Q} = \mathbf{D} + \frac{1}{2}(\mathbf{c}\mathbf{e}^T + \mathbf{e}\mathbf{c}^T)$ with $\mathbf{D}$ diagonal; see (Bomze & Locatelli, 2012)), StQP problems become $\mathcal{NP}$-hard when $\mathbf{Q}$ is an indefinite matrix. To see this, one might recall a classical result of (Motzkin & Strauss, 1965). Let $G = (V, E)$ be an undirected graph; a *clique* in G is a subset of nodes $C \subseteq V$ which induces a complete subgraph in G, i.e.,

$$(i, j) \in E \quad \forall i, j \in C, i \neq j.$$

The *maximum clique problem* aims at detecting the largest clique in G. The cardinality of the largest clique is denoted by $\omega(G)$. We first introduce the definition of Karush–Kuhn–Tucker (KKT) points, which will be needed in what follows, and later we will prove the result by Motzkin and Strauss.

Definition 2.19. *Given the optimization problem*

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ & c_i(\mathbf{x}) \leq 0, \quad i \in I_1, \\ & c_i(\mathbf{x}) = 0, \quad i \in I_2, \end{aligned} \tag{2.20}$$

with f, c_i, $i \in I_1 \cup I_2$, continuously differentiable, a point $\bar{\mathbf{x}}$ is a KKT point for (2.20) *if the following are satisfied:*

$$\begin{array}{ll} \nabla f(\bar{\mathbf{x}}) + \sum_{i\in I_1} \mu_i \nabla c_i(\bar{\mathbf{x}}) + \sum_{i\in I_2} \lambda_i \nabla c_i(\bar{\mathbf{x}}) = 0, & \\ c_i(\bar{\mathbf{x}}) \le 0, & i \in I_1, \\ c_i(\bar{\mathbf{x}}) = 0, & i \in I_2, \\ \mu_i \ge 0, & i \in I_1, \\ \mu_i c_i(\bar{\mathbf{x}}) = 0, & i \in I_1. \end{array}$$

The above conditions, called KKT conditions, are necessary for a point $\bar{\mathbf{x}}$ to be a local minimum of the optimization problem (2.20) provided that a regularity assumption is satisfied at $\bar{\mathbf{x}}$. *The values μ_i, $i \in I_1$, λ_i, $i \in I_2$, are called Lagrange multipliers. If f and c_i, $i \in I_1$, are convex functions, and c_i, $i \in I_2$, are affine functions, then the KKT conditions are also sufficient for the local optimality of $\bar{\mathbf{x}}$.*

If we denote by

$$\mathcal{A}(\bar{\mathbf{x}}) = \{i \in I_1 \cup I_2 \ : \ c_i(\bar{\mathbf{x}}) = 0\}$$

the set of active constraints at $\bar{\mathbf{x}}$, then examples of conditions under which the regularity assumption is satisfied at $\bar{\mathbf{x}}$ are the following:

- the functions c_i, $i \in \mathcal{A}(\bar{\mathbf{x}})$, are affine;
- the gradient vectors $\nabla c_i(\bar{\mathbf{x}})$, $i \in \mathcal{A}(\bar{\mathbf{x}})$, are linearly independent.

Theorem 2.20. *Let $\mathbf{A}_G$ be the adjacency matrix of the graph G. Then,*

$$\frac{1}{2}\left(1 - \frac{1}{\omega(G)}\right) = \max \quad \frac{1}{2}\mathbf{x}^T \mathbf{A}_G \mathbf{x} \qquad \mathbf{x} \in \Delta_n; \tag{2.21}$$

i.e., the maximum clique problem admits a reformulation as an StQP problem.

Proof. Let us denote by f^* the optimal value of the problem. Let C be a clique with maximum cardinality $k = \omega(G)$ and $\mathbf{x}^C \in \Delta_n$ be defined as follows:

$$x_i^C = \begin{cases} \frac{1}{k} & \text{if } i \in C, \\ 0 & \text{otherwise.} \end{cases}$$

Then,

$$f^* \ge \frac{1}{2}\mathbf{x}^{C^T} \mathbf{A}_G \mathbf{x}^C = \frac{1}{2}\left(1 - \frac{1}{k}\right).$$

The opposite inequality is proved by induction on the number n of nodes. If $n = 1$, then $f^* = 0$ and $k = 1$, so that the result is true. Next, for $n \ge 2$, we assume that the result is true for all graphs with $n-1$ nodes, and we prove it for the graphs with n nodes. Let $\mathbf{x}^* = (x_1^*, \ldots, x_n^*)$ be a globally optimal solution of the problem. If for some i we have $x_i^* = 0$, then the optimal value of the problem for the graph G is equal to the optimal value of the problem for the graph G' obtained from G by removing node i. Since G' has $n-1$ nodes, by induction we have

$$f^* = \frac{1}{2}\left(1-\frac{1}{k'}\right) \le \frac{1}{2}\left(1-\frac{1}{k}\right),$$

where $k' = \omega(G') \le k$. Otherwise, if $\mathbf{x}^* > \mathbf{0}$, we distinguish two cases:

- G is not the complete graph. Then, if we assume, without loss of generality, that $(1,2) \notin E$, for all $\alpha \le x_1^*$ and

  $$\mathbf{z} = \mathbf{z}(\alpha) = (x_1^* - \alpha, x_2^* + \alpha, x_3^*, \dots, x_n^*)$$

  we have

  $$\frac{1}{2}\mathbf{z}^T \mathbf{A}_G \mathbf{z} = \frac{1}{2}\mathbf{x}^{*T} \mathbf{A}_G \mathbf{x}^* - \alpha \sum_{j\,:\,(1,j)\in E} x_j^* + \alpha \sum_{j\,:\,(2,j)\in E} x_j^*. \qquad (2.22)$$

  The KKT conditions must hold at $\mathbf{x}^*$ because (i) it is a global and, thus, also a local minimum of the problem; and (ii) the regularity assumption is certainly satisfied at $\mathbf{x}^*$ since all the constraints are linear ones. Then, since $\mathbf{x}^* > \mathbf{0}$, we have that for all $i = 1, \dots, n$,

  $$\sum_{j\,:\,(i,j)\in E} x_j^* = \xi, \qquad (2.23)$$

  where ξ is the Lagrange multiplier of the constraint $\mathbf{e}^T\mathbf{x} = 1$. Therefore, in view of (2.22) and (2.23)

  $$\frac{1}{2}\mathbf{z}^T \mathbf{A}_G \mathbf{z} = \frac{1}{2}\mathbf{x}^{*T} \mathbf{A}_G \mathbf{x}^*;$$

  i.e., $\mathbf{z}$ also is an optimal solution of the problem. If we set $\alpha = x_1^*$, $z_1 = 0$, we are back to the situation where we have an optimal solution with a null coordinate.

- G is the complete graph. Then,

  $$\begin{aligned} f^* = \tfrac{1}{2}\mathbf{x}^{*T}\mathbf{A}_G\mathbf{x}^* &= \tfrac{1}{2}\left[\left(\sum_{i=1}^n x_i^*\right)^2 - \sum_{i=1}^n \left(x_i^*\right)^2\right] = \tfrac{1}{2}\left(1 - \|\mathbf{x}^*\|^2\right) \\ &\le \tfrac{1}{2}\left(1 - \min_{\mathbf{x}\in\Delta_n} \|\mathbf{x}\|\right) = \tfrac{1}{2}\left(1-\tfrac{1}{n}\right), \end{aligned}$$

  which proves the result. □

In view of the relation between the optimal value of the StQP problem (2.21) and the optimal value of the corresponding maximum clique problem, we can conclude that the $\mathcal{NP}$-completeness of the maximum clique problem implies the $\mathcal{NP}$-hardness for StQP problems. (To be more precise, we introduced StQP problems as minimization problems, while (2.21) is a maximization problem; but, obviously, we can also convert it into a minimization problem.) Moreover, an inapproximability result proven in (Hastad, 1999) for the maximum clique problem also shows that the StQP problems do not even admit a FPTAS, unless $\mathcal{NP} = \mathcal{ZPP}$, where $\mathcal{ZPP}$ denotes the set of problems which can be solved by a randomized algorithm whose average running time is polynomial. Another continuous reformulation of the maximum clique problem is the following (see (Nesterov, 2003)):

$$\sqrt{1-\frac{1}{\omega(G)}} = 3\sqrt{3}\max \quad \sum_{i_1<i_2:(i_1,i_2)\notin E} y_{i_1 i_2} x_{i_1} x_{i_2}$$

$$\|\mathbf{x}\|_2^2 + \|\mathbf{y}\|_2^2 = 1.$$

Similar to StQP, such a reformulation, combined with the inapproximability result proven in (Hastad, 1999), shows that, while quadratic minimization problems over spheres can be solved in polynomial time, cubic minimization problems over spheres do not admit a FPTAS. It is still an open question whether the minimization of a form of fixed degree d (a polynomial whose monomials all have degree d) over the unit sphere admits a PTAS, although this has been proven for some special forms (see (Barvinok, 2007)). (Ling, Nie, Qi, & Ye, 2009) have proposed two PTASs for the bi-quadratic optimization problem over spheres

$$\begin{aligned} \min \quad & \textstyle\sum_{i,j=1}^{n} \sum_{k,h=1}^{m} a_{ijkh} x_i x_j y_k y_h \\ & \|\mathbf{x}\|_2 = 1, \\ & \|\mathbf{y}\|_2 = 1, \end{aligned}$$

when $\min\{m,n\}$ is fixed. Getting back to StQP problems, existence of a PTAS for them has been proven by Bomze and de Klerk (Bomze & de Klerk, 2002). We will present such a PTAS in what follows, but first some preliminary results about exact and approximate reformulations of StQP problems through conic programming are needed.

2.5.1 StQP and conic programming

StQP problems can be reformulated as problems over the cone of completely positive matrices $\mathcal{C}_n^*$ (see Definition A.22). Indeed, if we denote by f_* the optimal value of (2.19), we have that (see (Bomze et al., 2000))

$$f_* = \min\{\mathbf{Q} \bullet \mathbf{X} : \ \mathbf{E} \bullet \mathbf{X} = 1, \mathbf{X} \in \mathcal{C}_n^*\}$$

(recall that $\bullet$ is the Frobenius inner product; see Definition A.17). Since $\frac{1}{n}\mathbf{I}_n$ is a strictly feasible solution for the problem above, i.e., Slater's condition is satisfied, we have that strong duality holds so that f_* is equal to the optimal value of the corresponding dual problem, i.e.,

$$f_* = \max\{\lambda \ : \ \mathbf{Q} - \lambda \mathbf{E} \in \mathcal{C}_n\}, \tag{2.24}$$

where $\mathcal{C}_n$ is the copositive cone (see Definition A.21).

In order to limit from below the value f_*, one might substitute the copositive cone with some other, more manageable cone. One such possibility is to replace $\mathcal{C}_n$ with the cone $\mathcal{C}_n^r$, where r is a nonnegative integer (see (de Klerk & Pasechnik, 2002)). The cone $\mathcal{C}_n^r$ is defined as the cone of the symmetric matrices $\mathbf{M}$ for which the polynomial

$$P^{(r)}(\mathbf{x}) = \sum_{i,j=1}^{n} M_{ij} x_i^2 x_j^2 \left(\sum_{k=1}^{n} x_k^2 \right)^r$$

has nonnegative coefficients, which implies that $\mathcal{C}_n^r \subset \mathcal{C}_n$. Therefore, for each value of r,

$$f_*^r = \max\{\lambda \; : \; \mathbf{Q} - \lambda\mathbf{E} \in \mathcal{C}_n^r\} \tag{2.25}$$

satisfies $f_*^r \leq f_*$. Moreover,

$$\mathcal{C}_n^r \subseteq \mathcal{C}_n^{r+1} \quad \forall\, r.$$

An alternative representation of $\mathcal{C}_n^r$ (see (Bomze & de Klerk, 2002)) is

$$\mathcal{C}_n^r = \left\{\mathbf{M} \in \mathcal{S}_n \; : \; \mathbf{x}^T\mathbf{M}\mathbf{x} - \mathbf{x}^T diag(\mathbf{M}) \geq 0 \;\; \forall \; \frac{1}{r+2}\mathbf{x} \in \Delta_n(r)\right\}, \tag{2.26}$$

where $diag(\mathbf{M})$ denotes the n-dimensional vector whose entries are equal to the diagonal entries of the matrix $\mathbf{M}$, and

$$\Delta_n(r) = \{\mathbf{x} \in \Delta_n \; : \; (r+2)\mathbf{x} \in \mathbb{N}_0^n\}$$

is a *uniform* grid over the unit simplex.

2.5.2 A PTAS for StQP

A quite simple PTAS for StQP is based on the uniform grid $\Delta_n(r)$. The following result, proven in (Bomze & de Klerk, 2002), is needed.

Theorem 2.21. *Let*

$$\mathbf{q}_r = \frac{1}{r+2} diag(\mathbf{Q}).$$

Then

$$f_*^r = \frac{r+2}{r+1} \min\left\{\mathbf{x}^T\mathbf{Q}\mathbf{x} - \mathbf{q}_r^T\mathbf{x} \; : \; \mathbf{x} \in \Delta_n(r)\right\}.$$

Proof. First we notice that for each $\mathbf{x}$ such that $\frac{1}{r+2}\mathbf{x} \in \Delta_n(r)$, we have that

$$\mathbf{x}^T\mathbf{E}\mathbf{x} = (r+2)^2, \quad \mathbf{x}^T\mathbf{e} = r+2.$$

Then, by employing the representation (2.26) in (2.25), we have that

$$\begin{aligned}
f_r^* &= \max\left\{\lambda \; : \; \mathbf{x}^T(\mathbf{Q} - \lambda\mathbf{E})\mathbf{x} - \mathbf{x}^T diag(\mathbf{Q} - \lambda\mathbf{E}) \geq 0 \;\; \forall \; \tfrac{1}{r+2}\mathbf{x} \in \Delta_n(r)\right\} \\
&= \max\left\{\lambda \; : \; \mathbf{x}^T\mathbf{Q}\mathbf{x} - \lambda\mathbf{x}^T\mathbf{E}\mathbf{x} - \mathbf{x}^T diag(\mathbf{Q}) + \lambda\mathbf{x}^T\mathbf{e} \geq 0 \;\; \forall \; \tfrac{1}{r+2}\mathbf{x} \in \Delta_n(r)\right\} \\
&= \max\left\{\lambda \; : \; \mathbf{x}^T\mathbf{Q}\mathbf{x} - \mathbf{x}^T diag(\mathbf{Q}) \geq \lambda(r+2)(r+1) \;\; \forall \; \tfrac{1}{r+2}\mathbf{x} \in \Delta_n(r)\right\} \\
&= \min\left\{\tfrac{1}{(r+1)(r+2)}[\mathbf{x}^T\mathbf{Q}\mathbf{x} - \mathbf{x}^T diag(\mathbf{Q})] : \;\; \tfrac{1}{r+2}\mathbf{x} \in \Delta_n(r)\right\}.
\end{aligned}$$

Then, by the change of variables

$$\mathbf{y} = \frac{1}{r+2}\mathbf{x},$$

we end up with

$$f_r^* = \frac{r+2}{r+1}\min\{\mathbf{y}^T Q\mathbf{y} - \mathbf{q}_r^T\mathbf{y} \;:\; \mathbf{y} \in \Delta_n(r)\},$$

as we wanted to prove. □

Next, in (Bomze & de Klerk, 2002) the following theorem has been proven, where f^* denotes the maximum value of $\mathbf{x}^T\mathbf{Q}\mathbf{x}$.

Theorem 2.22. *Let*

$$\begin{array}{rl} f_{\Delta_n(r)} = \min & \mathbf{x}^T\mathbf{Q}\mathbf{x} \\ & \mathbf{x} \in \Delta_n(r). \end{array}$$

Then

$$f_{\Delta_n(r)} - f_* \leq \frac{1}{r+2}(f^* - f_*).$$

Proof. It follows from Theorem 2.21 that

$$\begin{array}{rcl} f_{\Delta_n(r)} & \leq & \frac{r+1}{r+2} f_r^* + \frac{1}{r+2}\max_{i=1,\ldots,n} Q_{ii} \\ & \leq & \frac{r+1}{r+2} f_r^* + \frac{1}{r+2} f^* \leq \frac{r+1}{r+2} f_* + \frac{1}{r+2} f^*, \end{array}$$

so that

$$f_{\Delta_n(r)} - f_* \leq \frac{1}{r+2}(f^* - f_*),$$

as we wanted to prove. □

For any fixed $\varepsilon \in (0,1]$ we can choose $\bar{r}$ as follows:

$$\bar{r} = \min\left\{r \in \mathbb{N} \;:\; \frac{1}{r+2} \leq \varepsilon\right\}.$$

Such a choice for r guarantees that the algorithm, based on the evaluation of the objective function at points in $\Delta_n(\bar{r})$, is an ε-approximation algorithm. Since the number of points over the uniform grid $\Delta_n(\bar{r})$ is

$$\begin{pmatrix} n+\bar{r}+1 \\ \bar{r}+2 \end{pmatrix},$$

which, for fixed $\varepsilon \in (0,1]$ and, consequently, for fixed $\bar{r}$, is polynomial with respect to n, we can conclude that the algorithm is a PTAS for the StQP problems.

In a more recent paper (de Klerk, Laurent, & Parrillo, 2006), the above result for StQP has been extended to the minimization of polynomials of fixed degree d over the unit simplex. In such a case the PTAS evaluates the objective function over a grid $\Delta_n(r+d)$.

It is also worthwhile to underline at this point that different definitions of the ε-approximation problem may also lead to quite different outcomes from the point of view of complexity. In particular, de Klerk et al. (de Klerk et al., 2006), still exploiting the result in (Motzkin & Strauss, 1965) about the StQP reformulation of the maximum clique problem and inapproximability results for the same problem, have shown that when employing

Definition 2.7 of the approximation problem, no polynomial time solution can be obtained for any $\varepsilon > 0$, unless $\mathcal{P} = \mathcal{NP}$.

2.6 Approximability and inapproximability results

At a higher level of difficulty, we meet those $\mathcal{NP}$-hard problems for which even the ε-approximation turns out to be a difficult task for small enough ε values, although in some cases the ε-approximation problem becomes solvable in polynomial time when ε gets sufficiently large. Such problems also include some whose objective function and feasible region are of relatively simple form. For instance, Bellare and Rogaway (Bellare & Rogaway, 1995) proved that for the class of QP problems over a polyhedral subset of the unit hypercube

$$\begin{aligned} \min \quad & \mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x} \\ & \mathbf{A}\mathbf{x} \leq \mathbf{b}, \\ & \mathbf{x} \in [0,1]^n, \end{aligned} \tag{2.27}$$

the ε-approximation problem is not solvable in polynomial time for all $\varepsilon < 1/3$ unless $\mathcal{P} = \mathcal{NP}$. Moreover, in (Bellare & Rogaway, 1995) it is proven that for some $\delta > 0$ there exists no polynomial time ε-approximation algorithm for $\varepsilon = (1 - 2^{-\log^{\delta} n})$ if $\mathcal{NP} \not\subseteq \tilde{\mathcal{P}}$, where $\tilde{\mathcal{P}}$ is the class of problems solvable in quasipolynomial time, i.e., time $O(n^{log^k(n)})$. In this case, no polynomial time ε-approximation algorithm exists for any *constant* value $\varepsilon \in (0,1)$ (i.e., the problem is not in $\mathcal{APX}$). The latter result has been proven for cubic (Tardos, 1994) and quartic objective functions (Bellare, Goldwasser, Lund, & Russell, 1993) under the assumption that $\mathcal{P} \neq \mathcal{NP}$. Later results show that even the quadratic case does not belong to $\mathcal{APX}$, unless $\mathcal{P} = \mathcal{NP}$ (see (Feige & Kilian, 1994)), and this is true even when the objective function is a concave separable one like the opposite of the square of the Euclidean norm (see (Brieden, 2002)).

Things do not improve much even when we further restrict the class of QP problems under consideration. Consider the class of concave QP problems over a box, where we further impose that the objective function is concave quadratic and the feasible region is the unit box; i.e., in (2.27) $\mathbf{Q}$ is a negative semidefinite matrix and the constraints $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ are removed. This class includes as a special case a continuous reformulation of the *max cut* problem. Given a graph $G = (V,E)$, the max cut problem aims at detecting a subset $C \subseteq V$ such that the value

$$card(\{(i,j) \in E : \ i \in C, \ j \notin C\})$$

is maximum ($card(A)$ denotes the cardinality of a set A). Such a problem admits the following continuous reformulation. Let $\mathbf{L}$ denote the *Laplacian matrix* of the graph, i.e., the matrix whose entries are

$$L_{ij} = \begin{cases} deg(v_i) & \text{if } i = j, \\ -1 & \text{if } i \neq j, \text{ and } (i,j) \in E, \\ 0 & \text{otherwise,} \end{cases} \tag{2.28}$$

where $deg(v_i)$ is the degree of node $v_i \in V$.

Observation 2.1. *The max cut problem is equivalent to the following GO problem:*

$$
\begin{aligned}
-\min \quad & \frac{1}{4}(2\mathbf{x}-\mathbf{e})^T(-\mathbf{L})(2\mathbf{x}-\mathbf{e}) \\
& \mathbf{x} \in [0,1]^n.
\end{aligned}
\tag{2.29}
$$

Proof. First we notice that the Laplacian matrix $\mathbf{L}$ is positive semidefinite. Indeed, the following condition, which is sufficient for a matrix to be positive semidefinite, is satisfied:

$$
L_{ii} \geq \sum_{j \neq i} |L_{ij}|, \quad i = 1, \ldots, n.
$$

Then, $-\mathbf{L}$ is negative semidefinite and the problem above is a special case of concave QP over a box. Notice that, in view of Theorem 2.11, at least an optimal solution lies at a vertex of the box. For each vertex $\mathbf{y}$ of $[0,1]^n$, the objective function of (2.29) is equal to

$$
\begin{aligned}
& -\tfrac{1}{4}\textstyle\sum_{i=1}^{n}[deg(v_i) - card(\{j \,:\, (i,j) \in E,\ y_i = y_j\}) + card(\{j \,:\, (i,j) \in E,\ y_i \neq y_j\})] \\
& = -\tfrac{1}{2}\textstyle\sum_{i=1}^{n} card(\{j \,:\, (i,j) \in E \text{ and } y_i \neq y_j\}).
\end{aligned}
$$

Now, let $C = \{i \,:\, y_i = 1\}$. Then, the objective function evaluated at $\mathbf{y}$ is equal to

$$
-card(\{(i,j) \in E \,:\, i \in C,\ j \notin C\});
$$

i.e., it is equal to the opposite of the cardinality of the cut defined by C. Since we can associate a vertex $\mathbf{y}$ of the unit box to each $C \subseteq V$, the result immediately follows. □

The inapproximability result proven in (Hastad, 2001) for the max cut problem can be extended to problem (2.29), showing that for such problems (and, consequently, for the broader class of concave QPs over boxes) the ε-approximation problem cannot be solved in polynomial time, unless $\mathcal{P} = \mathcal{NP}$ for all $\varepsilon \leq (1 - \frac{16}{17})$.

On the positive side, a well-known result by Goemans and Williamson (see (Goemans & Williamson, 1995)) for the approximation of the max cut problem shows that an ε-approximation solution for problem (2.29) can be obtained in polynomial time by a randomized algorithm for $\varepsilon = 1 - 0.878$. The approach has been extended in (Ye, 1999) to the broader class of indefinite QPs over boxes. In this case the ε-approximation problem can be solved in polynomial time by a randomized algorithm for $\varepsilon = \frac{\pi}{2} - 1 < \frac{4}{7}$ (in fact, the result is also proven when some quadratic equality constraints are present). The randomized algorithm is now described. Following the literature, we describe it for the corresponding problem of maximizing a quadratic function over a box; in particular, we consider the problem

$$
\begin{aligned}
f^*(\mathbf{Q}) = \max \quad & \mathbf{x}^T\mathbf{Q}\mathbf{x} \\
& -\mathbf{e} \leq \mathbf{x} \leq \mathbf{e}.
\end{aligned}
\tag{2.30}
$$

Notice that omitting the linear terms in the above definition is not a real restriction. Indeed, a problem with a linear term in the objective function such as

$$
\begin{aligned}
\max \quad & \mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x} \\
& -\mathbf{e} \leq \mathbf{x} \leq \mathbf{e}
\end{aligned}
$$

can be rewritten in the form (2.30) by adding a new variable t as follows:

$$\begin{aligned} \max \quad & \mathbf{x}^T\mathbf{Q}\mathbf{x} + t\mathbf{c}^T\mathbf{x} \\ & -\mathbf{e} \le \mathbf{x} \le \mathbf{e}, \\ & -1 \le t \le 1, \end{aligned}$$

and by reformulating the problem as

$$\begin{aligned} \max \quad & \mathbf{y}^T\mathbf{Q}'\mathbf{y} \\ & -\mathbf{e} \le \mathbf{y} \le \mathbf{e}, \end{aligned}$$

where $\mathbf{y} = (\mathbf{x}\ t)$ and

$$\mathbf{Q}' = \begin{bmatrix} \mathbf{Q} & \frac{1}{2}\mathbf{c}^T \\ \frac{1}{2}\mathbf{c} & 0 \end{bmatrix}.$$

There is always an optimal solution $(\mathbf{x}^*\ t^*)$ of the above problem with $t^* = -1$ (in which case $-\mathbf{x}^*$ is an optimal solution of the original problem) or with $t^* = 1$ (in which case $\mathbf{x}^*$ is an optimal solution of the original problem). In order to describe the algorithm, first we need to consider a semidefinite programming relaxation of the problem. We note that for any $\mathbf{x} \in \mathbb{R}^n$

$$\mathbf{x}^T\mathbf{Q}\mathbf{x} = \mathbf{Q} \bullet (\mathbf{x}\mathbf{x}^T), \quad -\mathbf{e} \le \mathbf{x} \le \mathbf{e} \ \Leftrightarrow \ diag(\mathbf{x}\mathbf{x}^T) \le \mathbf{e}.$$

Therefore, (2.30) can be rewritten as

$$\begin{aligned} \max \quad & \mathbf{Q} \bullet \mathbf{X} \\ & diag(\mathbf{X}) \le \mathbf{e}, \\ & \mathbf{X} \succeq \mathbf{O}, \\ & rank(\mathbf{X}) = 1. \end{aligned}$$

If we remove the rank-1 constraint we have the following semidefinite programming relaxation of (2.30):

$$\begin{aligned} \max \quad & \mathbf{Q} \bullet \mathbf{X} \\ & diag(\mathbf{X}) \le \mathbf{e}, \qquad (2.31) \\ & \mathbf{X} \succeq \mathbf{O}. \end{aligned}$$

Now, let $\bar{\mathbf{X}}$ be an optimal solution of this problem, and let

$$\bar{\mathbf{V}} = (\mathbf{v}^1, \ldots, \mathbf{v}^n) \in \mathbb{R}^n$$

($\mathbf{v}^i$ is the ith column of matrix $\bar{\mathbf{V}}$) be such that

$$\bar{\mathbf{X}} = \bar{\mathbf{V}}^T\bar{\mathbf{V}}.$$

Let $\bar{\mathbf{D}}$ be the diagonal matrix whose diagonal entries are

$$D_{ii} = \|\mathbf{v}^i\|_2.$$

For each $\mathbf{x} \in \mathbb{R}^n$, let σ be a n-dimensional vector whose ith entry is defined as

$$\sigma(\mathbf{x})_i = \begin{cases} 1 & \text{if } x_i \geq 0, \\ -1 & \text{otherwise.} \end{cases}$$

Finally, let $\mathbf{u} \in \mathbb{R}^n$ be a vector uniformly sampled over the unit ball. Then, we define the feasible point

$$\hat{\mathbf{x}} = \bar{\mathbf{D}}\sigma(\bar{\mathbf{V}}^T\mathbf{u}).$$

In what follows we prove that the expected value of the objective function evaluated at $\hat{\mathbf{x}}$ differs from the maximum value of the objective function over the box by not more than $\frac{\pi}{2} - 1$ times the difference between the maximum and the minimum value of the objective function over the box. Before proving the result, we remark that the above approach has been further extended (see (Nesterov, Wolkowicz, & Ye, 2000)) to derive an ε-approximation algorithm again with $\varepsilon = \frac{\pi}{2} - 1$ when the box constraints are replaced by the constraints

$$\begin{aligned} &\textstyle\sum_{i=1}^n a_{ij}x_i^2 = b_j, \quad j = 1,\ldots,m, \\ &\textstyle\sum_{i=1}^n c_{ij}x_i^2 \leq d_j, \quad j = 1,\ldots,p. \end{aligned}$$

Let us consider the minimization counterpart of problem (2.30):

$$\begin{aligned} f_*(Q) = \min \quad & \mathbf{x}^T\mathbf{Q}\mathbf{x} \\ & -\mathbf{e} \leq \mathbf{x} \leq \mathbf{e}. \end{aligned}$$

Next, we introduce the dual of problem (2.31):

$$\begin{aligned} g^*(Q) = \min \quad & \mathbf{e}^T\mathbf{y} \\ & Diag(\mathbf{y}) - \mathbf{Q} \in \mathcal{P}_n, \\ & \mathbf{y} \geq \mathbf{0}, \end{aligned} \tag{2.32}$$

where $\mathcal{P}_n$ is the cone of semidefinite matrices (see Definition A.14). Note that strict duality holds since problem (2.31) satisfies Slater's condition (e.g., the matrix $\frac{1}{2}\mathbf{I}_n$ lies in the strict interior of the feasible region of (2.31)). Moreover, let us consider the minimization problem

$$\begin{aligned} g_*(Q) = \min \quad & \mathbf{Q} \bullet \mathbf{X} \\ & diag(\mathbf{X}) \leq \mathbf{e}, \\ & \mathbf{X} \in \mathcal{P}_n, \end{aligned} \tag{2.33}$$

and its dual

$$\begin{aligned} g_*(Q) = \max \quad & \mathbf{e}^T\mathbf{y} \\ & -Diag(\mathbf{y}) + \mathbf{Q} \in \mathcal{P}_n, \\ & \mathbf{y} \leq \mathbf{0}. \end{aligned} \tag{2.34}$$

We will denote by $\mathbf{y}^*$ and $\mathbf{y}_*$ the optimal solutions, respectively, of the dual problem (2.32) and of the dual problem (2.34) (i.e., $\mathbf{e}^T\mathbf{y}^* = g^*(\mathbf{Q})$ and $\mathbf{e}^T\mathbf{y}_* = g_*(\mathbf{Q})$). Obviously,

$$g_*(\mathbf{Q}) \le f_*(\mathbf{Q}) \le f^*(\mathbf{Q}) \le g^*(\mathbf{Q}).$$

In order to prove the main result, we need some lemmas.

Lemma 2.23. *We have that*

$$\begin{array}{rll} f^*(\mathbf{Q}) = & \max & E[\sigma(\mathbf{V}^T\mathbf{u})\mathbf{DQD}\sigma(\mathbf{V}^T\mathbf{u})] \\ & \|\mathbf{v}_i\| \le 1, & i = 1,\dots,n, \end{array} \tag{2.35}$$

where

$$\mathbf{D} = Diag(\|\mathbf{v}_1\|,\dots,\|\mathbf{v}_n\|).$$

Proof. For any feasible $\mathbf{V}$, $\mathbf{D}\sigma\left(\mathbf{V}^T\mathbf{u}\right)$ is feasible for problem (2.30). Then,

$$f^*(\mathbf{Q}) \ge E[\sigma(\mathbf{V}^T\mathbf{u})\mathbf{DQD}\sigma(\mathbf{V}^T\mathbf{u})]. \tag{2.36}$$

Now, for a fixed vector $\mathbf{u}$ with unit norm,

$$E[\sigma(\mathbf{V}^T\mathbf{u})\mathbf{DQD}\sigma(\mathbf{V}^T\mathbf{u})] = \sum_{i=1}^{n}\sum_{j=1}^{n} Q_{ij}\|\mathbf{v}_i\|\|\mathbf{v}_j\|E[\sigma(\mathbf{v}_i^T\mathbf{u})\sigma(\mathbf{v}_j^T\mathbf{u})]. \tag{2.37}$$

Let

$$\mathbf{v}_i = \frac{\bar{x}_i}{\|\bar{\mathbf{x}}\|}\bar{\mathbf{x}}, \quad i = 1,\dots,n,$$

which is feasible for (2.35). Then,

$$E[\sigma(\mathbf{v}_i^T\mathbf{u})\sigma(\mathbf{v}_j^T\mathbf{u})] = \begin{cases} 1 & \text{if } \sigma(\bar{x}_i) = \sigma(\bar{x}_j), \\ -1 & \text{otherwise,} \end{cases}$$

and

$$\|\mathbf{v}_i\|\|\mathbf{v}_j\|E[\sigma(\mathbf{v}_i^T\mathbf{u})\sigma(\mathbf{v}_j^T\mathbf{u})] = \bar{x}_i\bar{x}_j.$$

Then, for such feasible $\mathbf{V}$ we have that

$$f^*(\mathbf{Q}) = \bar{\mathbf{x}}^T\mathbf{Q}\bar{\mathbf{x}} \le E[\sigma(\mathbf{V}^T\mathbf{u})\mathbf{DQD}\sigma(\mathbf{V}^T\mathbf{u})],$$

which, combined with (2.36), concludes the proof. □

The next lemma is proven in (Nesterov, 1997).

Lemma 2.24. *If* $\mathbf{X} \in \mathcal{P}_n$ *and* $diag(\mathbf{X}) \le \mathbf{e}$, *then* $arcsin[\mathbf{X}] - \mathbf{X} \in \mathcal{P}_n$, *where* $arcsin[\mathbf{X}]$ *is the matrix whose entry* (i,j) *is equal to* $arcsin(X_{ij})$.

Proof. Noting that $\mathbf{X} \in \mathcal{P}_n$ and $diag(\mathbf{X}) \leq \mathbf{e}$ imply $|X_{ij}| \leq 1$ for all $i, j = 1, \ldots, n$, the result is an immediate consequence of the Taylor expansion for arcsin($\mathbf{X}$),

$$\arcsin(\mathbf{X}) = \mathbf{X} + \frac{[\mathbf{X}]^3}{6} + \frac{3[\mathbf{X}]^5}{40} + \cdots \quad \text{for } |X_{ij}| \leq 1,$$

where $[\mathbf{X}]^k$ denotes the matrix whose entry (i, j) is equal to X_{ij}^k. □

In (Goemans & Williamson, 1995) the following lemma is proven.

Lemma 2.25. *We have that*

$$P\left[\sigma\left(\frac{\mathbf{v}_i^T\mathbf{u}}{\|\mathbf{v}_i\|}\right) \neq \sigma\left(\frac{\mathbf{v}_j^T\mathbf{u}}{\|\mathbf{v}_j\|}\right)\right] = \frac{1}{\pi}\arccos\left(\frac{\mathbf{v}_i^T\mathbf{v}_j}{\|\mathbf{v}_i\|\|\mathbf{v}_j\|}\right).$$

Proof. Since

$$P\left[\sigma\left(\frac{\mathbf{v}_i^T\mathbf{u}}{\|\mathbf{v}_i\|}\right) \neq \sigma\left(\frac{\mathbf{v}_j^T\mathbf{u}}{\|\mathbf{v}_j\|}\right)\right] = P[\mathbf{v}_i^T\mathbf{u} \geq 0,\ \mathbf{v}_j^T\mathbf{u} < 0] + P[\mathbf{v}_i^T\mathbf{u} < 0,\ \mathbf{v}_j^T\mathbf{u} \geq 0],$$

by symmetry we have that

$$P\left[\sigma\left(\frac{\mathbf{v}_i^T\mathbf{u}}{\|\mathbf{v}_i\|}\right) \neq \sigma\left(\frac{\mathbf{v}_j^T\mathbf{u}}{\|\mathbf{v}_j\|}\right)\right] = 2P[\mathbf{v}_i^T\mathbf{u} \geq 0,\ \mathbf{v}_j^T\mathbf{u} < 0].$$

The set

$$\{\mathbf{v}_i^T\mathbf{u} \geq 0,\ \mathbf{v}_j^T\mathbf{u} < 0\}$$

is the intersection of two half-spaces, and the intersection of this set with the n-dimensional unit sphere is a digon whose measure is equal to

$$\frac{1}{\pi}\arccos\left(\frac{\mathbf{v}_i^T\mathbf{v}_j}{\|\mathbf{v}_i\|\|\mathbf{v}_j\|}\right)$$

times the measure of the unit sphere, from which the result follows. □

Next, we prove the following theorems.

Theorem 2.26. *We have that*

$$\begin{aligned} f^*(\mathbf{Q}) = \quad & \sup\ \tfrac{2}{\pi}\mathbf{Q} \bullet \mathbf{D}\arcsin\left[\mathbf{D}^{-1}\mathbf{X}\mathbf{D}^{-1}\right]\mathbf{D}, \\ & diag(\mathbf{X}) \leq \mathbf{e}, \\ & \mathbf{X} \succ \mathbf{O}, \end{aligned}$$

where

$$\mathbf{D} = Diag(\sqrt{X_{11}}, \ldots, \sqrt{X_{nn}}). \tag{2.38}$$

Proof. For all $\mathbf{X} = \mathbf{V}\mathbf{V}^T \succ \mathbf{O}$ with $diag(\mathbf{X}) \leq \mathbf{e}$,

$$\begin{aligned}
&E\left[\sigma\left(\mathbf{v}_i^T\mathbf{u}\right)\sigma\left(\mathbf{v}_j^T\mathbf{u}\right)\right] \\
&= P\left[\sigma\left(\frac{\mathbf{v}_i^T\mathbf{u}}{\|\mathbf{v}_i\|}\right) = \sigma\left(\frac{\mathbf{v}_j^T\mathbf{u}}{\|\mathbf{v}_j\|}\right)\right] - P\left[\sigma\left(\frac{\mathbf{v}_i^T\mathbf{u}}{\|\mathbf{v}_i\|}\right) \neq \sigma\left(\frac{\mathbf{v}_j^T\mathbf{u}}{\|\mathbf{v}_j\|}\right)\right] \\
&= 1 - 2P\left[\sigma\left(\frac{\mathbf{v}_i^T\mathbf{u}}{\|\mathbf{v}_i\|}\right) \neq \sigma\left(\frac{\mathbf{v}_j^T\mathbf{u}}{\|\mathbf{v}_j\|}\right)\right].
\end{aligned}$$

Then, the result follows from (2.37), Lemma 2.25, and $\arcsin(y) + \arccos(y) = \frac{\pi}{2}$. □

Theorem 2.27. *We have that:*

1. $$f^*(\mathbf{Q}) - g_*(\mathbf{Q}) \geq \frac{2}{\pi}\left(g^*(\mathbf{Q}) - g_*(\mathbf{Q})\right);$$

2. $$g^*(\mathbf{Q}) - f_*(\mathbf{Q}) \geq \frac{2}{\pi}\left(g^*(\mathbf{Q}) - g_*(\mathbf{Q})\right);$$

3. $$g^*(\mathbf{Q}) - g_*(\mathbf{Q}) \geq f^*(\mathbf{Q}) - f_*(\mathbf{Q}) \geq \frac{4-\pi}{\pi}\left(g^*(\mathbf{Q}) - g_*(\mathbf{Q})\right).$$

Proof. From Theorem 2.26 we have that for any $\mathbf{X} \succ \mathbf{O}$ with $diag(\mathbf{X}) \leq \mathbf{e}$, $\mathbf{D}$ defined as in (2.38), and recalling that $\mathbf{y}_*$ is the optimal solution of (2.34),

$$\begin{aligned}
\tfrac{\pi}{2} f^*(\mathbf{Q}) &\geq \mathbf{Q} \bullet \mathbf{D}\arcsin\left[\mathbf{D}^{-1}\mathbf{X}\mathbf{D}^{-1}\right]\mathbf{D} \\
&= (\mathbf{Q} - Diag(\mathbf{y}_*)) \bullet \mathbf{D}\arcsin\left[\mathbf{D}^{-1}\mathbf{X}\mathbf{D}^{-1}\right]\mathbf{D} + Diag(\mathbf{y}_*) \bullet \mathbf{D}\arcsin\left[\mathbf{D}^{-1}\mathbf{X}\mathbf{D}^{-1}\right]\mathbf{D} \\
&\geq (\mathbf{Q} - Diag(\mathbf{y}_*)) \bullet \mathbf{D}\mathbf{D}^{-1}\mathbf{X}\mathbf{D}^{-1}\mathbf{D} + Diag(\mathbf{y}_*) \bullet \mathbf{D}\arcsin\left[\mathbf{D}^{-1}\mathbf{X}\mathbf{D}^{-1}\right]\mathbf{D},
\end{aligned}$$

where the last inequality follows from $\mathbf{Q} - Diag(\mathbf{y}_*) \in \mathcal{P}_n$ and $\arcsin\left[\mathbf{D}^{-1}\mathbf{X}\mathbf{D}^{-1}\right] - \mathbf{D}^{-1}\mathbf{X}\mathbf{D}^{-1} \in \mathcal{P}_n$ (in view of Lemma 2.24). Thus,

$$\begin{aligned}
\tfrac{\pi}{2} f^*(\mathbf{Q}) &\geq (\mathbf{Q} - Diag(\mathbf{y}_*)) \bullet \mathbf{X} + Diag(\mathbf{y}_*) \bullet \mathbf{D}\arcsin\left[\mathbf{D}^{-1}\mathbf{X}\mathbf{D}^{-1}\right]\mathbf{D} \\
&= \mathbf{Q} \bullet \mathbf{X} - Diag(\mathbf{y}_*) \bullet \mathbf{X} + Diag(\mathbf{y}_*) \bullet \mathbf{D}\arcsin\left[\mathbf{D}^{-1}\mathbf{X}\mathbf{D}^{-1}\right]\mathbf{D} \\
&= \mathbf{Q} \bullet \mathbf{X} - \mathbf{y}_*^T diag(\mathbf{X}) + \mathbf{y}_*^T diag\left(\mathbf{D}\arcsin\left[\mathbf{D}^{-1}\mathbf{X}\mathbf{D}^{-1}\right]\mathbf{D}\right) \\
&= \mathbf{Q} \bullet \mathbf{X} - \mathbf{y}_*^T diag(\mathbf{X}) + \tfrac{\pi}{2}\mathbf{y}_*^T diag(\mathbf{X}),
\end{aligned}$$

where the last equality follows from the fact that, by definition of $\mathbf{D}$,

$$diag\left(\mathbf{D}\arcsin\left[\mathbf{D}^{-1}\mathbf{X}\mathbf{D}^{-1}\right]\mathbf{D}\right) = \frac{\pi}{2} diag(\mathbf{X}).$$

Therefore,

$$\begin{aligned}
\tfrac{\pi}{2} f^*(\mathbf{Q}) &\geq \mathbf{Q} \bullet \mathbf{X} + \left(\tfrac{\pi}{2} - 1\right)\mathbf{y}_*^T diag(\mathbf{X}) \\
&\geq \mathbf{Q} \bullet \mathbf{X} + \left(\tfrac{\pi}{2} - 1\right)\mathbf{y}_*^T \mathbf{e} \\
&= \mathbf{Q} \bullet \mathbf{X} + \left(\tfrac{\pi}{2} - 1\right) g_*(\mathbf{Q}),
\end{aligned}$$

where the last inequality follows from $\mathbf{0} \leq diag(\mathbf{X}) \leq \mathbf{e}$ and $\mathbf{y}_* \leq \mathbf{0}$. By taking $\mathbf{X} \to \bar{\mathbf{X}}$, we have $\mathbf{Q} \bullet \mathbf{X} \to g^*(\mathbf{Q})$, from which the first inequality follows. The second can be proven in a completely analogous way, while the third can be simply obtained by adding the first two. □

We also have the following corollary.

Corollary 2.28. *Let* $\mathbf{X} = \mathbf{V}\mathbf{V}^T \succ 0$, $diag(\mathbf{X}) \leq \mathbf{e}$, $\mathbf{D}$ *be defined as in* (2.38)*, and let*

$$\hat{\mathbf{x}} = \mathbf{D}\sigma\left(\mathbf{V}^T\mathbf{u}\right), \tag{2.39}$$

where $\mathbf{u}$ *is a vector uniformly randomly generated over the unit ball. Then*

$$\lim_{\mathbf{X}\to\bar{\mathbf{X}}} E[\hat{\mathbf{x}}^T\mathbf{Q}\hat{\mathbf{x}}] = \lim_{\mathbf{X}\to\bar{\mathbf{X}}} \frac{2}{\pi}\mathbf{Q} \bullet \mathbf{D}\arcsin\left[\mathbf{D}^{-1}\mathbf{X}\mathbf{D}^{-1}\right]\mathbf{D} \geq \frac{2}{\pi}g^*(\mathbf{Q}) + \left(1 - \frac{2}{\pi}\right)g_*(\mathbf{Q}).$$

Now we are ready for the final theorem.

Theorem 2.29. *Let* $\hat{\mathbf{x}}$ *be defined as in* (2.39) *for* $\mathbf{X} = \bar{\mathbf{X}}$. *Then*

$$\frac{f^*(\mathbf{Q}) - E[\hat{\mathbf{x}}^T\mathbf{Q}\hat{\mathbf{x}}]}{f^*(\mathbf{Q}) - f_*(\mathbf{Q})} \leq \frac{\pi}{2} - 1.$$

Proof. Since

$$\begin{aligned} g^*(\mathbf{Q}) \geq f^*(\mathbf{Q}) &\geq \frac{2}{\pi}g^*(\mathbf{Q}) + \left(1 - \frac{2}{\pi}\right)g_*(\mathbf{Q}) \\ &\geq \frac{2}{\pi}g_*(\mathbf{Q}) + \left(1 - \frac{2}{\pi}\right)g^*(\mathbf{Q}) \\ &\geq f_*(\mathbf{Q}) \geq g_*(\mathbf{Q}), \end{aligned}$$

where the next-to-last inequality follows from point 2 of Theorem 2.27, we have, also in view of Corollary 2.28,

$$\begin{aligned} \frac{f^*(\mathbf{Q})-E[\hat{\mathbf{x}}^T\mathbf{Q}\hat{\mathbf{x}}]}{f^*(\mathbf{Q})-f_*(\mathbf{Q})} &\leq \frac{f^*(\mathbf{Q})-\frac{2}{\pi}g^*(\mathbf{Q})-\left(1-\frac{2}{\pi}\right)g_*(\mathbf{Q})}{f^*(\mathbf{Q})-f_*(\mathbf{Q})} \\ &\leq \frac{f^*(\mathbf{Q})-\frac{2}{\pi}g^*(\mathbf{Q})-\left(1-\frac{2}{\pi}\right)g_*(\mathbf{Q})}{f^*(\mathbf{Q})-\frac{2}{\pi}g_*(\mathbf{Q})-\left(1-\frac{2}{\pi}\right)g^*(\mathbf{Q})} \\ &\leq \frac{g^*(\mathbf{Q})-\frac{2}{\pi}g^*(\mathbf{Q})-\left(1-\frac{2}{\pi}\right)g_*(\mathbf{Q})}{g^*(\mathbf{Q})-\frac{2}{\pi}g_*(\mathbf{Q})-\left(1-\frac{2}{\pi}\right)g^*(\mathbf{Q})} \\ &= \frac{\left(1-\frac{2}{\pi}\right)(g^*(\mathbf{Q})-g_*(\mathbf{Q}))}{\frac{2}{\pi}(g^*(\mathbf{Q})-g_*(\mathbf{Q}))} \\ &= \frac{\left(1-\frac{2}{\pi}\right)}{\frac{2}{\pi}} = \frac{\pi}{2} - 1. \end{aligned}$$ □

Other approximation algorithms, with the related worst-case performance ratios, have been discussed in (He, Li, & Zhang, 2010; Ling et al., 2009; Z. Q. Luo, Sidiropoulos, Tseng, & Zhang, 2007; Z. Q. Luo & Zhang, 2011; So, 2011; Y. Yang & Yang, 2012; X. Zhang, Ling, & Qi, 2011) for different problems with a polynomial objective function and quadratic constraints. Feasible regions for such problems include the border of the unit sphere and the intersection of a finite number of ellipsoids.

An inapproximability result has been proven in (Locatelli, 2009) for a class of problems defined as follows:

$$\begin{array}{lll} \min & -\sum_{i=1}^{n} f_i(x_i) & \\ & x_{i_1}+x_{i_2} \leq 1, & (i_1,i_2) \in A, \\ & x_i \geq 0, & i=1,\ldots,n, \end{array} \tag{2.40}$$

where $A \subseteq \{(i_1,i_2)\ :\ i_1,i_2=1,\ldots,n,\ \ i_1 \neq i_2\}$, and for the one-dimensional functions f_i we have that

- the functions f_i are convex and nondecreasing,
- $f_i(0)=0$ and $f_i(1)=1$.

Therefore, the objective function is concave separable, while the feasible region is defined by clique constraints (this name comes from the fact that these are the constraints appearing in the formulation of a clique problem as a binary linear program). In (Locatelli, 2009) it is proven that if $f_i(\frac{1}{2}) \leq \frac{1}{3}$ for all $i=1,\ldots,n$, then the ε-approximation problem for (2.40) cannot be solved in polynomial time, unless $\mathcal{P}=\mathcal{NP}$, for $\varepsilon=0.01492$. In particular, if we consider quadratic functions f_i defined as

$$f_i(x_i)=\alpha x_i^2+(1-\alpha)x_i,$$

where $\alpha \in [0,1]$ can be viewed as a weight of the quadratic term with respect to the linear one, the inapproximability result holds for all $\alpha \geq \frac{2}{3}$, i.e., as soon as the weight of the quadratic term is sufficiently large.

On the positive side, a very simple algorithm based on the solution of the linear program with the same feasible region as in (2.40) and objective function $\sum_{i=1}^{n} x_i$ returns in polynomial time a $(1-2\rho)$-approximate solution if

$$f_i\left(\frac{1}{2}\right) \geq \rho$$

for some $\rho>0$.

Inapproximability results also exist for other problems with a simple separable concave function and a highly structured feasible region. Let the feasible region X be a full-dimensional *parallelotope* centered at the origin, i.e.,

$$X=\left\{\mathbf{x} \in \mathbb{R}^n\ :\mathbf{x}=\sum_{i=1}^{n} \lambda_i \mathbf{v}_i,\ \ \lambda_i \in [-1,1]\right\},$$

or with one vertex at the origin, i.e.,

$$X = \left\{ \mathbf{x} \in \mathbb{R}^n \ : \mathbf{x} = \sum_{i=1}^{n} \lambda_i \mathbf{v}_i, \ \ \lambda_i \in [0,1] \right\},$$

where in both cases the vectors $\mathbf{v}_i$'s are linearly independent. By denoting with $\mathbf{V}$ the nonsingular matrix of order n whose ith column is the vector $\mathbf{v}_i$, we can also represent a parallelotope centered at the origin as

$$X = \left\{ \mathbf{x} \in \mathbb{R}^n \ : -\mathbf{e} \leq \mathbf{V}^{-1}\mathbf{x} \leq \mathbf{e} \right\}.$$

A similar definition can be given for a parallelotope with a vertex at the origin. Then, in (Bodlaender, Gritzmann, Klee, & van Leeuwen, 1990) it is proven that, unless $\mathcal{P} = \mathcal{NP}$, the opposite of the square of the Euclidean norm (a rather simple concave quadratic separable objective function) over X cannot be minimized in polynomial time, while in (Brieden, Gritzmann, & Klee, 2000) it has been proven that even the ε-approximation problem cannot be solved in polynomial time for $\varepsilon \leq 0.0825$. On the positive side, noting that the change of variables

$$\mathbf{y} = \mathbf{V}^{-1}\mathbf{x} \tag{2.41}$$

leads to a QP problem over a box, we can immediately conclude that the previously discussed algorithm proposed in (Ye, 1999) for the approximation of QPs over boxes can be applied to these problems and returns an ε-approximate solution in polynomial time for $\varepsilon = \frac{\pi}{2} - 1$. We also remark that the change of variables (2.41) leads to a QP over a box even when the objective function is a general indefinite quadratic function, so that the approximation result is valid also in this case.

2.7 An overview of complexity results for QP

In order to have a clearer picture and summarize some of the results previously discussed, here we present Table 2.1 with known (and unknown) results about the complexity of some subclasses of GO problems. Our attention is restricted to problems with a quadratic objective function and a polyhedral or spherical feasible region, because such cases already allow us to cover a broad variety of complexity results. In Table 2.1 rows refer to objective functions, while columns refer to feasible regions. The objective functions are distinguished in four classes:

- concave separable;
- general concave;
- indefinite separable;
- general indefinite.

Note that the separable cases are always subclasses of the corresponding general cases, while the concave cases are always subclasses of the corresponding indefinite cases.

Table 2.1. *Complexity of QPs over different feasible regions*

		Sphere	Box	Simplex	Knapsack	Clique	Parallelotope	Polytope
Concave	Separable	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$	FPTAS	$\mathcal{APX}$	$\mathcal{APX}$	$\notin \mathcal{APX}$
	General	$\mathcal{P}$	$\mathcal{APX}$	$\mathcal{P}$	?	?	$\mathcal{APX}$	$\notin \mathcal{APX}$
Indefinite	Separable	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$	FPTAS	?	$\mathcal{APX}$	$\notin \mathcal{APX}$
	General	$\mathcal{P}$	$\mathcal{APX}$	PTAS	?	?	$\mathcal{APX}$	$\notin \mathcal{APX}$

For the feasible regions, we consider

- sphere;
- unit simplex;
- box;
- knapsack;
- clique;
- parallelotope;
- polytope.

Finally, the entries in the table are the following:

- $\mathcal{P}$: the problem is solvable in polynomial time;
- FPTAS: the problem admits a FPTAS;
- PTAS: the problem admits a PTAS;
- $\mathcal{APX}$: the ε-approximation problem is solvable in polynomial time for sufficiently large ε;
- $\notin \mathcal{APX}$: the problem is not solvable in polynomial time for any ε, unless $\mathcal{P} = \mathcal{NP}$;
- ? : not aware of complexity results about this case in the literature.

Concerning the ? entry related to QPs over knapsack constraints when the objective function is concave and nonseparable, we observe that, for ε small enough, an ε-approximation solution cannot be detected in polynomial time (as usual, unless $\mathcal{P} = \mathcal{NP}$). This follows from the *max bisection* problem, i.e., the max cut problem with the additional requirement that the two subsets into which the set of nodes is partitioned have the same cardinality (obviously, the cardinality of the set of nodes must be even). Any max cut problem can be reformulated as a max bisection problem by adding a suitable number of isolated nodes, i.e., nodes which are not adjacent to any other node. We need only add a number of isolated nodes not larger than the cardinality of the original set of nodes. Such nodes give a null contribution to the objective function and can be added to one subset or the other in order to balance the cardinality of the two subsets. Therefore, the inapproximability result for max cut can be extended to max bisection. The max bisection problem can be formulated as follows:

$$\begin{aligned} -\min \quad & \tfrac{1}{4}(2\mathbf{x}-\mathbf{e})^T(-\mathbf{L})(2\mathbf{x}-\mathbf{e}) \\ & \textstyle\sum_{i=1}^n x_i = \frac{n}{2}, \\ & x \in [0,1]^n, \end{aligned}$$

where $\mathbf{L}$ is the Laplacian matrix defined in (2.28). Notice that the maximum value of this problem is 0 and is attained when $x_i = \frac{1}{2}$ for $i = 1,\ldots,n$. Also note that, since the objective function is concave, an optimal solution lies at a vertex of the feasible region (see Theorem 2.11), and any vertex of the feasible region has binary coordinates. Indeed, $n-1$ nonbasic variables must have value equal to 0 or 1, so that, $\frac{n}{2}$ being an integer value, the remaining basic variable must also have value equal to 0 or 1. This problem turns out to be an instance of concave QP over knapsack constraints, so that we can further extend the inapproximability result for max cut to this class.

We finally point out that the complexity results displayed in Table 2.1 can be improved if we impose further restrictions on the quadratic objective functions. For instance, the previously mentioned result in (Vavasis, 1992) shows that QPs with a fixed number of negative eigenvalues over a polytope admit a FPTAS, and different FPTASs have been obtained for the special case of a quadratic function obtained as the product of two affine functions, even when the feasible region is an unbounded polyhedron. As a side result of a more general theory which can be found in (Sharkey, Romeijn, & Geunes, 2011), we also have that, if the objective function of a QP with knapsack constraints has the special form

$$\mathbf{c}^T\mathbf{x} - (\mathbf{d}^T\mathbf{x} + d_0)^2,$$

then the problem can be solved in polynomial time $O(n^2 \log(n))$.

Chapter 3

Heuristics

The first section in this chapter is devoted to the presentation of some algorithmic schemes which can be used to find good solutions to GO problems. Methods included in this section presume that evaluating the objective function is sufficiently cheap—so cheap that it is often possible to perform local optimization runs. Thus most methods in this section will be based on the exploitation of local optimization and in the generation of samples of quite large cardinality. In the following section, models and methods are presented for GO problems in which evaluating the objective function is extremely expensive from a computational point of view. Thus the focus of Section 3.2 is on trying to save as much as possible in function evaluations, possibly devoting a large computational effort in deciding the few points where the objective function has to be evaluated. The last section in this chapter is devoted to some basic results concerning the capability of heuristic GO methods of eventually locating the global optimum.

3.1 Heuristic methods

Although there exists a huge number of methods for computing good, though not necessarily optimal, solutions to GO problems, most of them simply contain specializations and customizations of a few basic tools. It can be quite safely assumed that most methods for GO repeatedly switch between two phases, which we might denominate as *local* and *global* phases. Roughly speaking, during the global phase all of the feasible region is potentially explored, while in the local phase we are restricted to explore a (more or less small) portion of the feasible region.

During a local phase, given a tentative solution, which might even be infeasible, an exploration is performed by sampling more observations in a neighborhood of the current point, in the hope of finding an improved solution; here "improved" might mean different things and, in general, a criterion should also be defined in order to be able to discriminate between improving and nonimproving moves. Technically, *improvement* might be defined by introducing an *order relation* $\preccurlyeq$ in the space of solutions. More precisely, $\preccurlyeq$ will in general be a pre-order, i.e., a binary relation which is reflexive ($x \preccurlyeq x$) and transitive ($x \preccurlyeq y, y \preccurlyeq z$ implies $x \preccurlyeq z$). A *strict improvement* $x \prec y$ can be defined when $x \preccurlyeq y$ while $y \not\preccurlyeq x$. For feasible solutions, a strict improvement is usually defined in terms of a lower function value, although more stringent requirements might sometimes be imposed, such as that of a sufficient improvement. For infeasible solutions, an improvement might be defined taking into account total absolute constraint violation, or maximum constraint violation, or any other measure of violation, possibly including a mixture of infeasibility measure and function value. It is worth remembering that many successful methods in GO, while performing a local exploration, do not always require that points which are generated and accepted actually form a monotonic sequence with respect, e.g., to function value. Many methods prescribe, during their local phase, accepting moves which are nonimproving. In this sense, the definition of improved solution by means of the order $\preccurlyeq$ should be completed with a more general concept, which is that of an "acceptable" solution. Each algorithm defines a suitable acceptance criterion in order to judge whether a perturbation of the current solution within a prescribed neighborhood is worth being accepted; if the answer is yes, usually the accepted point replaces the current one and becomes the reference point for possibly further local phases. Examples of local phases are standard local searches performed by means of local optimization methods, or methods in which a ball of prescribed radius centered in the current solution is repeatedly sampled until either an acceptable point is found or a stopping criterion is met.

Concerning the global phase, its aim is usually that of exploring the search domain, as opposed to that of the local phase which is more concerned with refinement of the current solution. As such, the global phase usually does not depend on a single, current solution, but most of the time it consists of the generation of new candidate points in a search space which frequently is the feasible set, but which might also be different. As an example, for problems with nonlinear constraints, the search space might be a box containing the feasible set or any other relaxation of the constraints, so that random generation of points within the search space might be relatively easy even when generating a feasible solution might be difficult, or even impossible (in the case of empty feasible sets). Being composed of exploratory moves, global phases might depend on the history of the algorithm, but they are usually based on methods which generate points without restricting themselves to neighborhoods of past solutions. On the contrary, these methods might prescribe generating points which are as far as possible from past observations. Examples of global phases are random (uniform) generation of feasible points or the choice of points which maximize some measure of their distance from all previously sampled points.

Most good heuristic methods are composed of a clever mixture of these two phases; clearly, any method which does not perform a sufficient number of global phases incurs the risk of being trapped in a local minimum, even if run for a very long time. On the other hand, a method which just relies on global phases might be extremely slow to converge to a global solution, as it does not exploit any characteristic of the problem to be solved, such as the continuity of the objective function.

Browsing the huge literature on heuristic GO methods, it quickly becomes evident that another important characteristic of methods that should be taken into account concerns the sequential or parallel nature of the methods. By this we do not necessarily mean the use of parallel hardware, although this is currently an affordable and useful option. We mainly would like to distinguish between methods which are based on a single candidate solution, which we will call *sequential methods*, and methods which maintain a "population" of solutions (*population-based methods*). In other words, we distinguish between methods in which the "current" iterate is a single point in $\mathbb{R}^n$ and methods in which the iterate is a set of points in the same Euclidean space; usually it will be convenient to arrange such a set of points in a matrix whose columns are single points. Of course we might just consider that any method can be seen as population based, with sequential methods being defined as a special case in which the set of points (the population) has cardinality one. We will in some sense proceed in this way, although it is important to remark that there is a profound difference between sequential and population algorithms. Even population-based algorithms might be seen in terms of local and global phases: in this case both phases are built in order to generate sets of solutions. In a population-based local phase, every point in the set (every individual in the population) might be defined to belong to a prescribed neighborhood of a single point (its "parent"), but more elaborate methods might be defined in which a whole new set of points is generated taking into account the current population. Even the acceptance criterion in a population-based local phase might be radically different from the sequential one. A point generated within the population might be accepted if it is better than its parent(s); however, a generated point might also be accepted if, without being preferable with respect to its direct parents, it is preferable with respect to a different point in the set (a different member in the population). Very elaborate rules both for the generation and for the acceptance of new sets of points might be defined. It is usual to define a population-based acceptance criterion as a method which, given the current and the generated population, defines the new current population. Of course, when the set is composed of a single solution, these more elaborate forms for the acceptance criterion become useless, and this is why it is often easier to treat sequential and population-based methods separately.

Before defining a general scheme for GO heuristic methods, it seems worthwhile to emphasize that, along with the evolution of the set of solution points, it is usually convenient to enrich the structure of the algorithm by means of a "state" which can evolve from iteration to iteration. As an example, the state might contain the best feasible point observed so far (the so-called *record*); in more refined methods, the state might include information on parts of the feasible region which are not worth exploring anymore.

Algorithm 1 defines a general scheme for a population-based method in which local phases are nested within a cycle of global ones. This scheme, while not completely general, encompasses a large class of sequential as well as population-based methods.

```
$\mathcal{S} = Initialize()$ ;
while GlobalStoppingRule is false do
    $\mathbf{X} = GloballyGenerate()$;
    $\mathcal{S} = StateUpdate(\mathcal{S})$ ;
    while LocalStoppingRule is false do
        $\mathbf{Y} = LocallyGenerate(\mathbf{X}; \mathcal{S})$ ;
        $\mathbf{X} = Select(\mathbf{X}, \mathbf{Y}; \mathcal{S})$ ;
        $\mathcal{S} = StateUpdate(\mathcal{S})$ ;
    end
end
```

Algorithm 1: Generic population GO method

Let us introduce some notation: here, differently from the rest of this book, we used bold capital letters to denote the iterates. This is due to the fact that the above scheme is general enough to include both standard sequential algorithms and population-based algorithms. In practice, for sequential methods $\mathbf{X}$ will be an array in $\mathbb{R}^n$, but for population algorithms $\mathbf{X}$ will denote a matrix, or population, composed of a certain number, P, of vectors in $\mathbb{R}^n$. Of course, the sequential case can be retrieved immediately letting $P = 1$. For populations, $\mathbf{X}$ will be in $\mathbb{R}^{n \times P}$, or

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_P \end{bmatrix}$$

so that a connection between a population and one of its members will be obtained by using the same letter—uppercase for populations and lower case for vectors. When needed, the ith coordinate of the jth vector in the population $\mathbf{X}$ will be denoted by X_{ij}, as in standard matrix notation.

In what follows, we describe in more detail the main procedures of this algorithm.

Initialize

This is meant for general definition of parameters of the algorithm and initial settings, such as choosing a maximum number of steps to be performed, defining specific parameters for the algorithm, choosing a seed for random number generation, and defining the search space. This procedure initializes the state $\mathcal{S}$ of the method. The *StateUpdate* procedure is used to maintain the state of the algorithm in such a way that it is coherent with the evolution of the method. As an example, if the state consists of the current record, the updating rule simply consists in replacing the current record with a newly discovered, better solution.

GloballyGenerate

Here a global exploration is performed in order to generate points, or populations of points, whose aim is to sample parts of the domain which have not yet been thoroughly explored. As an example, this procedure might simply consist in drawing uniform random vectors in the search space. Another possible implementation of this procedure, when choosing a single point, might be that of generating, possibly after invoking a suitable solver, a point in the search space which is as far as possible from all points which have been sampled up to now. For a population-based method, this might consist in finding P points in such a way that after this choice the minimal distance between any newly generated point and its closest neighbor is maximized; i.e., we try to leave the smallest possible "holes" in the search domain. This procedure of finding points which are far enough from sampled ones might include knowledge of the objective function, in order to give possibly different weights to being close to a point with high or low function value. It is worth remarking that this kind of global generation might be a computationally demanding task. In a constrained optimization problem, this global phase might also consist in returning one (or more) feasible solution, or, e.g., some extreme points of the feasible domain, depending on the specific problem at hand.

LocallyGenerate

This is often the most characteristic part in the definition of a GO algorithm, the one which radically distinguishes one method from another. The term "locally" needs to be instantiated: this is usually obtained either through the definition of a set which defines the neighborhood of the current solution or in a procedural way through the introduction of "perturbation moves" to be applied to the current solution.

We distinguish between the case of sequential methods and population-based ones. In the first case, a neighborhood might be classical, e.g., a ball centered in the current point $\mathbf{x}_k$:

$$\{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x} - \mathbf{x}_k\| \leq \varepsilon\}.$$

Here the norm might be any norm and ε is a given positive quantity (which, in some algorithms, might also depend on the current iteration, giving rise to so-called *variable neighborhood search* methods (Mladenovic & Hansen, 1997)). Alternatively, a locally generated point $\mathbf{y}_k$ might be seen as the result of a move:

$$\mathbf{y}_k = \mathbf{x}_k + \mathbf{d},$$

where $\|\mathbf{d}\| \leq \varepsilon$. It is important to observe that what here is called a "neighborhood" might radically differ from the above definitions. A typical example of "local" modification is the result of the combination of two operations—one which generates a point in a prescribed ball centered at $\mathbf{x}_k$, and another which, starting from the generated point, executes a local optimization algorithm. In this case, the result of the local move is a point which hopefully corresponds to a local optimum of the objective function obtained by starting a descent algorithm from a point which is close to the current one. Algorithms based on this kind of neighborhood are known in the literature as *Basin Hopping* (BH) (Wales & Doye, 1997) or *Iterated Local Search* (ILS) (Lourenço, Martin, & Stützle, 2003) methods. This procedure, composed of a perturbation followed by a local optimization, is so common in good heuristic methods for GO that it is worth giving explicit names to its components, as done in Algorithm 2.

```
Data: A starting solution $\mathbf{X}$
Result: A local optimum $\mathbf{Z}$ obtained starting a local search from a neighbor of $\mathbf{X}$
begin
    $\mathbf{Y} = LocallyPerturb(\mathbf{X})$
    $\mathbf{Z} = LocallyOptimizeFrom(\mathbf{Y})$
    return $\mathbf{Z}$;
end
```

Algorithm 2: *LocallyGenerate*: An ILS-type local generation

Here and in what follows we used capital letters to denote populations of solutions; we did not give precise definitions of population related concepts, like that of local optimum, which might be generalized from those of sequential methods. No precise definition is given here of the two procedures *LocallyPerturb* and *LocallyOptimizeFrom*, so any perturbation of the current iterate might be seen as a special case within this framework. However, it is understood that *LocallyPerturb* is indeed a local move, obtained either through the definition of a suitable neighborhood $\mathcal{N}(\mathbf{X})$ of the current iterate or through some "simple" modification rule (e.g., changing only a subset of variables). On the other hand, *LocallyOptimizeFrom* is a possibly complex optimization routine which, given a starting point $\mathbf{Y}$, usually (i.e., unless it fails or it is artificially stopped before convergence) returns a *locally optimal solution* $\mathbf{Z}$, characterized by the fact that there exists a radius ε such that no feasible solution whose distance from $\mathbf{Z}$ is at most ε can be strictly "better" than $\mathbf{Z}$ (with respect to the pre-order $\preccurlyeq$):

$$\|\mathbf{X} - \mathbf{Z}\| \leq \varepsilon \Rightarrow \mathbf{Z} \preccurlyeq \mathbf{X}.$$

Here we assume that in order to define the concept of "local optimum" a ball defined by means of the Euclidean norm is used in order to check that $\mathbf{Z}$ is locally optimal. Other definitions of a neighborhood might be given as well, but it is important to stress that the neighborhood used in the definition of a local optimum does not coincide with the neighborhood structure used in Algorithm 2. By this we mean that the concept of local optimality is the usual one, which relies on the existence of a ball centered in the local optimum $\mathbf{Z}$ characterized by the fact that no other feasible point in the same ball is strictly better than $\mathbf{Z}$. The neighborhood structure introduced in the algorithm, however, is more general: given a solution $\mathbf{X}$, a neighbor is generated by first perturbing $\mathbf{X}$ and subsequently starting a local optimization method from the perturbed point. It can be easily seen that the point returned by this procedure can possibly be quite far from the original one.

It is worthwhile to introduce here some definitions which will become useful in various sections. Given a neighborhood structure $\mathcal{N}(\cdot)$ associated to each point in the search domain, a pre-order relation $\preccurlyeq$, and a local search procedure $\mathcal{L}(\cdot)$ (with $\mathcal{L}(\mathbf{Y}) =$ *LocallyOptimizeFrom*($\mathbf{Y}$)), it is possible to define a hierarchical structure in the space of local optima. We assume that the local search is monotonic with respect to the pre-order; i.e., when started from a solution $\mathbf{Y}$ it will return another solution $\mathcal{L}(\mathbf{Y}) \preccurlyeq \mathbf{Y}$. In particular, assume that the set of all possible local optima L is given, that is,

$$L = \{\mathbf{y} \in S : \exists \mathbf{x} \in S, \mathbf{y} = \mathcal{L}(\mathbf{x})\},$$

where S is the search domain. We assume that this set is finite. A directed graph can be defined whose node set is L and whose arc set is

$$E = \{(\mathbf{x},\mathbf{y}) \in L \times L : \exists \mathbf{z} \in \mathcal{N}(\mathbf{x}) : \mathcal{L}(\mathbf{z}) = \mathbf{y} \text{ and } \mathbf{y} \preccurlyeq \mathbf{x}\}. \tag{3.1}$$

In the above definition, a directed arc exists between a local optimum $\mathbf{x}$ and a better local optimum $\mathbf{y}$ if it is possible to reach $\mathbf{y}$ by means of a *LocallyGenerate* procedure like Algorithm 2, i.e., starting a local search from a point $\mathbf{z}$ in a neighborhood of $\mathbf{x}$.

It is now possible to reformulate a GO problem as a combinatorial optimization problem over the graph $\langle L, E\rangle$. Every local optimum $\mathbf{y} \in L$ with no outgoing arcs is a locally optimal solution for the combinatorial optimization problem. We denote by $L^\star$ the set of such local optima (note that if $\preccurlyeq$ is based on function values, $L^\star$ always contains the global optima). It obviously holds that $L^\star \subseteq L$, but, if $\mathcal{L}$ and $\mathcal{N}$ are suitably defined, the cardinality of $L^\star$ might be significantly lower than that of L. When $\mathcal{L}$ is a standard local search procedure, in (Locatelli, 2005) the set $L^\star$ was called the set of *local optima at level 1*, while the set L is the set of *local optima at level 0*. A local search over the graph $\langle L, E\rangle$ is called a local search at level 1, while a standard local search is called a local search at level 0. Note that the former is more expensive than the latter, since each node evaluation in the graph $\langle L, E\rangle$ requires a local search at level 0, but such additional cost is in some cases quite rewarding. In (Locatelli, 2005) local optima at higher levels are also introduced. For example, in order to define local optima at level 2, we need only substitute standard local searches with local searches at level 1 for the procedure $\mathcal{L}$ used in the definition (3.1).

Now let us restrict our attention to the case where $\mathcal{L}$ is a standard local search, and let, for any $\mathbf{x} \in L^\star$,

$$\begin{aligned}\mathcal{R}(\mathbf{x}) = \{\mathbf{y} \in L : \exists \mathbf{y}_0 = \mathbf{y}, \mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{k+1} = \mathbf{x} \in L :\\ (\mathbf{y}_i, \mathbf{y}_{i+1}) \in E \, \forall i = 0, \ldots, k\}.\end{aligned}$$

Thus $\mathcal{R}(\mathbf{x})$ is the set of local optima from which it is possible to follow a directed path in the graph leading to $\mathbf{x}$. The set $\mathcal{R}(\mathbf{x})$ can be viewed as a region of attraction for the level 1 local optimum $\mathbf{x}$. We notice that

$$\bigcup_{\mathbf{x}\in L^\star} \mathcal{R}(\mathbf{x}) = L,$$

but the sets $\mathcal{R}(\mathbf{x})$, $\mathbf{x} \in L^*$ do not induce a partition; i.e., a local optimum at level 0 might belong to the region of attraction of several different optima. The set $\mathcal{R}(\mathbf{x})$ is called a *funnel*, while a node with no outgoing arcs, i.e., a local optimum over the graph $\langle L, E\rangle$, is also defined as a *funnel bottom*; funnel bottoms are always local optima. Any GO algorithm for which the initial generation phase is capable of generating at least a point in each funnel basin is, in theory, capable of detecting the global optimum (with respect to the pre-order). If the pre-order $\preccurlyeq$ is, for feasible solutions, the natural order relation induced by the objective function f, then the globally optimal solution of a GO is always a funnel bottom. The structure of funnels is strongly influenced by the definition of $\mathcal{L}$ and $\mathcal{N}$; in particular, the cardinality of the set $L^\star$ might be anywhere between 1 and the cardinality of L: if, as an example, $\mathcal{N}$ is the identity operator, then $L = L^\star$. If, at the other extreme, $\mathcal{N}(\mathbf{x}) = L$ for every $\mathbf{x}$, then $L^\star$ just contains global optima. In the first case, exploring funnel bottoms is very easy, but their number can be prohibitively large; in the second case all funnel bottoms are global optima, so it is sufficient to explore one of them to solve the GO problem. However, descending to the funnel bottom in this case is extremely difficult as a consequence of the cardinality of the neighborhood which has to be explored. One should always search for neighborhood definitions which, on one hand, generate small neighborhoods, but, on the other hand, do not generate too many funnel bottoms. In this case one or a few local searches at level 1 are able to return the global minimum. However, while such a definition is possible for some GO problems, there is no hope of finding one which is valid for all GO problems. In (Locatelli, 2005) the difficulty of GO problems is related to the possibility of finding suitable neighborhood definitions.

When considering population-based methods, some of the above considerations are still valid, while others need some additional definitions. Also, for population methods a neighborhood structure can be easily defined, although the possibilities are many more than in the sequential case. As an example, given a set of solutions, a neighborhood for this set might be defined as the Cartesian product of neighborhoods for each solution in the set. This way, a new population which has been generated in such a neighborhood of the current one can be seen as a set of solutions (individuals), each of which has been generated by means of the perturbation of a single solution in the current set (its direct parent). However, more refined neighborhood structures might be defined in population methods, allowing, for example, for the generation of a new population in which each member depends on two or more parents in the population. As an example, when applied to conformation problems, which consist in optimally placing a number of two- or three-dimensional objects inside a suitable container (as we will see in Section 6.2.1), a new solution, defined as the set of two- or three-dimensional coordinates of each object, might be generated through a cut-and-paste technique from a pair of current individuals in the population. Some methods in this framework prescribe to cut each parent into two “halves” by means of a random plane passing through the geometric center of the structure; from each of the parents one half of the object is chosen and the union of these two halves forms the newly generated solution. In general optimization models, a new solution might be obtained by partitioning the variable set and taking each group of variable values from a specific solution in the current population. Many other population perturbation techniques can be defined, some of which might even cause the population to shrink or to grow.

Also in the case of population methods a pre-order should be defined in order to check whether the population is improving. In all known population implementations, the pre-order is based on a pre-order defined for pairs of solutions, although it might also be defined in such a way that a whole population is compared to its parent. The most common choice in population methods is that of introducing a pre-order for pairs of solutions and identifying, in the selection step, how to take a comparison between selected pairs of solutions into account in order to evolve the population.

Similar to sequential methods, many methods prescribe, after a perturbation of the population, a refinement obtained through the application of a local optimization algorithm to each solution in the perturbed population. We are not aware of local optimization methods globally applied to the whole population, although they might possibly be defined within this framework.

Closing this section, it seems worth recalling also that the idea of using local searches, in a quite general setting, within global exploration methods, is well known in combinatorial optimization. In the field of metaheuristic techniques for discrete problems, the term *memetic algorithms* is frequently used; see, e.g., (Moscato & Cotta, 2010) and, in the context of GO, (Molina, Lozano, Sànchez, & Herrera, 2011).

Select

The *selection operator* takes the current and the perturbed solution and generates a new solution which will become the current one in the next iteration. In sequential methods this selection is usually implemented according to an acceptance criterion; this criterion compares, through the usual pre-order relation, the two solutions and decides which of the two should be kept as current. In the easiest situation, when the pre-order is the natural order of objective function values (for feasible solutions), an acceptance rule might be that of substituting the generated solution in place of the current one if its function value is strictly better, or if it improves over the current one by at least a prefixed amount, or if it is the best in the whole neighborhood, or at least in the last few neighbors sampled so far. Methods inspired by *tabu-search* (see, e.g., (Cvijović & Klinowski, 2002; Hedar & Fukushima, 2006)) might also consider as acceptable a solution which does not improve with respect to the current one but, e.g., which is the best in a neighborhood which does not include the current solution. *Simulated annealing* (see Section 3.1.10) based acceptance might allow for the substitution of the generated solution in place of the current one even if, according to the pre-order, it is worse. In this case, a probabilistic criterion is used in order to accept with smaller probabilities solutions which are significantly worse than the current one.

In population methods, the *Select* operation might be much more refined, and, in some cases, it is one of the most relevant elements in the definition of a population-based method. Again, it might simply consist of a replication of the analogous operation performed for each pair of parent and child solutions (when applicable). However, in many population methods, the main criterion in accepting a newly generated solution is that of a general improvement of the whole population. In this framework, methods can be defined for which each newly generated solution is compared not only with its parents but also with all solutions in the current population, and each replaces, if possible, one of the current solutions. This way it is possible to generate methods in which a generated solution which is worse than its parent (or its parents) can survive in the next population, as it substitutes another element in the current one. Similarly, if a generated solution is better than its parent(s), a selection method might be defined in such a way that it will enter the new population, but without replacing the parent(s), which survives in the population. Both these survival strategies might be crucial in designing population algorithms in which some backtracking is allowed which prevents abandoning solutions which are worse than newly generated ones, thus designing new methods which avoid the greedy behavior of their sequential counterparts. More specifically, in (Grosso, Locatelli, & Schoen, 2007a) a detailed experimental analysis was performed in order to more deeply understand the behavior of a population-based method. There, it was experimentally observed that in a suitably defined population-based method, two phenomena occur which might strongly influence its capability of discovering the global minimum: *survival* and *backtracking*. Survival occurs when an element of the current population generates a better solution: in a standard, sequential algorithm, usually acceptance rules are defined in such a way that the parent will be substituted by the generated child. In a population environment, however, thanks to the specific selection rule used, the newly generated solution does not necessarily compete with its parent in order to find a place in the new population. Because of this the (worse) parent might survive in the next generation; this mechanism, in some cases, is very helpful, as it maintains in the population a set of good generators and thus avoids a too-greedy behavior of the population itself. The other observed phenomenon, backtracking, is related to the situation in which a generated solution might find its place in the new population even if it is worse than its generator. Again, this is a consequence of the fact that many selection rules do not impose direct comparison between a child and its parent.

The most frequently used selection rules in population methods are as follows.

***Child-parent*:** If each solution on the population has been obtained through the perturbation of a single solution in the current population, then it is compared with its direct parent. A standard, sequential selection rule is used, similar to those outlined above for nonpopulation strategies.

***Child-population*:** Each solution in the new population is compared with all members in the population, and it will enter the new population if it is preferable with respect to at least one element in the current population. Frequently, a dissimilarity criterion is introduced between pairs of solutions, and, in this case, the child solution will enter the new population if no preferable solution exists among parents which are close enough (i.e., not too dissimilar). Or, if the generated solution is radically new, i.e., its dissimilarity with respect to each individual in the current set of solutions is sufficiently high, it will be accepted in the new population. For the current population, some solutions might "*survive*" and enter the new population. In general, when checking the quality of the newly generated solution, if it satisfies some of the criteria for being included, such as that of being better than a close parent, the current solution with which it is compared will be deleted from the population. Other survival schemes, however, are possible.

***Child-population "Gauss–Seidel"*:** This selection strategy is similar to the previous one, except that when a solution in the new population is compared with the current one and accepted, it replaces one of the current solutions, or it is added to the current population. This way, when the next solution is checked for inclusion in the next current population, it will be compared with a partially modified set of solutions.

A cleverly chosen selection scheme for a population-based method might make it radically different from the sequential method on which it is based and might be the reason behind its success.

Population selection methods might even include extreme cases, such as that in which the whole new population replaces the current one, or another case in which the new population is added to the current one.

3.1.1 Elementary methods

In this section some of the most elementary methods are briefly recalled, mostly to show how their scheme fits into the general framework introduced. Let the problem to be solved be defined as

$$\min_{\mathbf{x}\in S\subseteq\mathbb{R}^n} f(\mathbf{x}).$$

Pure Random Search. In this method points are sequentially generated by means of a possibly uniform random generator in the feasible set. If such a generator is available, then this method can be obtained by simply choosing:

- *GloballyGenerate*: $\mathbf{x} = \mathcal{U}(S)$, where $\mathcal{U}$ represents a uniform pseudorandom generator.
- *StateUpdate*: Consists in possibly updating the record found so far. In particular, if the state $\mathcal{S}$ is defined as the pair $\mathcal{S} = \langle \mathbf{x}^\star, f^\star \rangle$, then updating the state corresponds to checking whether $\mathbf{x} \prec \mathbf{x}^\star$, in which case

  $$\begin{aligned} \mathbf{x}^\star &= \mathbf{x}, \\ f^\star &= f(\mathbf{x}^\star). \end{aligned}$$

  In this as well as in any other method, if the pre-order does not coincide with the natural order of function values, the state might be extended in order to contain two pairs—one consisting of the best observation so far according to the pre-order, and another containing the best *feasible* solution found so far according to the objective function value.
- *LocalStoppingRule*: This is always **true**, i.e., no local refinement is performed.

Pure random search thus is a sequential, purely global method. It is the most elementary method for global optimization, although, when the feasible set S is nontrivial, its implementation might be quite difficult. Even when the feasible set consists of a polytope, generating a sequence of uniformly distributed points on its boundary is a nontrivial task (see, for an algorithm which produces asymptotically uniformly distributed points, (Boender et al., 1991)). Its definition might be extended so that the random generator $\mathcal{U}$ produces a solution in a search space (e.g., in a box) containing S; in this case, the method might be seen as a standard random search method with an acceptance-rejection mechanism for the generation of feasible solutions.

This method is extremely inefficient, as it is easily checked from its theoretical convergence behavior: if we consider the global minimum point as having been discovered whenever an iterate has been placed in a neighborhood of the global optimum whose volume (Lebesgue measure) is equal to $\varepsilon > 0$, then the probability of hitting the global minimum in a single (feasible) iteration is

$$\frac{\varepsilon}{vol(S)},$$

and, thus, the probability of finding it in N iterations is

$$1 - \left(1 - \frac{\varepsilon}{vol(S)}\right)^N.$$

Thus, in order to be able to guarantee that the global minimum has been observed with a prescribed confidence level α, at least

$$\frac{\log(1-\alpha)}{\log(1-\varepsilon/vol(S))}$$

observations need to be taken. If we assume that the feasible space is a unit box (of unit volume) and that the neighborhood of the global minimum is a box whose edge length is ℓ, then the required number of iterations is

$$\frac{\log(1-\alpha)}{\log(1-\ell^n)} = O\left(\frac{1}{\ell^n}\right),$$

which is a clear indication of the so-called *curse of dimensionality*. Even worse, perhaps, is the fact that the method has no memory and nothing is learned from the sample except the record. Thus the method will be guaranteed to be successful only if it will place at least one observation in each ball within a set of balls of volume ε covering S.

Best Start. This is identical to Pure Random Search, except that a single local optimization is performed from the record point prior to stopping. The state update procedure just compares its current state with the last generated feasible solution and keeps the one which has the lowest function value. The local stopping rule might be set to **true** at each iteration except the last one; at the last iteration the local generation method is invoked starting from the current record, which is kept in the state.

Multistart. This is one of the basic strategies in GO. It consists simply in repeatedly sampling in S (or in some region of simple form, such as a box containing S) and in starting a local optimization from each sampled point. It is the same method as Pure Random Search, but with the addition of a local phase. According to our scheme,

- *LocalStoppingRule* is defined so that it becomes **true** after one execution of the body of the while cycle;
- *LocallyGenerate*(**x**) is a local optimization method initialized at **x**;
- *Select*, as before, has no role;

- *StateUpdate* keeps the record solution. It might also be used to set *LocalStoppingRule* to **false** when it is **true** and vice versa, so that a single local search is performed after each global generation.

This method is quite basic, and, for those problems for which a local optimization is possible, it is the baseline for efficiency and efficacy comparison. It has advantages with respect to Pure Random Search, as the probability of successfully returning the global optimum is connected with the probability of placing a single observation in the region of attraction of the global minimum. This is usually a significant advantage with respect to Pure Random Search, as the bottleneck here is no longer the predefined volume of an acceptable region around the global minimum but the relative volume of the region of attraction of the global minimum.

It is worth observing also that what is called here a local optimization method can indeed be any (global) optimization algorithm. In fact, one of the most popular uses of Multistart is in repeatedly performing a possibly quite complex optimization method with randomly generated starting points.

3.1.2 Clustering methods

After the first books on GO were published (Dixon & Szegö, 1975, 1978), great attention was given to a class of methods known as *clustering techniques*, which were introduced and analyzed in (Rinnooy Kan & Timmer, 1987a, 1987b). The idea behind such methods was to try to identify, through statistical clustering, the regions of attraction of all local optima in a GO problem in such a way that a single local search started from a point in each region would "efficiently" lead to the global minimum. Those methods were very interesting at that time, but today their applicability is quite low. In fact, given the computing power of computers in the 1980's, local optimization for problems with few variables was already a demanding task, so that one of the main goals in designing a global optimization algorithm was to try to mimic the behavior of Multistart but avoid the repeated execution of expensive local searches leading to already discovered local minima. Although not evident at that time, the original clustering methods were indeed population algorithms. Their scheme was the following:

1. Sample N uniform points in the search space and add them to previously sampled points.

2. Concentrate the sample, either by performing a few steps of a gradient-based descent algorithm or by temporarily discarding from the sample a fraction $\gamma \in (0,1)$ of all points those with the worst function value. Call the resulting set of points the "transformed sample."

3. Use a statistical technique to identify clusters in the transformed sample.

4. Choose a representative from each cluster and start a local search from it.

If considered within the framework introduced in this chapter, clustering methods can be considered as population-based methods where the single procedures are defined as follows.

- *GloballyGenerate*: This procedure uniformly samples N points from the feasible domain and adds them to the sample.

- *LocallyGenerate*: This procedure consists of two distinct phases. In the first a procedure is executed to obtain a higher concentration of the sampled points in the basins of attraction of low level minima. Then, using a statistical clustering technique, points of the transformed sample are grouped together and a local search is started from exactly one point chosen in each cluster.

  The concentration phase is usually performed in one of two possible ways. One possibility is to perform a few descent (gradient) steps starting from the last sampled points; the reduced sample is composed of the points obtained this way, together with those obtained in previous iterations of the same algorithm. Another possibility to concentrate the sample is to retain, for the following clustering phase, only the $(1-\gamma)\%$ best observations (in terms of function value), where $\gamma \in (0,1)$ is a parameter of the algorithm.

  In both cases the resulting set of concentrated points is used to identify the regions of attraction of local minima. In fact, after this concentration step, the sample is no longer uniform but (we hope) tends to be concentrated inside the regions of attraction of local minima. In the *multilevel single linkage* (MLSL) clustering methodology, starting from a point called the "seed," new points are recursively added to the same cluster in a sequential way. A sample point $\mathbf{x}_i$ is clustered with another sample point $\mathbf{x}_j$ if

  1. $\mathbf{x}_j \preccurlyeq \mathbf{x}_i$;
  2. $\mathbf{x}_j$ is within a threshold r_{kN} from $\mathbf{x}_i$.

  The rationale behind this choice is that the algorithm tries to identify monotone paths which might be reasonably associated to a descent within a single basin of attraction. The threshold r_{kN} is updated at every cycle of the algorithm and is usually defined as

  $$\pi^{-1/2}\left(\Gamma(1+n/2)\mu(S)\sigma\frac{\log kN}{kN}\right)^{1/n}, \tag{3.2}$$

  where Γ denotes the Gamma function, a well-known extension of the factorial, μ denotes the Lebesgue measure, $\sigma > 0$ is a parameter, and k is the iteration counter, so that kN is the total sample size at iteration k. This formula derives from statistical considerations, but here it is used in a quite heuristic way.

  The seeds chosen either are local optima discovered in previous iterations or, after all clusters around them have been formed, are selected from the best points still to be clustered.

  After cluster formation, a local optimization is run from each seed of each cluster, unless it is a local optimum.

- *Select*: This procedure does not do anything, as the whole sample is retained in the following iterations.

- *StateUpdate*: The N points sampled as well as all the newly discovered local optima are added to the population, which thus consists of all points sampled so far as well as all local optima discovered.

Figures 3.1–3.4 show an example of the application of MLSL to a classical test function in two variables; this function has three local as well as global minima whose location can be easily identified from the figures. In Figure 3.1 some level curves of the function are reported, from which it is possible to notice the regions of attraction of the three distinct local and global minima of the function. In the same figure we also report 100 points, drawn at random in the feasible box. In Figure 3.2 the results of the concentration step are presented: here we select the 50% best points, according to function values. As can be seen, these points are concentrated in low level sets and no longer form a uniform sample. Using the threshold (3.2) with $\sigma = 4$, a graph can be built in which nodes are the reduced sample and directed arcs connect $\mathbf{x}_i$ and $\mathbf{x}_j$ if the objective function at $\mathbf{x}_j$ is not worse than at $\mathbf{x}_i$ and the Euclidean distance between the two points is below the threshold (3.2). This graph is represented in Figure 3.3. Finally, in Figure 3.4, the leaves of the graph, i.e., those points characterized by the fact that no better point exists in a ball of radius r_{kN}, are represented. It can be seen that, in this fairly easy test case, the procedure is highly efficient, as, with only 5 runs of a local optimization method, it is capable of finding all three local (and global) optima of the function.

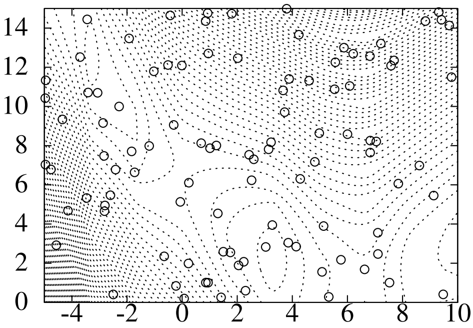


Figure 3.1. *Level curves of Branin's function with a sample of* $N = 100$ *uniform points*

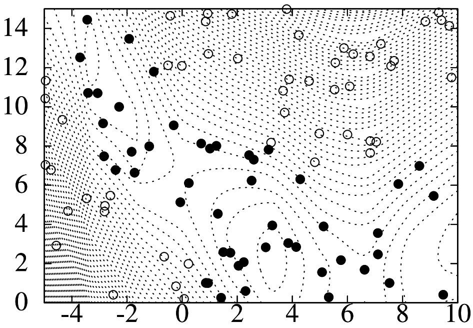


Figure 3.2. *Selection of the* 50% *best sampled points (filled circles)*

MLSL is a population-based method, but it does not make much use of the characteristics of population methods; in particular, the simultaneous sampling of the objective

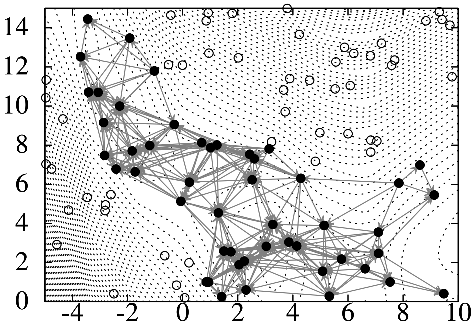


Figure 3.3. *Graph induced by the MLSL criterion (gray arrows)*

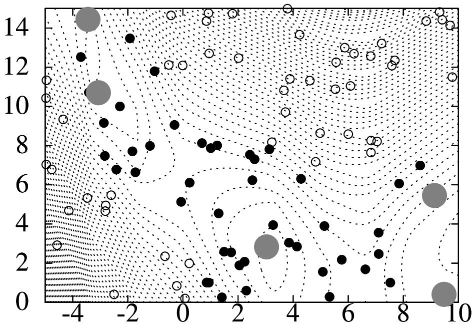


Figure 3.4. *Seeds of the MLSL graph, from which a local search will be started (large gray dots)*

function at N points is just performed in order to avoid too frequent runs of the clustering phase, which might be computationally expensive. There are, like in many population methods, backtrack moves, as the decision of not starting a local search from a point in the transformed sample can be revised in later iterations. In particular, even if a point had been clustered to a better one in the first phases of the method, thanks to the fact that the threshold is decreasing, it might become "unclustered" at later stages, and thus it might become a candidate for starting a local optimization. These backtrack moves, not too frequent in practice, are the only characteristics of a population method which are enjoyed by clustering techniques.

Simple linkage

It is possible to build a sequential version of MLSL which does not require the simultaneous sampling of several feasible points, while avoiding clustering at all. A method named *simple linkage* (SL) within this framework was proposed by (Locatelli & Schoen, 1996) and later generalized in (Locatelli & Schoen, 1999) as an alternative to MLSL. In SL no backtracking is performed. In fact, SL chooses whether or not to start a local search only from the current iterate, with no possibility of starting from a previously sampled one.

The method proposed in the cited paper prescribes that a local search be started from the current point $\mathbf{x}_k$ only if it is "far enough" from better ones:

$$\min_{j<k:\mathbf{x}_j \preccurlyeq \mathbf{x}_k} \{\|\mathbf{x}_j - \mathbf{x}_k\|\} > r_k, \tag{3.3}$$

where the threshold r_k is computed in a way quite similar to that appearing in the definition of MLSL. This approach was later extended to include a randomization in the decision of starting a local search: in (Locatelli & Schoen, 1999) the decision of starting a local search is taken with a probability which depends on the left-hand side of (3.3). It was shown that most of the properties of MLSL can be easily generalized to these methods. These properties include guarantees such as

1. convergence, with probability 1, of the best observed function value to the global optimum;

2. convergence to 0 of the probability of starting a local search from a sampled point, provided that the parameter σ in (3.2) is large enough;

3. a finite total number of local searches performed, with probability 1, even if the algorithm is allowed to run forever.

3.1.3 Basin hopping—iterated local search

This is a fundamental method which can be considered as the basis of most more elaborate GO heuristics. Indeed, it is the method which inspired our general definition of sequential as well as population-based heuristics outlined in Algorithm 1. We outline here its basic scheme in order to easily check the similarity with the general scheme of Algorithm 3.

```
while GlobalStoppingRule is false do
    $\mathbf{x} = GloballyGenerate()$;
    $\mathbf{x} = \mathcal{L}(\mathbf{x})$;
    while LocalStoppingRule is false do
        $\mathbf{z} = \mathcal{U}(\mathcal{N}(\mathbf{x}))$ ;
        $\mathbf{y} = \mathcal{L}(\mathbf{z})$;
        if IsAcceptable($\mathbf{x}, \mathbf{y}$) then
            $\mathbf{x} = \mathbf{y}$;
        end
    end
end
```

Algorithm 3: Basin hopping method

As can be seen, this method is very close to Multistart, the main difference being that while in Multistart each global generation is performed in the whole feasible set, here it is restricted to a neighborhood $\mathcal{N}(\mathbf{x})$ of the current solution. It is worth observing that the two central steps, $\mathbf{z} = \mathcal{U}(\mathcal{N}(\mathbf{x}))$ and $\mathbf{y} = \mathcal{L}(\mathbf{z})$, are equivalent to a single run of Algorithm 2. From a different point of view, basin hopping can be seen as a pure Multistart method in which the local search employed is not a traditional local, gradient-type descent, but a method which, adopting the terminology of (Locatelli, 2005), is a local search at Level 1, i.e., following a descent path in the subset of (standard) local optima (see the discussion on the level structure of GO earlier in this chapter on page 46).

Basin hopping methods were mainly developed in the field of molecular conformation problems. The first and most relevant references in the field are (Wales & Doye, 1997; Wales & Scheraga, 1999) and (Leary, 2000) (who developed the monotonic version of BH). The name BH comes from the fact that if the objective function could be graphically represented, the path followed by a BH method could be seen as jumping from one local minimum to another nearby local minimum, thus "hopping" from one basin to another. From another point of view, BH might be considered as a standard GO method which is directly applied not to the objective function f but to the composition between the objective function and a local search procedure. This way the objective function as "perceived" by BH is a piecewise constant landscape, as is seen in Figure 3.5.

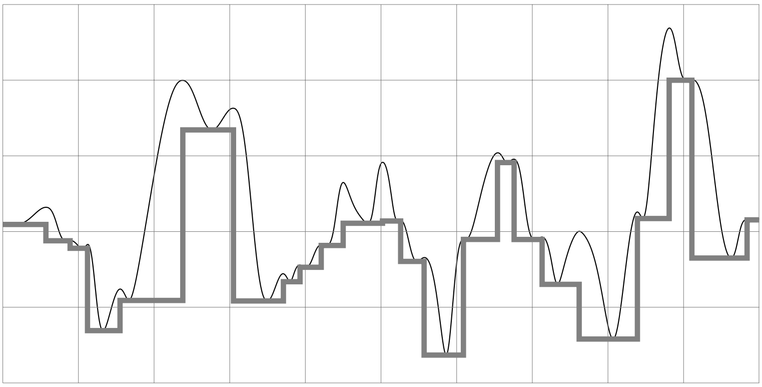


Figure 3.5. *An objective function and its transformation obtained through the execution of local searches*

We remark that, according to the terminology used in the field of combinatorial optimization, BH can also be viewed as an *iterated local search* (ILS) method (Lourenço et al., 2003).

Many variants are possible within this basic scheme, leading to many different implementations. Among the many existing possibilities, we cite the following:

- Acceptance criteria: The two most widely used variants of this step are as follows:

  1. MBH (*monotonic basin hopping*): Here, assuming all generated solutions $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ in Algorithm 3 are kept feasible, the acceptance criterion simply consists in setting *IsAcceptable*$(\mathbf{x}, \mathbf{y})$ to true if and only if

     $$f(\mathbf{y}) < f(\mathbf{x}),$$

     i.e., if $\mathbf{y}$ is a strict improvement over the current point. This way the method, within an inner Multistart iteration, performs a monotonic descent in the set of local optima. Taking into account the graph whose edge set is (3.1) and the definition of funnels, this variant of the method corresponds to following a directed path in the graph based on the perturbation operator and the neighborhood used in BH.

2. BH (generic basin hopping): The original version of BH used a criterion similar to the one commonly used in simulated annealing (see Section 3.1.10). *IsAcceptable*$(\mathbf{x},\mathbf{y})$ returns true if and only if

$$\mathcal{U}[0,1] \le \exp(-(f(\mathbf{y}) - f(\mathbf{x}))/T),$$

where T is a nonnegative parameter called "temperature" which is usually monotically decreased during the iterations. The rationale behind this variant is that, while all improving moves from $\mathbf{x}$ to $\mathbf{y}$ are always accepted, nonimproving moves can also be accepted, albeit with a probability which decreases as a function of the difference $f(\mathbf{y}) - f(\mathbf{x})$. The reason for accepting nonimproving moves is that this way it might be possible to escape from the region of attraction of a local minimum not only through the BH mechanism (perturbation plus local optimization) but also through the substitution of the current local minimum with one slightly worse. It should be clear that this variant allows for a wider state space exploration than the monotonic one, so that its capability of eventually finding the global minimum should increase. However, there is a danger of inefficiency caused by the fact that a nonimproving move leads to a random worse neighbor, with no guarantee that this will be a good direction toward the global minimum. Moreover, there is the danger of cycling among a subset of local optima, repeatedly visiting some of them. It is these authors' experience that, usually, MBH is more efficient than BH, as it can quickly reach a funnel bottom, after which a new starting point is generated.

- *LocalStoppingRule*. Frequently the inner loop of BH and MBH is terminated when a certain number of iterations with no improvements have been performed. Thus a positive constant *MaxNoImprove* is defined, and the loop becomes that outlined in Algorithm 4. Obviously, many other criteria might be used, taking into account other events, such as the total number of local searches performed, the CPU time spent, and the statistics on past successful iterations.

```
$It = 0$;
while $It \le MaxNoImprove$ do
    $It = It + 1$;
    $\mathbf{z} = \mathcal{U}(\mathcal{N}(\mathbf{x}))$ ;
    $\mathbf{y} = \mathcal{L}(\mathbf{z})$;
    if IsAcceptable($\mathbf{x},\mathbf{y}$) then
        $\mathbf{x} = \mathbf{y}$;
        $It = 0$;
    end
end
```

Algorithm 4: Inner loop in BH method

- Local search $\mathcal{L}$: Usually, the local search used in BH is a standard (continuous) local optimization method, like a gradient-based descent strategy. However, there is no real restriction in the definition of the local search method to be used, and indeed many applications of BH make use of more refined local search techniques. In some applications, notably those in computational chemistry and computational geometry, besides standard descent-based local methods, other, problem-dependent, techniques are used which, quite often, are based on some sort of combinatorial exploration of different neighbors. As an example, when looking for the "optimal" two- or three-dimensional conformation of a set of objects, after a standard local search has been performed, it is often beneficial to start an exploration through the systematic relocation of few objects from their current position to a different one. In fact, it is frequently possible to define each object's contribution to the objective function. A *direct move* is thus defined which consists in changing the coordinates of a few objects which give the "worst" contribution to the objective function to a new location which, according to the objective function, is preferable (see, e.g., (Cheng, Feng, Yang, & Yang, 2009) and Section 6.2.1). After this direct relocation, usually a standard local search is performed again. This way, the local search becomes quite a complex procedure, composed of a sequence of different local optimizations, each based on its own definition of neighborhood. It has been observed that a clever usage of these direct moves can greatly improve the performance of BH.

Population BH

Among the most successful implementations of BH methods, a prominent place has to be given to their population variants. Again, as in pure, sequential BH, the idea of population basin hopping (PBH) methods is so general that it encompasses most population-based frameworks. In its most basic version, PBH can be formulated as described in Algorithm 5.

```
while GlobalStoppingRule is false do
    $\mathbf{X} = GloballyGenerate()$;
    foreach $p \in 1, \ldots, P$ do
        $\mathbf{x}_p = \mathcal{L}(\mathbf{x}_p)$;
    end
    while LocalStoppingRule is false do
        foreach $p \in 1, \ldots, P$ do
            $\mathbf{z} = \mathcal{U}(\mathcal{N}(\mathbf{x}_p))$ ;
            $\mathbf{y}_p = \mathcal{L}(\mathbf{z})$;
        end
        $\mathbf{X} = Select(\mathbf{X}, \mathbf{Y})$;
    end
end
```

Algorithm 5: Population BH

As can be seen, PBH is a straightforward population-based version of BH. Every element in the population, exactly as in BH, is first randomly generated and then relaxed to a local optimum, and during the inner cycle, every member in the current population **X** is randomly perturbed (by means of local moves) and then locally optimized. The unique, relevant difference from the sequential version is in the *Select* operator; here the power of a population-based method comes into play. Usually this phase of the algorithm is designed in such a way as to guarantee an improvement of the whole population, as opposed to improvements of single elements. Here are some popular implementations of the selection operator:

***Embarassing parallelism*:** In this case the population has no role, as it just consists of the simultaneous execution of P independent instances of the same algorithm. The sketch in Algorithm 6 shows that the pth element in the population just evolves as a standard BH method, e.g., by accepting a new, perturbed, and optimized solution $\mathbf{y}_p$ if and only if it is acceptable with respect to $\mathbf{x}_p$. With this implementation there is no advantage to using a population method; no learning takes places during the execution of the algorithm.

```
Data: X: the current population; Y: the perturbed one
Result: X = Select(X, Y): the updated population
foreach $p \in 1,\dots,P$ do
    if IsAcceptable($\mathbf{x}_p, \mathbf{y}_p$) then
        $\mathbf{x}_p = \mathbf{y}_p$;
    end
end
```

Algorithm 6: PBH: Embarassing parallel

```
Data: X: the current population; Y: the perturbed one
Result: X = Select(X, Y): the updated population
foreach $p \in 1,\dots,P$ do
    let $c \in \arg\max_{i \in 1,\dots,P} f(\mathbf{x}_i)$ ;
    if $f(\mathbf{y}_p) < f(\mathbf{x}_c)$ then
        $\mathbf{x}_c = \mathbf{y}_p$;
    end
end
```

Algorithm 7: PBH: Greedy

***Greedy*:** This update rule prescribes that newly generated solutions possibly replace the worst elements in the population. Algorithm 7 shows an implementation of this strategy, under the simplifying assumption that acceptance is performed through the comparison of function values (even if more general acceptance rules might be easily accounted for). As is seen from the algorithmic scheme, the greedy update rule prescribes that, sequentially, the worst elements in the population are discarded and replaced by better elements taken from the perturbed one. The scheme outlined here is still not completely defined. In fact, the order by which the perturbed population $\mathbf{Y}$ is scanned is not defined. Among various possibilities, we cite random element extraction, ordered scan (either in nondecreasing or in nonincreasing order induced by the objective function value); another possibility might be that of examining the elements in $\mathbf{Y}$ following an order induced by the objective function value of the parent $\mathbf{x}_p$ of each element $\mathbf{y}_p$. Each of these strategies gives rise to a different evolution of the method; apparently no computational study has performed a numerical comparison of these strategies.

***Dissimilarity based*:** This seems to be the most interesting approach in PBH and the one which, at least in some applied fields, displays the most interesting numerical performance. The definition of this PBH approach relies on the introduction of a *dissimilarity measure* in the space of feasible solutions. Let $\mathbf{x}_i$ and $\mathbf{x}_j$ be any two solutions of a GO problem; a dissimilarity $d(\cdot,\cdot)$ is a real-valued function with the following characteristics:

$$\begin{aligned} d(\mathbf{x}_i,\mathbf{x}_j) &\geq 0 && \forall \mathbf{x}_i,\mathbf{x}_j, \\ d(\mathbf{x}_i,\mathbf{x}_i) &= 0 && \forall \mathbf{x}_i, \\ d(\mathbf{x}_i,\mathbf{x}_j) &= d(\mathbf{x}_j,\mathbf{x}_i) && \forall \mathbf{x}_i,\mathbf{x}_j. \end{aligned}$$

As an example, a norm induces a dissimilarity in a quite natural way. However, in many applied contexts it might be preferable to use a different definition of dissimilarity; one of the reasons for not using the norm might be the desire to take into account symmetries in the solutions. As an example, in the disk packing problem (see Section 6.2.3) a solution is defined by the two-dimensional coordinates of the centers of N disks. When trying to discriminate among different solutions, any permutation of the disks, although giving rise to a different array of centers, actually represents the same solution. In this case a dissimilarity based on the (Euclidean) distance between vectors representing the $2N$ coordinates of the centers might fail to reveal similarity of configurations which differ only as a consequence of a different labeling of the disk centers. In these cases it is preferable to use a different function in order to capture relevant geometric characteristics of the solution. Given a dissimilarity function d, a possible scheme for this method is reported in Algorithm 8. In this scheme it can be seen that each element $\mathbf{y}_p$ in the perturbed population is compared with the least dissimilar one in the current population $\mathbf{X}$. A possible implementation would be that of replacing the least dissimilar element $\mathbf{x}_c$ with $\mathbf{y}_p$

```
Data: $\mathbf{X}$: the current population; $\mathbf{Y}$: the perturbed one; $D_{cut}$: a nonnegative
      threshold
Result: $\mathbf{X} = Select(\mathbf{X},\mathbf{Y})$: the updated population
foreach $p \in 1,\ldots,P$ do
    let $c \in \arg\min_{i\in 1,\ldots,P} d(\mathbf{x}_i,\mathbf{y}_p)$ ;
    if $d(\mathbf{x}_c,\mathbf{y}_p) > D_{cut}$ then
        let $c \in \arg\max_{i\in 1,\ldots,P} f(\mathbf{x}_i)$ ;
    end
    if $f(\mathbf{y}_p) < f(\mathbf{x}_c)$ then
        $\mathbf{x}_c = \mathbf{y}_p$;
    end
end
```

Algorithm 8: PBH: Dissimilarity based

if and only if this replacement produces an improvement in the objective function value. Most frequently, however, the procedure is implemented as described in the reported scheme: a threshold value D_{cut} is given which is used to decide whether the similarity between $\mathbf{x}_c$ and $\mathbf{y}_p$ is sufficiently strong. If this is not the case, the substitution based upon dissimilarity is abandoned in favor of a standard greedy approach in which the solution $\mathbf{y}_p$ replaces the worst current element in the population, provided strict preference holds. A common choice for D_{cut} is the average dissimilarity in the initial population, possibly updated periodically. If D_{cut} is chosen too small (possibly even 0), the overall algorithm tends to resemble the greedy version of PBH. In some implementations, in order to avoid a too-greedy evolution of the method, if the greedy branch in the algorithm is chosen, the solution $\mathbf{y}_p$ instead of substituting the worst element in the population is added to the population itself, as sketched in Algorithm 9. In this case some attention is necessary in order to avoid the population increase with no limit. Usually some kind of housekeeping is performed by means of a periodic check of dissimilarities, followed by substituting a pair of very similar solutions with a single one.

```
Data: $\mathbf{X}$: the current population; $\mathbf{Y}$: the perturbed one; $D_{cut}$: a nonnegative
      threshold
Result: $\mathbf{X} = Select(\mathbf{X}, \mathbf{Y})$: the updated population
foreach $\mathbf{y} \in \mathbf{Y}$ do
    let $c \in \arg\min_{i \in 1,\ldots,P} d(\mathbf{x}_i, \mathbf{y})$ ;
    if $d(\mathbf{x}_c, \mathbf{y}) > D_{cut}$ then
        let $c \in \arg\max_{i \in 1,\ldots,P} f(\mathbf{x}_i)$ ;
        if $f(\mathbf{y}) < f(\mathbf{x}_c)$ then
            let $P = P + 1$;
            $\mathbf{x}_P = \mathbf{y}$;
        end
    else
        if $f(\mathbf{y}) < f(\mathbf{x}_c)$ then
            $\mathbf{x}_c = \mathbf{y}$;
        end
    end
end
```

Algorithm 9: PBH: Dissimilarity based, with growing population

The literature on PBH methods is quite large, although most papers deal with specific problem domains, such as molecular clusters or disk packing problems. A general framework for the analysis and comparison of PBH methods was introduced in (Grosso, Locatelli, & Schoen, 2007b; Grosso et al., 2007a; Cassioli, Locatelli, & Schoen, 2010). The first appearance of PBH methods in molecular cluster optimization can be traced to the *conformational space annealing* method (J. Lee et al., 2000, 2001; J. Lee, Lee, & Lee, 2003). Applications of various implementations of PBH are quite common in some specific fields, such as molecular conformation problems (Locatelli & Schoen, 2003; Doye, Leary, Locatelli, & Schoen, 2004; Pullan, 2005; Cassioli, Locatelli, & Schoen, 2009), distance geometry (Grosso, Locatelli, & Schoen, 2009), disk packing (Addis, Locatelli, & Schoen, 2008a; Grosso, Jamali, Locatelli, & Schoen, 2010), and space trajectory design (Addis, Cassioli, Locatelli, & Schoen, 2011).

3.1.4 Variable neighborhood search

Variable neighborhood search (VNS), a quite well-known heuristic, can be seen as a generalization of BH. The main difference between the two approaches is that in VNS, instead of defining a single, possibly complex, neighborhood structure $\mathcal{N}$, a whole family of neighborhoods is defined. Typically this family is arranged into a set of nested neighborhoods $\mathcal{N}_1 \subseteq \mathcal{N}_2 \subseteq \cdots \subseteq \mathcal{N}_k$. In the generic BH scheme, VNS comes into play when, after a local perturbation and local optimization, the newly generated solution $\mathbf{y}$ is compared with the original one $\mathbf{x}$ in order to check if it is acceptable. While in BH if the answer is yes, then $\mathbf{y}$ replaces $\mathbf{x}$, and otherwise a new trial is performed; in VNS, if the new solution $\mathbf{y}$ is acceptable, similar to BH, it replaces the current one. However, in this case, the current neighborhood is reset to the initial one, say, $\mathcal{N}_1$. On the other hand, if the generated solution is not acceptable, a different neighborhood is explored by switching from the current $\mathcal{N}_i$ to the next one, $\mathcal{N}_{i+1}$. There are many papers dealing with VNS and its applications and variants. We refer the reader to (Hansen, Mladenović, & Moreno Pérez, 2008) for a detailed survey on VNS definitions and applications.

3.1.5 Particle swarm optimization

Particle swarm optimization (PSO) is a global optimization framework which was first introduced in (Kennedy & Eberhart, 1995), taking inspiration from the movement of bird swarms and analyzing their social behavior. Although it is quite clear that the naturalistic inspiration has been the starting point for this algorithm family and possibly might have contributed to its popularity, here we prefer to introduce the method within the strict framework of heuristic methods for GO. Many papers and books have been devoted to PSO, to its variants, to its implementation, and to its practical use in solving real-life problems. Choosing among the vast literature, it seems worthwhile to cite at least (Banks, Vincent, & Anyakoha, 2007, 2008) and (Poli, Kennedy, & Blackwell, 2007) for interesting surveys, (Clerc & Kennedy, 2002) for one of the very few rigorous and deep analyses on the properties of the method, and (Cooren, Clerc, & Siarry, 2009) and (Vaz & Vicente, 2007) for interesting variations of the basic scheme.

PSO methods are population-based heuristics in which the evolution of the population is governed by an attempt of population individuals to make descent steps. In trying to identify descent steps, each solution vector is updated by means of a transformation which takes into account both the previous step performed and the directions which point toward good solutions found by the population. These good solutions are typically either the best solution observed during the path followed by a single solution or the best solution found by all neighbors of each solution vector; often the neighbor considered here is composed of all elements in the population, so that the best neighboring solution is the current population record.

Using the general scheme introduced in this chapter, a quite general PSO method might be represented as in Algorithm 10.

The basic idea of the method is to evolve the current population by performing steps in directions that are composed of a positive linear combination of the previous step

```
1  Initialize: Choose an integer $P > 0$ and a subset $W \subseteq \mathbb{R}^n$ ;
2  Choose positive real numbers $\omega, \phi_1, \phi_2, \beta_{\max}$ ;
3  for $i \in 1, \dots, P$ do
4      $\mathbf{x}_i = \mathcal{U}(S)$ ;
5      $\mathbf{p}_i = \mathbf{x}_i$ ;
6      $\mathbf{v}_i = \mathcal{U}(W)$ ;
7  end
8  Let $\mathbf{X} := [\mathbf{x}_1, \mathbf{x}_2 \cdots \mathbf{x}_P]$ ;
9  Let $\mathbf{V} := [\mathbf{v}_1, \mathbf{v}_2 \cdots \mathbf{v}_P]$ ;
10 Let $\mathbf{P} := [\mathbf{p}_1, \mathbf{p}_2 \cdots \mathbf{p}_P]$ ;
11 Let $\mathbf{p}^\star \in \arg\min_{i=1,\dots,P} f(\mathbf{x}_i)$ ;
12 Let $State := \{\mathbf{X}, \mathbf{V}, \mathbf{P}, \mathbf{p}^\star\}$ ;
13 while LocalStoppingRule is false do
14     for $i \in 1, \dots, P$ do
15         Let $\beta_1, \beta_2 = \mathcal{U}(0, \beta_{\max})$ ;
16         Let $\mathbf{v}_i = \omega \mathbf{v}_i + \beta_1 \phi_1 (\mathbf{p}_i - \mathbf{x}_i) + \beta_2 \phi_2 (\mathbf{p}^\star - \mathbf{x}_i)$ ;
17         Let $\mathbf{x}_i = \mathbf{x}_i + \mathbf{v}_i$ ;
18         Let $\mathbf{p}_i \in \arg\min\{f(\mathbf{p}_i), f(\mathbf{x}_i)\}$
19     end
20     Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2 \cdots \mathbf{x}_P]$;
21     Let $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2 \cdots \mathbf{v}_P]$;
22     Let $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2 \cdots \mathbf{p}_P]$;
23     Let $\mathbf{p}^\star \in \arg\min_{i=1,\dots,P} f(\mathbf{x}_i)$ ;
24     Let $State := \{\mathbf{X}, \mathbf{V}, \mathbf{P}, \mathbf{p}^\star\}$ ;
25 end
```

Algorithm 10: Generic particle swarm algorithm

performed by the same solution vector and the steps which would lead the current solution either to the current record point or to the local record, defined as the best point observed by the current individual. In line 1 a population size is chosen and a set W is defined which will be used as a sample space for initial "velocities," i.e., for initial step directions. In line 2 some constants are chosen which will influence the behavior of the method. The first three are connected with the coefficients of the linear combination of the three directions: past, best, local best.

The cycle starting at line 3 initializes each solution in the initial population to a random position in the feasible space and the initial set of directions to a random set of vectors. $\mathbf{p}_i$ is the local record, i.e., the best observation found by the ith individual; a few lines later, $\mathbf{p}^\star$ is the current global record point. Of course, although here we presumed that the term "best" is used in connection with the objective function value, other possibilities do exist, depending on the definition of the pre-order $\preccurlyeq$. The $State$ of the algorithm is defined as the current population $\mathbf{X}$, the current steps $\mathbf{V}$, the local records $\{\mathbf{p}_i\}$, and the global record $\mathbf{p}^\star$.

The main loop of the method, starting at line 13, is the local search part of the algorithm. Iteratively each individual $\mathbf{x}_i$ in the current solution $\mathbf{X}$ is updated through a linear combination of $\mathbf{v}_i, \mathbf{p}_i, \mathbf{p}^\star$. The coefficients used in this combination are usually chosen randomly (between 0 and $\beta_{\max}\phi_1$ and $\beta_{\max}\phi_2$, respectively, for the local and global directions), while the coefficient for the previously performed step is usually chosen as a parameter ω which can be possibly updated during the iterations. At the end of the loop the *State* is updated, possibly modifying the local and global records.

It is easy to see that this method perfectly fits our general population scheme. In fact the outer loop is executed just once. The GlobalStoppingRule allows for a single execution of the loop—this can be easily accomplished by letting the *StateUpdate* procedure immediately set *GlobalStoppingRule* to false. The inner **while** loop in Algorithm 10 is exactly the inner **while** loop in the general scheme presented in Algorithm 1. The *GloballyGenerate* procedure randomly initializes the population in the feasible space; the *LocallyGenerate* method corresponds to the generation of the direction for each component of the population and application of this step to each individual. The *Select* step simply accepts the new population and sets it as the current one, while the *StateUpdate* procedure maintains the current step, local and global records coherent with the current situation.

It is a truly population-based method, although the use of the population is rather limited. The unique information from the whole population that is used to change the behavior of each solution in the population is the global record. At each iteration, every solution is moved to a new position which is influenced by the global record localization, with a general tendency of all members in the population of moving toward that goal.

In some papers and implementations the step generation mechanism is slightly, but significantly, different. In our scheme two random numbers are drawn (line 15) and used as scalar weights for the directions toward the local and the global records, respectively. Other authors use different random coefficients for each component of those two directions, so that the set of directions which can be generated is much larger than in the scheme we outlined.

Quite a few parameters have to be defined in PSO which govern the behavior of the method. Usually ω is chosen as slightly less than 1 in the first iterations and then it is gradually decreased in order to promote convergence of the population. As an alterative, some authors prescribe that the step be updated as

$$\begin{aligned}
&\mathbf{v}_i = \xi\left(\mathbf{v}_i + \beta_1\phi_1(\mathbf{p}_i - \mathbf{x}_i) + \beta_2\phi_2(\mathbf{p}^\star - \mathbf{x}_i)\right), \text{ where}\\
&\xi = \frac{2}{|2-\phi-\sqrt{\phi^2-4\phi}|},\\
&\phi = \phi_1 + \phi_2,\\
&\phi > 4.
\end{aligned}$$

These values were obtained in (Clerc & Kennedy, 2002) from an accurate analysis of the behavior of a simplified PSO algorithm with just one monodimensional particle. This simplification allows the authors to analyze the dynamical system whose state is defined by the current iterate and the step. The values reported above, as well as many others suggested in the paper, arise from the analysis of the eigenvalues of the matrix of the dynamical system, following standard stability analysis for discrete and continuous systems. According to the analysis, this choice prevents the tendency of standard implementations to "explode," i.e., to generate ever-increasing steps which eventually cause the population to diverge. The damping obtained thanks to the coefficient ξ is, under the stated assumptions, sufficient to guarantee that steps are limited and convergence is attained. Although no similar analysis has been performed with higher-dimensional problems and with true populations containing more than one element, these settings have gained popularity and are quite often used in applications.

The number of papers dealing with variants of PSO and their applications is really huge; however, a definitive and comprehensive comparison of these methods with more traditional methods, such as those based on the use of local searches, is still lacking. One of the most successful implementations of PSO, PSwarm, is reported in (Vaz & Vicente, 2007). Here an attempt is made to combine the exploratory capabilities of PSO methods with the refinement obtainable by means of a local search technique. As is common in most PSO literature, it is assumed that no information on the gradient of the objective function is available, and thus the proposal is to use a derivative-free local search method. Although any such method might have been employed, the authors choose pattern search; see (Kolda, Lewis, & Torczon, 2003) for a very detailed overview of pattern search techniques. What is proposed in (Vaz & Vicente, 2007) is indeed a mixture of the two methods, in which a PSO algorithm is used to evolve a population of solutions until a stopping criterion is satisfied. After each iteration, if no improvement has been observed in the objective function value, then a pattern search is performed from the currently best solution in an attempt to refine the best estimate of the global optimum.

3.1.6 Differential evolution

Differential evolution (DE) is another very popular method which does not use, in its basic definition, gradient information or local search methods. It has some similarities with PSO and with the general framework of genetic algorithms. DE is a population-based method in which the elements in the population evolve taking into account the positions of the whole population. In this respect it differs from PSO, where the evolution of each element in the population depends on the element itself and on the position of just a few good solutions.

DE, as introduced in (Storn & Price, 1997), can be described as in Algorithm 11 (note that $\mathbf{x}_i^{(j)}$ and $Trial^{(j)}$ denote the jth coordinate of $\mathbf{x}_i$ and $Trial$, respectively). Borrowing from the language of genetic algorithms, DE can be seen as a population method in

```
Data: $F \in (0,2)$, a real constant; $CR \in (0,1)$, a probability threshold
foreach $i \in 1,\dots,P$ do
    let $\bar{\imath} \in \mathcal{U}(1,\dots,n)$;
    randomly choose $k_1,k_2,k_3 \in \{1,\dots,P\} \setminus \{i\}$, all different;
    let $Trial := \mathbf{x}_{k_1} + F(\mathbf{x}_{k_2} - \mathbf{x}_{k_3})$;
    for $j \in 1,\dots,n : j \neq \bar{\imath}$ do
        if $\mathcal{U}(0,1) > CR$ then
            let $Trial^{(j)} := \mathbf{x}_i^{(j)}$;
        end
    end
    if $f(Trial) < f(\mathbf{x}_i)$ then
        $\mathbf{x}_i = Trial$;
    end
end
```

Algorithm 11: Differential evolution

which both mutation and selection are performed, as in most population-based algorithms. Mutation is performed by substituting a few components of each solution vector, randomly chosen with a prefixed probability CR, with the corresponding components of a trial solution obtained by displacing one element in the population by a step that depends on the difference between two other solutions in the population. This way, some components of the current solution are replaced by a linear combination of the analogous components of three different solutions. After a trial solution has been generated this way, the objective function is evaluated and the trial solution replaces the current element in the population if it generates a strict improvement. This selection mechanism induces a monotonically improving behavior of the population. It can be observed that the method has a vague resemblance with respect to the classical Nelder–Mead simplex method, which performs steps by which the currently worst solution in the population is displaced, taking into account the barycenter of the population and the direction toward the best element. That method is a local one, with serious convergence problems (see (Lagarias, Reeds, Wright, & Wright, 1998)). In contrast, DE is a GO method and its evolution does not greedily depend on the position of the best element in the population. The mutation operator generates trial solutions which depend on a few randomly chosen elements and, eventually, all elements in the population will contribute to updating the population itself.

In order to easily distinguish among different DE variants, a taxonomy has become popular; this takes the form of $DE/x/y/z$, where x, which takes the values *rand* or *best*, represents the solution to be perturbed (e.g., a random one or the current record). The second field, y, stands for the number of difference vectors used in the perturbation of x, where a difference vector is the difference between two distinct randomly selected elements in the population. Finally, z identifies the recombination (crossover) operator; its value is typically *bin* when the choice is made following a binomial probability distribution function, as in the scheme reported in Algorithm 11, while it can be, e.g., *exp* when the choice on which components to keep from the current population is based on an exponential distribution. The strategy outlined in Algorithm 11 can be classified as DE/rand/1/bin. DE can be seen as a generalized PSO method, which randomizes the choice of the step both by basing its choice not only on the currently best population elements, but also by the fact that not all coordinates of the current solution are indeed changed this way. Of course these choices have both advantages and disadvantages; disregarding the best population elements makes the algorithm less greedy, which can be a good or a bad choice. Being less greedy enables a wider state space exploration, with an increased probability of placing observations in unexplored regions. On the other hand, the absence of a greedy step choice might slow down the convergence of the method toward good optima, once they have been approximately localized. Another characteristic which might prove not to be beneficial is the decision to keep unaltered a component of the current solution vector, or to change it quite radically to a different one, according to the trial vector generated. This way, some "memory" of the current vector is maintained in the new trial solution, but this memory is rather extreme: either a component is *exactly* kept unchanged, or it is radically changed into a new one. For some problems this might prove useful in exploring the feasible region, but in some other situations this behavior might produce quite random solutions with small likelihood of producing an improvement. A situation like this one arises, for example, in geometric conformation problems, where the same solution, representing the centers of N objects, corresponds to an infinite number of two- or three-dimensional coordinate vectors, differing for rigid rotation and displacement. In these cases the absolute value of each coordinate has no real meaning, while all the information is within the pairwise distances.

From the point of view of the convergence analysis, some phenomena like the collapse of the population into a single point or the freezing of the population might occur. The latter phenomenon is a consequence of the finite number of possible candidate points generated by DE: if at some iteration none of them improves the members of the current population, this will not change in all the subsequent iterations. As remarked, e.g., in (Zaharie, 2002), increasing the population size (and, thus, also the cardinality of the set of candidates) usually strongly reduces the probability of these phenomena, but they cannot be completely excluded. In (Vasile, Minisci, & Locatelli, 2011) it has been shown that if we allow for $k_1 = k_3$ in Algorithm 11, then convergence to a single point is guaranteed if f is strictly convex. However, this point is not necessarily a local (global) minimum. In spite of the limited theoretical results about the convergence properties of DE, the literature dealing with variants of DE and reports on its successful application abound. It is still not clear whether the success of DE is due to the easiness of implementation with respect to more refined methods or to some special characteristics of the method. Indeed the structure of the algorithm is interesting, as the mutation operator initially tends to produce solutions which spread around in the feasible set, but as the population evolves, some sort of automatic coordination among elements in the population occurs thanks to the special form of the mutation operator. However, it appears that the convergence of population elements toward a small region of the feasible space is an indirect way to perform a local search. Other methods, like, e.g., PBH, are based on a radically different idea—that of keeping the population as diverse as possible. It seems that some popular methods like DE and PSO mix local and global optimization within the same framework. However, it is felt that exploitation, i.e., refinement toward local optima, is much more efficiently performed by means of special purpose local optimization tools than through population evolution. A step toward this direction is made in (Vaz & Vicente, 2007) for PSO. In these authors' opinion, the best performing methods should be based on the best available local optimization methods coupled with a clever global exploration strategy. Mixing the roles of these two phases into a single, although simple, algorithm does not seem to be an efficient and effective strategy. This by no means implies that we should not trust DE methods and the like: some implementations for specific problem domains display excellent performance. However, more research seems to be needed in order to assess the theoretical properties of this family of methods, and new algorithmic schemes should be implemented which exploit the efficiency of available local methods.

3.1.7 Continuous GRASP

The *Greedy randomized adaptive search* or GRASP method is a quite well known technique first introduced in the combinatorial optimization literature. It appears with this name in (Feo & Resende, 1995). In its basic form, GRASP is a Multistart method with local searches, equipped with a specialized *GloballyGenerate* procedure. This generation procedure consists in sequentially building a solution adopting a greedy scheme which consists in adding one part at a time of the solution, guided by a greedy cost evaluation procedure and randomly drawing part of the solution from a *restricted candidate list* (RCL). Recently (Hirsch, Meneses, Pardalos, & Resende, 2007) proposed a version of GRASP for continuous GO problems; their procedure is simply a Multistart method which does not use gradients and is based on a special-purpose random generation phase. In particular, the authors propose using the procedure represented in Algorithm 12 to generate starting point solutions. After having randomly generated a uniform feasible solution **x**, an RCL is initialized as consisting of all of the n components of **x**, line searches are performed along the n coordinate-axes starting from **x**. In the scheme of the procedure, CoordinateLineSearch(i) denotes a local optimization method which returns a local optimum $\mathbf{z}_i$ found along the ith coordinate direction starting from **x**. Based on the best and worst function values obtained during these searches, a random component is chosen among those which contributed to a set of sufficiently good observed function values. This component is then fixed and deleted from the RCL. A new iteration is then started from the current point, with the updated restricted candidate list, and the whole procedure is iterated until all coordinates have been fixed. This method can be seen as a randomized version of the standard coordinate descent local optimization procedure.

```
Fixed = $\emptyset$;
$\mathbf{x} \in \mathcal{U}(S)$
while $Fixed \neq \{1,2,\dots,n\}$ do
    for $i \notin Fixed$ do
        $\mathbf{z}_i$ = CoordinateLineSearch($i$);
    end
    $\mathbf{z}^\star = \max_i f(\mathbf{z}_i)$;
    $\mathbf{z}_\star = \min_i f(\mathbf{z}_i)$;
    Threshold = $\alpha \mathbf{z}^\star + (1-\alpha)\mathbf{z}_\star$;
    RCL = $\{i \notin$ Fixed : $f(\mathbf{z}_i) \leq$ Threshold$\}$;
    Choose $i$ randomly in RCL;
    $\mathbf{x} = \mathbf{z}_i$;
    Fixed = Fixed $\bigcup \{i\}$;
end
```

Algorithm 12: Continuous GRASP construction method

After a solution has been generated by means of this construction heuristic, a local optimization is performed; in the cited paper, this local optimization is a randomized, derivative-free, coordinate descent method, but it is clear that, depending on the information available, any local optimization scheme might be used.

This method can be as well considered as a BH procedure, in which the GRASP construction phase consists in the loop of local perturbation and local optimization, which are the characteristics of BH. The neighborhood of the current solution is implicitly defined as the set of "good enough" (depending on the parameter α) local optima along the n coordinate axes associated with coordinates which are currently not fixed.

3.1.8 DIRECT

DIRECT (Jones, Perttunen, & Stuckman, 1993) is a simple and quite popular heuristic. The idea of the method is that of a pure branching method, with no bounding unless some a priori information on the objective function is available. Branching is performed by maintaining a partition of the original feasible set as a union of hyperrectangles (it is usually assumed that the original feasible set is a hyperrectangle). After each branching decision, one or more hyperrectangles are subdivided and the objective function is evaluated at the centers of each newly generated hyperrectangle, unless the function had been previously evaluated at that point. As there is no bounding, the method never prunes the BB tree (see Chapter 5), which grows at each iteration (and this is one of the disadvantages of the method, in particular at large dimension). Every leaf of the tree is a hyperrectangle whose "quality" is given by two parameters: the value of the objective function at the center point of the region, and a proxy for the volume of the region given by the diameter of the hyperrectangle. At each iteration, a few rectangles are candidates for being subdivided: the decision on which region to partition is guided by two criteria, namely having a low value at the center or a large diameter. This is a bi-criteria selection rule, and, as it is quite natural, a strategy inspired by *Pareto* optimization is adopted: an efficient frontier is built in the space of the two criteria and all regions which correspond to efficient points on the frontier are selected for subdivision.

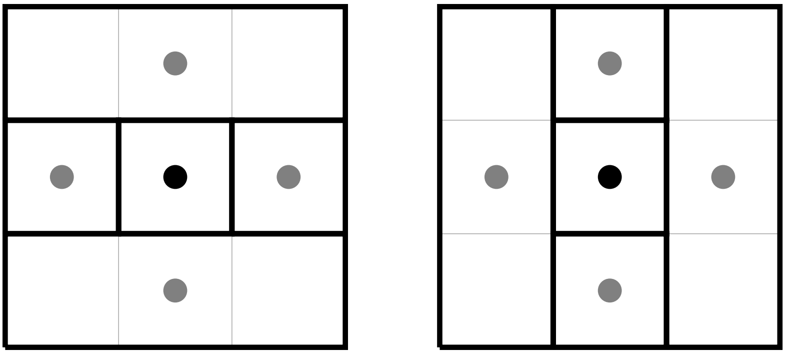


Figure 3.6. *Possible subdivisions in DIRECT*

In the original paper, this criterion was justified on the basis of Lipschitz optimization (see Section 4.9), but indeed it is just a bi-objective optimization criterion. Concerning the partition of a selected region, the idea of the method is from one side to make "simple" subdivisions which reduce the length of the longest edges. A longest edge is partitioned into three equal ones, so that, after division along such an edge, the original central point of the region is still the center of a smaller hyperrectangle. When multiple longest edges exist, e.g., in cubic regions, the order of subdivisions along the longest edges has to be determined, as different orders produce different regions, as can be seen in Figure 3.6. In the original paper, the criterion chosen by (Jones et al., 1993) was the following: starting from the center $\mathbf{x}_k$ of the current region, if ℓ is the length of the longest edges and I the index set of such longest edges, the objective function is evaluated at a regular pattern of points $\mathbf{x}_k \pm \mathbf{e}_i\ell/3$, $i \in I$, where $\mathbf{e}_i$ is the ith unit vector. According to function values at these points, the order of subdivisions was chosen; again, the choice of the original author was to somewhat give priority to good function values, and, according to this criterion, the direction of the first subdivision was the one which left the point with the best function value of the pattern within the largest region; the idea behind this choice is that the two facts of having a low function value at the center and of being in a relatively large region in some way "push" the associated region toward the frontier of the bi-objective criterion, so that the region is a more likely candidate for one of the next subdivisions. Figure 3.7

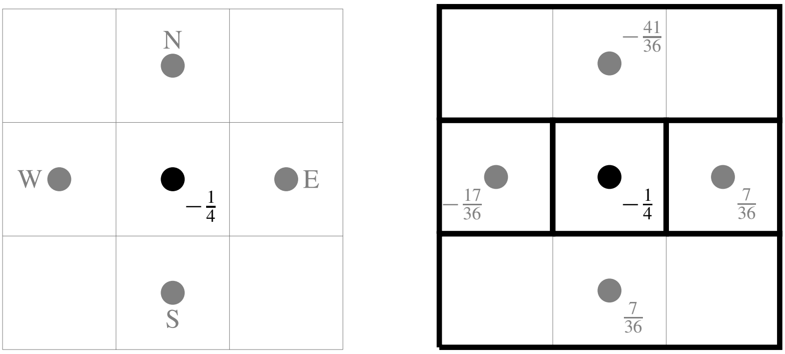


Figure 3.7. *Hyperrectangle subdivision guided by function value*

shows the idea of splitting based on function values, taking as an example the bi-variate nonconvex function $x^2 - 2y^2$ over the unit square.

In more recent versions of the algorithm (see, e.g., (Jones, 2001a)) this search for the best edge to split is relaxed and a single longest edge is chosen for subdivision without devoting too much effort in choosing the “best” one.

The overall algorithm, although it possesses many possibilities for parallelization, is a pure sequential method. The usual mix between local and global searches is performed thanks to the Pareto criterion: exploration is performed when subdividing large regions, while refinement is obtained subdividing regions associated to low function values.

Algorithm 13 presents our view of the original DIRECT method. It is assumed that the feasible set is the unit hypercube $[0,1]^n$, a condition which is easily obtained through scaling and translation if the original feasible set is a hyperrectangle. In this procedure, the

```
$k = 1$;
$\mathbf{x}_k = [1/2, 1/2, \ldots, 1/2]^T$;
$f_k = f(\mathbf{x}_k)$;
$Box_k = [0,1] \times [0,1] \times \cdots \times [0,1]$;
$d_k = Diameter(Box_k)$;
**while** *GlobalStoppingRule is false* **do**
    let $P := ParetoSet(\{(f_i, d_i), i \in \{1, \ldots, k\}\})$;
    **foreach** $p \in P$ **do**
        let $\ell_p$ = length of the longest edge of $Box_p$;
        Choose a direction $\mathbf{e}_i$ parallel to a longest edge of $Box_p$;
        Evaluate $f(\mathbf{x}_p \pm \mathbf{e}_i \ell_p / 3)$;
        Split $Box_p$ along direction $\mathbf{e}_i$ into three equal boxes;
        let $k := k + 2$;
        Update the list of boxes;
    **end**
**end**
```

Algorithm 13: DIRECT

Pareto set $P = ParetoSet(\{(f_i,d_i), i \in \{1,\ldots,k\}\})$, where the definitions of d_i and f_i are given in Algorithm 13, is defined as a set of indices of boxes such that $h \in P$ if and only if

$$\nexists \bar{h} \in \{1,\ldots,k\} : f_{\bar{h}} \leq f_h \text{ and } d_{\bar{h}} \leq d_h$$

with at least one of the two inequalities being strict.

This method, although publicized as exact, has been put in this chapter on heuristic methods as it does not provide any guarantee or any error measure unless some strong assumptions are imposed on the problem. Convergence is proven, as is common in these kinds of approaches, under very mild assumptions, and it just consists in showing that the subdivision procedure will eventually generate a dense set of observations in the feasible domain. This type of convergence property is common to all GO methods which do not use any prior information on the structure of the problem: more on this point will be presented in Section 3.3. The method is usually stopped after some resource is exhausted, like the number of function evaluations or the CPU time, or when a sufficiently good function value has been observed. Thus its behavior is exactly that of a heuristic method.

DIRECT, although very popular, thanks also to the available software implementations and to the relative simplicity of its scheme, suffers from serious defects for modern GO. In fact, although memory tends to be a cheap asset, the necessity of storing the center and edge lengths of every box makes the method unfeasible for large-scale optimization and confines it to problems with at most a few tens of variables. Moreover, no use is made of information which might be available, like gradients, or of the possibility of performing local optimization runs. The method is indeed a derivative-free technique, although it is not suitable for expensive objective functions, as it basically performs a grid search; no attempt is made to choose the least possible number of function evaluations. Recently an attempt to extend the method in such a way that it can also be used at larger dimensions has been proposed in (Liuzzi, Lucidi, & Piccialli, 2010). The authors assume that a local search algorithm is available and apply the descent procedure to each evaluation point; in practice, instead of optimizing the objective function f, the strategy is adopted of filtering the objective function through local optimization. Using the notation introduced in this chapter, given a local optimization method $\mathcal{L}$, they optimize function $f(\mathcal{L}(\mathbf{x}))$. This strategy, which is quite commonly used in modern GO heuristic methods, significantly improves the refinement phase of the method, at the cost of a considerably more expensive function evaluation phase. Another modification introduced in the proposed approach is that of restarting the method after enough function values have been observed. When the number of function evaluations performed reaches a threshold, the best local optimum is retained and an affine transformation is performed so that the best local optimum becomes the center of the initial feasible box. Restarting this way, unless the best local optimum was already located in the center of the original box, a new subdivision is performed, with new evaluation points generated; it can also be observed that the initial phases of the DIRECT method are quite exploratory, as they generate new evaluation points which are quite far from the center of the box. This way, restarting produces some sort of exploration in which regions that are far from the current best point are considered. Interesting numerical results are presented in which the authors apply their method to molecular cluster optimization according to the *Morse potential*, a very challenging GO problem which will be described in Section 6.2.1.

This paper is interesting as it contributes to showing that even quite elementary methods can be significantly improved when local optimization is added to the basic strategy: as we already pointed out, a common error in many simple heuristics is to have a single, simple, scheme both for the global exploration and for the local refinement. In our view, when available, local optimization methods should be exploited as much as possible within GO schemes, and good GO methods should concentrate more on the global exploration phase, possibly through population-based methods, rather than on the local refinement. Of course there are exceptions to this view, the most evident being that of very expensive objective functions (see Section 3.2).

3.1.9 Genetic algorithms

In this book very little space will be devoted to genetic algorithms (see, for example, (Goldberg, 1989)) and other well-known metaheuristic algorithms mainly known in the combinatorial optimization framework. There are a couple of reasons for this: first, the literature on these methods is so vast, so scattered, and with such a variable level of scientific relevance that it is quite difficult to present a detailed and rigorous analysis. Second, most of the details on metaheuristic algorithms are composed of tiny modifications of a scheme which is extremely close to our population-based algorithm. The most relevant characteristics of a genetic algorithm for continuous GO are, as in most population-based methods, the definitions of the specific methods *LocallyGenerate* and *Select*. Concerning local generation of a new population $\mathbf{Y}$ from an existing population $\mathbf{X}$, the literature on genetic methods usually identifies the following operators.

Mutation: An element $\mathbf{y}_p \in \mathbf{Y}$ in the newly generated population is obtained by randomly perturbing its direct parent $\mathbf{x}_p \in \mathbf{X}$ in the starting population. This perturbation can be seen as a standard generation of a solution in a prescribed neighborhood of another one. In this case the population plays no role in defining the new element.

Crossover: In this case a new element $\mathbf{y}_p$ is generated through a mechanism which depends on two (or, seldomly, more than two) elements in the original population. Many different combination rules might be adopted, some of which are strongly linked to the specific problem at hand. A simple possibility is, given two "parents" $\mathbf{x}_i$ and $\mathbf{x}_j$, to use a randomization method in order to choose, for each component of $\mathbf{y}_p$, whether to copy that component from $\mathbf{x}_i$ or from $\mathbf{x}_j$. This way the "offspring" $\mathbf{y}_p$ inherits its components from two different solutions; it is clear that this, as well as other, crossover operations might produce infeasible solutions, and, thus, a feasibility restoration phase usually follows the generation. Quite often, also, in continuous GO, after a new element is produced, it is relaxed to a nearby local minimum through the application of a standard local optimization method. If the elements in the population represent the two- or three-dimensional coordinates of N objects, a frequently used crossover operator (see, e.g., (Roberts, Johnston, & Wilson, 2000; Hartke, 2006)) consists in generating a random line in $\mathbb{R}^2$ or a plane in $\mathbb{R}^3$ through the geometric center of the set of objects and using this line or plane to split it into two halves. The offspring $\mathbf{y}_p$ is obtained by taking one half of parent $\mathbf{x}_i$ and one half of $\mathbf{x}_j$, with some care devoted to guaranteeing that the resulting solution still corresponds to N objects; after this "cut-and-paste" operation, local optimization is employed to generate a reasonable configuration.

Here, as in other parts in this chapter, it is useful to recall that the definitions of local and global perturbation do not permit a clear-cut distinction: crossover is a typical perturbation which can generate a new solution which is very far from the two original solutions. So, in a certain sense, it is a global move. We call it "local," as it is the result of a perturbation of the whole current population, in contrast to "global" moves, which generate new populations from scratch.

Selection: This part enables us to decide how to combine the original population $\mathbf{X}$ and the newly generated population, $\mathbf{Y}$, in order to produce the set of solutions constituting the next generation. It is often presented as a fundamental characteristic of genetic methods, but it simply consists in the acceptance criterion used in general population-based methods in the *Select* procedure.

In the vast literature on genetic algorithms, it is our opinion that crossover operators are the most specific characteristics. For some problems, e.g., those already mentioned where elements in the population correspond to geometric objects, highly effective crossover operations can be defined. On the other hand, in some other cases crossover operations, at least effective ones, might be difficult (or even impossible) to define.

As frequently happens, the use of the term "genetic" and the attempt to refer the algorithm to some kind of naturalistic behavior have contributed to the widespread acceptance of these methods but, in our opinion, have quite often been overemphasized, leading to their sometimes acritical acceptance. What we are proposing here is to look at genetic methods as standard population-based GO algorithms, with a specific generation mechanism; proceeding this way, without sacrificing the peculiarity of the methods, allows us to build quite efficient approaches which combine the interesting generation mechanism of crossover operators with the efficient exploration tools of local optimization.

3.1.10 Simulated annealing

From a continuous optimization point of view, *simulated annealing* is nothing more than an acceptance rule to be applied within a general GO method based on local perturbations. The algorithm which goes under the name of simulated annealing just consists in sampling the objective function in a neighborhood of the current solution $\mathbf{x}_k$; sampling is not necessarily performed by means of the uniform distribution in a neighborhood, but any probabilistic sampling rule can in principle be used. Once a new solution $\mathbf{y}$ has been generated this way, it is accepted, i.e., $\mathbf{y}$ replaces the current solution $\mathbf{x}_k$, with probability

$$\min\{1, \exp(-\beta_k(f(\mathbf{y}) - f(\mathbf{x}_k)))\}. \tag{3.4}$$

Thus, as can be seen, according to the rule (3.4), the generated solution $\mathbf{y}$ surely replaces the current one if it improves over $\mathbf{x}_k$, i.e., if $f(\mathbf{y}) \leq f(\mathbf{x}_k)$. However, uphill moves are also possible, thanks to the fact that, even if $f(\mathbf{y}) > f(\mathbf{x}_k)$, the new point might be accepted, although with a probability which decreases to zero as the gap between the new and the current solutions' values grows. The probability of accepting a nonimproving move is also governed by a parameter sequence $\beta_k > 0$, which can be chosen in many different ways. The effect of β_k is quite easily seen looking at (3.4): if it is close to zero, then almost any perturbation is accepted, and the algorithm just performs a sort of pure random search, the only difference being that of generating any new point in a neighborhood of the current one. As the parameter is increased, which is typically done in implementations of this method through a so-called *cooling schedule*, the resulting algorithm gradually switches from a global exploration phase to a local one, and eventually it becomes a local optimization method based on random sampling in the neighborhood of the current solution. The method has been widely studied, both in its combinatorial version as well as in the continuous one; for the latter see, e.g., (Locatelli, 2002). Many papers analyze the convergence behavior of the method, which can be studied within the framework of *Markov stochastic processes*; all the theoretical analyses agree in requiring that, in order to be able to prove convergence, it is necessary that the control parameter β_k is increased "slowly enough." This is expected, since if the parameter is increased too quickly, there is a high risk of the method being trapped in the region of attraction of a local minimum which is not the global one. All theoretical studies of simulated annealing obtain the fundamental result that the stationary, or invariant, measure $\mu_\beta(\cdot)$ of the Markov process associated to the evolution of the method when the β parameter is kept fixed for long enough is such that

$$\mu_\beta(\mathbf{x}) \propto \exp(-\beta f(\mathbf{x})). \tag{3.5}$$

The right-hand side of (3.5) is the *Gibbs* or *Boltzmann distribution* with energy function $f(\mathbf{x})$ and *temperature* β^{-1}. As in other parts of the book, we avoid putting too much emphasis on the "naturalistic" inspiration of GO methods; here we just recall that the term *temperature*, as well as the name of the method, come from an analogy from metallurgy. From (3.5) it can be seen that, as $\beta \to \infty$, this measure converges to one which is null everywhere except at the global optima. From this it can be understood that the method, proceeding through a slow enough cooling schedule $\beta_k \to \infty$, is based on the idea of simulating the Gibbs distribution by letting the algorithm perform iterations with a somewhat fixed parameter β until the distribution of the iterates approaches the Gibbs one. By slowly varying β, the Gibbs measure becomes more and more concentrated around good local minima and, eventually, the global minima.

From the point of view adopted in this volume, simulated annealing can be seen as a sequential GO heuristic with a probabilistic acceptance rule. We do not wish to diminish the importance of the method and the extremely refined and profound theoretical analyses which have been done on its behavior; however, it is felt that in modern GO methods, the importance of simulated annealing is less relevant than it used to be. Its acceptance rule is sometimes adopted as a tool for generating a nonmonotonic sequence of iterates. However, the most successful applications are those in which the criterion of simulated annealing is used within, e.g., a basin hopping scheme, as described in Section 3.1.3. Even in that context, however, it is questionable, from the practical point of view, whether or not it is more convenient to perform a monotonic basin hopping method (MBH) which is restarted from a random initial solution whenever it reaches a funnel bottom. According to the analysis in (Schön, 1997) (see also (Locatelli, 2002)), simulated annealing can be successfully applied when the barriers separating the local minima are low with respect to the "depth" of the valleys where such local minima lie. However, in general, it seems that the time necessary to move uphill in order to escape from the region of attraction of a local minimum is too high, at least for hard GO problems, and might be more efficiently used in restarting the search from scratch. Of course this is just a feeling of the authors which, although confirmed by vast numerical experience, cannot be accepted as a truth and might be easily disconfirmed in specific cases.

3.1.11 Methods based on tunneling and filled functions

The *tunneling method*, first described in (Levy & Montalvo, 1985), is based on coupling a local search algorithm with a clever way to generate a good starting point. The idea is that of starting, from a randomly chosen point, a descent algorithm which leads to a local minimum $\mathbf{x}^\star$. Then a procedure is started whose aim is to find a point $\mathbf{x}_0 \neq \mathbf{x}^\star$ such that

$$f(\mathbf{x}_0) \leq f(\mathbf{x}^\star). \tag{3.6}$$

If such a point is found, a new local search started from it will eventually lead to a different local minimum, and, if strict inequality holds, this minimum will be an improvement over the current one. In Figure 3.8 a simple, monodimensional, illustration of the tunneling phase is presented. There, given a randomly generated starting local optimum $x_0^\star$, the tunneling phase leads to x_0. A local search started from this point will eventually lead to $x_1^\star$, and, in this simple example, the next tunneling phase, followed by a local minimization, will get to the global minimum point.

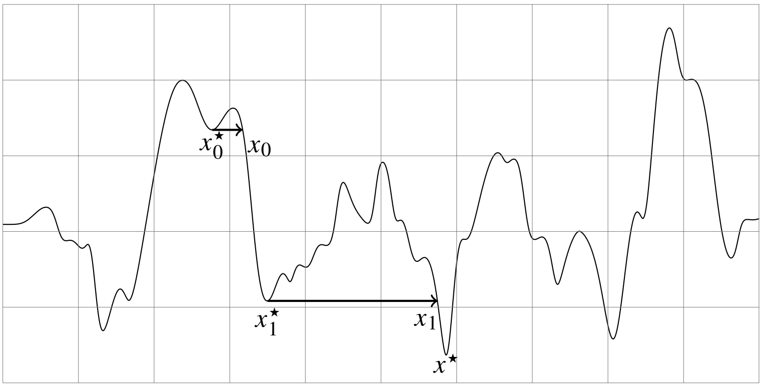


Figure 3.8. *A simple illustration of tunneling: arrows denote two successive tunneling phases*

Unfortunately, although very appealing for simple one-dimensional functions, the method incurs some serious drawbacks when applied to higher-dimensional GO problems. In fact, finding a point satisfying (3.6) is in general not a trivial task and is quite often as complex as GO itself. Moreover, once the global minimum has been found, proving that the strict inequality cannot hold is, again, a very difficult task. In the original paper, the problem of finding a solution to (3.6) is transformed into the problem of finding a point in which an auxiliary function, called the *tunneling function*, is negative. Given already observed local optima $\{\mathbf{x}_1^\star, \mathbf{x}_2^\star, \ldots, \mathbf{x}_k^\star\}$ and the best function value observed so far, $f_k^\star = \min_{i=1,\ldots,k} f(\mathbf{x}_i^\star)$, the tunneling function is defined as

$$T(\mathbf{x}; \mathbf{x}_1^\star, \mathbf{x}_2^\star, \ldots, \mathbf{x}_k^\star) = \frac{f(\mathbf{x}) - f_k^\star}{\prod_i \|\mathbf{x} - \mathbf{x}_i^\star\|^{\gamma_i}}. \tag{3.7}$$

It is readily seen that, if constants γ_i are chosen as sufficiently large positive numbers, a pole is generated at each observed local optimum. Thus the tunneling function has the property of being nonnegative at all feasible points at which the objective function is not lower than the current record $f_k^\star$ and, when approaching the already observed local minima, it diverges to $+\infty$. Thus a GO method applied to this function should be able either to find an improving point or to show that the global minimum has been reached. Note, however, that this tunneling phase might be quite time consuming.

The method has been quite popular and has stimulated some further research inspired by the idea of tunneling; see, e.g., (Lucidi & Piccioni, 1989).

Filled functions

Based on an idea which, in a certain sense, is similar to that of the tunneling method, algorithms based on filled functions have been introduced in (Renpu, 1990) and subsequently refined and extended by many authors (see, e.g., (Wu, Lee, Zhang, & Yang, 2006; Wu, Bai, Lee, & Yang, 2007)). The idea is again that of trying to escape from a local minimum and either find another local minimum with lower function value or prove that no other such minimum exists, as the current one is the global optimum.

The concept of a filled function is that of an auxiliary function which helps in escaping the current local minimum, possibly leading to a point with lower function value, from which a new local search can be started.

Many definitions of a filled function have been given; here we refer to that in (Wu et al., 2006):

Definition 3.1. *A continuously differentiable function* $p(\mathbf{x})$ *is a* filled function *corresponding to a local minimum* $\mathbf{x}^\star$ *of an objective function* $f(\mathbf{x})$ *if the following hold:*

1. $\mathbf{x}^\star$ *is a strict local* maximum *for* $p(\mathbf{x})$*;*
2. *if* $\mathbf{x}$ *is such that* $f(\mathbf{x}) \geq f(\mathbf{x}^\star)$*, then* $\nabla p(\mathbf{x}) \neq 0$*;*
3. *if a local minimum* $\bar{\mathbf{x}}$ *for* f *exists such that* $f(\bar{\mathbf{x}}) < f(\mathbf{x}^\star)$*, then* $\bar{\mathbf{x}}$ *is also a local minimizer for* $p(\mathbf{x})$.

The definition in the cited paper has some extra technicalities, but the main arguments are those presented above, i.e., the fact that the filled function turns the current best local optimum into a maximum and that there cannot be any stationary point for the filled function at points at which the original function has higher values than the current record. Moreover, strictly better local optima for f are also local minima for p. Thus any descent method, initialized close to the current optimum, applied to the filled function, should eventually lead to a point where the original function is lower and, possibly, at a new local optimum. The descent for the filled function cannot start at the current point, as it is stationary for p, so that most local methods fail to start. As in tunneling methods, once the global minimum has been reached, it might not be easy to detect that no better points exist. In Figure 3.9 we report an example of a filled function relative to the local minimum located at $\mathbf{x}_1^\star$; the specific form of the filled function, which is quite involved, is that found in (Wu et al., 2006). Basically it consists of a function of the form

$$p(\mathbf{x}) = q\left(\exp\left(-\|\mathbf{x}-\mathbf{x}^\star\|^2/q\right) g_r(f(\mathbf{x}) - f(\mathbf{x}^\star)) + f_r(f(\mathbf{x}) - f(\mathbf{x}^\star))\right),$$

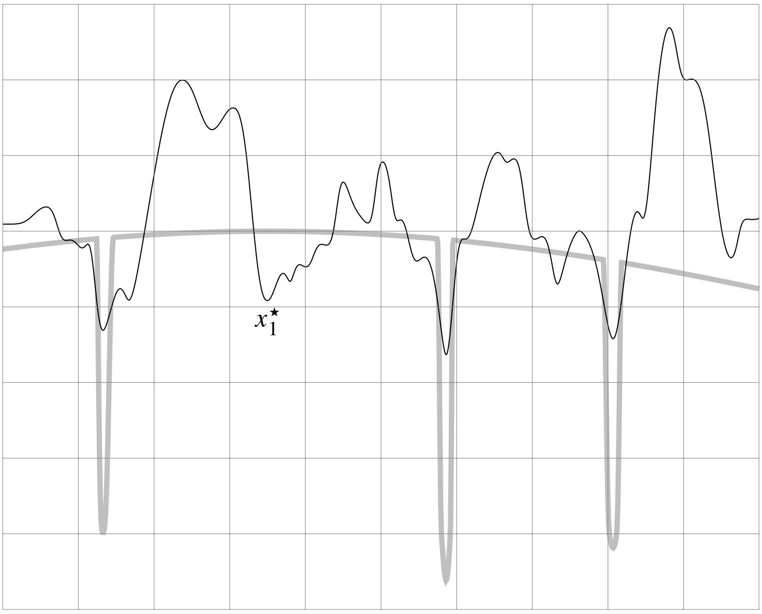


Figure 3.9. *An example of a filled function based on the local optimum* $\mathbf{x}_1^\star$

where $q > 0$ is a parameter and f_r, g_r are two scalar functions depending on a real positive parameter r, designed in such a way as to satisfy the definition of a filled function. In particular, both g_r and f_r are equal to 1 for all points $\mathbf{x}$ such that $f(\mathbf{x}) \geq f(\mathbf{x}^\star)$. g_r is 0 if $f(\mathbf{x}) \leq f(\mathbf{x}^\star) - r$. For the same $\mathbf{x}$ values f_r is equal to $f(\mathbf{x}) - f(\mathbf{x}^\star) + r$. For other values of their argument, both f_r and g_r are defined as smooth cubic interpolation functions.

The theory behind the filled function is surely more sound than that of tunneling and attracted quite a large body of research. The practical usability of these methods for large-scale or difficult problems is not particularly extensive; some of the difficulties are related to the need to suitably choose the parameters of the filled function, to optimize it and escape from the initial point, and to detect that the global optimum has been observed. Nonetheless, the method deserves interest.

3.1.12 Pure adaptive search and hit-and-run

Pure adaptive search (PAS) (see (Wood & Zabinsky, 2002)) is an abstract method which differs from pure random search (see Section 3.1.1) in the criterion to generate and accept new points. While PRS generates points uniformly in the search domain, PAS, at each iteration, generates new points according to a distribution which is the restriction of the uniform one to the level set of the current point. In other words, after a point $\mathbf{x}_k$ has been generated by the algorithm, the new point $\mathbf{x}_{k+1}$ will be uniformly generated in the set

$$\{\mathbf{x} \in S : f(\mathbf{x}) \leq f(\mathbf{x}_k)\}.$$

This method enjoys nice properties such as, as shown in (Zabinsky & Smith, 1992), the fact that the number of iterations needed to see the global minimum increases only linearly as the dimension of the problem increases, under quite mild assumptions.

Unfortunately, implementation of PAS is usually impossible, and simulations obtained, e.g., by means of acceptance-rejection criteria, incur the usual "curse of dimensionality."

Hit-and-run (Zabinsky & Wood, 2002) is, on the contrary, relatively easy to implement. Starting from an initial point $\mathbf{x}_0$, usually randomly generated in the feasible set, a random direction $\mathbf{d} \in \mathbb{R}^n$ is chosen according to a uniform distribution on the unit n-dimensional sphere. If the feasible set is assumed to be convex and compact, points along the line which originates from $\mathbf{x}_0$ in direction $\mathbf{d}$,

$$\mathbf{x}_0 + \lambda \mathbf{d},$$

are feasible for all $\lambda \in [0, \bar{\lambda}]$, where $\bar{\lambda}$ is a positive scalar (unless $\mathbf{x}_0$ is at the border of the feasible set). The hit-and-run method prescribes choosing a random uniform value for λ between 0 and $\bar{\lambda}$ and assigning the next point $\mathbf{x}_1$ to $\mathbf{x}_0 + \lambda \mathbf{d}$.

The two methods, PAS and hit-and-run, have also been combined in the improving hit-and-run algorithm, which differs from hit-ad-run simply by the fact that the point generated along the line with direction $\mathbf{d}$ is accepted only if it is an improvement over the current solution. A further variant of the method prescribes accepting even a nonimproving point according to some probabilistic criterion, similar to what is done in simulated annealing.

The advantage of these methods is that their implementation can be relatively easy, at least when the feasible region has a "simple" shape, like a polytope or a hypersphere. In general, no use is made of local optimization (although it can be very easy to extend these methods in this direction), and their performance in large-scale, highly multimodal examples is somewhat limited.

3.1.13 Methods based on smoothing and homotopies

The idea of substituting the objective function with a different, easier to optimize one is quite natural. In Section 4.2 convex envelopes are introduced, and their properties are discussed. The convex envelope of a function is the best possible convex underestimator of that function over some region. If finding it or, at least, a convex underestimator of the objective function is not too difficult, then methods can be built which optimize such an approximation in order to approach the global minimum. This is a possible way to smooth out local optima. Of course, finding a convex underrestimator and, even more, the convex envelope of a function is not an easy task in general, so in the literature some attempts have been made to define different smooth approximations to the objective function.

The principle of *smoothing methods* is that local optima can be seen as "high-frequency" perturbations of a signal, the objective function, which can be suitably filtered by means of a convolution operation. In (Moré & Wu, 1996, 1997a, 1997b) the smoothing approach is introduced and applied with interesting results to problems in molecular conformation. Many authors had already introduced the idea of smoothing the objective function by means of a suitable convolution. Some of them worked on stochastic approximation techniques in the 1980s, among which here we just cite, as an example, (Rubinstein, 1983). In all those papers it is observed that, given an objective function f, it is possible, at least in theory, to define its transform (here we limit the analysis to the Gaussian transform, but other possibilities exist choosing a different density function) as

$$\langle f \rangle_\lambda(\mathbf{x}) = \frac{1}{\lambda^n \pi^{n/2}} \int_{\mathbb{R}^n} f(\mathbf{y}) \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{\lambda^2} \right) d\mathbf{y}, \tag{3.8}$$

or, equivalently, as

$$\langle f\rangle_\lambda(\mathbf{x}) = \frac{1}{\pi^{n/2}} \int_{\mathbb{R}^n} f(\mathbf{x}+\lambda\mathbf{z}) \exp\left(-\|\mathbf{z}\|^2\right) d\mathbf{z}. \tag{3.9}$$

An example of smoothing a one-dimensional function is reported in Figures 3.10–3.12.

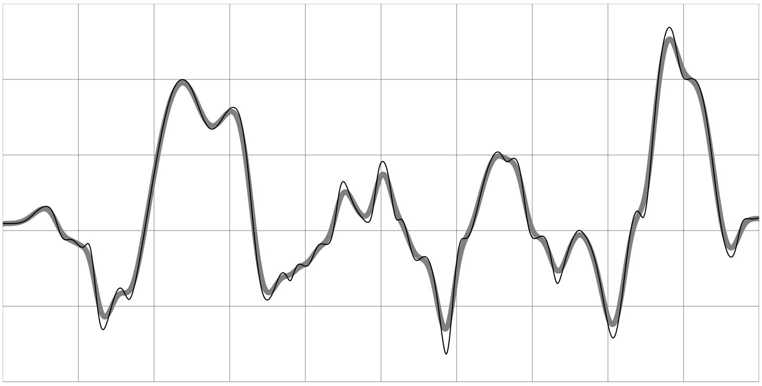


Figure 3.10. *An example of smoothing at* $\lambda = 0.01$

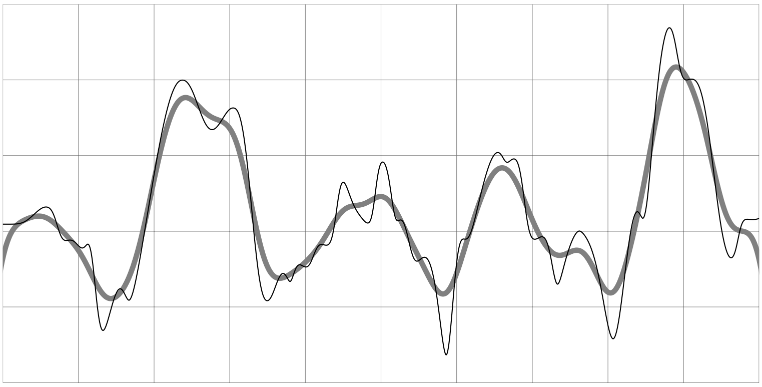


Figure 3.11. *Smoothing at* $\lambda = 0.025$

As often happens, one-dimensional examples on one hand are useful but on the other hand might be misleading in their simplicity. In fact, from the figures it is seen that smoothing has the effect of eliminating local minima and giving the "overall picture" of the graph of the function, something like looking at the function from "very far," so that fine details and oscillations are smoothed out. While this continues to be true in higher dimensions, some difficulties arise. First it is not guaranteed that the global minimum of smoothed functions converges to a global minimum of the original one as $\lambda \to 0$. Tracking all local minima of the smoothed function as λ decreases can be very hard in high dimensions. Moreover, the multidimensional integral which has to be computed is usually impossible to

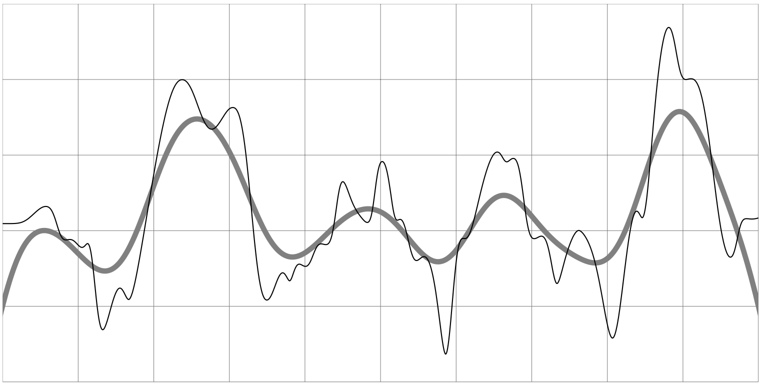


Figure 3.12. *Smoothing at* $\lambda = 0.05$

obtain in analytic form, so that even the evaluation of the smoothed function can be computationally complex and numerically difficult. It is true that, for some important classes of functions, in (Moré & Wu, 1997a) it is shown that the multidimensional integral can be reduced to a one-dimensional one when the function depends on $\mathbf{x}$ only through $\|\mathbf{x}\|$.

In (Addis, Locatelli, & Schoen, 2005) basin hopping and smoothing methods are jointly considered in order to build a novel algorithm, ALSO, based on the concept of *local smoothing*. The idea is that a first important smoothing is already obtained when using local optimization, as is commonly done in BH methods. We already remarked that the objective function can be "filtered" through local searches, reducing to a piecewise constant function. In the cited paper, the authors proposed using a smoothing approach in order to obtain good descent directions in this piecewise constant function. In order to obtain this effect, a local smoothing operator is defined in order to obtain a smooth approximation of the function

$$f(\mathcal{L}(\mathbf{x})),$$

where $\mathcal{L}$ is, as in Section 3.1.3, a local optimization procedure. In Figure 3.13 an objective function is reported as well as the function obtained if we could apply a local optimization to each feasible point. It can be easily seen from the figure that, if it were possible to perform a "descent" step in the transformed function, great advantages might be gained. Indeed, the basins of the function reported in this figure change from the original 23 to only 5 in the objective transformed by means of local search. Unfortunately, this last statement is imprecise: in fact in the piecewise constant function resulting from the application of local searches, there is an uncountable number of local optima. The idea, first introduced in (Addis et al., 2005) and later refined in (Addis & Leyffer, 2006), was that if smoothing could be applied to the transformed function, descent directions could be found. Clearly, smoothing is highly complex for a generic function, so applying it to the result of local optimization is even more difficult. In the cited papers a proposal was made to use only a local approximation to the smoothing function, performed in a sort of trust region around the current iteration. Numerical results were quite interesting even for moderately large test problems. As an illustration of the idea of this algorithm, in Figures 3.14–3.16 the result which could be obtained if a smoothing transform were applied to the piecewise constant function $f(\mathcal{L}(\mathbf{x}))$ is shown.

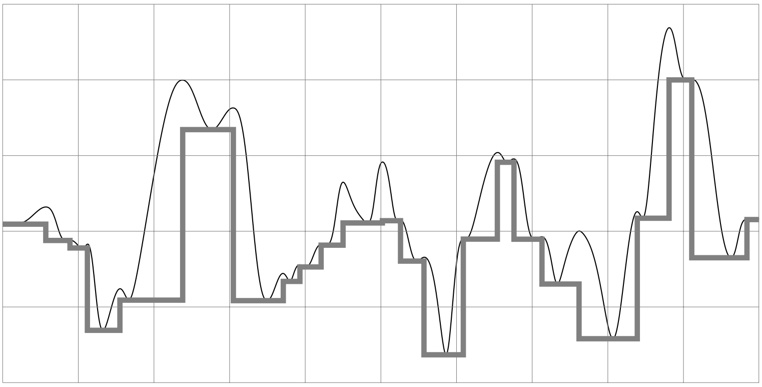


Figure 3.13. *An objective function $f(x)$ and its transformation through a local search operator (in gray)*

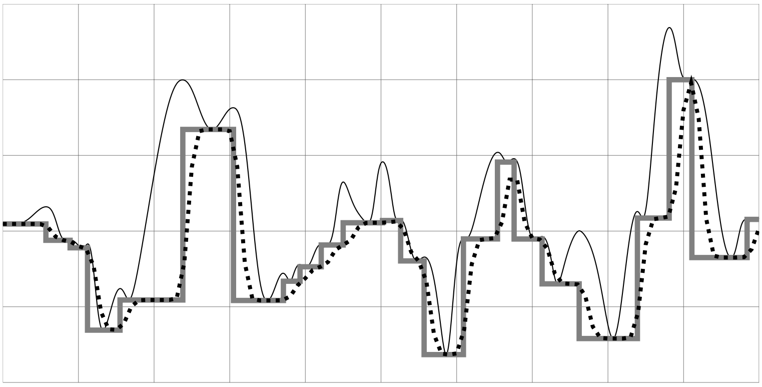


Figure 3.14. *An example of smoothing (dotted line) $f(\mathcal{L}(x))$ with $\lambda = 0.01$*

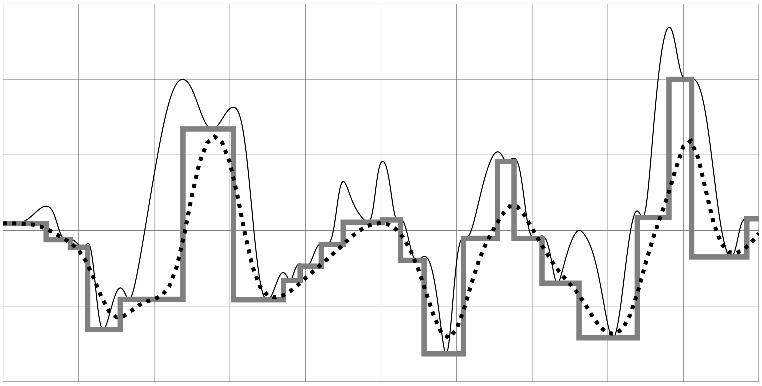


Figure 3.15. *Smoothing $f(\mathcal{L}(x))$ with $\lambda = 0.025$*

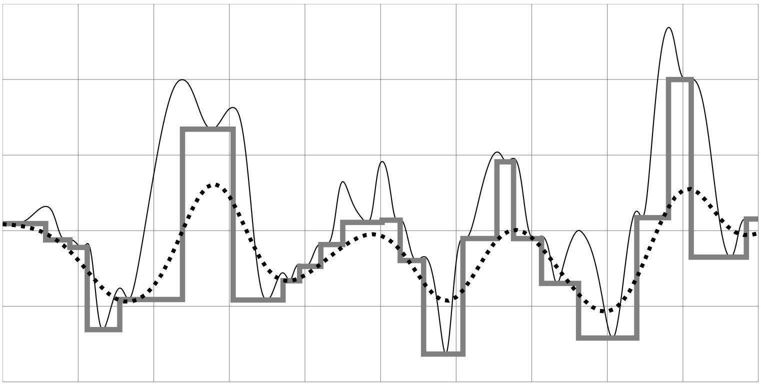


Figure 3.16. *Smoothing* $f(\mathcal{L}(x))$ *with* $\lambda = 0.05$

Somewhat related to smoothing are methods based on the idea of *homotopy* or of *continuation*. In these approaches, similar to that suggested in smoothing algorithms, instead of solving the original problem, a family of problems is defined, depending on a scalar parameter, in the hope that the solution of a perturbed problem will eventually lead to the solution of the original one. Homotopy methods (see, e.g., (Watson, 1986, 2001; Allgower & Georg, 2003)) can be more easily seen in the context of the solution of nonlinear systems of equations. Let $F(\mathbf{x}) = 0$ be a system to be solved, and let $G(\mathbf{x}) = 0$ be another, significantly easier, nonlinear system of equations. A *homotopy* is a continuous map $\rho(\lambda;\mathbf{x})$ from the real interval $[0,1]$ into a functional space; it is assumed that

$$\begin{aligned} \rho(0;\mathbf{x}) &= G(\mathbf{x}) && \forall \mathbf{x}, \\ \rho(1;\mathbf{x}) &= F(\mathbf{x}) && \forall \mathbf{x}. \end{aligned}$$

An easy example of an homotopy is the convex combination of F and G:

$$\rho(\lambda;\mathbf{x}) = \lambda F(\mathbf{x}) + (1-\lambda)G(\mathbf{x}), \qquad \lambda \in [0,1].$$

The idea of these methods is to follow the path of a solution $\mathbf{x}^\star(\lambda)$ to the equation

$$\rho(\lambda;\mathbf{x}^\star(\lambda)) = 0,$$

starting at $\lambda = 0$ and gradually increasing it to 1. Suitable assumptions have to be made in order to be able to guarantee that any path of solutions to $\rho(\lambda;\mathbf{x}) = 0$ continuously connects the zeros of G with those of F. Moreover, it is usually assumed that the Jacobian matrix of ρ with respect to $\mathbf{x}$ is nonsingular along the set of solutions to $\rho(\lambda;\mathbf{x}) = 0$. Failure to follow a path of zeros may occur because the path might have turning points or might bifurcate into two or more paths; moreover, it may happen that for some value of the parameter λ no solution exists, or that the path diverges when $\lambda \to 1$. Applications of homotopy methods to optimization can be readily imagined when F and G are seen as the gradients of $f(\mathbf{x})$ and of a suitable approximation $g(\mathbf{x})$, possibly a convex function. Considered as a tool to solve nonlinear equations, a homotopy method seems to be a local optimization approach. However, depending on the choice of $g(\mathbf{x})$, it might behave as a GO method which avoids being trapped in local minima thanks to the fact that starting from a convex function g and gradually deforming it toward f, high-frequency oscillations are somewhat filtered away.

These methods, similar to those based on filtering, have been applied quite often, mainly in the field of distance geometry and molecular conformation problems. They share some of the defects of filtering methods, as it is already very difficult to guarantee that the path of stationary points of the deformed function converges to a stationary point of f; even more difficult is to have it to converge to a minimum of f and, even worse, to a global optimum.

3.1.14 Other methods inspired by combinatorial optimization

It can be quite safely stated that any general-purpose heuristic developed in the combinatorial optimization framework can be adapted to deal with continuous GO problems. For instance, *scatter search* (see, e.g., (Resende, Ribeiro, Glover, & Marti, 2010)) is a population-based metaheuristic based on a set of basic elements:

1. A diversification generation method, which creates a new population from scratch or perturbs an existing one.

2. An improvement method, which refines the elements of the current population in order to improve their quality (function value, feasibility, etc.).

3. A reference set update method, which maintains a subset of solutions which are currently identified as the "best" because their function value is very good, or they are feasible (or close to feasibility), or they are sufficiently dissimilar with respect to other elements in the same set.

4. A subset generation and combination method, which, operating on groups, e.g., pairs, of solutions belonging to the reference set, generates solutions for the next iteration.

As is readily seen, many of the elements of this scheme have already been seen in other heuristics, and the whole procedure fits the general scheme we presented here. Scatter search has already been used in GO ((Herrera, Lozano, & Molina, 2006; Egea, Martí, & Banga, 2010; Laguna, Molina, Peréz, Caballero, & Hernàndez-Dìaz, 2010)) with interesting results.

Other, even more popular, metaheuristics have been applied to continuous GO problems, such as Tabu search ((Cvijović & Klinowski, 2002; Hedar & Fukushima, 2006)) and Ant Colonies Optimization ((Shelokar, Siarry, Jayaraman, & Kulkarni, 2007; Socha & Dorigo, 2008)). We refer the interested reader to the cited papers and references therein.

3.1.15 Machine learning and global optimization

We conclude this section by very briefly mentioning a recently developed approach, which might deserve further investigation in order to fully exploit its potential. As has been shown in this section, many heuristic techniques for GO problems are based on repeated trials performed by the same algorithm initialized by means of a single solution or a whole initial population. Usually these methods proceed in a Multistart-like fashion, as they are just repeated starting with different randomly generated solutions until some global stopping rule asks for termination of the procedure. After termination, the best solution is returned as an estimate of the global optimum.

It has been observed that a lot of information is wasted in this way; in particular, all of the runs which did not lead to the current best solution are forgotten. As computer memory is quite a cheap asset today, it seems worthwhile to keep some of the information collected during all the performed runs in order to learn something about the problem. This idea is the starting point of the *LeGO* (Learning for GO) approach introduced in (Cassioli, Di Lorenzo, Locatelli, Schoen, & Sciandrone, 2012). The main idea of this approach is that, after a number of runs performed by any GO method, it is possible to train a *support vector machine* (SVM), or any other machine learning tool, in order to find a correlation between the starting point(s) used to initialize the algorithm and the quality of the resulting solution. This way, after sufficient training has been performed, the trained SVM can be used as a predictor of the quality of a new starting point. In particular, the *GloballyGenerate* procedure of Algorithm 1 might be implemented as an acceptance/rejection procedure in which a new point (or a population) is randomly generated according to the usual technique employed by the algorithm, but the point is accepted as a starting point for the following steps if and only if it is accepted by the trained SVM; otherwise a different point is generated, until some stopping criterion is met.

Some numerical experiments have been performed and their results are presented and commented on in (Cassioli, Di Lorenzo, Locatelli, Schoen, & Sciandrone, 2012); there it is shown that the method, if designed with some care, can deliver very interesting results in terms of quality of the solution and in terms of efficiency.

3.2 Optimization methods for problems with expensive functions

When dealing with very expensive objective (or constraint) functions, it is natural to try to avoid unnecessary function evaluations as much as possible. A particularly interesting and well-studied method designed to achieve this goal is to use as much as possible the information so far collected, both on the location of points at which the function was observed and on function values, and to use this information to build a suitable *surrogate model* of the function. After a model is built, it can be used to extrapolate and guess the behavior of the expensive function at points which are different from the observed ones; various approaches are then available which enable us to choose the next evaluation point (or even the next group of points) based on the model. Assume that an expensive, possibly black-box function f has been evaluated at *k pairwise distinct* points $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^k \in \mathbb{R}^n$, and denote the observed function values as $f_j := f(\mathbf{x}^j), j = 1, k$. Let the sample at a generic step be defined as the set of pairs $S_k = \{(\mathbf{x}^1, f_1), (\mathbf{x}^2, f_2), \ldots, (\mathbf{x}^k, f_k)\}$. Usually at each iteration of a method a new observation is taken, so that usually k is both the cardinality of S_k and the iteration counter. However, sometimes more observations are chosen at each individual step, in which case the iteration counter will be different from the cardinality of S_k. In order to be able to effectively use a model-based sampling criterion, the following procedures should be suitably defined:

Initialization: A strategy for selecting the number k_0 and the location of a set of initial points $\mathbf{x}^1, \ldots, \mathbf{x}^{k_0}$. After function evaluation, an initial set S_{k_0} is defined as $S_{k_0} = \{(\mathbf{x}^1, f_1), (\mathbf{x}^2, f_2), \ldots, (\mathbf{x}^{k_0}, f_{k_0})\}$. Not all methods need an initialization, but most do.

Often the initial choice has to be blind; i.e., no model can be constructed until a certain number of points has been chosen and the black-box function evaluated. These initial points must usually satisfy some requirements characterizing their geometry, such as that of not being degenerate in some subspace (e.g., they must be all pairwise different, not aligned, not on the same hyperplane, etc.).

Model building: Based on the sample S_k, a model function $s(\cdot) = s(\cdot|S_k) : \mathbb{R}^n \to \mathbb{R}$ is built. The literature on *interpolation* and *approximation* (or *regression*) is huge, and any attempt to review it would require an extensive survey. Here it might suffice to recall that interpolation is usually confined to modeling functions whose evaluation is noise-free, while approximation is the choice when random errors are incurred when observing f. However, sometimes approximation is used also in noiseless environments in order to smooth out the oscillations of the true function f and to better capture its "global" behavior.

Sampling: After a model has been built, it can be used to choose one or more points for successive function evaluations. There are two main families of strategies to exploit the information in the surrogate model. These families are called, in the excellent surveys in (Jones, 2001b; Forrester & Keane, 2009), respectively, *two-stage* and *one-stage* methods, but here a different choice for their name is preferred. Both methods are based on the choice of a suitable *merit function*, which is used to drive the search toward promising points. The difference between the two approaches is that while in the first the approximation function is trusted as a good model and directly used to choose the next observation(s), in the second a guess for a new point is made and a new approximation is built taking into account this new point also. We will further discuss these methods below.

Most methods also prescribe choosing a real value $\hat{f}$ which represents a guess on the global minimum value for f, or at least an "*aspiration level*" corresponding to a desired sufficient decrease with respect to the current record $f_k^\star := \min_{j=1,k} f_j$. In many situations an estimate of the global minimum is available. As an example, when using optimization to calibrate parameters in a simulation model based upon, e.g., the minimization of the sum of squared errors between predicted and observed data, a natural choice for an aspiration level would be $\hat{f} = 0$ (i.e., no error in prediction). Some methods thus use this value as a guide to search for new evaluation points. It can be remarked that this aspiration level $\hat{f}$ is indeed a parameter which can be used to balance the exploratory and the local phases. Methods which use this level usually tend to be local when $\hat{f} \approx f_k^\star$; indeed, as both the objective function and the approximant are usually chosen to be continuous, small deviations from the observed minimum will often, but not necessarily, lead to small displacement in the next chosen point. On the other hand, when $\hat{f}$ is chosen far from the observed minimum or, as a limit, it is set to $-\infty$, less and less importance is given to the sample and aspiration level–based methods will tend to generate observations which are as far as possible from previous ones. In this way, global exploration is achieved.

The two families of methods are as follows.

Sample-based: No information except the sample and the surrogate model based on the sample is used. After a model is built, it is in some way considered as the true function f, and the decision on the next evaluation point(s) is based exclusively on the sample and the model. A *merit function* is defined which depends exclusively on the model so far obtained. Depending on the context, a merit function is usually either maximized or minimized. The easiest possibility is to choose the interpolant to be also the merit function. In this case the global minimizer of the model is used as the next evaluation point for f; other possibilities exist, in which, for example, the merit function is built as a linear combination of the current interpolation with a function which takes into account the distance to the closest point in the sample. In this case the merit function is defined so that it can balance the greedy search based on the interpolant alone, with an exploratory search, based on the desire to choose a point which is far enough from past observations.

Some of these methods also use an aspiration level $\hat{f}$ and try to choose the next point where, in some sense to be made more precise, it is more likely to observe that level. Usually these methods rely on a stochastic model of the objective function.

Extended sample-based: Given an aspiration level, the model is extended assuming the sample is

$$S_k(\hat{\mathbf{x}}) := S_k \bigcup \{(\hat{\mathbf{x}}, \hat{f})\},$$

where $\hat{\mathbf{x}} \in \mathbb{R}^n$. It is thus assumed that the aspiration level is attained at a point $\hat{\mathbf{x}}$. The model is thus forced to pass through $(\hat{\mathbf{x}}, \hat{f})$ if it is interpolation based, or in any case to incorporate this new point if it is regression based. Of course, even if the value $\hat{f}$ might be trusted, the point $\hat{\mathbf{x}}$ is unknown, so that the resulting model $s(\cdot|S_k \bigcup \{(\hat{\mathbf{x}}, \hat{f})\})$ is parameterized by the vector $\hat{\mathbf{x}}$. Many strategies can then be employed to define a merit function and thus to choose the "most likely" location for $\hat{\mathbf{x}}$. As an example, $\hat{\mathbf{x}}$ might be chosen so that the resulting model is the least oscillating in a family of analogous models (or the least "bumpy" as in (Gutmann, 2001a)). In this case the merit function is a measure of the total curvature, or bumpiness, of the function. In other cases, assuming the objective function is the realization of a stochastic process, the merit function can be defined as the expected value of this process, conditioned upon past observations, or as the probability of improving with respect to the current record.

In both cases, once new points are chosen, function f is evaluated and the sample S_k enlarged to include the new information. The whole procedure is then repeated with a new model, until a stopping criterion is met, e.g., a maximum number of function evaluations has been reached.

The above procedure is quite general and in this section some more details will be given on how to implement its components. It is important to recall that, as in most heuristic techniques for GO, the two conflicting requirements of exploration and refinement should be taken into account here also. Exploration is attained by choosing new sample points sufficiently far from the others, in unexplored yet promising areas; refinement is obtained when concentrating the sample around good sample points. When choosing the new point in the above procedure, there is a danger of trusting the model too much. This leads to an imbalance between exploration and refinement, with a tendency to overemphasize the latter. Thus, frequently in those approaches it is chosen to draw some of the samples disregarding the model and choosing new observations in large unexplored regions. Choosing the correct mix of exploratory and refinement moves is not easy, but this is a characteristic of most GO heuristics, not just of those based on model building. An interesting possibility, which was proposed in (Jones, 2001b) but has up to now received only partial attention, is that of trying to explore all of the possibilities, i.e., exploring all possible choices obtained by letting $\hat{f}$ vary continuously from $f_k^\star$ to $-\infty$. Of course, in practice only a finite sample of possible values for $\hat{f}$ can be observed. The idea is that the resulting points will form distinct clusters, which might be identified by means of a statistical tool; a single representative point in each cluster is then chosen for the evaluation of the objective function f.

The pseudo code represented in Algorithm 14 summarizes the main parts of the general scheme outlined in this section; in this scheme a single new observation is chosen at each iteration of the final loop. It is easy to outline a similar scheme for the case in which blocks of observations are chosen instead.

```
Let $k_0 \geq 0$ ;
for $k \in 1, k_0$ do
    Choose $\mathbf{x}_k \in \mathbb{R}^n$;
    Evaluate $f_k = f(\mathbf{x}_k)$;
end
Let $S_{k_0} := \{(\mathbf{x}_1, f_1), \ldots, (\mathbf{x}_{k_0}, f_{k_0})\}$ $(S_0 = \emptyset$ if $k_0 = 0)$;
Let $k := k_0$;
repeat
    Let $s(x) = s(x|S_k)$ be a model for $f$ given $S_k$;
    Let $\mathcal{M}(x|S_k)$ be a “merit function,” depending on $s(\cdot)$;
    Choose $\mathbf{x}_{k+1} \in \arg\max_{\mathbf{x}} \mathcal{M}(x|S_k)$ ;
    Evaluate $f_{k+1} = f(\mathbf{x}_{k+1})$;
    Let $S_{k+1} = S_k \cup \{(\mathbf{x}_{k+1}, f_{k+1})\}$;
    Let $k = k + 1$
until Stop Condition;
```

Algorithm 14: A general framework for optimization based on surrogate models

In the following sections various possibilities for model-based optimization will be presented. First, however, it seems necessary to give some basic notions on the theory and techniques for building interpolation and approximation functions in $\mathbb{R}^n$, as this topic is the basis upon which most of the methods are built.

3.2.1 Surrogate model building

An introduction to multivariate interpolation

This section contains some fundamental results which form the basis for the construction of interpolating surrogate models for a function f. Usually this function is the objective to be minimized, although it might equally be the left-hand side of an expensive constraint. Assume that f has been evaluated at *k pairwise distinct* points $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^k \in \mathbb{R}^n$, and denote the observed function values as $f_j := f(\mathbf{x}^j), j = 1, k$.

The space $\mathbb{R}_m[\mathbf{x}]$, $\mathbf{x} = (x_1 \ldots x_n)$, of polynomials in $\mathbb{R}^n$ whose degree is at most $m \in \mathbb{N}$ is a linear space which can be finitely generated. Let $p_1, p_2, \ldots, p_{\hat{m}}$ be a basis for $\mathbb{R}_m[\mathbf{x}]$;

the dimension $\dim(\mathbb{R}_m[\mathbf{x}])$ of this space will be denoted as $\hat{m}$. $\hat{m} = \dim(\mathbb{R}_m[\mathbf{x}])$ is a function both of the maximum degree m of the polynomials and of the dimension n of the embedding space, and it can be proven (see, e.g., (Wendland, 2005)) that

$$\hat{m} = \binom{m+n}{n}.$$

For example, for any n the space of constant polynomials ($m = 0$) has dimension 1, as it can be generated by the constant polynomial 1. The space of polynomials of degree at most 1 can be generated by the basis

$$1, x_1, x_2, \dots, x_n;$$

thus its dimension is $1+n$. For quadratic polynomials, the basis should include also bilinear and quadratic terms

$$x_1^2, x_1x_2, \dots, x_1x_n, x_2^2, x_2x_3, \dots, x_2x_n, \dots, x_n^2;$$

thus its dimension is $1+n+n(n+1)/2 = \binom{n+2}{n}$, and so on.

Thus any polynomial $\pi(\mathbf{x}) \in \mathbb{R}_m[\mathbf{x}]$ can be obtained in a unique way as a linear combination of the basis; i.e., there exist uniquely determined coefficients $c_1, c_2, \dots, c_{\hat{m}}$ such that

$$\pi(\mathbf{x}) = \sum_{\ell=1}^{\hat{m}} c_\ell p_\ell(\mathbf{x}) \qquad \forall \mathbf{x} \in \mathbb{R}^n.$$

A *polynomial interpolant* in $\mathbb{R}_m[\mathbf{x}]$ is any polynomial $\pi \in \mathbb{R}_m[\mathbf{x}]$ such that

$$\pi(\mathbf{x}^j) = f_j, \qquad j = 1, k.$$

Such an interpolating polynomial can be found solving the following linear system in the unknowns c_ℓ:

$$\sum_{\ell=1}^{\hat{m}} c_\ell p_\ell(\mathbf{x}^j) = f_j, \qquad j = 1, k. \tag{3.10}$$

Let $\mathbf{P} \in \mathbb{R}^{k \times \hat{m}}$ be the matrix whose rows correspond to the values of the basic polynomials at each of the observation points:

$$\mathbf{P} = \begin{bmatrix} p_1(\mathbf{x}^1) & p_2(\mathbf{x}^1) & \cdots & p_{\hat{m}}(\mathbf{x}^1) \\ p_1(\mathbf{x}^2) & p_2(\mathbf{x}^2) & \cdots & p_{\hat{m}}(\mathbf{x}^2) \\ \vdots & & & \vdots \\ p_1(\mathbf{x}^k) & p_2(\mathbf{x}^k) & \cdots & p_{\hat{m}}(\mathbf{x}^k) \end{bmatrix}. \tag{3.11}$$

As an example, for polynomials in $\mathbb{R}_1[\mathbf{x}]$, the matrix is

$$\mathbf{P} = \begin{bmatrix} 1 & \mathbf{x}^{1^T} \\ 1 & \mathbf{x}^{2^T} \\ \vdots & \vdots \\ 1 & \mathbf{x}^{k^T} \end{bmatrix},$$

or, using the compact notation $\mathbf{e}$ to indicate a column vector whose elements are all equal to 1 and $\mathbf{X}$ for a matrix whose jth column is $\mathbf{x}^j$, then

$$\mathbf{P} = \begin{bmatrix}\mathbf{e} & \mathbf{X}^T\end{bmatrix} \in \mathbb{R}^{k\times(n+1)}.$$

Given these definitions, an interpolating polynomial can be found solving for the unknown vector $\mathbf{c}$ the system

$$\mathbf{P}\mathbf{c} = \mathbf{f}, \tag{3.12}$$

where $\mathbf{f} \in \mathbb{R}^k$ is the column vector whose jth component is f_j. It is evident that the interpolating polynomial is uniquely defined if and only if the above linear system admits a unique solution. In order to characterize the situations in which a unique polynomial interpolation exists, the following definition and theorem are useful.

Definition 3.2. *A set of vectors* $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^k$ *in* $\mathbb{R}^n$ *is called* $\mathbb{R}_m[\mathbf{x}]$-unisolvent *if the unique polynomial* $q \in \mathbb{R}_m[\mathbf{x}]$ *such that*

$$q(\mathbf{x}^j) = 0, \qquad j = 1, k,$$

is the null polynomial.

This definition is linked to interpolation thanks to the following, whose proof is elementary.

Theorem 3.3. *A set of vectors* $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^k$ *in* $\mathbb{R}^n$ *is* $\mathbb{R}_m[\mathbf{x}]$*-unisolvent if and only if* $\mathrm{rank}(\mathbf{P}) = \hat{m}$.

From this it follows that a unique polynomial interpolation might exist only if at least $\hat{m}$ interpolation points are available.

Although quite simple, polynomial interpolation should in general be avoided; in fact, as the number k of point increases, polynomial interpolants usually will require a higher degree; this fact has the negative consequence that the resulting function will be extremely oscillating (or "bumpy"), and, in general, although correctly interpolating at the observation points, it will often be a very bad predictor of function behavior far from sampled points.

In the univariate cases ($n = 1$), splines are frequently used to obtain smooth interpolation without too many oscillations; however, an analogue to the multivariate case is not easy to define. Radial bases have been proposed as a very effective and interesting possibility. A *radial function* is defined as

$$s(\mathbf{x}) = \sum_{j=1}^{h} \lambda_j \varphi(\|\mathbf{x} - \mathbf{y}^j\|), \tag{3.13}$$

where λ_j are real numbers, $\{\mathbf{y}^j\}_{j=1}^h \in \mathbb{R}^n$ are called *centers* of the radial basis, and

$$\varphi : \mathbb{R}^+ \to \mathbb{R}$$

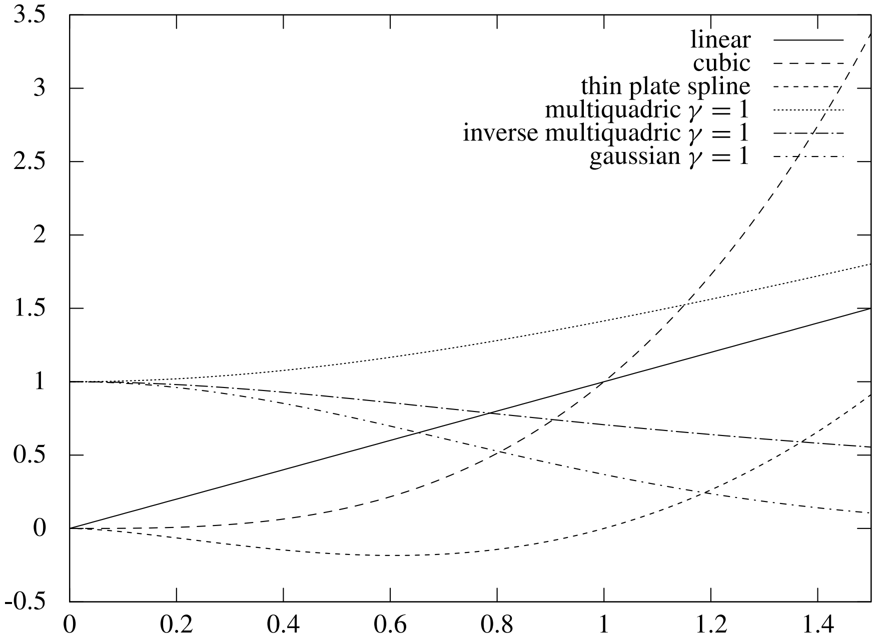


Figure 3.17. *Common radial basis functions*

is a univariate function. Although there is no restriction on the norm used, usually it is understood that $\|\cdot\|$ represents the Euclidean norm. A radial function is thus a linear combination of basis functions which are radially symmetric around the chosen centers. Function φ is usually referred to as *radial basis function* (RBF). Some of the most common choices for the RBFs, sketched in Figure 3.17, are the following (Gutmann, 2001a; Buhmann, 2003):

- **linear:** $\varphi(r) = r$;
- **cubic:** $\varphi(r) = r^3$;
- **thin plate spline:** $\varphi(r) = r^2 \log r$;
- **multiquadric:** $\varphi(r) = \sqrt{r^2 + \gamma^2}$, with $\gamma \neq 0$;
- **inverse multiquadric:** $\varphi(r) = (r^2 + \gamma^2)^{-1/2}$, with $\gamma \neq 0$; and
- **Gaussian:** $\varphi(r) = \exp - \gamma r^2$, with $\gamma > 0$.

A very common choice in interpolation is to let the set of centers coincide with the set of interpolation points $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^k$. Denoting by $\Phi \in \mathbb{R}^{k \times k}$ the matrix whose (i, j)th element is $\varphi(\|\mathbf{x}^i - \mathbf{x}^j\|), i, j = 1, k$,

$$\Phi = \begin{bmatrix} \varphi(0) & \varphi(\|\mathbf{x}^1 - \mathbf{x}^2\|) & \cdots & \varphi(\|\mathbf{x}^1 - \mathbf{x}^k\|) \\ \varphi(\|\mathbf{x}^1 - \mathbf{x}^2\|) & \varphi(0) & \cdots & \varphi(\|\mathbf{x}^2 - \mathbf{x}^k\|) \\ \vdots & & & \vdots \\ \varphi(\|\mathbf{x}^1 - \mathbf{x}^k\|) & \varphi(\|\mathbf{x}^2 - \mathbf{x}^k\|) & \cdots & \varphi(0) \end{bmatrix}, \tag{3.14}$$

it is immediately seen that a uniquely defined radial basis interpolant exists if and only if the linear system

$$\Phi \boldsymbol{\lambda} = \mathbf{f} \tag{3.15}$$

has a unique solution in $\boldsymbol{\lambda}$, i.e., if and only if matrix Φ is invertible. In some cases this condition can be very easily obtained. As an example, for the multiquadric RBF it can be shown that matrix Φ is always invertible, provided that $k \geq 2$, i.e., that there are at least two distinct points in the sample, which is an extremely weak requirement. However, there are exceptions: for the thin plate spline, if points $\mathbf{x}^2, \mathbf{x}^3, \ldots, \mathbf{x}^k$ are chosen on the unit sphere centered at $\mathbf{x}^1$, then a whole row and column of matrix Φ will vanish, so that no matter how many centers there are, a unique interpolant will not exist. Even worse, if $\mathbf{x}^1, \ldots, \mathbf{x}^{n+1}$ form a regular simplex in which every pairwise distance is equal to 1, then the corresponding Φ matrix is null.

The standard approach used to guarantee a unique interpolant is to mix the two approaches by forming a function s which is the sum of a radial function and a (low degree) polynomial:

$$s(\mathbf{x}) = \sum_{j=1}^{k} \lambda_j \varphi(\|\mathbf{x} - \mathbf{x}^j\|) + \pi(\mathbf{x}), \tag{3.16}$$

$$\pi(\mathbf{x}) := \sum_{\ell=1}^{\hat{m}} c_\ell p_\ell(\mathbf{x}). \tag{3.17}$$

From all the above it is obtained that a unique interpolant might exist only if the system

$$\Phi \boldsymbol{\lambda} + \mathbf{P}\mathbf{c} = \mathbf{f}$$

has a unique solution. However, this system has k equations, one for each interpolation point, and $k + \hat{m}$ variables, one for each radial basis and one for each polynomial basis. Thus, with the exception of the case in which no polynomial is added (a case in which we conventionally put $\hat{m} = 0$), the system is underdetermined. The $\hat{m}$ further conditions usually imposed for interpolation are the following:

$$\sum_{j=1}^{k} \lambda_j p(\mathbf{x}^j) = 0 \qquad \forall p \in \mathbb{R}_m[\mathbf{x}] \tag{3.18}$$

or, equivalently,

$$\sum_{j=1}^{k} \lambda_j p_\ell(\mathbf{x}^j) = 0, \qquad \ell = 1, \hat{m}.$$

One of the motivations for introducing these conditions, apart from the desire to obtain a unique interpolant, will become clearer later, when it will be shown that thanks to this assumption it will be relatively easy to derive a measure for the "bumpiness," or total oscillation, of the resulting function (see Section 3.2.2). This measure will then be used to guide the search toward the next evaluation points in a GO algorithm.

Thus the system to be solved is

$$\Phi\boldsymbol{\lambda} + \mathbf{P}\mathbf{c} = \mathbf{f}, \tag{3.19}$$

$$\boldsymbol{\lambda}^T\mathbf{P} = \mathbf{0} \tag{3.20}$$

or, in matrix form,

$$\begin{bmatrix} \Phi & \mathbf{P} \\ \mathbf{P}^T & \mathbf{O}_{m,m} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \mathbf{c} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0}_m \end{bmatrix}, \tag{3.21}$$

where $\mathbf{O}$ represents the zero matrix of appropriate dimension. The theory behind the unique solution of the above system and the unicity of the radial basis interpolation is well developed (see, e.g., (Buhmann, 2003; Wendland, 2005)). We report here some of the main results relevant to what follows, where it is understood that $\mathbb{R}_{-1}[\mathbf{x}] = \emptyset$.

Definition 3.4. *A symmetric function* $\phi : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ *is* strictly conditional positive definite *of order* $m \geq 0$ *in* $\mathbb{R}^n$ *if for every set* $\mathbf{X} = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^k\}$ *of distinct points and all choices of* $\boldsymbol{\lambda} \neq 0$ *satisfying the orthogonality condition*

$$\sum_{j=1}^{k} \lambda_j q(\mathbf{x}^j) = 0 \qquad\qquad \forall q \in \mathbb{R}_{m-1}[\mathbf{x}],$$

it holds that the quadratic form

$$\boldsymbol{\lambda}^T \Phi \boldsymbol{\lambda} = \sum_{i=1}^{k} \sum_{j=1}^{k} \lambda_i \lambda_j \phi(\mathbf{x}^i, \mathbf{x}^j)$$

is positive.

A function ϕ which is strictly conditional positive of order 0 is also called *strictly positive definite*. The relevance of strict conditional positive definiteness in the context of radial basis interpolation is given by the following theorem.

Theorem 3.5. *If* $\phi(\mathbf{x}, \mathbf{y}) = \varphi(\|\mathbf{x} - \mathbf{y}\|)$ *is strictly positive definite of order* $m \geq 0$ *and* $\mathbf{X}$ *is unisolvent for* $\mathbb{R}_{m-1}[\mathbf{x}]$*, then the linear system*

$$\begin{bmatrix} \Phi & \mathbf{P} \\ \mathbf{P}^T & \mathbf{O}_{m,m} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \mathbf{c} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0}_m \end{bmatrix}$$

admits a unique solution.

Proof. Consider the homogeneous linear system

$$\begin{bmatrix} \Phi & \mathbf{P} \\ \mathbf{P}^T & \mathbf{O}_{m,m} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} = \mathbf{0}.$$

Multiplying the first block of equations on the left by $\boldsymbol{\alpha}^T$,

$$\begin{aligned}&\boldsymbol{\alpha}^T\Phi\boldsymbol{\alpha}+\boldsymbol{\alpha}^T\mathbf{P}\boldsymbol{\beta}\\ &\quad=\boldsymbol{\alpha}^T\Phi\boldsymbol{\alpha}+(\mathbf{P}^T\boldsymbol{\alpha})\boldsymbol{\beta}\\ &\quad=\boldsymbol{\alpha}^T\Phi\boldsymbol{\alpha},\end{aligned}$$

which, thanks to the conditional positive definiteness of Φ can be null if and only if $\boldsymbol{\alpha}=\mathbf{0}$. Moreover, given this, from the same first block of equations

$$\mathbf{P}\boldsymbol{\beta}=\mathbf{0},$$

which implies $\boldsymbol{\beta}=\mathbf{0}$ thanks to the fact that $\mathbf{X}$ is unisolvent. Thus the null space of the linear system is the zero vector, and thus uniqueness is proven. □

It should be observed that the assumption on the unisolvency of $\mathbf{X}$ is required only to prove the unicity of the interpolant, while it is not required to prove the existence. In order to stress the dependency of m on the radial basis function φ, in what follows m_φ will denote the order of φ. Let us introduce the following definition.

Definition 3.6. *A function $g\in\mathcal{C}^\infty(\mathbb{R}_+)$ is* completely monotonic *if its derivatives satisfy the alternating sign condition*

$$(-1)^h\frac{d^h}{dt^h}g(t)\geq 0 \qquad\qquad \forall t>0, h\geq 0.$$

Then, a sufficient condition for strictly positive definiteness is the following.

Theorem 3.7. *If φ is a continuous function and there exists an integer k such that, for $t>0$,*

$$g(t)=(-1)^k\frac{d^k}{dt^k}\varphi(\sqrt{t})$$

is completely monotonic but not constant, then

$$\phi(\mathbf{x},\mathbf{y})=\varphi(\|\mathbf{x}-\mathbf{y}\|)$$

is strictly conditionally positive definite of order k on $\mathbb{R}^n$.

As an application of the above theorem, it is possible to check the condition for the most popular RBFs. For the linear RBF,

$$\phi(\sqrt{r})=\sqrt{r},$$

the first derivative is positive, while all its higher-order derivatives have alternating signs. So the kernel defined by $\varphi(r)=r$ is strictly positive definite of order 1. For the cubic basis, the second derivative of $\varphi(\sqrt{r})$ is positive, and then the signs alternate, so that the cubic radial basis is strictly positive definite of order 2. Analogously, the thin plate can be seen to satisfy the above theorem with order $k=2$, while, e.g., the Gaussian is strictly

Table 3.1. *Order of common strictly positive definite kernel functions*

Basis	$\varphi(r)$	Order
linear	r	1
cubic	r^3	2
thin plate spline	$r^2 \log r$	2
multiquadric	$\sqrt{r^2+\gamma^2}$	1
inverse multiquadric	$(r^2+\gamma^2)^{-1/2}$	0
Gaussian	$\exp(-\gamma r^2)$	0

positive definite (of order 0). Table 3.1 summarizes the results of the application of the above theorem to the most common basis functions.

Putting together the last conditions with Theorem 3.5, we obtain that, in order to obtain a unique RBF interpolant for cubic and thin plate spline radial bases, it is sufficient to add a polynomial of degree at least 1, while for the linear and multiquadric a constant polynomial has to be added. For the Gaussian case no condition is assumed, and no polynomial is required.

Concerning the requirements on the geometry of sample points, nothing is required for the linear, multiquadric, Gaussian bases, while for the cubic and thin plate splines (by far the most frequently chosen) in order to have a unique interpolant it is required that

$$\mathbf{P} = \begin{bmatrix} \mathbf{e} & \mathbf{X}^T \end{bmatrix} \in \mathbb{R}^{k\times(n+1)}$$

have rank $n+1$, or, in other words, that the set of points $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^k$ be $\mathbb{R}_1[\mathbf{x}]$-unisolvent. Thus, as a first trivial consequence, no unique interpolation can be found until at least $k = n+1$ points have been sampled. Moreover, these first $n+1$ points are unisolvent if and only if they are *affinely independent*. That is,

$$(\mathbf{x}^1 - \mathbf{x}^{n+1}), (\mathbf{x}^2 - \mathbf{x}^{n+1}), \ldots, (\mathbf{x}^n - \mathbf{x}^{n+1})$$

should be linearly independent, or, in other words, the points $\mathbf{x}^1, \ldots, \mathbf{x}^{n+1}$ should form a nondegenerate simplex. The proof of this fact is relatively easy. Indeed, matrix $\mathbf{P}$, when $k = n+1$, has full rank if and only if

$$\sum_{j=1}^{n+1} \lambda_j \mathbf{x}^j = 0,$$

$$\sum_{j=1}^{n+1} \lambda_j = 0$$

implies $\lambda_j = 0$ for all j. But the system can be written as

$$\sum_{j=1}^{n} \lambda_j \mathbf{x}^j + \lambda_{n+1}\mathbf{x}^{n+1} = 0,$$

$$\sum_{j=1}^{n} \lambda_j + \lambda_{n+1} = 0,$$

from which, by substitution,

$$\sum_{j=1}^{n} \lambda_j \mathbf{x}^j - \sum_{j=1}^{n} \lambda_j \mathbf{x}^{n+1} = 0,$$

which implies $\lambda_j = 0$ if and only if the vectors are affinely independent. Thus, in order to start a radial basis interpolation with cubic or thin spline basis functions, a starting n-dimensional simplex has to be provided. Of course different, and more demanding, assumptions on the initial points have to be required if the degree m of the polynomial used in the interpolation is chosen greater than 1. This choice from one side might be seen as not reasonable, as the theory of interpolation guarantees the existence of unique interpolation with the minimal requirements given above. However, there might be an interest in including higher-order polynomials in order to be able to obtain a better fit to larger classes of functions. As an example, if f happens to be a quadratic function, it will never be perfectly interpolated by a finite number of thin plate–based radial bases with the addition of only a polynomial of degree one.

To derive the updated interpolant which includes the pair $(\hat{\mathbf{x}}, \hat{f})$ it is useful to reconsider the definition of the radial basis interpolation function as an iterative procedure.

Sequential updating of the RBF interpolation

It might be observed that, given a unisolvent set of observations $\mathbf{X} = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^k\}$ and a strictly positive definite RBF φ of order m, the linear system giving rise to the unique interpolant has dimension $k + \hat{m}$; however, the resulting interpolation is the sum of k functions only, one for each interpolation center. It might be argued then that a basis consisting of only k functions might be employed to generate the same interpolant. Indeed this is the case; let

$$\hat{\mathcal{L}}_j(\mathbf{x}), \qquad\qquad j = 1, k,$$

be a set of functions characterized by the interpolation requirements

$$\hat{\mathcal{L}}_j(\mathbf{x}^j) = 1, \tag{3.22a}$$

$$\hat{\mathcal{L}}_j(\mathbf{x}^i) = 0, \qquad\qquad i \neq j. \tag{3.22b}$$

Then an interpolation to the data might be obtained through the linear combination

$$s(\mathbf{x}) = \sum_{j=1}^{k} f_j \hat{\mathcal{L}}_j(\mathbf{x}).$$

This form of interpolation is closely reminiscent of the classical Lagrange polynomial interpolation, so that functions $\hat{\mathcal{L}}_j(\cdot)$ are often referred to as *Lagrange functions*. Conditions (3.22) are interpolation conditions over the set $\mathbf{X}$ with binary values. If functions $\hat{\mathcal{L}}$ are chosen within the same family as those used for radial basis interpolation, i.e., if

$$\hat{\mathcal{L}}_i(\mathbf{x}) = \sum_{j=1}^{k} \lambda_{ij} \varphi(\|\mathbf{x} - \mathbf{x}^j\|) + \sum_{\ell=1}^{\hat{m}} c_{i\ell} p_\ell(\mathbf{x}),$$

then the coefficients of each Lagrange function can be found by solving a linear system similar to (3.21). This approach can be used if each Lagrange function is built after all points **X** have been sampled. In what follows, it seems more useful to derive an incremental Lagrange expansion which enables us to retain all already evaluated Lagrange functions when a new observation becomes available. In this case, letting $s_0(\mathbf{x}) := 0$, the following expression can be used to derive the interpolant:

$$s(\mathbf{x}) = s_k(\mathbf{x}|\{(\mathbf{x}^1, f_1), \ldots, (\mathbf{x}^k, f_k)\}) = s_{k-1}(\mathbf{x}) + (f_k - s_{k-1}(\mathbf{x}^n))\mathcal{L}_k(\mathbf{x}),$$

where, similar to the preceding case, the $\mathcal{L}_j$'s are Lagrange functions, but the interpolation condition is imposed only on points with indices not greater than k. In other words, for $j = 1, k$, they satisfy

$$\mathcal{L}_j(\mathbf{x}^j) = 1, \tag{3.23a}$$

$$\mathcal{L}_j(\mathbf{x}^i) = 0, \qquad i \leq j-1. \tag{3.23b}$$

Thanks to this recursive definition, a single Lagrange function has to be determined after each new observation.

Let $(\mathbf{x}^{k+1}, f_{k+1})$ be a new observation; then it is possible to iteratively define the interpolation based on $S_{k+1} = S_k \bigcup(\mathbf{x}^{k+1}, f_{k+1})$ as

$$s(\mathbf{x}|S_{k+1}) = s(\mathbf{x}|S_k) + (f_{k+1} - s(\mathbf{x}^{k+1}|S_k))\mathcal{L}_{k+1}(\mathbf{x}),$$

where $\mathcal{L}_{k+1}(\mathbf{x})$ is an interpolation function in the same family $\mathcal{F}_{\varphi,m}$ of s, where $\mathcal{F}_{\varphi,m}$ is the family of all radial basis interpolants built through the RBF φ and a polynomial of degree at most m. By this definition $s(\mathbf{x}|S_{k+1})$ is an interpolant based on the data S_{k+1}; moreover, being a linear combination of two functions from $\mathcal{F}_{\varphi,m}$ it belongs to the same space, and, thanks to the unicity of the interpolation, it is the required function. One of the advantages of building the interpolation in sequential steps is that the coefficient updates can now be defined independently from the values $f_1, \ldots, f_{k+1}$. Indeed, imposing (3.23) on $\mathcal{L}_{k+1}(\mathbf{x})$ corresponds to finding a radial basis interpolation with value 0 at $\{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^k\}$ and value 1 at $\mathbf{x}^{k+1}$.

Function $\mathcal{L}_{k+1}$ can be written explicitly as an element in $\mathcal{F}_{\varphi,m}$:

$$\begin{aligned}\mathcal{L}_{k+1}(\mathbf{x}) = &\sum_{j=1}^{k} \alpha_j(\mathbf{x}^{k+1})\varphi(\|\mathbf{x} - \mathbf{x}^j\|) + \beta(\mathbf{x}^{k+1})\varphi(\|\mathbf{x} - \mathbf{x}^{k+1}\|) \\ &+ \sum_{\ell=1}^{\hat{m}} b_\ell(\mathbf{x}^{k+1}) p_\ell(\mathbf{x}).\end{aligned}$$

Recalling the general theory of interpolation by radial basis, the unknown coefficients are the unique solution of a linear system similar to (3.21). Indeed, let

$$\boldsymbol{\phi}_{k+1}(\mathbf{x}) = \begin{bmatrix} \varphi(\|\mathbf{x}^1 - \mathbf{x}\|) \\ \varphi(\|\mathbf{x}^2 - \mathbf{x}\|) \\ \vdots \\ \varphi(\|\mathbf{x}^k - \mathbf{x}\|) \end{bmatrix}$$

and

$$\boldsymbol{\pi}_{k+1}(\mathbf{x}) = \begin{bmatrix} p_1(\mathbf{x}) \\ p_2(\mathbf{x}) \\ \vdots \\ p_{\hat{m}}(\mathbf{x}) \end{bmatrix}.$$

Then, the coefficients of $\mathcal{L}_{k+1}(\mathbf{x})$ can be uniquely obtained as the solution of the linear system

$$\begin{bmatrix} \Phi & \boldsymbol{\phi}_{k+1}(\mathbf{x}_{k+1}) & \mathbf{P} \\ \boldsymbol{\phi}^T_{k+1}(\mathbf{x}_{k+1}) & \varphi(0) & \boldsymbol{\pi}^T_{k+1}(\mathbf{x}_{k+1}) \\ \mathbf{P}^T & \boldsymbol{\pi}_{k+1}(\mathbf{x}_{k+1}) & \mathbf{O} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}(\mathbf{x}^{k+1}) \\ \beta(\mathbf{x}^{k+1}) \\ \mathbf{b}(\mathbf{x}^{k+1}) \end{bmatrix} = \begin{bmatrix} \mathbf{0}_k \\ 1 \\ \mathbf{0}_{\hat{m}} \end{bmatrix}. \tag{3.24}$$

Thus each time a new point is sampled and the objective function is evaluated, it is sufficient to solve system (3.24) to obtain the new interpolation formula. This on one hand is useful as a standard update method for interpolation, but it also forms the basis for the development of the GO method. Notice also that the matrix of the above system is exactly the same matrix which appears in the standard definition of RBF interpolation for $k+1$ points; the unique change is the right-hand side.

3.2.2 A "bumpiness" measure

RBFs generally possess the property of generating smooth interpolation without too many oscillations, and this is one of the main differences with respect to pure polynomial interpolation. Moreover, while in polynomial interpolation only a finite number of points can be exactly interpolated within the space of polynomials of fixed degree, in radial basis interpolation every new observation can be included and correctly interpolated, obviously at the expense of additional terms in the summation (3.16).

It is not easy to characterize the notion of bumpiness in the multivariate case. However (Gutmann, 2001b) provides an analogy with unidimensional natural splines which motivates a definition which can be assumed also in the multidimensional case. As can be seen in any elementary book on numerical analysis, given a finite set of points in $\mathbb{R}^1$ and the corresponding function evaluations

$$(x_1, f_1), (x_2, f_2), \ldots, (x_k, f_k),$$

a *cubic spline* is defined as a function $S(x)$ which, in each interval $[x_1,x_2]$, $[x_2,x_3], \ldots,$ $[x_{k-1},x_k]$ is a cubic polynomial. Function $S(x)$ moreover should satisfy the following requirements:

- $S(x_j) = f_j$, $j = 1,k$, i.e., S interpolates f at each point; and
- S is in $\mathcal{C}^2([x_1,x_k])$, i.e., at each interpolation point $x_2,\ldots,x_{k-1}$ there should be perfect agreement between left and right first and second derivatives.

If, in addition to these conditions, S also satisfies

- $S''(x_1) = S''(x_k) = 0,$

then the spline is called *natural.* It is known that natural cubic splines exist and are uniquely defined by these requirements. Moreover, they possess the interesting property of minimizing the quantity

$$I(g) := \int_{-\infty}^{\infty} g''(x)^2 dx$$

among all interpolants g such that $I(g)$ exists and is finite.

Natural cubic splines can be seen as special cases of radial basis interpolation in $\mathbb{R}^1$. Indeed, choosing the cubic basis function $\varphi(r) = r^3$, a one-dimensional radial basis interpolation is given by

$$s(x) = \sum_{j=1}^{k} \lambda_j |x - x_j|^3 + c_1 + c_2 x.$$

It is elementary to check that $s(x)$ restricted to each subinterval $[x_j, x_{j+1}]$ is a cubic function that is everywhere continuous and differentiable up to the second order. Thus it is a cubic spline interpolant. Moreover, the second derivative at the leftmost extremum x_1 is readily computed as

$$s''(x_1) = 6 \sum_{j=2}^{k} \lambda_j (x_j - x_1).$$

Recalling that $\boldsymbol{\lambda}^T \mathbf{P} = 0$, i.e.,

$$\sum_{j=1}^{k} \lambda_j = 0,$$

$$\sum_{j=1}^{k} \lambda_j x_j = 0,$$

it holds that

$$\begin{aligned} s''(x_1) &= 6 \sum_{j=2}^{k} \lambda_j (x_j - x_1) \\ &= 6 \left(\sum_{j=2}^{k} \lambda_j x_j - \left(\sum_{j=2}^{k} \lambda_j \right) x_1 \right) \\ &= 6 \left(\sum_{j=2}^{k} \lambda_j x_j + \lambda_1 x_1 \right) = 0. \end{aligned}$$

An analogous development leads to $s''(x_k) = 0$, and thus $s(x)$ is a *natural* cubic spline. Recalling the uniqueness of natural cubic splines, the cubic radial basis interpolant in $\mathbb{R}^1$

is *the* natural cubic spline. Along the same lines it can be seen that the second derivative vanishes everywhere outside the interval $[x_1, x_k]$.

By the way, this elementary analysis sheds some light on the meaning of the additional constraints (3.18). In the univariate case, for the cubic basis function, those conditions are equivalent to being natural, i.e., to having null curvature outside the interpolation interval. This also has the consequence of avoiding the introduction of unnecessary oscillation outside the interpolation interval.

Elaborating on the correspondence between natural splines and radial bases, (Gutmann, 2001b) further analyzes the total curvature expression for the natural cubic spline and proves the following result (which can be derived, integrating by parts, after tedious but elementary computations):

$$\begin{aligned}
I(s) &= \int_{-\infty}^{\infty} s''(x)^2\,dx \\
&= 12\sum_{j=1}^{k} \lambda_j s(x_j) \\
&= 12\sum_{j=1}^{k} \lambda_j \left(\sum_{i=1}^{k} \lambda_i \varphi(|x_i - x_j|) + p(x_j)\right) \\
&= 12\sum_{j=1}^{k}\sum_{i=1}^{k} \lambda_j \lambda_i \varphi(|x_i - x_j|),
\end{aligned}$$

where the assumption $\sum_{j=1}^{k} \lambda_j p(x_j) = 0$ was used. Then, at least for cubic splines in $\mathbb{R}^1$,

$$I(s) = 12\boldsymbol{\lambda}^T \Phi \boldsymbol{\lambda} \tag{3.25}$$

is a measure of the bumpiness of the interpolating function; this measure, for the chosen interpolation form, is minimal among all interpolants of the same points.

The analogy with the natural cubic splines can be carried over to more general interpolations obtained through RBFs. In fact, given two RBFs in the same family (i.e., defined by means of the same radial function and based on the addition of a polynomial of the same degree),

$$\begin{aligned}
u(\mathbf{x}) &= \sum_{j=1}^{k} \lambda_j \varphi(\|\mathbf{x} - \mathbf{x}^j\|) + p(\mathbf{x}), \\
v(\mathbf{x}) &= \sum_{j=1}^{k} \mu_j \varphi(\|\mathbf{x} - \mathbf{x}^j\|) + q(\mathbf{x}),
\end{aligned}$$

and defining

$$\langle u, v \rangle := (-1)^{m_\varphi} \sum_{j=1}^{k} \lambda_j v(\mathbf{x}^j),$$

it can be proven that

- $\langle u, v \rangle = \langle v, u \rangle$; and
- $\langle u, u \rangle > 0$, provided that the assumptions which guarantee existence and uniqueness of the interplant are satisfied.

Thus $\langle u, u \rangle^{1/2}$ is a seminorm which can be computed through

$$\langle u, u \rangle = (-1)^{m_\varphi} \boldsymbol{\lambda}^T \Phi \boldsymbol{\lambda} > 0. \tag{3.26}$$

This quantity is a measure of the total variation of u (actually, the definition of the inner product can be more general, allowing for different centers for u and v, but here this definition seems to be sufficient). The following theorem (Schaback, 1993) states that the unique interpolant defined in (3.16) attains the minimum value of the bumpiness measure.

Theorem 3.8. *Let φ be an RBF of order m_φ, and let $m \geq m_\varphi - 1$. Let $(\mathbf{x}^1, f_1), (\mathbf{x}^2, f_2), \ldots, (\mathbf{x}^k, f_k)$ be a set of interpolation points and values. If the interpolation points are such that the matrix*

$$\mathbf{P} = \begin{bmatrix} \mathbf{e} & \mathbf{X}^T \end{bmatrix} \in \mathbb{R}^{k \times (n+1)}$$

has rank $\hat{m} = \dim(\mathbb{R}_{m-1}[\mathbf{x}])$ (i.e., the sample points are $\mathbb{R}_{m-1}[\mathbf{x}]$-unisolvent), then the radial basis $s(\mathbf{x})$ of the form (3.16) uniquely obtained by solving the system

$$\begin{bmatrix} \Phi & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0}_{m,m} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \mathbf{c} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0}_m \end{bmatrix}$$

is such that

$$\langle s, s \rangle \leq \langle g, g \rangle \tag{3.27}$$

for every interpolating function g of the same family (i.e., obtained with the same radial function and a polynomial of the same degree as s).

Thus, if the seminorm $\langle g, g \rangle$ is used as a measure for bumpiness, then among all radial basis interpolations of a sample S_k, the one which satisfies the additional condition $\mathbf{P}^T \boldsymbol{\lambda} = \mathbf{0}$, which generalizes the property of natural splines, is the best, in the sense that it minimizes the bumpiness measure.

3.2.3 Kriging

Radial basis and polynomials are not unique possibilities for the interpolation of a function $f(\mathbf{x})$ observed at a finite set of points in $\mathbb{R}^n$, although they are so general and powerful that most popular interpolation methods can be seen as special cases of RBF interpolation. A very popular family of interpolation tools which, even if born from statistical analysis, can be viewed as a special case of RBF interpolation goes under the name of *Kriging*. This name, which is well known in statistics and engineering, was coined by (Matheron, 1963) in honor of Danie Krige, a mining engineer from South Africa who seems to have been the first to introduce this estimation technique while studying correlation patterns in spatial distributions of gold samples (Forrester & Keane, 2009). There are many variations of Kriging, and here the simplest, and most useful, will be presented.

The general framework for the development of Kriging interpolation (and, as will be shown later, regression) is the assumption of a stochastic model for the objective function f. By this it is meant that $f(\cdot)$ is considered not as a deterministic function but as a realization of a stochastic process $f(\cdot;\omega)$. This assumption forms the basis of quite a few GO algorithms, starting from the pioneering work of (Kushner, 1964) and (Kiefer & Wolfowitz, 1952). In the context of Kriging, using wherever possible the same notation introduced before in the context of RBF, it is assumed that

$$\begin{aligned} f(\mathbf{x};\omega) &= \sum_{i=1}^{m} c_i p_i(\mathbf{x}) + Z(\mathbf{x};\omega) + \varepsilon(\omega) \qquad (3.28)\\ &= \pi(\mathbf{x}) + Z(\mathbf{x};\omega) + \varepsilon(\omega), \end{aligned}$$

where, as before, $\{p_i\}_{i=1}^{m}$ is a polynomial basis and $\omega \in \Omega$ is an element of a suitably defined sample space. In what follows, wherever unnecessary for comprehension, the dependance on ω will be understood. In the Kriging literature it is assumed that the polynomial basis is sufficient for giving a reasonably accurate model for the objective function. However, two error terms are considered. One, $Z(\mathbf{x})$, is called the *systematic error* or *bias* and represents a discrepancy between the polynomial model and the actual function f; this discrepancy is a function of $\mathbf{x}$. The other term $\varepsilon(\omega)$ is a random noise, which accounts for the situation in which repeated measurements, or "observations," of the stochastic process give different observed values due to the presence of measurement errors. Differently from the bias, the measurement error is in general assumed to be independent of $\mathbf{x}$. In Kriging it is postulated that $Z(\mathbf{x})$ is the realization of a Gaussian stochastic process with 0 mean and with variance/covariance described by the following relation:

$$\begin{aligned} \mathrm{Cov}(Z(\mathbf{x}), Z(\mathbf{y})) &= \sigma^2 \exp\left(-\sum_{h=1}^{n} \theta_h |x^{(h)} - y^{(h)}|^{\psi_h}\right) && (3.29)\\ &= \sigma^2 \mathbf{R}(Z(\mathbf{x}), Z(\mathbf{y})) \qquad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n, && (3.30) \end{aligned}$$

where $\sigma^2, \{\theta_h\}, \{\psi_h\}$ are deterministic parameters and $\mathbf{R}$ represents the correlation between the stochastic models at two points. Thus it is assumed that, at each feasible point $\mathbf{x}$, $Z(\mathbf{x})$ is an $\mathcal{N}(0,\sigma^2)$ random variable (i.e., a random variable whose distribution is normal with average equal to 0 and variance equal to σ^2) and that the correlation between two points $\mathbf{x}_i$ and $\mathbf{x}_j$ is an exponentially decreasing function of a generalized distance between the two points. The parameters $\sigma^2 > 0, \theta_h > 0$, and $\psi_h \geq 1$ can be used to generate a wide variety of interpolation functions. In particular, θ_h is a scale parameter associated to the hth coordinate; large values of this parameter correspond to low correlation between different points (something which geologists associate with more "activity" in component h). ψ_h regulates the smoothness of the correlation around a point $\mathbf{x}$, with values close to 1 generating a non-differentiable model, while values higher than 2 correspond to flatter and flatter functions. Most papers dealing with Kriging assume $\psi_h = 2$ for all h. When no noise affects observations, which is the assumption made up until now throughout this chapter, the term $\varepsilon(\omega)$ is fixed to 0. By using standard statistical techniques and assuming that the parameters of the model are known, the best linear predictor of the value of $f(\mathbf{x};\omega)$ conditioned upon a sample $S_k = \{(\mathbf{x}^1, f_1), (\mathbf{x}^2, f_2), \ldots, (\mathbf{x}^k, f_k)\}$ is given by

$$s(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x})^T \boldsymbol{\Phi}^{-1}(\mathbf{f} - \mathbf{P}\hat{\mathbf{c}}) + \boldsymbol{\pi}^T(\mathbf{x})\hat{\mathbf{c}},$$

where

$$\begin{aligned}
\boldsymbol{\pi}(\mathbf{x}) &= [p_1(\mathbf{x}), p_2(\mathbf{x}), \ldots, p_m(\mathbf{x})]^T, \\
\boldsymbol{\phi}(\mathbf{x}) &= [\mathrm{Cov}(\mathbf{x}_1, \mathbf{x}), \mathrm{Cov}(\mathbf{x}_2, \mathbf{x}), \ldots, \mathrm{Cov}(\mathbf{x}_k, \mathbf{x})]^T, \\
\Phi_{ij} &= \mathrm{Cov}(\mathbf{x}_i, \mathbf{x}_j), && i, j = 1, k, \\
\mathbf{f} &= [f_1, f_2, \ldots, f_k]^T, \\
P_{ij} &= p_i(\mathbf{x}_j), && i = 1, m, j = 1, k,
\end{aligned}$$

and

$$\hat{\mathbf{c}} = \left(\mathbf{P}^T \boldsymbol{\Phi}^{-1} \mathbf{P}\right)^{-1} \mathbf{P}^T \boldsymbol{\Phi}^{-1} \mathbf{f}. \tag{3.31}$$

These formulae are derived through standard statistical estimation techniques based on the assumptions made on the stochastic process $Z(\cdot)$; the analytical derivations can be found in many textbooks and papers, such as (Sacks, Welch, Mitchell, & Wynn, 1989). In this volume, however, we would like to stress the fact that Kriging is just a specialization of RBF methods; this way something is lost and something is gained. What is lost considering Kriging as just as another RBF interpolation method is a statistical interpretation of the model. On the other hand, looking at Kriging in an RBF context opens the way to different uses of the interpolation itself, as will be seen, e.g., in Section 3.2.5. As a side note on Kriging from a deterministic perspective, it should be recalled that the assumption of normality on the observed function values is very strong and almost never justified in practice, when the objective function is indeed a deterministic one. Furthermore, the idea itself that the objective function is just the realization of a stochastic process is almost never realistic in an optimization context; or, if it can find some justification, it is usually confined to measurement errors (i.e., on the term $\varepsilon()$) and not in the bias. So, although extremely elegant, the Gaussian assumption is usually very far from reasonable, and all the statistical considerations which are derived from this assumption have only a heuristic justification. By this we are not dismissing their importance, but we are simply asking the reader to accept with some caution some statements found in the Kriging literature, such as that this interpolation method is superior to others because it is able to provide an explicit analytical form for the prediction error. Of course, this can be true only if the assumptions on the originating stochastic process can be verified; otherwise the estimate of the prediction error is nothing more than a merit function which can be used, heuristically, to drive the search toward interesting regions of the feasible space.

Looking at the similarity between the covariance function $\mathrm{Cov}(Z(\mathbf{x}_i), Z(\mathbf{x}_j))$ and the RBFs, it is reasonable to derive an RBF interpolation of the observed sample using the following RBF:

$$\varphi(\mathbf{x}_i; \mathbf{x}_j) = \sigma^2 \exp\left(-\sum_{h=1}^{n} \theta_h |x_i^{(h)} - x_j^{(h)}|^{\psi_h}\right)$$

(here a slight abuse in notation is assumed, where function φ in (3.13) instead of being a function of a single nonnegative variable is extended to a general function of two vectors). By using these bases for RBF interpolation as well as a polynomial basis represented by matrix $\mathbf{P}$, it has already been shown that the coefficients of the interpolation are the unique (under suitable assumptions) solution of the linear system

$$\begin{bmatrix} \mathbf{\Phi} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{O} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \mathbf{c} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix}.$$

If it is assumed that $\mathbf{\Phi}$ is invertible, then the first equation above is equivalent to

$$\boldsymbol{\lambda} + \mathbf{\Phi}^{-1}\mathbf{P}\mathbf{c} = \mathbf{\Phi}^{-1}\mathbf{f}.$$

Multiplying by $\mathbf{P}^T$ on the left and recalling that the second equation is $\mathbf{P}^T\boldsymbol{\lambda} = 0$, then

$$\mathbf{P}^T\mathbf{\Phi}^{-1}\mathbf{P}\mathbf{c} = \mathbf{P}^T\mathbf{\Phi}^{-1}\mathbf{f},$$

and, given the full rank assumption on $\mathbf{P}$ and the invertibility of $\mathbf{\Phi}$,

$$\mathbf{c} = \left(\mathbf{P}^T\mathbf{\Phi}^{-1}\mathbf{P}\right)^{-1}\mathbf{P}^T\mathbf{\Phi}^{-1}\mathbf{f},$$

which is exactly the Kriging best linear estimate $\hat{\mathbf{c}}$ of the $\mathbf{c}$ parameter (3.31). Using this estimate, the RBF coefficients are immediately obtained as

$$\hat{\boldsymbol{\lambda}} = \mathbf{\Phi}^{-1}(\mathbf{f} - \mathbf{P}\hat{\mathbf{c}}),$$

and through these coefficients, the RBF predictor of $f(\mathbf{x})$ turns out to be

$$s(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x})^T\mathbf{\Phi}^{-1}(\mathbf{f} - \mathbf{P}\hat{\mathbf{c}}) + \boldsymbol{\pi}^T(\mathbf{x})\hat{\mathbf{c}},$$

which is again exactly the Kriging predictor of $f(\mathbf{x})$. Thus it has been shown that Kriging is just a special, although particularly important, case of RBF interpolation. This immediately implies that Kriging, developed under the no-noise assumption $\varepsilon = 0$, generates an interpolant. What remains in order to be able to fully exploit the potential of Kriging is to devise a procedure to calibrate the large number of parameters in the model. In fact, $1 + 2n$ parameters, with n equal to the dimension of the space, have to be chosen in order to obtain a workable definition of a Kriging interpolant. This is both the weakness as well as the strength of Kriging. Having so many parameters to estimate is surely a defect, since, on one hand every estimation, or calibration, procedure is error prone, and, on the other hand, a reliable estimate can be obtained only when a sufficiently large sample has been observed. This is somewhat in contrast to the main assumption of this chapter, namely, that the observation of function values is a computationally expensive process. The richness in parameters, however, gives the family a remarkable flexibility, which makes these models extremely precise in the modeling of quite complex functions.

It can be seen that the Kriging kernel function is a generalized Gaussian kernel of order 0. Thus, as has been proven in this chapter, there is no need to introduce a polynomial into the interpolation. However, in the literature it is quite often assumed that a constant polynomial is added to the pure RBF interpolation. Thus in the literature the term Kriging is usually restricted to a predictor of the form

$$s(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x})^T\mathbf{\Phi}^{-1}(\mathbf{f} - \mathbf{e}\hat{c}) + \hat{c}, \tag{3.32}$$

where

$$\hat{c} = \frac{\mathbf{e}^T \mathbf{\Phi}^{-1}\mathbf{f}}{\mathbf{e}^T \mathbf{\Phi}^{-1}\mathbf{e}} \tag{3.33}$$

$$= \frac{\mathbf{e}^T \mathbf{R}^{-1}\mathbf{f}}{\mathbf{e}^T \mathbf{R}^{-1}\mathbf{e}}, \tag{3.34}$$

while the more general regression based on higher-degree polynomials is referred to as *universal Kriging*.

In what follows, in order to keep notation simpler, only ordinary Kriging will be considered, even if it is easy to generalize the results to universal Kriging interpolation as well.

The estimates for the unknown parameters are derived by means of maximum likelihood estimation derived under the normality assumptions. Assuming, as before, that a sample of k observation is available, it can be easily verified that the likelihood is given by the following expression:

$$\begin{aligned}\mathcal{L}(S_k) &= \frac{1}{(2\pi)^{k/2}\det(\mathbf{\Phi})^{1/2}} \exp - \left\{ \frac{1}{2}\left((\mathbf{f}-\mathbf{e}c)^T \mathbf{\Phi}^{-1} (\mathbf{f}-\mathbf{e}c) \right) \right\} \\ &= \frac{1}{(2\pi\sigma^2)^{k/2}\det(\mathbf{R})^{1/2}} \exp \left\{ -\frac{1}{2\sigma^2}\left((\mathbf{f}-\mathbf{e}c)^T \mathbf{R}^{-1} (\mathbf{f}-\mathbf{e}c) \right) \right\}.\end{aligned}$$

In order to derive maximum likelihood estimators it is more convenient to work with the logarithm of the likelihood:

$$\log \mathcal{L}(S_k) = -\frac{k}{2}\log(2\pi\sigma^2) - \frac{1}{2}\log\det(\mathbf{R}) - \frac{1}{2\sigma^2}\left((\mathbf{f}-\mathbf{e}c)^T \mathbf{R}^{-1} (\mathbf{f}-\mathbf{e}c) \right).$$

Equating the partial derivatives of the log likelihood with respect to σ^2 and to c to zero, the optimal estimate for the variance parameter is readily obtained as

$$\hat{\sigma}^2 = \frac{1}{k}\left((\mathbf{f}-\mathbf{e}c)^T \mathbf{R}^{-1} (\mathbf{f}-\mathbf{e}c) \right), \tag{3.35}$$

while the optimal estimate for c turns out to be exactly the $\hat{c}$ found previously in (3.34). These estimates can now be substituted in the log likelihood function

$$-\frac{k}{2}\log(2\pi) - \frac{k}{2}\log\left(\frac{1}{k}(\mathbf{f}-\mathbf{e}c)^T \mathbf{R}^{-1} (\mathbf{f}-\mathbf{e}c) \right) - \frac{1}{2}\log\det(\mathbf{R}) - \frac{k}{2}.$$

In order to derive optimal estimates of the shape and smoothness parameters $\{\theta_i\}_{i=1}^n$ and $\{\psi_i\}_{i=1}^n$, it is enough to consider the so-called *concentrated log likelihood* function (obtained disregarding constant terms)

$$-\frac{k}{2}\log\left((\mathbf{f}-\mathbf{e}c)^T \mathbf{R}^{-1} (\mathbf{f}-\mathbf{e}c) \right) - \frac{1}{2}\log\det(\mathbf{R}). \tag{3.36}$$

If the stochastic model can be at least partially trusted, then an advantage of the Kriging method is that it also provides an estimate of the prediction error, so that a reliability measure can be associated to each point and the associated estimated function value.

In fact, it can be found by standard statistical methods (see, e.g., (Sacks et al., 1989)) that the variance of the estimate made at point $\mathbf{x}$ is given by

$$\hat{\sigma}^2(\mathbf{x}) = \hat{\sigma}^2 \left(1 - \boldsymbol{\phi}(\mathbf{x})^T \mathbf{R}^{-1} \boldsymbol{\phi}(\mathbf{x}) + \frac{1 - \mathbf{e}^T \mathbf{R}^{-1} \boldsymbol{\phi}(\mathbf{x})}{\mathbf{e}^T \mathbf{R}^{-1} \mathbf{e}} \right). \tag{3.37}$$

Thus, according to the assumptions of Kriging models, at each point $\mathbf{x}$ the objective function is Gaussian, with mean (3.32) and variance (3.37); in both formulae, the maximum likelihood estimates for all the parameters of the model have to be used.

It should be immediately clear that finding the maximum likelihood estimates for these parameters is a significantly expensive computational task, as function (3.36) is a nonconvex $2n$-dimensional function which has to be globally maximized in an unbounded domain, as the shape parameters are required only to be positive, while the smoothness parameters are usually chosen not smaller than 2. This computational cost is usually assumed to be much lower than that related to the observation of function f; however, it is an additional burden which has to be carefully taken into account. Moreover, the reliability of these parameter estimations, and, as a consequence, the reliability of the whole interpolation, increases with sample size; this means that during the first iterations not much confidence can be given to the interpolation obtained by Kriging. Of course similar reasoning might apply also for general RBF (or any other basis) interpolation. However, it seems that during the initial iterations the additional computational burden required to train Kriging is not justified and mixed procedures, which start with regular RBF and switch to Kriging as soon as the sample size is sufficiently large, might be preferable.

It has thus been shown that Kriging is just another RBF interpolation method with the additional benefit of a statistical interpretation and the disadvantage of a demanding computational task required to tune the parameters at each iteration. As with any RBF-based method, once a reasonably reliable model has been built, many merit functions can be defined, as will be seen in Section 3.2.5, in order to guide the search toward promising regions. However, the statistical motivations on which Kriging is based allow, at least heuristically, for more specific methods which will be reviewed in Section 3.2.5. These methods exploit the statistical assumptions in order to define a merit function which estimates the probability of improving the current record. Although, as already pointed out, this statistical interpretation is usually not justified, the resulting methods are indeed interesting, as they provide a reasonable mix between exploratory and refinement moves, which are the heart of all GO methods.

3.2.4 Regression

In some cases, when an interpolation either is not possible or is not required, an approximation model, i.e., a *regression*, is chosen as a model for the objective function f. It has already been observed that existence and uniqueness of interpolation are subject to some assumptions: when they are not satisfied, such as when sample size is small, different methods should be used. More importantly, quite often even if interpolation is possible, it might be *undesirable*. There are at least two situations in which this might happen. The first is when the objective function is noisy, e.g., when observation of $f(\mathbf{x})$ is the outcome of a process involving some stochastic element. One of the best-known cases is when the objective function is the result of a simulation experiment. In these cases, knowing that the observed function values are perturbed by noise, it would be incorrect to fit an interpolation model; this would imply that the fitted model would interpolate function values which are almost surely “wrong.” Seen from a different perspective, in this case too much emphasis would be given to the observed function values, with the danger of overfitting the observations. This way we are trying to “learn” also the noise and not only the expected behavior of the objective function. As a trivial example, if f were a linear function affected by noise, a polynomial interpolation would generate a widely oscillating model, missing the approximation of the average linear trend. A similar situation happens when no noise is present, but the objective function has many “high-frequency” small oscillations. It is in this case interesting to try to “smooth out” these oscillations in order to let the algorithm learn the overall behavior of the function without trying to fit every small oscillation.

Another situation in which it might be desirable to avoid interpolation in favor of approximation is when the sample size grows. It has been shown that the number of terms in an RBF interpolation grows proportionally to the sample size. This growth is not a problem per se, but it implies that larger matrices will be needed and larger systems of equations will have to be solved to evaluate the coefficients of the RBF. As will be seen in Section 3.2.5, most methods for GO prescribe the evaluation of those coefficients a large number of times during the search for new sample points. Thus the interpolation of a sufficiently large sample makes accessory computations quite demanding. This is even more remarkable in the case of Kriging, where, as has been shown, parameter estimation also requires a complex optimization procedure. Another problem with interpolation is that, although theoretically the matrices which appear in the linear systems which define interpolation coefficients are positive definite, in practice, as sample points tend to cluster around best observations, there is a significant tendency toward singular matrices.

In all of the above situations, although with different motivations, there is a need for a less precise approximation. In other words, instead of asking for

$$s(\mathbf{x}_j) = f(\mathbf{x}_j) = f_j \qquad \forall j \in 1,k,$$

a model is sought for which

$$s(\mathbf{x}_j) \approx f_j \qquad \forall j \in 1,k,$$

where the meaning of $\approx$ has to be defined.

When sample data are insufficient to generate an interpolation, the method usually chosen in approximation literature is a least squares approximation. In practice, instead of solving the infeasible linear system

$$\mathbf{\Phi}\boldsymbol{\lambda} = \mathbf{f} \tag{3.38}$$

(where for simplicity we refer to the situation in which no polynomial is added to the RBF model), an estimate of $\boldsymbol{\lambda}$ is found as the solution of the problem

$$\min_{\boldsymbol{\lambda}} \| \mathbf{\Phi}\boldsymbol{\lambda} - \mathbf{f} \|_2^2$$

which, if the columns of $\mathbf{\Phi}$ are linearly independent, is

$$(\mathbf{\Phi}^T \mathbf{\Phi})^{-1} \mathbf{\Phi} \mathbf{f}.$$

When, however, the coefficient matrix $\mathbf{\Phi}$ (or its analogue in case a polynomial is added) is invertible, the above formula reduces to standard interpolation. If a regression is sought, some authors simply introduce a *regularization parameter* η and substitute (3.38) with

$$(\mathbf{\Phi}+\eta\mathbf{I})\boldsymbol{\lambda}=\mathbf{f}. \tag{3.39}$$

Here η is a parameter which can be estimated by means of maximum likelihood techniques in case f is noisy. In other situations, it has to be considered as an additional parameter of the model to be calibrated.

Another possibility, which is especially useful in later stages of an optimization algorithm, when sample density begins to grow in the neighborhood of some local optima, is choosing a set of centers for the radial basis and finding a least squares solution of the resulting approximation problem. Letting $\{\mathbf{y}_1,\dots,\mathbf{y}_h\}$ be a set of centers, the problem becomes choosing $\boldsymbol{\lambda}$ as the solution of the least squares problem

$$\min_{\boldsymbol{\lambda}} \|\mathbf{\Phi}\boldsymbol{\lambda}-\mathbf{f}\|_2^2,$$

where, in this case, $\mathbf{\Phi}$ is redefined as

$$\Phi_{ij}=\varphi(\|\mathbf{y}_i-\mathbf{x}_j\|) \qquad \forall i\in 1,k, j\in 1,h.$$

Apart from this definition, the regression coefficients can be found in the same way as before, by simply exploiting the optimality conditions, as

$$\hat{\boldsymbol{\lambda}}=(\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}\mathbf{f}.$$

In this case the problem remains choosing a suitable set of centers; one possibility is to adopt a greedy strategy, in which one center, extracted from the sample, at a time is added according to a criterion based on an error measure. The center which gives the minimum error estimate is chosen, until either a prefixed number of centers has been chosen, or the variation in the error measure falls below a threshold. This way, somewhat automatically, points which are too close will be discarded, as their inclusion in the approximation does not significantly alter the regression.

Another possibility for explicitly taking into account the presence of noise in building a surrogate model has been outlined in the previous section on Kriging. Indeed, there the original model included an error term, which we neglected just in order to obtain an interpolation. Working out all the necessary computations based on the same maximum likelihood estimation procedures, a regression can be found when including a zero-mean, constant variance error term in (3.28). The details are left as an exercise.

Recently a more elaborate and quite interesting approach has been proposed in (Jakobsson, Patriksson, Rudholm, & Wojciechowski, 2010). The approach explicitly refers to the definition of the bumpiness measure which has been introduced in Section 3.2.2. Assume, as before, that a functional space $\mathcal{F}_{\varphi,m}$ is defined, which consists of all RBFs built on the radial function φ with the addition of a polynomial of degree m at most. The "bumpiness" measure of an interpolant s in this family has been defined as $|\boldsymbol{\lambda}^T\mathbf{\Phi}\boldsymbol{\lambda}|$, where

$\boldsymbol{\lambda}$ is obtained solving the linear system (3.21):

$$\begin{aligned} &\min(-1)^{m_\varphi}\boldsymbol{\lambda}^T\boldsymbol{\Phi}\boldsymbol{\lambda} \\ &\qquad s(\mathbf{x}_j) = f_j \qquad\qquad \forall j = 1,k, \\ &\qquad s(\cdot) \in \mathcal{F}_{\varphi,m}. \end{aligned}$$

If noise is present, a similar optimization problem can be formulated as

$$\begin{aligned} &\min\eta(-1)^{m_\varphi}\boldsymbol{\lambda}^T\boldsymbol{\Phi}\boldsymbol{\lambda} + (1-\eta)\|\epsilon\|_2^2 \\ &\qquad s(\mathbf{x}_j) = f_j + \epsilon_j \qquad\qquad \forall j = 1,k, \\ &\qquad \epsilon \in \mathbb{R}^k, \\ &\qquad s(\cdot) \in \mathcal{F}_{\varphi,m}. \end{aligned}$$

In this approach a balance is sought between the goodness measure of s and its fitting quality. The model permits approximation in place of exact fitting, but large approximation errors are discouraged through the inclusion of a penalty term in the objective, whose importance is regulated by means of a parameter $\eta \in [0,1]$. In what follows, in order to simplify the presentation, it will be assumed that an approximation is chosen in $\mathcal{F}_{\varphi,-1}$, i.e., no polynomial is added to the definition of the approximation function. Moreover, it is assumed that matrix $\boldsymbol{\Phi}$ in (3.21) is invertible; for the general case we refer the reader to (Jakobsson et al., 2010). The optimization problem thus becomes

$$\begin{aligned} &\min\eta\boldsymbol{\lambda}^T\boldsymbol{\Phi}\boldsymbol{\lambda} + (1-\eta)\epsilon^T\epsilon \\ &\qquad \boldsymbol{\Phi}\boldsymbol{\lambda} = \mathbf{f} + \epsilon. \end{aligned}$$

Solving the equations for $\boldsymbol{\lambda}$,

$$\boldsymbol{\lambda} = \boldsymbol{\Phi}^{-1}(\mathbf{f} + \epsilon),$$

and substituting into the objective function, the following quadratic form in ϵ is obtained:

$$\min_{\epsilon}\eta(\mathbf{f}+\epsilon)^T\boldsymbol{\Phi}^{-1}(\mathbf{f}+\epsilon) + (1-\eta)\epsilon^T\epsilon.$$

This is minimized choosing

$$\epsilon = -\left(\boldsymbol{\Phi}^{-1} + \frac{1-\eta}{\eta}\mathbf{I}\right)^{-1}\boldsymbol{\Phi}^{-1}\mathbf{f},$$

where it can be observed that, thanks to the positive definiteness of $\boldsymbol{\Phi}$, all matrix inversions in the above formula are well defined. Thus it has been shown that it is possible to take into account errors in function evaluation simply by substituting the interpolation of the observed values $\mathbf{f}$ with the interpolation of perturbed values

$$\left(\mathbf{I} - \left(\boldsymbol{\Phi}^{-1} + \frac{1-\eta}{\eta}\mathbf{I}\right)^{-1}\boldsymbol{\Phi}^{-1}\right)\mathbf{f}.$$

Very similar, although only marginally more complicated, computations enable us to find the optimal error estimates for general function spaces $\mathcal{F}_{\varphi,m}$.

The value of η in the above formula can be considered as a parameter to be chosen in order to balance the requirements of low bumpiness and of accurate representation of the observations. It is also possible to estimate η through a cross-validation procedure, e.g., by sequentially leaving one observation out of the sample and estimating its value with different choices of η. The "optimal" η is then chosen as the one which gives the minimum possible error estimate in this cross-validation procedure.

Finally, it is important to recall that a very general and powerful regression tool is now available thanks to the developments in support vector machines classification.

3.2.5 Merit functions and sampling

RBF-based global optimization based on extended samples

Some important references for GO methods based on radial basis interpolation and their bumpiness measure are (Gutmann, 2001a, 2001b; Jones, 2001b). Most sample-based methods just require an interpolation or an approximation to be built and base their decision directly on the information conveyed by the model. In some cases these methods make use of an aspiration level $\hat{f}$, but they do not force the model to include a new point in the sample whose value is $\hat{f}$. On the contrary, extended sample-based models try to extrapolate, on the basis of the current sample as well as the level $\hat{f}$, a new model which takes into account this information. This section will present some details on how to include this value in an RBF model within a GO context. Gutmann in his research proposes the use of radial basis interpolation to build GO methods based on sample data and aspiration levels. The theory and methods which enable us to build a radial basis interpolant can be extended to include the aspiration level and lead to a quite elegant method. The implementation details of this method are not trivial, and this might be the reason for its quite limited application; in the recent survey by (Forrester & Keane, 2009) Gutmann is correctly referenced, but nothing is said about his bumpiness measure, which is his main contribution. Let, as in Section 3.2.1, φ be a radial basis function of order m_φ, and let $m \geq m_\varphi - 1$ be the degree of the polynomial used for the interpolation:

$$s(\mathbf{x}) = s(\mathbf{x}|S_k) = \sum_{j=1}^{k} \lambda_j \varphi(\|\mathbf{x} - \mathbf{x}^j\|) + \sum_{i=1}^{\hat{m}} c_i p_i(\mathbf{x}),$$

where $S_k = \{\mathbf{x}^j, f_j\}_{j=1}^k$ is the observed sample, $\hat{m} = \binom{m+n}{n}$ is the dimension of the basis for polynomials of degree at most m, and the coefficients $\boldsymbol{\lambda}, \mathbf{c}$ are the unique solution, under suitable conditions, of the linear system

$$\begin{bmatrix} \Phi & \mathbf{P} \\ \mathbf{P}^T & \mathbf{O} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \mathbf{c} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix}.$$

Let an aspiration level $\hat{f}$ be chosen and the pair $(\hat{\mathbf{x}}, \hat{f})$ be symbolically added to the current sample.

Now let $\hat{\mathbf{x}} = \mathbf{x}^{k+1}$ be an *unknown* evaluation point. Then (3.24) gives an implicit expression for the coefficients of the radial basis interpolation through $(\hat{\mathbf{x}}, \hat{f})$. Following the theory presented in Section 3.2.2, a good choice for $\mathbf{x}^{k+1}$ is the point $\hat{\mathbf{x}}$ which minimizes the bumpiness measure (3.26) of the new interpolant

$$\begin{aligned}&\langle s(\mathbf{x}|S_k \cup \{\hat{\mathbf{x}}, \hat{f}\}), s(\mathbf{x}|S_k \cup \{\hat{\mathbf{x}}, \hat{f}\})\rangle \\ &\quad = \langle s(\mathbf{x}|S_k), s(\mathbf{x}|S_k)\rangle + 2(\hat{f} - s(\hat{\mathbf{x}}|S_k))\langle s(\mathbf{x}|S_k), \mathcal{L}_{k+1}(\mathbf{x};\hat{\mathbf{x}})\rangle \qquad (3.40)\\ &\quad\quad + (\hat{f} - s(\hat{\mathbf{x}}|S_k))^2 \langle \mathcal{L}_{k+1}(\mathbf{x};\hat{\mathbf{x}}), \mathcal{L}_{k+1}(\mathbf{x};\hat{\mathbf{x}})\rangle. \qquad (3.41)\end{aligned}$$

Notice that in the above expression we have explicitly represented the dependence of $\mathcal{L}_{k+1}$ on the unknown new observation $\hat{\mathbf{x}}$. The first term in the above expression is a constant with respect to $\hat{\mathbf{x}}$; the second term is also constant, as

$$\begin{aligned}&\langle s(\mathbf{x}|S_k), \mathcal{L}_{k+1}(\mathbf{x};\hat{\mathbf{x}})\rangle \\ &\quad = (-1)^{m_\phi} \sum_{j=1}^{k} \lambda_j \mathcal{L}_{k+1}(\mathbf{x}^j;\hat{\mathbf{x}}) = 0\end{aligned}$$

thanks to (3.23).

Thus, searching for the global minimizer of the bumpiness measure is equivalent to looking for an $\hat{\mathbf{x}}$ which minimizes

$$(\hat{f} - s(\hat{\mathbf{x}}|S_k))^2 \mu_k(\hat{\mathbf{x}}), \qquad (3.42)$$

where

$$\mu_k(\hat{\mathbf{x}}) = \langle \mathcal{L}_{k+1}(\mathbf{x};\hat{\mathbf{x}}), \mathcal{L}_{k+1}(\mathbf{x};\hat{\mathbf{x}})\rangle \qquad (3.43)$$

represents the square of the seminorm of the binary valued interpolant through S_k and $\hat{\mathbf{x}}$. Recalling (3.26), it holds that

$$\mu_k(\hat{\mathbf{x}}) = (-1)^{m_\varphi} \sum_{i=1}^{k+1}\sum_{j=1}^{k+1} \lambda_i \lambda_j \begin{bmatrix} \mathbf{\Phi} & \boldsymbol{\phi}_{k+1}(\hat{\mathbf{x}}) \\ \boldsymbol{\phi}_{k+1}^T(\hat{\mathbf{x}}) & \varphi(0) \end{bmatrix}_{ij}.$$

Denoting by $\boldsymbol{\alpha}$ the first k components of $\boldsymbol{\lambda}$ and by β the last component associated with the new point $\hat{\mathbf{x}}$, we have

$$\begin{aligned}\mu_k(\hat{\mathbf{x}}) &= (-1)^{m_\varphi} \left(\sum_{i=1}^{k}\sum_{j=1}^{k} \alpha_i(\hat{\mathbf{x}})\alpha_j(\hat{\mathbf{x}})\varphi(\|\mathbf{x}^i - \mathbf{x}^j\|) \right.\\ &\quad \left. + 2\sum_{j=1}^{k} \alpha_j(\hat{\mathbf{x}})\beta(\hat{\mathbf{x}})\varphi(\|\mathbf{x}^j - \hat{\mathbf{x}}\|) + \beta(\hat{\mathbf{x}})^2\varphi(0) \right) \\ &= (-1)^{m_\varphi} \left(\sum_{j=1}^{k} \alpha_j(\hat{\mathbf{x}})\mathcal{L}_{k+1}(\mathbf{x}^j;\hat{\mathbf{x}}) + \beta(\hat{\mathbf{x}})\mathcal{L}_{k+1}(\hat{\mathbf{x}};\hat{\mathbf{x}}) \right).\end{aligned}$$

Recalling that

$$\begin{aligned} \mathcal{L}_{k+1}(\hat{\mathbf{x}};\mathbf{x}^j) &= 0, \qquad\qquad j = 1,k, \\ \mathcal{L}_{k+1}(\hat{\mathbf{x}};\hat{\mathbf{x}}) &= 1, \end{aligned}$$

the squared seminorm is just

$$(-1)^{m_\varphi}\beta(\hat{\mathbf{x}}).$$

Thus, to compute the bumpiness measure it is sufficient to solve system (3.24) for β, which, by direct application of Cramer's rule, turns out to be

$$\begin{aligned} \mu_k(\hat{\mathbf{x}}) &= (-1)^{m_\varphi} \frac{\det\begin{pmatrix} \Phi & \mathbf{0} & \mathbf{P} \\ \boldsymbol{\phi}_{k+1}^T(\hat{\mathbf{x}}) & 1 & \boldsymbol{\pi}_{k+1}^T(\hat{\mathbf{x}}) \\ \mathbf{P}^T & \mathbf{0} & \mathbf{O} \end{pmatrix}}{\det\begin{pmatrix} \Phi & \boldsymbol{\phi}_{k+1}(\hat{\mathbf{x}}) & \mathbf{P} \\ \boldsymbol{\phi}_{k+1}^T(\hat{\mathbf{x}}) & \varphi(0) & \boldsymbol{\pi}_{k+1}^T(\hat{\mathbf{x}}) \\ \mathbf{P}^T & \boldsymbol{\pi}_{k+1}(\hat{\mathbf{x}}) & \mathbf{O} \end{pmatrix}} \\ &= (-1)^{m_\varphi} \frac{\det\begin{pmatrix} \Phi & \mathbf{P} \\ \mathbf{P}^T & \mathbf{O} \end{pmatrix}}{\det\begin{pmatrix} \Phi & \boldsymbol{\phi}_{k+1} & \mathbf{P} \\ \boldsymbol{\phi}_{k+1}^T & \varphi(0) & \boldsymbol{\pi}_{k+1}^T \\ \mathbf{P}^T & \boldsymbol{\pi}_{k+1} & \mathbf{O} \end{pmatrix}}. \end{aligned} \tag{3.44}$$

In (3.44) the numerator is a constant, independent of $\hat{\mathbf{x}}$, while it is easy to see that the denominator tends to vanish as $\hat{\mathbf{x}} \to \mathbf{x}^j$ for any $j = 1,k$; thus the bumpiness diverges to $+\infty$ close to observation points. Moreover, this quantity, being a seminorm, is always nonnegative. Function $\mu_k(\hat{\mathbf{x}})$ is continuous everywhere except at $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^k$.

Observe also that if for some $\hat{\mathbf{x}} = \tilde{\mathbf{x}}$ it happens that $\hat{f} - s(\hat{\tilde{\mathbf{x}}}|S_k) = 0$, then in (3.41) all terms depending on $\hat{\mathbf{x}}$ vanish, so that the bumpiness measure is constant and equal to the same measure computed before the addition of the new point $\tilde{\mathbf{x}}$. As the bumpiness measure after the addition of $\hat{\mathbf{x}}$ cannot be lower than the bumpiness measure before this addition,

$$\begin{aligned} \langle s(\mathbf{x}|S_k \cup \{\hat{\mathbf{x}}, \hat{f}\}), s(\mathbf{x}|S_k \cup \{\hat{\mathbf{x}}, \hat{f}\})\rangle &= \langle s(\mathbf{x}|S_k), s(\mathbf{x}|S_k)\rangle + (\hat{f} - s(\hat{\mathbf{x}}|S_k))^2 \mu_k(\hat{\mathbf{x}}) \\ &\geq \langle s(\mathbf{x}|S_k), s(\mathbf{x}|S_k)\rangle, \end{aligned}$$

then, if $\hat{f}$ is chosen in such a way that

$$\exists \tilde{\mathbf{x}} : s(\tilde{\mathbf{x}}|S_k) = \hat{f},$$

i.e.,

$$\hat{f} \in [\min_{\hat{\mathbf{x}}} s(\hat{\mathbf{x}}|S_k), \max_{\hat{\mathbf{x}}} s(\hat{\mathbf{x}}|S_k)],$$

the new bumpiness is minimized by choosing $\hat{\mathbf{x}} = \tilde{\mathbf{x}}$. This should be avoided because, in this case, the predicted interpolant will not change, and *every* point in which $s(\mathbf{x}) = \hat{f}$

would minimize the bumpiness measure. In other words, if the aspiration level is already attained by the interpolant, it is only too natural to choose the point at which it is attained as the next observation. This should be avoided, particularly when the aspiration level is equal to a previously observed function value; in this case the optimal location of the next observation coincides with a sample point. On the contrary, if the target $\hat{f}$ is chosen as a strict lower bound of $s(\hat{\mathbf{x}}|S_k)$, it is guaranteed that the point which minimizes the bumpiness will be different from all sampled points.

A consequence of the above developments can thus be summarized in the following.

Theorem 3.9. *Consider the GO problem*

$$\min_{x\in S\subset\mathbb{R}^n} f(x),$$

where the feasible set S is compact. Let φ be a radial basis function of order m_φ and $m \geq m_\varphi - 1$. If, after the evaluation of f at k points $\mathbf{x}^1,\mathbf{x}^2,\dots,\mathbf{x}^k$,

1. *the points $\{\mathbf{x}^1,\mathbf{x}^2,\dots,\mathbf{x}^k\}$ form a unisolvent set;*
2. *a reference value $\hat{f}$ is chosen in such a way that*
$$\hat{f} < \min_{\hat{\mathbf{x}}\in S} s(\hat{\mathbf{x}}|S_k); \text{ and}$$
3. *$\mathbf{x}^{k+1}$ is chosen as*
$$\begin{aligned}\mathbf{x}^{k+1} &\in \arg\min_{\hat{\mathbf{x}}\in S}(\hat{f} - s(\hat{\mathbf{x}}|S_k))^2\mu_k(\hat{\mathbf{x}})\\ &= \arg\min_{\hat{\mathbf{x}}\in S}(-1)^{m_\varphi}(\hat{f} - s(\hat{\mathbf{x}}|S_k))^2 \det\begin{pmatrix}\Phi & \mathbf{P}\\ \mathbf{P}^T & \mathbf{O}\end{pmatrix}/\det\begin{pmatrix}\Phi & \boldsymbol{\phi}_{k+1} & \mathbf{P}\\ \boldsymbol{\phi}_{k+1}^T & \varphi(0) & \boldsymbol{\pi}_{k+1}^T\\ \mathbf{P}^T & \boldsymbol{\pi}_{k+1} & \mathbf{O}\end{pmatrix},\end{aligned}$$

then the point $\mathbf{x}^{k+1}$ is well defined and distinct from all points $\mathbf{x}^1,\mathbf{x}^2,\dots,\mathbf{x}^k$.

The above theorem simply requires that the reference value $\hat{f}$ be small enough; thus it might be of interest to check what happens when this value diverges to $-\infty$. If $\mathbf{x}^{k+1}$ is chosen according to the preceding theorem, then

$$(\hat{f} - s(\mathbf{x}^{k+1}|S_k))^2\mu_k(\mathbf{x}^{k+1}) \leq (\hat{f} - s(\mathbf{y}|S_k))^2\mu_k(\mathbf{y}) \qquad \forall\mathbf{y}\in S,$$

or

$$\mu_k(\mathbf{x}^{k+1}) \leq \frac{(\hat{f} - s(\mathbf{y}|S_k))^2}{(\hat{f} - s(\mathbf{x}^{k+1}|S_k))^2}\mu_k(\mathbf{y}) \qquad \forall\mathbf{y}\in S.$$

Letting $\hat{f} \to -\infty$, we obtain

$$\mu_k(\mathbf{x}^{k+1}) \leq \mu_k(\mathbf{y}) \qquad \forall\mathbf{y}\in S.$$

Thus, in Theorem 3.9, it is possible to add the specification that if $\hat{f}$ is chosen equal to $-\infty$, then the new evaluation point should be defined as

$$\mathbf{x}^{k+1} \in \arg\min_{\hat{\mathbf{x}}\in S}\mu_k(\hat{\mathbf{x}}).$$

As a general remark, we may comment at this point that the aspiration level $\hat{f}$ acts as a parameter which regulates the balance between local and global exploration. In fact, choosing $\hat{f} \approx \min_{j=1,k} f_j$ leads to a point $\mathbf{x}^{k+1}$ which will be close to the current record, thus performing some sort of local search around it. Choosing $\hat{f} \to -\infty$, on the contrary, will lead to a point $\mathbf{x}^{k+1}$ which is as far as possible from all other observations, thus leading to a global search in the most unexplored region, somewhat independently on the observed function values.

From the point of view of practical implementation of the method described in this section, having to deal with the GO of a function like (3.42) is likely to cause difficulties, as this function is undefined at $\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^k$. It is thus more sensible, e.g., to maximize its inverse

$$\frac{1}{(\hat{f} - s(\hat{\mathbf{x}}|S_k))^2 \mu_k(\hat{\mathbf{x}})},$$

which can be proven to be continuous on all the feasible set S. Moreover, $1/\mu_k(\hat{\mathbf{x}})$ can be expressed as

$$(-1)^{m_\varphi} \left(\varphi(0) - \begin{bmatrix} \boldsymbol{\phi}_{k+1}(\hat{\mathbf{x}}) & \boldsymbol{\pi}_{k+1}(\hat{\mathbf{x}}) \end{bmatrix}^T \begin{bmatrix} \Phi & \mathbf{P} \\ \mathbf{P}^T & \mathbf{O} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{\phi}_{k+1}(\hat{\mathbf{x}}) \\ \boldsymbol{\pi}_{k+1}(\hat{\mathbf{x}}) \end{bmatrix} \right).$$

The proof of the validity of this expression is quite elementary and can be found in (Gutmann, 2001b). Using this definition the method thus prescribes finding the maximizers of the inverse merit function or, equivalently, solving the GO problem

$$\max_{\hat{\mathbf{x}} \in S, \hat{\phi}, \hat{\pi}} \frac{(-1)^{m_\varphi} \left(\varphi(0) - \begin{bmatrix} \boldsymbol{\phi}_{k+1}(\hat{\mathbf{x}}) & \boldsymbol{\pi}_{k+1}(\hat{\mathbf{x}}) \end{bmatrix}^T \begin{bmatrix} \hat{\boldsymbol{\phi}} \\ \hat{\boldsymbol{\pi}} \end{bmatrix} \right)}{(\hat{f} - s(\hat{\mathbf{x}}|S_k))^2}$$

$$\begin{bmatrix} \boldsymbol{\Phi} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{O} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\phi}} \\ \hat{\boldsymbol{\pi}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\phi}_{k+1}(\hat{\mathbf{x}}) \\ \boldsymbol{\pi}_{k+1}(\hat{\mathbf{x}}) \end{bmatrix}. \tag{3.45}$$

Clearly it is not necessary to solve this problem as a constrained optimization problem, as the explicit constraint (3.45) can be used implicitly, through the solution of a linear system with constant coefficient matrix and a right-hand side which varies as a function of $\hat{\mathbf{x}}$; thus, it might be worth devoting some computational time to transforming this linear system in such a way as to ease its repeated solution. A possibility is performing a Cholesky decomposition of the coefficient matrix; other possibilities, developed in order to keep the solution of the system numerically stable, require some form of preconditioning in order to lower the condition number of the matrix which, otherwise, for large k usually tends to be quite high. Finally, another possibility is using some iterative technique, possibly stopping the iterations soon enough. Indeed, it should be recalled that devoting too much effort to the exact estimation of the parameters of a surely approximate model might be a waste; quite often, especially when the cardinality of the sample grows, a few steps with an iterative method will give a sufficiently good model on which decisions might be based.

Recent literature on RBF interpolation might be used for GO in different ways. A possibility is that of exploiting the local nature of RBF interpolation. This means that, although the support of each RBF is in general the whole $\mathbb{R}_+$, the influence of radial bases centered at some point far from the point at which the model is evaluated is usually small. Thus some recent approaches try to exploit this fact through so-called fast multipole methods, in which the radial basis interpolant at a point is decomposed into the sum of contributions of close samples and a sort of an averaged sum of contributions from points which are sufficiently far away. This stream of research has not yet seen applications in the GO context, possibly because the GO problems which can be solved by means of surrogate-based models are so expensive that, in practice, no run will require more than a few hundred function evaluations.

Merit functions for RBF-based optimization

All of the developments introduced in the preceding section lead to a family of merit functions $\mathcal{M}$ which can be used for driving the search for the next observation $\mathbf{x}_{k+1} \in \arg\min_{\hat{\mathbf{x}} \in S} \mathcal{M}(\hat{\mathbf{x}})$. The choices made available by the analysis of RBF interpolation are as follows:

1. $\mathcal{M}(\hat{\mathbf{x}}) = s(\hat{\mathbf{x}}|S_k)$. This choice is purely local; the merit function corresponds to trusting the model, and, thus, the next evaluation point for the real objective function is chosen as the global minimizer of the interpolation model. This gives rise to a sample-based criterion; it might seem that this criterion is to be confined to later stages of the optimizations, when few more observations are to be done and it is more reasonable to refine the search around good points rather than exploring new parts of the feasible domain. However, although the intensification of local refinement is indeed a good strategy in some parts of a GO algorithm, it seems very reasonable, as suggested in (Holmström, 2008), to use this criterion when there is a big discrepancy between the best observation in the sample and the minimum of the interpolant, i.e., when the quantity

   $$\Delta_k = \min_{j \in 1,k} f_j - \min_{\hat{x} \in S} s(\hat{x}|S_k)$$

   is large. In fact, in this case there seems to be a large discrepancy between the model and the fitted response surface, particularly concerning the estimate of the global minimum. Thus it seems more reasonable to place observations at the minimum of the interpolant. This will have one of two possible effects: either the model is a good predictor and the observed function value will be a good one, leading to an improved estimation of the global minimum, or the global minimum of $s()$ will not correspond to a low value for f. In the latter case the next interpolation will take this into consideration, and the error in the model will be in general reduced.

   Using this strategy, of course, introduces a new parameter which has to be suitably calibrated, namely, the largest admissible Δ_k. This can be seen as a parameter which regulates, again, the balance between exploratory and refinement searches.

2. $\mathcal{M}(\hat{\mathbf{x}}; \hat{f}) = (\hat{f} - s(\hat{\mathbf{x}}|S_k))^2 \mu_k(\hat{\mathbf{x}})$. This is the natural choice coming from the desire to minimize the bumpiness and, thus, to place the next observation where it seems reasonable, if we assume that the true objective function does not oscillate too much. This choice is preferred when dealing with expensive objective functions for which a quite trustable surrogate model has been built. By choosing a target value which is strictly lower than the minimum of the interpolant, a balance between exploration

and refinement is automatically obtained. In fact, when the target value is close enough to the interpolation minimum, the effect will be similar to that obtained with the preceding merit function; thus new observations will tend to cluster around the best already observed. After a quite thorough exploration of the region around the best observation in the sample, however, keeping the target below the minimum of the interpolant will lead us to explore different regions. A very low estimate $\hat{f}$ will drive the search toward parts of the feasible region which are as far as possible from those already explored.

The choice of $\hat{f}$ at each iteration is quite critical and is linked to the choice of a suitable balance between exploration and refinement. Recent tendencies in this framework do not prescribe a fixed choice but favor the analysis of more than one possible choice. If it were possible to track the trajectory of the optimal choice

$$\hat{\mathbf{x}}^\star = \hat{\mathbf{x}}^\star(\hat{f}) \in \arg\min_{\hat{\mathbf{x}} \in S} (\hat{f} - s(\hat{\mathbf{x}}|S_k))^2 \mu_k(\hat{\mathbf{x}})$$

when $\hat{f}$ is changed continuously from $\min_{\mathbf{x} \in S} s(\mathbf{x}|S_k)$ to $-\infty$, it should be clear that this trajectory will be in general discontinuous, with points clustered in a finite number of connected regions. One of these regions corresponds to local exploration around the current record point; others correspond to regions which are far from all points in the sample. The idea, proposed in (Jones, 2001b) and further analyzed in (Holmström, 2008), is then to sample this trajectory by solving a family of optimization problems, one for each of a finite grid of possible values for $\hat{f}$ in the merit function. Then, by employing some sort of a clustering technique, a few significant points can be chosen and the objective function can be evaluated, possibly exploiting parallel computation. This stream of research is quite promising, and further developments might be expected in this direction.

3. $\mathcal{M}(\hat{\mathbf{x}}) = \mu_k(\hat{\mathbf{x}})$. This case, which is a limiting case of the preceding one obtained by letting $\hat{f} \downarrow -\infty$, is an extreme choice in which the sample is partially disregarded, in the sense that only the location of function observations is taken into strong consideration, while function values become less important. By using this merit function, a purely exploratory move is performed. Sometimes this is desirable or even necessary when, because of numerical instability, no candidate solution can be found by other means. However, it is easy to observe that when the sample size is relatively small, optimizing this merit function will typically lead to a point on the border of the feasible region. This is often a bad choice, as usually the feasible set S is just a box whose extremes are exactly "extreme" values for the variables. As an example, we can refer to many cases in which the black box to be optimized is the outcome of a numerical simulation code which depends on some parameters: in these cases, parameters have to be calibrated in such a way that the simulated output is as close as possible to real measurements obtained from the system we wish to simulate. Usually the bounds on these parameters correspond to values which are not expected to be found in an optimal configuration but are just set to delimit the search space and avoid unreasonable parameter choices. This means that, often, the global optimum is assumed to be in the interior of S, often quite far from the boundary. Thus, as soon as the dimension increases, a large number of observations will be uselessly put near the extreme points of the feasible set if this merit function is frequently called.

Merit functions derived from Kriging

When an RBF interpolation is built using the Kriging kernel, as described in Section 3.2.3, it is possible to derive alternative merit functions based on the statistical origin of the method. It has already been observed that the statistical motivation behind Kriging is, in most cases, not motivated when dealing with the optimization of deterministic objective functions. However, methods derived by exploiting the statistical model of the objective function give rise to merit functions which allow us to automatically balance global and local search, similar to those seen previously and, thus, deserve interest in developing GO algorithms. All of the derivations in this section follow directly from the developments on the Kriging estimator and, in particular, the observation that, conditionally on the sample, the function value at a new point $\mathbf{x}$ is a normal random variable with mean (3.32) and variance (3.37).

The best known merit function for GO based on Kriging is that proposed in (Jones, Schonlau, & Welch, 1998), which was previously introduced by Mockus in the 1970s, and recalled in (Mockus, Eddy, & Reklaitis, 1996). Given a statistical model for the objective function, it seems worthwhile to place the next observation where the *expected improvement* is maximized. In formulae, the merit function used is defined as

$$\mathcal{M}(\hat{\mathbf{x}}) = E(\max(f_k^\star - f(\hat{\mathbf{x}}),0)), \tag{3.46}$$

where $f_k^\star = \min_{j=1,k}\{f_j\}$ is the current record and E denotes expectation with respect to a suitable probability distribution function. In general, the so-called *Bayesian approach* to GO prescribes the use of the conditional, a posteriori, distribution function given the observed sample. For Kriging this means that the probability distribution which has to be used in (3.46) is the normal distribution with mean (3.32) and variance (3.35). It has been reported in (Jones et al., 1998) that for the Kriging interpolation, (3.46) can be computed as

$$\mathcal{M}(\hat{\mathbf{x}}) = (f_k^\star - \hat{f}(\mathbf{x}))P\left(Z \le \frac{f_k^\star - s(\mathbf{x})}{\hat{\sigma}(\mathbf{x})}\right) + \hat{\sigma}(\mathbf{x})\frac{1}{\sqrt{2\pi}}\exp -\frac{1}{2}\left(\frac{f_k^\star - s(\mathbf{x})}{\hat{\sigma}(\mathbf{x})}\right)^2, \tag{3.47}$$

where Z represents a standard normal random variable.

This merit function has to be maximized and this, once again, is a nonconvex GO problem which has the added inconvenience of not being representable in explicit form due to the presence of a term related to the Gaussian probability distribution function (pdf). However, there exist many very accurate numerical approximations for the normal pdf, so that merit function (3.47) can be reasonably approximated. But it should be evident that the use of Kriging, while providing very accurate predictions, is computationally very demanding, as after each evaluation of the objective function two different GO problems have to be solved, one in order to find the maximum likelihood estimates of the Kriging parameters, and the other to choose the best location of the next evaluation point. On the positive side, it can be observed that criterion (3.47) takes into account in a clever way both the expected function value $\hat{f}(\mathbf{x})$ and the uncertainty $\hat{\sigma}(\mathbf{x})$ around this value. Recalling that Kriging provides an exact interpolation, it is readily seen that the variance of the Kriging predictor is null at each sample point, while it increases far from sample points. Thus in regions which are unexplored the variance tends to be high; in formula (3.47) a balance is sought between looking for points in which the expected improvement $f_k^\star - \hat{f}(\mathbf{x})$ is high and points in which the uncertainty is large. This way a GO algorithm based on the maximization of the expected improvement regulates itself switching from local approximation phases to global exploration phases, in the very spirit of well-designed GO methods. In the literature different suggestions are given on how to optimize this merit function, ranging from simple Multistart to more expensive exact branch-and-bound schemes. Like any other method in this chapter, it should be observed that, lacking any kind of knowledge on the objective function, there is no point in exaggerating the precision of optimization in this phase, so that any reasonably accurate heuristic coupled with efficient local search should be preferred to expensive exact optimization.

Another merit function which is used in a Kriging framework is that of choosing the next point where the probability, instead of the expected value, of improvement over the aspiration level $\hat{f}$ is maximized:

$$\mathcal{M}(\hat{\mathbf{x}}) = \frac{1}{\sqrt{2\pi}\hat{\sigma}(\hat{\mathbf{x}})} \int_{-\infty}^{\hat{f}} \exp\left(-\frac{(\mathbf{y} - s(\hat{\mathbf{x}}))^T(\mathbf{y} - s(\hat{\mathbf{x}}))}{2\hat{\sigma}^2(\hat{\mathbf{x}})} \right) d\mathbf{y}.$$

This method, although not born within what is now known as Kriging, was already proposed by (Kushner, 1964), who can be considered the pioneer in the use of stochastic models for GO.

Finally, as an extreme choice, the new point might also be chosen where there is maximum uncertainty in order to improve the approximation of the objective function. In this case it is natural to choose

$$\mathcal{M}(\hat{\mathbf{x}}) = \hat{\sigma}^2(\mathbf{x}),$$

but this choice significantly disregards sample values and tends to just favor global exploration.

3.3 Convergence

By definition heuristic methods do not give any guarantee about the quality of the solution returned. However, if stopping rules are removed and the heuristics are run for an infinite number of iterations, we might wonder under which conditions convergence to the globally optimal solutions is guaranteed. The complexity issues discussed in Chapter 2 have already clarified that GO is a difficult task and the detection of global optima, even for classes of highly structured problems (such as those with a quadratic objective function and box constraints), might require a computation time which increases exponentially with the dimension of the problems. In this section we will show that for "poorly structured" problems the situation is even worse. In particular, we will show that in most cases the unique possibility to "solve" a GO problem is to generate an everywhere dense set of solutions.

Despite the very large quantity and diversity of heuristic methods for GO, the theoretical results on their convergence properties are usually either quite elementary or, often, neglected in the literature. One reason for the relatively scarce attention to convergence results for heuristic techniques might be the quite depressing consideration of the "practical impossibility" of GO. By this we refer to the commonly used argument that, given any algorithm, after the algorithm has been stopped, it is in general possible to build an artificial objective function which, if given as an input to the same algorithm, would cause the algorithm to generate the same sequence of points and to stop with an arbitrarily large error in the location and value of the global optimum.

A simple example of this line of reasoning is, for example, the case in which the algorithm is based only on observed function values and is deterministic. Let us assume for simplicity that the global optimum of a function f of a single variable $x \in \mathbb{R}$ is sought over a nonempty interval $[a,b] \subset \mathbb{R}$. Assume that, after stopping, the algorithm produced the set of observations

$$(x_1, f_1), (x_2, f_2), \ldots, (x_N, f_N)$$

with $f_i = f(x_i)$ for all i. It is very easy to build a smooth function ϕ such that

$$\phi(x_i) = f_i \qquad\qquad \forall i \in 1, \ldots, N.$$

Moreover, given any $\overline{x} \in [a,b]$ such that $\overline{x} \neq x_i$ for all i, and any real constant M, it is very easy to constrain ϕ to interpolate the value M at $\overline{x}$. This way, the same algorithm, once run on function ϕ, will produce exactly the same sequence of iterates and will stop at iteration N, yet the best observed value

$$\min_{i=1,\ldots,N} f_i$$

will be arbitrarily far from M. A simple example for function ϕ is a polynomial interpolant of degree N. Thus it seems that GO is inherently intractable, and no finite algorithm can be designed which guarantees a fixed precision on the estimate of the global optimum. On the other hand, when the class of objective functions or constraints satisfies some additional requirements, finite convergence can be proven and error estimates can be made available. Therefore, the ability to detect globally optimal solutions, or, at least, solutions sufficiently close to the global ones, is strictly related to the structure of the class of problems at hand. In this section we would like to make more precise the definition of "poorly structured" classes of problems and show that for them the unique possibility to observe the global optimum is to generate a dense sequence.

An instance of an *optimization problem* will be given as a pair (f, S), where $S \subseteq \mathbb{R}^n$ is a set and f is a function defined over S. It is understood that the direction of optimization is that of minimization of f over S. These definitions, as well as most of this section, were inspired by (Baritompa & Stephens, 1998).

Definition 3.10. *A set $\mathcal{P}$ of optimization problems is a* sufficiently rich class of problems *if, for every $(f, S) \in \mathcal{P}$,*

- *S is a compact set;*
- *f is continuous over S; and*
- *for all $\bar{\mathbf{x}} \in S$, for each open neighborhood $\mathcal{N}(\bar{\mathbf{x}})$, for every real value M, there exists a function g such that*
  - $(g, S) \in \mathcal{P}$,
  - $g(\mathbf{x}) = f(\mathbf{x})$ *for all* $\mathbf{x} \in S : \mathbf{x} \notin \mathcal{N}(\bar{\mathbf{x}})$,
  - $g(\bar{\mathbf{x}}) = M$.

This definition says that a sufficiently rich class of problems is characterized by the fact that, given a problem, we can always find another problem in the same class so that the two problems are identical everywhere, except in a small neighborhood of a generic feasible point, where the two objective functions may arbitrarily differ. As an example, the class of optimization problems where continuous functions are minimized over a compact set S is sufficiently rich, but also the optimization of $\mathcal{C}^1(S)$ or even $\mathcal{C}^\infty(S)$ functions corresponds to sufficiently rich classes; piecewise linear minimization over an interval is a sufficiently rich class. The minimization of Lipschitz functions over compact sets is another example of a sufficiently rich class, but as soon as we restrict the class by imposing a finite upper bound for the Lipschitz constant, the class is no longer sufficiently rich. The class of continuous function minimization *possessing a unique global minimum* over a compact set is a sufficiently rich class, while a similar class that is characterized by having a unique *local (and, thus, global)* minimum is not. This latter class contains hidden global information. The class of problems where f is a concave function and S a compact convex set is not sufficiently rich.

In many papers algorithms are classified on the basis of the "information" they possess on the problem. In particular, many authors suggest that if some "global" information is available, such as knowing that f is concave and S is a polyhedron, or knowing the value of an upper bound to the Lipschitz constant of f, then it is possible to build methods which guarantee convergence, even in a finite number of steps, to the global optimum or to an arbitrarily precise estimate of the global optimum. Sufficiently large classes are the natural bases for defining classes of problems for which we cannot assume to know any kind of "global" information. As we will see, when working with a sufficiently large class, all the information available for the definition of the iterates of an algorithm is local, in some sense to be made more precise.

In what follows we formalize the notion of *local information* in such a way that it is compatible with the intuitive notion.

Let $\mathcal{P}$ be a sufficiently rich family of problems, and consider $(f,S) \in \mathcal{P}$; let S^N be the set of all N-sequences of feasible points, and denote by $S^\infty = \bigcup_N S^N$ the set of all *finite* sequences of points in S.

Definition 3.11. *A* local information *is a function* $LocInf = LocInf(f, S^\infty)$ *with range in a generic set such that, choosing any finite sequence* $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \in S^\infty$ *for every open set* $\mathcal{N}$ *such that*

$$\mathbf{x}_i \in \mathcal{N} \qquad \forall i \in 1, \ldots, N,$$

if $(g,S) \in \mathcal{P}$ *is any problem such that*

$$f(\mathbf{x}) = g(\mathbf{x}) \qquad \forall \mathbf{x} \in \mathcal{N},$$

then

$$LocInf(f, \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}) = LocInf(g, \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}).$$

Thus a local information is any kind of information which depends on the knowledge we can collect on an objective function "close" to the points where, as an example, the function has been evaluated. Examples of local information are the set of observed function values or the gradients evaluated at any finite number of points. The current record, i.e., the sampled point with minimum function value, is local information. A quadratic model of the objective function interpolating the function, its gradient, and its hessian at a specific point is local information. The global optimum, however, is not local information, and neither is the global minimum value. This is coherent with the previous discussion: after sampling a finite number of points, a function which is not distinguishable from the objective function "close" to sampled points can be built so that its minimum is arbitrarily far from the global minimum. In fact, if $S = [a,b] \in \mathbb{R}$ is a nonempty interval, given $a \le x_1 < x_2 < \cdots < x_N \le b$, if, for all $i = 1,\ldots,N-1$, $\varepsilon < (x_{i+1} - x_i)/2$, it is trivial to build two continuous functions f and g which are identical in any open set $(x_i - \varepsilon, x_i + \varepsilon)$ yet possess different global minima. Another example of information which is nonlocal is the number of local minima. A function defined on $\mathcal{P}$ which is not local information is called *global information* and is denoted by $GlobInf(f,\{\mathbf{x}_1,\ldots,\mathbf{x}_N\})$.

For instance, if the sufficiently rich class is composed of all Lipschitz-continuous functions, a function which maps the current problem (f,S) to the value of the Lipschitz constant of f over S (or any upper bound of this value) is an example of global information.

It is of interest to distinguish between local and global information because, informally speaking, if an algorithm cannot have access to global information, there is no hope of guaranteeing that it will approximately find the global minimum unless it is run forever and it generates a dense set of observations in the feasible set.

Before stating the main results concerning local and global information, some additional definitions are required.

Without going into more formal and abstract definitions, we define (optimization) algorithms as functions which generate sequences of iterates.

Definition 3.12. *A* deterministic sequential algorithm *on a class of problems $\mathcal{P}$ is a mapping characterized by the fact that, given a problem $(f,S) \in \mathcal{P}$, there exists a local information function $LocInf$ such that, for any choice of $\mathbf{x}_0 \in S$,*

$$\mathbf{x}_{k+1} = \mathbf{x}_{k+1}(LocInf(f,\{\mathbf{x}_0,\mathbf{x}_1,\ldots,\mathbf{x}_k\})) \qquad \forall k \ge 0.$$

Thus, in a deterministic sequential algorithm, each iterate is generated on the basis of the available information, which consists in the local information collected during the execution of the method itself or available a priori. Parallel and population-based algorithms might be defined in a similar way. It is worthwhile to emphasize that according to this definition an algorithm cannot make use of any global information, unless it is a characteristic of the whole class of functions. For example, if an algorithm is applied to a class of Lipschitz-continuous functions, a sequential algorithm cannot be defined in such a way that, if, for a specific function, an upper bound on the Lipschitz constant is known, then it is used. This is an example of global information which is not contained in the whole class of functions and, thus, cannot be used by a sequential algorithm. Or, stated in a probably more familiar way, if a sequential algorithm is built which uses an estimate of the Lipschitz constant, it can again be called a "sequential algorithm" according to the above definition; however, its applicability is limited to a class of functions which is not sufficiently rich, so that the algorithm can be applied only to a set of problems for which some global information is available. In other words, a sequential algorithm should be defined in such a way that its behavior depends on general characteristics of the whole function class to which it is applied; the only exploitable information is local information collected during the execution of the method itself.

Concerning randomized heuristics, we have the following definition.

Definition 3.13. *A* stochastic sequential algorithm *on a class of problems $\mathcal{P}$ is a mapping characterized by the fact that, given a problem $(f, S) \in \mathcal{P}$ and a feasible set $S \subseteq \mathbb{R}^n$, there exists a local information function $LocInf$ such that*

$$\mathbf{x}_{k+1} = \mathbf{x}_{k+1}(LocInf(f, \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_k\}), \omega_{k+1}) \qquad \forall k \geq 0,$$

where $\mathbf{x}_0 \in S$ is given and $\{\omega_k\}$ is a sequence of random numbers whose probability distribution function does not depend on the optimization problem.

If we assume that a sequential algorithm is run forever, without stopping, then we may define the set of limit points of a deterministic algorithm (or of a sample in the case of a stochastic method). If we denote by $X_{f,S}$ the infinite sequence of iterates generated by a sequential method (or by $X_{f,S}(\omega)$, the analogous one for stochastic methods), then we can denote by $X'_{f,S}$ the set of accumulation points of the sequence, while the closure of $X_{f,S}$ will be indicated by $\overline{X}_{f,S}$.

Definition 3.14. *An algorithm* sees the global optimum *of f if*

$$\overline{X}_{f,S} \bigcap X^\star_{f,S} \neq \emptyset,$$

where $X^\star_{f,S}$ is the set of global minima of the problem (f, S).

Definition 3.15. *An algorithm* localizes *the global optima if*

$$X'_{f,S} = X^\star_{f,S}.$$

A weaker definition which can be quite useful is obtained when the above equality is turned into

$$X'_{f,S} \subseteq X^\star_{f,S}.$$

An algorithm which sees the global optimum does not necessarily converge to a global minimum; however, if we keep track of the record, i.e., of the best feasible solution found, an algorithm which sees the global optimum in the limit will produce a record which is optimal. The stronger notion of localizing the global optima is connected with the fact that all limit points of subsequences generated by the algorithm are global optima, and, vice versa, every global minimum is the limit of a subsequence generated by the algorithm. Analogous definitions can be given for stochastic methods.

In (Baritompa & Stephens, 1998), the following theorem was proven.

Theorem 3.16. *A deterministic sequential sampling algorithm sees the global optimum for every problem (f, S) in a sufficiently rich class if and only if*

$$\overline{X}_{f,S} = S.$$

Thus, if the class of problems is sufficiently large, so that it is not possible to use any kind of global information on the problem, then the unique possibility of guaranteeing, in the limit, that one of the global minima has been reached at least once is that the algorithm generates a dense set of observations.

When we go to the stronger requirement of localizing the global minima, the situation is even worse, as we see in the following theorem.

Theorem 3.17. *Given a deterministic sequential algorithm applied to a sufficiently rich class of problems, there always exists a problem* (f, S) *for which the algorithm does not localize the global optima.*

The situation is not much improved when considering stochastic algorithms. In fact, the following results can be proven.

Theorem 3.18. *Given a stochastic sequential algorithm and any problem* (f, S) *in a sufficiently large class, given any* $p \in (0, 1)$*, then the probability that the algorithm sees the global optimum of* (f, S) *is at least* p *if and only if*

$$P(\mathbf{x} \in \overline{X}_{f,S}) \geq p \qquad\qquad \forall \mathbf{x} \in S.$$

Theorem 3.19. *For every stochastic sequential algorithm and* $\varepsilon > 0$*, there exists a problem* (f, S) *such that the probability that the algorithm localizes the global optima is smaller than* ε*.*

These results, whose proofs can be found in (Baritompa & Stephens, 1998), state that only algorithms which generate a dense set of observations are capable of guaranteeing the discovery of the global optimum, unless the algorithm is allowed to make use of some nonlocal information.

Chapter 4

Lower Bounds

4.1 Introduction

While in the previous chapter we discussed strategies to compute upper bounds for a GO problem, here we are now interested in strategies to compute, in a relatively cheap way, lower bounds for the same problem. As we will see in Chapter 5, lower bounds are an essential component of branch-and-bound (BB) approaches to the solution of GO problems. They are usually computed through the solution of a *relaxation*.

Definition 4.1. *Given an optimization problem*

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

a relaxation *is any optimization problem*

$$\min_{\mathbf{x}\in \hat{S}} \hat{f}(\mathbf{x})$$

such that

- $S \subseteq \hat{S}$*;*
- $\mathbf{x} \in S \Rightarrow \hat{f}(\mathbf{x}) \leq f(\mathbf{x})$.

If $\hat{S}$ is a convex set and $\hat{f}$ is convex on $\hat{S}$, then we have a convex relaxation.

If the feasible set S is explicitly defined as

$$S = \{\mathbf{x} \in \mathbb{R}^n \; : \; g_i(\mathbf{x}) \leq 0, \; i = 1,\dots,m\},$$

then a convex relaxation may have the form

$$\min_{\mathbf{x}\in \hat{S}} \hat{f}(\mathbf{x}), \tag{4.1}$$

where $\hat{f}$ is a convex function such that

$$\hat{f}(\mathbf{x}) \leq f(\mathbf{x}) \ \ \forall\, \mathbf{x} \in S,$$

i.e., $\hat{f}$ is a *convex underestimator* for the function f over S, while

$$\hat{S} = \{\mathbf{x} \in \mathbb{R}^n \ : \ \hat{g}_i(\mathbf{x}) \leq 0,\ i = 1,\dots,m\},$$

where, for each $i = 1,\dots,m$, $\hat{g}_i$ is a convex function such that

$$\hat{g}_i(\mathbf{x}) \leq g_i(\mathbf{x}) \ \ \forall\, \mathbf{x} \in S,$$

i.e., the functions $\hat{g}_i$, $i = 1,\dots,m$, are convex underestimators for the functions g_i, $i = 1,\dots,m$, over S. Thus $\hat{S}$ is a *convex outer approximation* of S. Then, the optimal value of the problem (4.1), which is a convex programming problem and can thus be solved, under mild assumptions, in polynomial time, is a lower bound for the optimal value of the GO problem. This is an obvious consequence of the fact that

$$\mathbf{x} \in S \quad \Rightarrow \quad \mathbf{x} \in \hat{S},\ \hat{f}(\mathbf{x}) \leq f(\mathbf{x}).$$

In fact, a convex relaxation can be defined in a more general way. First, what we need is a convex set $\hat{S}$ which outer approximates the feasible region S, and this is not necessarily obtained by replacing the functions g_i with convex underestimators (a different approach will be seen, e.g., in Section 4.14). Moreover, in the above definition we have assumed that the convex relaxation has the same variables as the GO problem. This is also not strictly necessary. A convex relaxation can be defined as

$$\min_{\mathbf{z} \in Z} \bar{f}(\mathbf{z}), \tag{4.2}$$

where Z is a convex set, $\bar{f}$ is a convex function, and the vector of variables $\mathbf{z}$ may (but do not necessarily) include the original variables $\mathbf{x}$. What is required in order to have a convex relaxation is that we are able to associate some point $\mathbf{z} \in Z$ to each $\mathbf{x} \in S$ such that $\bar{f}(\mathbf{z}) \leq f(\mathbf{x})$.

Of course, different relaxations can be defined for a given problem. Such relaxations differ under two respects: the quality of the bound (the higher the bound, the better the quality), and the time needed to compute it (which usually increases with the quality of the bound).

In this chapter we will discuss several issues related to the definition of convex relaxations and, thus, related to the computation of lower bounds. We will move from relaxations for highly structured problems toward those for mildly structured ones. In particular, the structure of the chapter is the following. In Section 4.2 we introduce the notion of best possible convex underestimator (convex envelope), and we discuss different general as well as specific results about this subject. In Section 4.3 we discuss the reformulation-linearization technique, which is employed to derive relaxations for problems with polynomial objective and/or constraint functions. In Section 4.4 we present different reformulations and relaxations for problems with a quadratic objective function and linear or quadratic constraints. In Section 4.5 we do the same for unconstrained problems with a polynomial objective function. In Section 4.6 we move to more general functions by introducing α-BB approaches, which allow us to define convex underestimators for twice-continuously differentiable functions. In Section 4.7 the notion of difference-of-convex decomposition of a function is introduced, and it is shown how this leads to possible relaxations for GO problems. Lower bounding techniques for problems involving Lipschitz functions are briefly discussed in Section 4.9. In Section 4.10 we present lower bounding techniques based on interval arithmetic, which, though usually returning bounds of lower quality with respect to the previously discussed ones, can be applied to a wide range of problems. In Section 4.11 we discuss dual bounds. In Section 4.12 we introduce McCormick relaxations, which allow us to derive convex underestimators for factorable functions by combining convex underestimators of simpler functions. In Section 4.13 we make the practically relevant observation that it is often more convenient to work with polyhedral convex underestimators rather than sharper nonpolyhedral convex estimators. Indeed, the former allow us to compute bounds through the solution of linear programming problems, so that the higher speed and stability of linear programming solvers with respect to nonlinear solvers can be exploited. As already mentioned, in Section 4.14 we will discuss an approach, alternative to the replacement of a constraint function with a convex underestimator, for the definition of a convex outer approximation of the feasible region. Finally, in Section 4.15 we state some concluding remarks about relaxations.

4.2 Convex envelopes

In this section we discuss convex envelopes. We first introduce some definitions and results in Section 4.2.1. Next, we discuss polyhedral (Section 4.2.2) and nonpolyhedral (Section 4.2.3) convex envelopes. In Section 4.2.4 we present techniques to compute the value and a supporting hyperplane of the convex envelope at some point. In Section 4.2.5 we discuss the relation between convex envelopes over polytopes and polyhedral subdivisions. In Section 4.2.6 we consider some results about convex envelopes for sums of functions. Finally, in Section 4.2.7 we cite a few results for convex envelopes of special functions.

4.2.1 Some definitions and results

The *convex envelope* of a lower semicontinuous function f over a compact, nonempty convex set $X \subset \mathbb{R}^n$ is denoted by $conv_{f,X}$ and is the pointwise supremum of all possible convex underestimators of f over X, i.e., for all $\mathbf{x} \in X$

$$conv_{f,X}(\mathbf{x}) = \sup\{c(\mathbf{x}) \ : \ c(\mathbf{x}') \leq f(\mathbf{x}') \ \forall\, \mathbf{x}' \in X, \ \ c \text{ is convex}\}. \tag{4.3}$$

In fact, the condition "c is convex" can be substituted by the condition "c is affine;" i.e., we have

$$conv_{f,X}(\mathbf{x}) = \sup\{c(\mathbf{x}) \ : \ c(\mathbf{x}') \leq f(\mathbf{x}') \ \forall\, \mathbf{x}' \in X, \ \ c \text{ is affine}\}. \tag{4.4}$$

Similarly, the *concave envelope* $conc_{f,X}$ of an upper semicontinuous function f over X is the pointwise infimum of all the concave overestimators of f over X. We will not further discuss concave envelopes, since we have the following relation:

$$conv_{f,X} \equiv -conc_{-f,X}.$$

We remark that the functions f and $conv_{f,X}$ have the same minimum value over X (simply observe that the constant function whose value is equal to the minimum value of f over X is a convex underestimator of f over X), and the intersection between the global minima of these two functions is never empty. More precisely, the set of global minima of f over X is always a nonempty subset of the set of global minima of $conv_{f,X}$ over X. This observation already clarifies that computing convex envelopes is a hard task, and that is true even when f and X are of quite special forms. For instance, in (Crama, 1989) it was proven that computing the convex envelope of a multilinear function (see Section 4.2.7) over the unit hypercube is $\mathcal{NP}$-hard.

In addition to the definition given in (4.3), we can give two alternative definitions for the convex envelope of f over X. The first is based on the convex hull of the epigraph of f over X:

$$conv_{f,X}(\mathbf{x}) = \inf\{\mu \ : \ (\mathbf{x},\mu) \in chull(epi_X[f])\}.$$

Then, in order to compute the value of $conv_{f,X}$ at some point $\mathbf{x} \in X$, we can solve the following optimization problem:

$$\begin{array}{lll} conv_{f,X}(\mathbf{x}) = \inf & \sum_{i=1}^k \lambda_i f(\mathbf{x}_i) & \\ & \sum_{i=1}^k \lambda_i \mathbf{x}_i = \mathbf{x}, & \\ & \sum_{i=1}^k \lambda_i = 1, & \\ & \boldsymbol{\lambda} \geq \mathbf{0}, & \\ & \mathbf{x}_i \in X, & i = 1,\dots,k, \\ & k \in \mathbb{N}. & \end{array} \tag{4.5}$$

In fact, Carathéodory's theorem (see Theorem A.5) allows us to impose the restriction $k \leq n+1$.

The second alternative definition for the convex envelope of a function is through conjugate functions. The *conjugate function* of f over X (see, e.g., (Rockafellar, 1970)) is defined as

$$f^\star(\mathbf{y}) = \sup_{\mathbf{x}\in X} \left[\mathbf{x}^T\mathbf{y} - f(\mathbf{x})\right].$$

Function $f^\star$, being the pointwise supremum of a family of affine functions, is convex. It turns out that the second conjugate $f^{\star\star}$ of f is equivalent to the convex envelope of f over X (a proof of this result can be found, e.g., in (Falk, 1969; Rockafellar, 1970)), i.e.,

$$conv_{f,X}(\mathbf{x}) = f^{\star\star}(\mathbf{x}) \quad \forall\, \mathbf{x} \in X. \tag{4.6}$$

We remark that having different, although equivalent, definitions for the convex envelope turns out to be useful. Indeed, some result about convex envelopes is easily derived from a given definition, while the proof of the same result might be less obvious with other definitions.

A result which can be immediately seen, e.g., from (4.5), is that

$$X' \subseteq X \ : \ Ext[X] \subseteq X' \quad \Rightarrow \quad conv_{f,X}(\mathbf{x}) \leq conv_{f,X'}(\mathbf{x}) \quad \forall\, \mathbf{x} \in X, \tag{4.7}$$

where $Ext[X]$ denotes the set of extreme points of X (see Definition A.7). When dealing with convex envelopes, a relevant concept is that of a *generating set* of f over X, denoted as $G(f,X)$. This is defined as

$$G(f,X) = \{\mathbf{x} \in X \ : \ (\mathbf{x}, f(\mathbf{x})) \in Ext[chull(epi_X[f])]\}.$$

Stated in another way, $G(f,X)$ is the smallest subset of X such that constraints $\mathbf{x}_i \in X$ in (4.5) can be replaced by $\mathbf{x}_i \in G(f,X)$, or, equivalently, such that we have an equality in (4.7). Note that $Ext[X] \subseteq G(f,X)$.

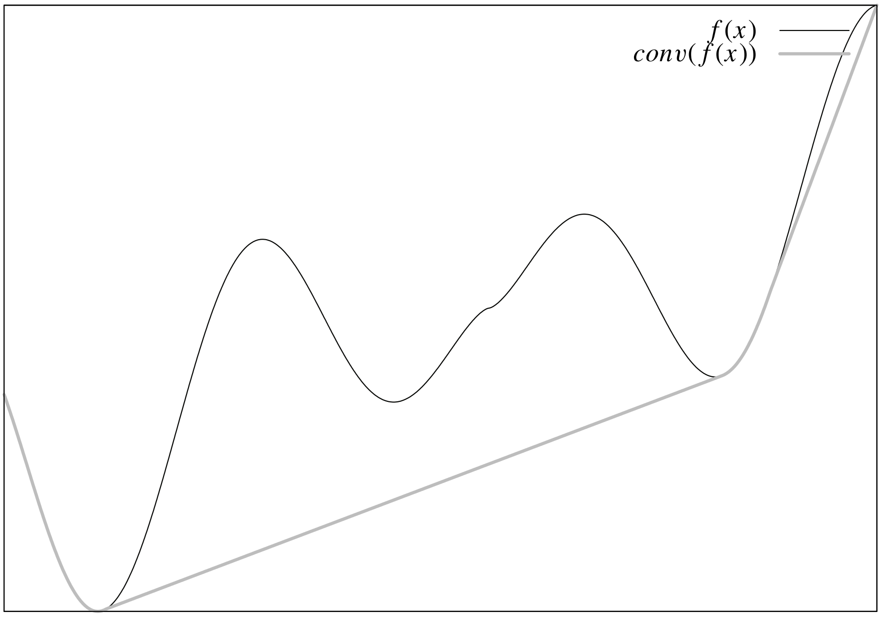


Figure 4.1. *An example of a nonconvex function and its convex envelope*

As an example, consider the simple one dimensional function in Figure 4.1, where, in gray, the convex envelope is reported. The generating set, in this example, can be seen in Figure 4.2.

The following theorem generalizes Theorem 1.2 in (Rikun, 1997) (see also Corollary 5 in (Tawarmalani & Sahinidis, 2002a)) and allows us to identify points which do *not* belong to $G(f,X)$.

Theorem 4.2. *Let f be a lower semicontinuous function and X be a polytope. Let $\bar{\mathbf{x}} \notin V(X)$, where $V(X)$ denotes the vertex set of X. If there exists a line ℓ such that*

- *$\bar{\mathbf{x}} \in ri(\ell \cap X)$ (recall that ri denotes the relative interior);*
- *f is concave in a neighborhood of $\bar{\mathbf{x}}$ over $ri(\ell \cap X)$,*

then $\bar{\mathbf{x}} \notin G(f,X)$.

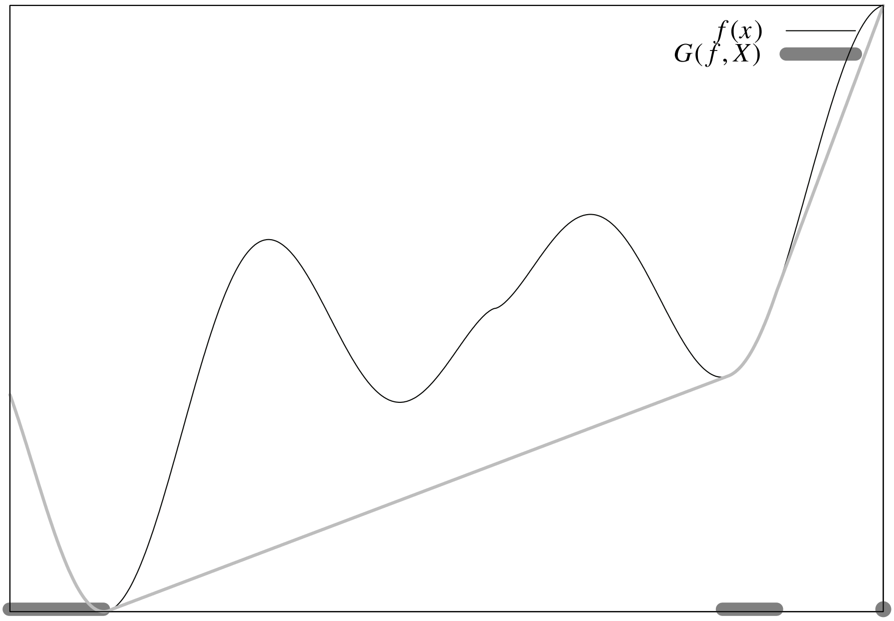


Figure 4.2. *An example of a generating set*

Proof. Consider problem (4.5), and let $\mathbf{z}_1,\mathbf{z}_2 \in ri(\ell \cap X)$ be two distinct points such that

- $\bar{\mathbf{x}} = \mu\mathbf{z}_1 + (1-\mu)\mathbf{z}_2$ for some $\mu \in (0,1)$;
- $\mathbf{z}_1,\mathbf{z}_2$ lie in the neighborhood in $ri(\ell \cap X)$ over which f is concave.

Then, by concavity,

$$f(\bar{\mathbf{x}}) \geq \mu f(\mathbf{z}_1) + (1-\mu) f(\mathbf{z}_2),$$

and in (4.5) any solution with $\mathbf{x}_i = \bar{\mathbf{x}}$ for some i can be replaced by a not worse solution with $\mathbf{x}_{i_1} = \mathbf{z}_1$ and $\mathbf{x}_{i_2} = \mathbf{z}_2$ for some i_1, i_2, $i_1 \neq i_2$. In particular, this means that $\bar{\mathbf{x}} \notin G(f,X)$. □

4.2.2 Polyhedral convex envelopes

Definition 4.3. *A function f admits a* polyhedral convex envelope *over a set X if*

$$conv_{f,X}(\mathbf{x}) = \max_{j=1,\dots,s} h_j(\mathbf{x}), \tag{4.8}$$

where the functions h_j, $j = 1,\dots,s$, form a finite set of affine underestimators of f over X.

In case X is a polytope with vertex set $V(X)$, a further relevant subclass is that of the functions f admitting a *vertex polyhedral convex envelope* over X, i.e., those for which the generating set $G(f,X)$ is equal to the vertex set $V(X)$. The following theorem has been proven in (Rikun, 1997) and states that differentiable functions admit a polyhedral convex envelope if and only if they admit a vertex polyhedral convex envelope.

Theorem 4.4. *If f is a continuously differentiable function and X is a polytope, then f has a polyhedral convex envelope if and only if*

$$G(f,X) = V(X),$$

i.e., f has a vertex polyhedral convex envelope.

Proof. First, assume that $G(f,X) = V(X)$ and prove that f admits a polyhedral convex envelope over X. In (4.5) we can replace $\mathbf{x}_i \in X$ with $\mathbf{x}_i \in V(X)$. Taking the dual of (4.5), we have

$$\begin{array}{lll} conv_{f,X}(\mathbf{x}) = \max & \mathbf{w}^T\mathbf{x} + w_0 & \\ & \mathbf{w}^T\mathbf{x}_i + w_0 \leq f(\mathbf{x}_i) & \forall\, \mathbf{x}_i \in V(X). \end{array} \tag{4.9}$$

Equivalently, if we denote by $(\mathbf{w}_j\ w_{0j})$, $j = 1,\ldots,s$, the s vertices of the polyhedron

$$\{(\mathbf{w}\ w_0)\ :\ \mathbf{w}^T\mathbf{x}_i + w_0 \leq f(\mathbf{x}_i)\ \ \forall\, \mathbf{x}_i \in V(X)\}, \tag{4.10}$$

we have

$$conv_{f,X}(\mathbf{x}) = \max_{j=1,\ldots,s} \mathbf{w}_j^T\mathbf{x} + w_{0j}; \tag{4.11}$$

i.e., $conv_{f,X}$ is the maximum of s affine functions and, thus, f admits a polyhedral convex envelope.

Conversely, assume by contradiction that f admits a polyhedral convex envelope, but

$$\exists\, \bar{\mathbf{x}} \in G(f,X) \setminus V(X).$$

If $\bar{\mathbf{x}} \in int(X)$, there are at least two distinct affine functions h_i and h_j, $i \neq j$, in (4.8) such that

$$h_i(\mathbf{x}) = f(\bar{\mathbf{x}}) + \mathbf{a}_i^T(\mathbf{x} - \bar{\mathbf{x}}), \quad h_j(\mathbf{x}) = f(\bar{\mathbf{x}}) + \mathbf{a}_j^T(\mathbf{x} - \bar{\mathbf{x}}).$$

This follows from the definition of $G(f,X)$ as the projection over X of the extreme points of $chull(epi_X[f])$, which is a polyhedral region in view of the assumption that f admits a polyhedral convex envelope. Therefore, $h_i(\mathbf{x})$ and $h_j(\mathbf{x})$ are supporting hyperplanes for f at $\bar{\mathbf{x}}$, and the vectors $\mathbf{a}_i$, $\mathbf{a}_j$ are subgradients of f at $\bar{\mathbf{x}}$. But in view of the continuous differentiability of f, we have $\mathbf{a}_i = \mathbf{a}_j$, thus contradicting the fact that h_i and h_j are distinct affine functions.

If $\bar{\mathbf{x}} \in X \setminus int(X)$, then consider the face F of X such that $\bar{\mathbf{x}} \in ri(F)$, i.e., $\bar{\mathbf{x}}$ is in the relative interior of F. Next, we consider the restriction f_F of f to $ahull(ri(F))$:

$$f_F(\mathbf{x}) = f(\mathbf{x})\ \ \forall\, \mathbf{x} \in ahull(ri(F)).$$

Since $\bar{\mathbf{x}} \in ri(F)$, we can repeat the same proof as above by replacing f with f_F and X with F (note that $conv_{f,X}(\mathbf{x}) = conv_{f_F,F}(\mathbf{x})$ for all $\mathbf{x} \in F$). □

To be more precise, in (Rikun, 1997) the assumption of continuous differentiability for f is replaced by the more general assumption of continuous differentiability *over the polytope* X; i.e., the function f is required to be

- continuous on X;
- for all $\mathbf{x} \in X \setminus V(X)$, the function f is continuously differentiable over $ri(F_{\mathbf{x}})$, where $F_{\mathbf{x}}$ is the lowest dimensional face such that $\mathbf{x} \in F_{\mathbf{x}}$.

We remark that the *if* part of the theorem does not require that f be continuously differentiable over X. Instead, the *only if* part may not be true for nondifferentiable functions. An easy example is the absolute value function $f(x) = |x| = \max\{x, -x\}$ over $X = [-1, 1]$. This function is convex and thus equal to its convex envelope over X. It is also polyhedral, but its generating set $G(f, X)$ is $\{-1, 0, 1\} \supset V(X) = \{-1, 1\}$.

As a consequence of Theorem 4.2, if the assumptions of such theorem are satisfied for all $\mathbf{x} \notin V(X)$, we can conclude that $G(f, X) = V(X)$, i.e., f admits a vertex polyhedral convex envelope over X. In particular, if f is a concave function, then it admits a vertex polyhedral convex envelope over any polytope X. A property more general than concavity is edge-concavity (see (Tardella, 2003, 2008)).

Definition 4.5. *A function f is said to be* edge-concave *over a polytope X if it is concave over each segment in X parallel to an edge of X.*

It is an immediate consequence of Theorem 4.2 that an edge-concave function over a polytope X also admits a vertex polyhedral convex envelope over X. However, as remarked in (Tardella, 2008), there exist functions which are not edge-concave but admit a vertex polyhedral convex envelope, as the function $f(x) = \sin^2(x)$ over $X = [0, 2\pi]$ shows.

A well-known example of vertex polyhedral convex envelope is that of the bilinear function x_1x_2 over the unit square $[0, 1]^2$. Such a function is edge-concave over $X = [0, 1]^2$ (and actually also edge-convex, since by fixing either x_1 or x_2 we get a linear function). Therefore, we can apply formulae (4.9), (4.10), and (4.11) to derive the convex envelope. First we need to detect the basic feasible solutions for

$$\begin{aligned}
&w_0 \le 0,\\
&w_1 + w_0 \le 0,\\
&w_2 + w_0 \le 0,\\
&w_1 + w_2 + w_0 \le 1.
\end{aligned}$$

These are

$$(w_0^1\ w_1^1\ w_2^1) = (0\ 0\ 0), \quad (w_0^2\ w_1^2\ w_2^2) = (-1\ 1\ 1).$$

Then,

$$conv_{x_1x_2,[0,1]^2} = \max\{0, x_1 + x_2 - 1\}. \tag{4.12}$$

Such a result was first derived in (McCormick, 1976) together with the companion result

$$conc_{x_1x_2,[0,1]^2} = \min\{x_1, x_2\}.$$

Now, consider the two triangles

$$\Delta_1 = chull\{(0\ 0),\ (1\ 0),\ (0\ 1)\}, \quad \Delta_2 = chull\{(1\ 0),\ (0\ 1),\ (1\ 1)\}. \tag{4.13}$$

Let $L_{x_1x_2,\Delta_i}$, $i = 1,2$, be the affine functions interpolating x_1x_2 at the vertices of Δ_i. Then, we have that

$$conv_{x_1x_2,[0,1]^2}(\mathbf{x}) = \begin{cases} L_{x_1x_2,\Delta_1}(\mathbf{x}) & \text{if } \mathbf{x} \in \Delta_1, \\ L_{x_1x_2,\Delta_2}(\mathbf{x}) & \text{if } \mathbf{x} \in \Delta_2. \end{cases}$$

Let us introduce the following definitions.

Definition 4.6. *Let $\mathbf{x}_1,\ldots,\mathbf{x}_{n+1}$ be $n+1$ affinely independent vectors. Then, the polytope*

$$\begin{aligned} \Delta &= \Delta(\mathbf{x}_1,\ldots,\mathbf{x}_{n+1}) \\ &= \left\{ \mathbf{x} \in \mathbb{R}^n \ : \ \mathbf{x} = \sum_{i=1}^{n+1} \lambda_i \mathbf{x}_i, \ \sum_{i=1}^{n+1} \lambda_i = 1, \ \lambda_i \geq 0, \ i = 1,\ldots,n+1 \right\} \end{aligned}$$

*is called an n-*simplex*, and its vertex set is made up by the $n+1$ points $\mathbf{x}_1,\ldots,\mathbf{x}_{n+1}$.*

Definition 4.7. *Given a polytope $X \subseteq \mathbb{R}^n$ with vertex set $V(X)$ and the n-simplices Δ_i, $i = 1,\ldots,t$, the collection $T = \{\Delta_i\}_{i=1}^t$ is a* triangulation *of X not adding vertices if*

- *$V(\Delta_i) \subseteq V(X)$, $i = 1,\ldots,t$, i.e., the vertices of the simplices are also vertices of X;*
- *$int(\Delta_i) \cap int(\Delta_j) = \emptyset$ for any $i \neq j$, i.e., two distinct simplices have no common interior point;*
- *$X = \cup_{i=1}^t \Delta_i$, i.e., the union of all the simplices is equal to the polytope X;*
- *$\Delta_i \cap \Delta_j$, $i \neq j$, is a (possibly empty) face both of Δ_i and of Δ_j.*

According to these definitions, the collection of two 2-simplices (i.e., triangles) $\{\Delta_1, \Delta_2\}$ defined in (4.13) is a triangulation of $X = [0,1]^2$, and we have seen that at each point $\mathbf{x} = (x_1 \ x_2) \in [0,1]^2$, $conv_{x_1x_2,[0,1]^2}$ coincides with the affine function interpolating x_1x_2 at the vertices of a triangle within the triangulation which contains $\mathbf{x}$. This result has been extended to all functions admitting a vertex polyhedral convex envelope over a polytope X (see (Meyer & Floudas, 2005a)). To see this, we first need to introduce some definitions and results.

Definition 4.8. *Given a polytope $X \subseteq \mathbb{R}^n$ with vertex set $V(X) = \{\mathbf{v}_1,\ldots,\mathbf{v}_k\}$, the* affine dependencies *of $V(X)$ are all the vectors $\lambda \in \mathbb{R}^k \setminus \{\mathbf{0}\}$ such that*

$$\sum_{i=1}^k \lambda_i = 0 \quad and \quad \sum_{i=1}^k \lambda_i \mathbf{v}_i = 0.$$

An affine dependency is called minimal *if its support (i.e., its set of nonzero components) is minimal with respect to inclusion within the set of affine dependencies.*

If we consider $X = [0,1]^2$, we notice that the affine dependencies are only the multiples of vector $(1 \ -1 \ -1 \ 1)$. For a given affine dependency $\boldsymbol{\lambda}$, let

$$P(\boldsymbol{\lambda}) = \{i \ : \ \lambda_i > 0\}, \quad N(\boldsymbol{\lambda}) = \{i \ : \ \lambda_i < 0\}.$$

In (Meyer & Floudas, 2005a) the following proposition is stated.

Proposition 4.9. *Each affine dependency $\boldsymbol{\lambda}$ for $V(X)$ can be expressed as a sum of minimal affine dependencies $\boldsymbol{\lambda}_i$, $i=1,\ldots,r$, for $V(X)$ with the property that*

$$P(\boldsymbol{\lambda}_i) \subseteq P(\boldsymbol{\lambda}), \quad N(\boldsymbol{\lambda}_i) \subseteq N(\boldsymbol{\lambda}), \quad i=1,\ldots,r.$$

Definition 4.10. *For a given minimal affine dependency $\boldsymbol{\lambda}$, let $\Lambda = \sum_{i \in P(\boldsymbol{\lambda})} \lambda_i$. The two sets*

$$chull\{\mathbf{v}_i \ : \ i \in P(\boldsymbol{\lambda})\}, \quad chull\{\mathbf{v}_i \ : \ i \in N(\boldsymbol{\lambda})\}$$

have relative interiors intersecting at the unique point

$$\bar{\mathbf{x}} = \sum_{i \in P(\boldsymbol{\lambda})} \frac{\lambda_i}{\Lambda} \mathbf{v}_i = \sum_{i \in N(\boldsymbol{\lambda})} -\frac{\lambda_i}{\Lambda} \mathbf{v}_i. \tag{4.14}$$

Such point is called a circuit intersection point*. We denote by $C(X)$ the set of all circuit intersection points of X.*

For $X=[0,1]^2$ we have that

$$chull\{\mathbf{v}_i \ : \ i \in P(\boldsymbol{\lambda})\} = \{(\gamma,\gamma) \ : \ \gamma \in [0,1]\},$$
$$chull\{\mathbf{v}_i \ : \ i \in N(\boldsymbol{\lambda})\} = \{(\gamma,1-\gamma) \ : \ \gamma \in [0,1]\},$$

where the first set is the segment joining the vertices (0 0) and (1 1), while the second set is the segment joining the vertices (0 1) and (1 0). Then, $C(X)$ is made up of a single circuit intersection point, namely, $(\frac{1}{2}\ \frac{1}{2})$. The following lemma has been proven in (Meyer & Floudas, 2005a).

Lemma 4.11. *Let Δ_1 and Δ_2 be two n-simplices such that $V(\Delta_1), V(\Delta_2) \subseteq X$. We have that $V(\Delta_1 \cap \Delta_2)$, i.e., the vertex set of the intersection between Δ_1 and Δ_2 is made up of the common vertices of Δ_1 and Δ_2 and by circuit intersection points.*

For $X=[0,1]^2$, if we consider the triangle Δ_1 as defined in (4.13) and the triangle

$$\Delta_3 = chull\{(0\ 0),\ (1\ 0),\ (1\ 1)\},$$

we have that

$$V(\Delta_1 \cap \Delta_3) = \left\{(0\ 0), (1\ 0), \left(\frac{1}{2}\ \frac{1}{2}\right)\right\};$$

i.e., the vertices are the two common vertices (0 0) and (1 0) plus the circuit intersection point $(\frac{1}{2}\ \frac{1}{2})$. We are now ready for the following theorem, proven in (Meyer & Floudas, 2005a), but for which we give a slightly different proof here.

Theorem 4.12. *Let $X \subseteq \mathbb{R}^n$ be a polytope with vertex set $V(X) = \{\mathbf{v}_1,\ldots,\mathbf{v}_k\}$. Let*

$$f : \mathbb{R}^n \to \mathbb{R}$$

be a function admitting a vertex polyhedral convex envelope over X. Let T be a triangulation of X and g_T be the function defined as

$$g_T(\mathbf{x}) = L_{f,\Delta}(\mathbf{x}) \quad if \quad \mathbf{x} \in \Delta,\ \Delta \in T,$$

i.e., over each n-simplex $\Delta \in T$, g_T *coincides with the affine function interpolating* f *at the vertices of* Δ. *Then,* g_T *is the convex envelope of* f *over* X *if and only if for all minimal affine dependencies* $\boldsymbol{\lambda}$ *such that*

$$\{\mathbf{v}_i\ :\ i \in P(\boldsymbol{\lambda})\} \subseteq \Delta'' \in T, \quad \{\mathbf{v}_i\ :\ i \in N(\boldsymbol{\lambda})\} \subseteq \Delta' \notin T, \tag{4.15}$$

we have that

$$\sum_{i\in P(\boldsymbol{\lambda})} \lambda_i f(\mathbf{v}_i) \leq -\sum_{i\in N(\boldsymbol{\lambda})} \lambda_i f(\mathbf{v}_i). \tag{4.16}$$

Proof. First assume that (4.15) implies (4.16) and prove that this implies that $g_T = conv_{f,X}$. Since we are assuming that f admits a vertex polyhedral convex envelope over X, it follows from (4.5) that

$$conv_{f,X}(\mathbf{x}) = \sum_{i=1}^{n+1} \bar{\lambda}_i f(\mathbf{z}_i),$$

where $\mathbf{z}_1,\dots,\mathbf{z}_{n+1} \in V(X)$ are $n+1$ affinely independent vertices of X, and

$$\sum_{i=1}^{n+1} \bar{\lambda}_i = 1, \quad \bar{\lambda}_i \geq 0,\ i = 1,\dots,n+1.$$

In particular, we have that

$$\mathbf{x} \in \Delta' = chull\{\mathbf{z}_1,\dots,\mathbf{z}_{n+1}\}.$$

If we can choose $\Delta' \in T$, then $g_T(\mathbf{x}) = conv_{f,X}(\mathbf{x})$. Otherwise, assume by contradiction that we cannot choose $\Delta' \in T$. By definition of triangulation, there must exist in T at least an n-simplex

$$\Delta'' = chull\{\mathbf{w}_1,\dots,\mathbf{w}_{n+1}\},$$

with $\mathbf{w}_1,\dots,\mathbf{w}_{n+1} \in V(X)$ and such that $\mathbf{x} \in \Delta''$. In view of the assumption by contradiction, we must have

$$L_{f,\Delta'}(\mathbf{x}) < L_{f,\Delta''}(\mathbf{x}). \tag{4.17}$$

Consider $\Delta' \cap \Delta''$. According to Lemma 4.11, we can split the r vertices $\{\mathbf{s}_1,\dots,\mathbf{s}_r\}$ of $\Delta' \cap \Delta''$ into two sets:

$$I_1(\Delta' \cap \Delta'') = V(\Delta') \cap V(\Delta''),$$

$$I_2(\Delta' \cap \Delta'') \subset C(X).$$

Since $\mathbf{x} \in \Delta' \cap \Delta''$, we can represent $\mathbf{x}$ as

$$\mathbf{x} = \sum_{i\in I_1(\Delta',\Delta'')} \gamma_i \mathbf{s}_i + \sum_{i\in I_2(\Delta',\Delta'')} \gamma_i \mathbf{s}_i,$$

where $\gamma_i \geq 0$, $i = 1,\dots,r$, and $\sum_{i=1}^r \gamma_i = 1$. We also have that

$$\begin{aligned} L_{f,\Delta'}(\mathbf{x}) &= \textstyle\sum_{i\in I_1(\Delta',\Delta'')} \gamma_i f(\mathbf{s}_i) + \sum_{i\in I_2(\Delta',\Delta'')} \gamma_i L_{f,\Delta'}(\mathbf{s}_i), \\ L_{f,\Delta''}(\mathbf{x}) &= \textstyle\sum_{i\in I_1(\Delta',\Delta'')} \gamma_i f(\mathbf{s}_i) + \sum_{i\in I_2(\Delta',\Delta'')} \gamma_i L_{f,\Delta''}(\mathbf{s}_i). \end{aligned}$$

Then, in view of (4.17), for some $\bar{\mathbf{s}} \in \{\mathbf{s}_i : i \in I_2(\Delta', \Delta'')\}$ we must have

$$L_{f,\Delta'}(\bar{\mathbf{s}}) < L_{f,\Delta''}(\bar{\mathbf{s}}). \tag{4.18}$$

Since

$$\bar{\mathbf{s}} = \sum_{j\,:\,\mathbf{v}_j \in V(\Delta')} \eta_j \mathbf{v}_j = \sum_{j\,:\,\mathbf{v}_j \in V(\Delta'')} \xi_j \mathbf{v}_j,$$

where

$$\sum_{j\,:\,\mathbf{v}_j \in V(\Delta')} \eta_j = 1, \quad \eta_j \geq 0, \;\; j : \mathbf{v}_j \in V(\Delta'),$$

$$\sum_{j\,:\,\mathbf{v}_j \in V(\Delta'')} \xi_j = 1, \quad \xi_j \geq 0, \;\; j : \mathbf{v}_j \in V(\Delta''),$$

there must exist an affine dependency $\boldsymbol{\lambda}$ for $V(\Delta') \cup V(\Delta'')$ such that

$$P(\boldsymbol{\lambda}) \subseteq \{j : \mathbf{v}_j \in V(\Delta'')\}, \quad N(\boldsymbol{\lambda}) \subseteq \{j : \mathbf{v}_j \in V(\Delta^{'})\}.$$

As a consequence of Proposition 4.9, we can write the affine dependency $\boldsymbol{\lambda}$ as a sum of minimal affine dependencies $\boldsymbol{\lambda}_i$, $i = 1, \ldots, t$, such that for all i

$$P(\boldsymbol{\lambda}_i) \subseteq P(\boldsymbol{\lambda}) \subseteq \{j : \mathbf{v}_j \in V(\Delta'')\}, \quad N(\boldsymbol{\lambda}_i) \subseteq N(\boldsymbol{\lambda}) \subseteq \{j : \mathbf{v}_j \in V(\Delta^{'})\}.$$

Since (4.15) implies (4.16), then, for each $i = 1, \ldots, t$,

$$-\sum_{j \in N(\boldsymbol{\lambda}_i)} \lambda_{ij} f(\mathbf{v}_j) \geq \sum_{j \in P(\boldsymbol{\lambda}_i)} \lambda_{ij} f(\mathbf{v}_j),$$

and, consequently, we should also have

$$L_{f,\Delta'}(\bar{\mathbf{s}}) \geq L_{f,\Delta''}(\bar{\mathbf{s}}),$$

which contradicts (4.18).

Conversely, assume by contradiction that g_T is the convex envelope of f over X, but (4.15) does not imply (4.16), i.e., there exists a minimal affine dependency $\boldsymbol{\lambda}$ such that (4.15) is true for some $\Delta'' \in T$ and $\Delta^{'} \notin T$, but

$$-\sum_{j \in N(\boldsymbol{\lambda})} \lambda_j f(\mathbf{v}_j) < \sum_{j \in P(\boldsymbol{\lambda})} \lambda_j f(\mathbf{v}_j). \tag{4.19}$$

Let $\mathbf{s} \in C(X)$ be the circuit intersection point corresponding to the minimal affine dependency $\boldsymbol{\lambda}$. Then, in view of (4.14), we have that

$$L_{f,\Delta''}(\mathbf{s}) = \sum_{j \in P(\boldsymbol{\lambda})} \eta_j f(\mathbf{v}_j),$$

where

$$\eta_j = \frac{\lambda_j}{\sum_{j \in P(\boldsymbol{\lambda})} \lambda_j}, \quad j \in P(\boldsymbol{\lambda}),$$

and

$$L_{f,\Delta'}(\mathbf{s}) = \sum_{j \in N(\boldsymbol{\lambda})} \xi_j f(\mathbf{v}_j),$$

where

$$\xi_j = -\frac{\lambda_j}{\sum_{j \in P(\boldsymbol{\lambda})} \lambda_j}, \quad j \in N(\boldsymbol{\lambda}).$$

Then, in view of (4.19),

$$L_{f,\Delta'}(\mathbf{s}) < L_{f,\Delta''}(\mathbf{s}) = conv_{f,X}(\mathbf{s}),$$

which contradicts the fact that $L_{f,\Delta''}(\mathbf{s})$ is the optimal value of (4.5) for $\mathbf{x} = \mathbf{s}$. □

It is worthwhile to add some comments to Theorem 4.12. The theorem establishes that in order to define vertex polyhedral convex envelopes over a polytope X, it is enough to search for a suitable triangulation of X. However, the size of the triangulation, i.e., the number of simplices it is composed of, may be extremely large. For instance, triangulations of the unit hypercube $[0,1]^n$ may have a size of order $n!$. Moreover, it is usually difficult to find the triangulation corresponding to the convex envelope. However, there are some cases for which we are able to derive it explicitly. We first need to introduce some definitions.

Definition 4.13. *A function f is* submodular *over a lattice Z if*

$$\forall\, \mathbf{z}, \mathbf{z}' \in Z \;:\; f(\max\{\mathbf{z}, \mathbf{z}'\}) + f(\min\{\mathbf{z}, \mathbf{z}'\}) \le f(\mathbf{z}) + f(\mathbf{z}'),$$

where $\max\{\mathbf{z}, \mathbf{z}'\}$ *and* $\min\{\mathbf{z}, \mathbf{z}'\}$ *denote, respectively, the componentwise maximum and minimum of* $\mathbf{z}$ *and* $\mathbf{z}'$.

Now, let

$$\pi \;:\; \{1, \ldots, n\} \to \{1, \ldots, n\}$$

be a permutation of $\{1, \ldots, n\}$, and consider the simplex Δ_π whose vertices are

$$\mathbf{v}_0 = \mathbf{0},\; \mathbf{v}_k = \sum_{j=1}^{k} \mathbf{e}_{\pi(j)}, \;\; k = 1, \ldots, n,$$

where $\mathbf{e}$ denotes the vector whose ith coordinate is equal to 1, while all the other coordinates are null. Then, we have the following definition.

Definition 4.14. *The triangulation of the unit hypercube* $[0,1]^n$ *with the* $n!$ *simplices*

$$\{\Delta_\pi \;:\; \pi \text{ permutation of } \{1, \ldots, n\}\}$$

is called Kuhn's triangulation.

Now, for some $\mathbf{x} \in [0,1]^n$, consider a permutation π of $\{1, \ldots, n\}$ such that

$$x_{\pi(1)} \ge x_{\pi(2)} \ge \cdots \ge x_{\pi(n)}.$$

Then, $\mathbf{x}$ can be expressed as the following convex combination of the vertices of Δ_π:

$$\mathbf{x} = (1 - x_{\pi(1)})\mathbf{0} + \sum_{k=1}^{n-1}(x_{\pi(k)} - x_{\pi(k+1)})\sum_{j=1}^{k}\mathbf{e}_{\pi(j)} + x_{\pi(n)}\sum_{j=1}^{n}\mathbf{e}_{\pi(j)},$$

from which (see (Lovász, 1982)) we understand the following definition.

Definition 4.15. *The function*

$$f^L(\mathbf{x}) = (1 - x_{\pi(1)})f(\mathbf{0}) + \sum_{k=1}^{n-1}(x_{\pi(k)} - x_{\pi(k+1)})f\left(\sum_{j=1}^{k}\mathbf{e}_{\pi(j)}\right) + x_{\pi(n)}f\left(\sum_{j=1}^{n}\mathbf{e}_{\pi(j)}\right)$$

is called the Lovász extension *of f.*

It turns out (see (Lovász, 1982)) that f^L is convex if and only if the restriction of f over $V([0,1]^n)$ is submodular. Moreover, in (Tawarmalani, Richard, & Xiong, 2012) the following theorem was proven.

Theorem 4.16. *The convex envelope of f over $[0,1]^n$ is its Lovász extension if and only if $G(f,[0,1]^n) = V([0,1]^n)$ and f is submodular over $V([0,1]^n)$.*

In particular, this means that under the assumptions of Theorem 4.16 the triangulation corresponding to the convex envelope of f over $[0,1]^n$ is Kuhn's triangulation (in (Tawarmalani et al., 2012) the result was also extended to some subsets of the unit hypercube). We illustrate this through a simple example.

Example 4.17. Let $f(x_1,x_2) = -x_1x_2$ and $X = [0,1]^2$. Kuhn's triangulation is made up of the two simplices

$$\Delta_1 = chull\left\{\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}1\\1\end{pmatrix}\right\},$$
$$\Delta_2 = chull\left\{\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}1\\1\end{pmatrix}\right\},$$

and the Lovász extension is

$$f^L(x_1,x_2) = \begin{cases} -x_2 & \text{if } (x_1\ x_2) \in \Delta_1, \\ -x_1 & \text{if } (x_1\ x_2) \in \Delta_2, \end{cases}$$

or, in compact form,

$$f^L(x_1,x_2) = -\min\{x_1,x_2\}.$$

It is easy to see that $G(f,[0,1]^2) = V([0,1]^2)$ and that f is submodular when restricted to $V([0,1]^2)$. Therefore, we can conclude that $conv_{f,[0,1]^2}(x_1,x_2) = f^L(x_1,x_2)$. ∎

4.2.3 Nonpolyhedral convex envelopes

We have previously seen that for functions admitting vertex polyhedral convex envelopes over a polytope, the computation of the convex envelope is based on the search of a proper triangulation of the polytope itself. Things are more complicated when a function does not admit a polyhedral convex envelope. According to (4.5), even the computation of the convex envelope at a given point may require the solution of a nonconvex problem. However, in some cases the computation can be carried out by solving a relatively simple problem. Here we first discuss three different but related results, where in the computation of $conv_{f,X}$ at some point $\mathbf{x}$ through (4.5) we might restrict our attention to representations of $\mathbf{x}$ by convex combinations of just two points $\mathbf{x}_1$ and $\mathbf{x}_2$ belonging to $G(f,X)$; i.e., we can impose $k \leq 2$ in (4.5).

The first such result was discussed in (Tawarmalani & Sahinidis, 2001). In that paper convex envelopes over boxes $X = \prod_{i=1}^{n}[l_i,u_i]$ of n-dimensional functions f, which are convex if the value of one variable is fixed and concave if we fix the remaining $n-1$ variables, are considered. Without loss of generality, we assume that f is convex with respect to $x_2,\dots,x_n$ if we fix x_1 and concave with respect to x_1 if we fix $x_2,\dots,x_n$. If we denote by

$$F_1 = X \cap \{\mathbf{x} \in \mathbb{R}^n \ : \ x_1 = l_1\}, \quad F_2 = X \cap \{\mathbf{x} \in \mathbb{R}^n \ : \ x_1 = u_1\}$$

the two parallel facets of X obtained by fixing x_1 at its lower and upper limits, respectively, then Theorem 4.2 shows that

$$G(f,X) \subseteq F_1 \cup F_2.$$

Now we prove the following theorem.

Theorem 4.18. *Let f be a function with the following properties:*

(i) *f is convex if the value of x_1 is fixed;*

(ii) *f is concave if the remaining $n-1$ variables are fixed.*

Let $\mathbf{x}_1^\star,\mathbf{x}_2^\star,\dots,\mathbf{x}_k^\star \in G(f,X)$, $k \leq n+1$, be the points in an optimal solution of the problem (4.5)*, which defines the value at $\mathbf{x}$ of the convex envelope of f over the box $X = \prod_{i=1}^{n}[l_i,u_i]$. There always exists an optimal solution with $k \leq 2$ and, when $k = 2$, $\mathbf{x}_1^\star \in F_1$ and $\mathbf{x}_2^\star \in F_2$ must hold.*

Proof. Let us assume by contradiction that for some $\mathbf{x} \in X$ only solutions with $k \geq 3$ are possible. Then, at least two points must belong to the same facet F_1 or F_2. Without loss of generality assume that this is true for F_1 (the proof is completely analogous for F_2). Denote

$$I_j = \{i \ : \ \mathbf{x}_i^\star \in F_j,\ i = 1,\dots,k\}, \quad j = 1,2.$$

The optimal solution of (4.5) is completed by values $\lambda_i^\star > 0$, $i = 1,\dots,k$, such that $\sum_{i=1}^{k}\lambda_i^\star = 1$ and

$$\mathbf{x} = \sum_{i \in I_1}\lambda_i^\star \mathbf{x}_i^\star + \sum_{i \in I_2}\lambda_i^\star \mathbf{x}_i^\star.$$

Now, consider the point

$$\tilde{\mathbf{x}}_1 = \frac{\sum_{i \in I_1} \lambda_i^\star \mathbf{x}_i^\star}{\sum_{i \in I_1} \lambda_i^\star} \in F_1.$$

Then, we can also write

$$\mathbf{x} = \left(\sum_{i \in I_1} \lambda_i^\star \right) \tilde{\mathbf{x}}_1 + \sum_{i \in I_2} \lambda_i^\star \mathbf{x}_i^\star.$$

Moreover, as when x_1 is fixed, function f is convex, we have, in particular, that f is convex over F_1, we also have that

$$\left(\sum_{i \in I_1} \lambda_i^\star \right) f(\tilde{\mathbf{x}}_1) \leq \sum_{i \in I_1} \lambda_i^\star f(\mathbf{x}_i^\star).$$

Thus, replacing in the optimal solution $\mathbf{x}_1^\star, \mathbf{x}_2^\star, \ldots, \mathbf{x}_k^\star$ of problem (4.5) all points $\mathbf{x}_i^\star$, $i \in I_1$, i.e., all points in F_1, with the single point $\tilde{\mathbf{x}}_1$, a solution is obtained which is at least as good as the optimal one, and is thus also optimal. By possibly repeating the same replacement with points in F_2, we end up with an optimal solution with just two points, one in F_1 and the other in F_2, thus contradicting that only optimal solutions with $k \geq 3$ are possible. □

In view of this result, for functions satisfying the properties of the theorem, problem (4.5) can be written as

$$\begin{aligned} conv_{f,X}(\mathbf{x}) = \min \quad & \lambda f(\mathbf{x}_1) + (1-\lambda) f(\mathbf{x}_2) \\ & \lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2 = \mathbf{x}, \\ & \mathbf{x}_1 \in F_1, \\ & \mathbf{x}_2 \in F_2, \end{aligned} \tag{4.20}$$

where λ is a constant value given by the following formula:

$$\lambda = \frac{u_1 - x_1}{u_1 - l_1} \in (0,1) \tag{4.21}$$

(for $x_1 = l_1$ or $x_1 = u_1$ the value of the convex envelope is always equal to $f(\mathbf{x})$). It turns out that, in view of the convexity of f over the two facets F_1 and F_2, problem (4.20) is a convex problem. For bivariate functions we can even reduce this problem to a one-dimensional problem. Indeed, we can rewrite (4.20) as

$$\begin{aligned} conv_{f,X}(x_1,x_2) = \min \quad & \lambda f(l_1, x_{12}) + (1-\lambda) f(u_1, x_{22}) \\ & \lambda x_{12} + (1-\lambda) x_{22} = x_2, \\ & l_2 \leq x_{12}, x_{22} \leq u_2, \end{aligned}$$

where λ is defined in (4.21), and one of the two variables x_{12} or x_{22} can be eliminated. In particular, by eliminating x_{12}, we end up with the following one-dimensional problem:

$$\begin{aligned} conv_{f,X}(x_1,x_2) = \min \quad & \lambda f\left(l_1, \tfrac{x_2 - (1-\lambda)x_{22}}{\lambda}\right) + (1-\lambda) f(u_1, x_{22}) \\ & \max\left\{l_2, \tfrac{x_2 - \lambda u_2}{1-\lambda}\right\} \leq x_{22} \leq \min\left\{l_2, \tfrac{x_2 - \lambda l_2}{1-\lambda}\right\}. \end{aligned}$$

In (Jach, Michaels, & Weismantel, 2008) a similar development was carried out for the computation of the convex envelope over a box X for n-dimensional functions f satisfying the following properties:

- f is $(n-1)$-convex; i.e., if we fix the value of one variable, the function is convex with respect to the remaining $n-1$ variables; and
- f is indefinite over X; i.e., its Hessian $\nabla^2 f(\mathbf{x})$ has at least one negative eigenvalue for all $\mathbf{x} \in X$.

First observe that indefiniteness of f over X together with Theorem 4.2 guarantees that no interior point of X belongs to $G(f,X)$. Therefore, $G(f,X)$ is (a subset of) the union of all the facets of X. Moreover, convexity over each facet of X implies that $conv_{f,X}(\mathbf{x}) = f(\mathbf{x})$ for each point $\mathbf{x}$ belonging to a facet of X. Also in this case it is possible to prove that there always exists an optimal solution of problem (4.5) with at most two points in $G(f,X)$. In particular, in (Jach et al., 2008) it is proven that

$$
\begin{aligned}
conv_{f,X}(\mathbf{x}) = \min \quad & \lambda f(\mathbf{x}_1) + (1-\lambda) f(\mathbf{x}_2) \\
& \lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2 = \mathbf{x}, \\
& \mathbf{x}_1 \in F_1, \\
& \mathbf{x}_2 \in F_2, \\
& 0 \leq \lambda \leq 1, \\
& F_1, F_2 \text{ distinct facets of } X.
\end{aligned}
$$

Therefore, the computation of the convex envelope requires the separate solution of different subproblems, one for each pair of distinct facets of X. The case of parallel facets F_1 and F_2 reduces to the convex problem (4.20) with λ defined by a formula similar to (4.21). If F_1 and F_2 are adjacent facets, in (Jach et al., 2008) it is first observed that once $\mathbf{x}_1 \in F_1$ is fixed, then λ and $\mathbf{x}_2$ are uniquely determined. Therefore, we can see λ and $\mathbf{x}_2$ as functions of $\mathbf{x}_1$ and solve the problem

$$
\begin{aligned}
\min \quad & \phi(\mathbf{x}_1) = \lambda(\mathbf{x}_1) f(\mathbf{x}_1) + (1-\lambda(\mathbf{x}_1)) f(\mathbf{x}_2(\mathbf{x}_1)) \\
& \lambda(\mathbf{x}_1)\mathbf{x}_1 + (1-\lambda(\mathbf{x}_1))\mathbf{x}_2(\mathbf{x}_1) = \mathbf{x}, \\
& \mathbf{x}_1 \in F_1, \\
& \mathbf{x}_2(\mathbf{x}_1) \in F_2.
\end{aligned}
$$

Next, in (Jach et al., 2008) it is proven that function ϕ turns out to be unimodal, while the set of feasible vectors $\mathbf{x}_1$ is the intersection of facet F_1 with the pointed cone with vertex $\mathbf{x}$ and directions

$$\{\mathbf{x} - \mathbf{v} \ : \ \mathbf{v} \in V(F_2)\},$$

so that the set is a convex one. Therefore, local solvers are able to return an optimal solution of the problem above. Again, for bivariate functions this problem can be reduced to a one-dimensional problem. Indeed, given $X = [l_1, u_1] \times [l_2, u_2]$, consider, e.g., the two adjacent facets obtained by fixing x_1 to l_1 and x_2 to l_2, respectively. Then, problem (4.5) becomes

$$\begin{aligned}\min\quad & (1-\lambda)f(l_1,x_{12})+\lambda f(x_{21},l_2)\\ & \lambda x_{21}+(1-\lambda)l_1=x_1,\\ & \lambda l_2+(1-\lambda)x_{12}=x_2,\\ & 0<\lambda<1,\\ & l_2\le x_{12}\le u_2,\\ & l_1\le x_{21}\le u_1,\end{aligned}$$

which can be rewritten as a problem with respect to the single variable λ:

$$\begin{aligned}\min\quad & \phi(\lambda)=(1-\lambda)f\left(l_1,\tfrac{x_2-\lambda l_2}{1-\lambda}\right)+\lambda f\left(\tfrac{x_1-(1-\lambda)l_1}{\lambda},l_2\right)\\ & \tfrac{x_1-l_1}{u_1-l_1}\le\lambda\le\tfrac{u_2-x_2}{u_2-l_2}.\end{aligned}$$

It can be checked that ϕ has nonnegative second derivative over $(0,1)$; i.e., it is a convex function.

In (Locatelli & Schoen, 2010) a method is presented to derive the convex envelope for twice-differentiable bivariate functions satisfying some conditions when $X\subset\mathbb{R}^2$ is a triangle. The required conditions for f are as follows.

Condition 1. The Hessian of f is indefinite in the interior of the triangle.

Condition 2. The restriction of f along each edge of the triangle is either concave or strictly convex.

Condition 3. If f is strictly convex over all the three edges, then there exist two edges such that f is also strictly convex along each segment joining two points belonging to such edges.

For instance, the bilinear function $f(x_1,x_2)=x_1x_2$ satisfies the above conditions for all possible triangles X. In this case edges along which the function is strictly convex are those lying along lines with positive slope, and the two edges satisfying Condition 3 are those forming an obtuse angle. Theorem 4.2 tells us that $G(f,X)$ might contain at most all the vertices of the triangle plus all the edges along which the function f is strictly convex. In fact, the proof of the theorem can be slightly modified to show that $G(f,X)$ is made up *exactly* of all the vertices of the triangle plus all the edges along which the function f is strictly convex. Indeed, under the assumption of strict convexity along an edge, for each point $\mathbf{x}$ along the edge the optimal value of (4.5) can only be attained for $k=1$ and $\mathbf{x}_1=\mathbf{x}$. In addition to the simplest vertex polyhedral case, we need to deal with three more cases—those where the number of edges along which f is strictly convex is equal, respectively, to 1, 2, or 3. The following lemma will be useful in dealing with such cases and, as in the previous cases, it states that in the nonpolyhedral cases, under the given assumptions the optimal solution of (4.5) is always attained with at most two points $\mathbf{x}_1,\mathbf{x}_2\in G(f,X)$. The proof of the lemma is not reported, as it is similar to that of Theorem 4.18.

Lemma 4.19. *Given a triangle with vertices $\mathbf{v}_1,\mathbf{v}_2,\mathbf{v}_3$, let ℓ be the number of edges along which the function is strictly convex. Then the minimum in (4.5) for a point $\mathbf{x}\notin G(f,X)$ and belonging to the triangle is attained:*

1. *with* $\mathbf{x}_i = \mathbf{v}_i$*,* $i = 1,2,3$*, i.e., the points* $\mathbf{x}_i$*'s are the three vertices of the triangle, for* $\ell = 0$*;*
2. *with just two points* $\mathbf{x}_1, \mathbf{x}_2 \in G(f,X)$*, one of which being the vertex opposite to the convex edge and the other lying along the convex edge itself, for* $\ell = 1$*;*
3. *with just two points* $\mathbf{x}_1, \mathbf{x}_2 \in G(f,X)$*, one belonging to a convex edge and the other lying along the other convex edge, for* $\ell = 2$*;*
4. *with just two points* $\mathbf{x}_1, \mathbf{x}_2 \in G(f,X)$*, one belonging to one of the two edges satisfying Condition 3 and the other belonging to the third edge for* $\ell = 3$*.*

The lemma above can be employed to derive the convex envelope of a function f, satisfying the given assumptions, over triangles. It turns out that when $\ell = 0,1$, for a given point within the triangle, we are able to immediately identify the points giving the minimum value in (4.5), while for the cases $\ell = 2,3$ the identification of the points requires the solution of a one-dimensional problem.

Theorem 4.20. *For a given point* $\mathbf{x}$ *in the triangle, the convex envelope of* f *at* $\mathbf{x}$ *is*

$\ell = 0$: *the value at* $\mathbf{x}$ *of the affine function interpolating* f *at the three vertices of the triangle;*

$\ell = 1$: *the value*

$$\lambda^\star f(\mathbf{v}_1) + (1-\lambda^\star) f(\mathbf{h}),$$

where $\mathbf{h}$ *is the point at the intersection of the convex edge and the line through* $\mathbf{v}_1$ *and* $\mathbf{x}$ *and* $\lambda^\star \in [0,1]$ *is the unique value such that*

$$\mathbf{x} = \lambda^\star \mathbf{v}_1 + (1-\lambda^\star)\mathbf{h};$$

$\ell = 2,3$: *the solution of a suitably defined one-dimensional minimization problem.*

Proof.

$\ell = 0$: This is easily seen, e.g., from Theorem 4.12.

$\ell = 1$: The result is an immediate consequence of point 2 in Lemma 4.19, stating that the minimum of (4.5) is attained at the points $\mathbf{v}_1$ and $\mathbf{h}$, where $\mathbf{h}$ is the unique point along the convex edge lying along the line through $\mathbf{v}_1$ and $\mathbf{x}$.

$\ell = 2$: Let $\overline{\mathbf{v}_1\mathbf{v}_3}$ and $\overline{\mathbf{v}_2\mathbf{v}_3}$ be the two edges along which the function f is convex. There exist unique values $\lambda_1, \lambda_2, \lambda_3 \geq 0$, $\sum_{i=1}^3 \lambda_i = 1$, such that

$$\mathbf{x} = \lambda_1 \mathbf{v}_1 + \lambda_2 \mathbf{v}_2 + \lambda_3 \mathbf{v}_3.$$

In view of point 3 in Lemma 4.19, we have that the optimal value of (4.5) is attained at two points,

$$\mathbf{q} = \mu \mathbf{v}_1 + (1-\mu)\mathbf{v}_3 \quad \mu \in [0,1],$$

$$\mathbf{q}' = \alpha \mathbf{v}_2 + (1-\alpha)\mathbf{v}_3 \quad \alpha \in [0,1],$$

such that $\mathbf{x} \in \overline{\mathbf{q}\mathbf{q}'}$, i.e.,

$$\mathbf{x} = \xi \mathbf{q} + (1-\xi)\mathbf{q}', \quad \xi \in [0,1].$$

By substitution we get

$$\begin{aligned}\mathbf{x} &= \xi(\mu \mathbf{v}_1 + (1-\mu)\mathbf{v}_3) + (1-\xi)(\alpha \mathbf{v}_2 + (1-\alpha)\mathbf{v}_3) \\ &= \mu\xi \mathbf{v}_1 + (1-\xi)\alpha \mathbf{v}_2 + [(1-\mu)\xi + (1-\xi)(1-\alpha)]\mathbf{v}_3.\end{aligned}$$

By the uniqueness of the λ_i values, we must have that

$$\begin{aligned}\xi\mu &= \lambda_1, \\ \alpha(1-\xi) &= \lambda_2\end{aligned}$$

and, consequently,

$$\begin{aligned}\xi &= \lambda_1/\mu, \\ \alpha &= \frac{\mu\lambda_2}{\mu - \lambda_1}.\end{aligned}$$

Note that

$$\xi \in [0,1] \Rightarrow \mu \geq \lambda_1, \quad \alpha \in [0,1] \Rightarrow \mu \geq \lambda_1/(1-\lambda_2).$$

Then, in order to detect the value of the convex envelope at $\mathbf{x}$, we need to solve the one-dimensional minimization problem

$$\min_{\mu \in [\lambda_1/(1-\lambda_2),1]} \frac{\lambda_1}{\mu} f(\mathbf{q}) + \left(1 - \frac{\lambda_1}{\mu}\right) f(\mathbf{q}')$$

or, equivalently,

$$\begin{aligned}\min_{\mu \in [\lambda_1/(1-\lambda_2),1]} & \frac{\lambda_1}{\mu} f\left(\mu \mathbf{v}_1 + (1-\mu)\mathbf{v}_3\right) \\ & + \left(1 - \frac{\lambda_1}{\mu}\right) f\left(\frac{\mu\lambda_2}{\mu-\lambda_1}\mathbf{v}_2 + \left(1 - \frac{\mu\lambda_2}{\mu-\lambda_1}\right)\mathbf{v}_3\right).\end{aligned}$$

$\ell = 3$: Let $\overline{\mathbf{v}_1\mathbf{v}_2}$ and $\overline{\mathbf{v}_2\mathbf{v}_3}$ be the two edges satisfying Condition 3. As before, there exist unique values $\lambda_1, \lambda_2, \lambda_3 \geq 0$, $\sum_{i=1}^3 \lambda_i = 1$, such that

$$\mathbf{x} = \lambda_1 \mathbf{v}_1 + \lambda_2 \mathbf{v}_2 + \lambda_3 \mathbf{v}_3.$$

In view of point 4 in Lemma 4.19, we have that the optimal value of (4.5) is attained at two points, one belonging to the edge $\overline{\mathbf{v}_1\mathbf{v}_3}$ different from the two satisfying Condition 3,

$$\mathbf{q} = \mu \mathbf{v}_1 + (1-\mu)\mathbf{v}_3, \quad \mu \in [0,1],$$

and the other one either belonging to the edge $\overline{\mathbf{v}_2\mathbf{v}_3}$,

$$\mathbf{q}' = \alpha\mathbf{v}_2 + (1-\alpha)\mathbf{v}_3, \quad \alpha \in [0,1],$$

or to the edge $\overline{\mathbf{v}_1\mathbf{v}_2}$,

$$\mathbf{q}'' = \beta\mathbf{v}_1 + (1-\beta)\mathbf{v}_2, \quad \beta \in [0,1].$$

Therefore, either $\mathbf{x} \in \overline{\mathbf{q}\mathbf{q}'}$, i.e.,

$$\mathbf{x} = \xi\mathbf{q} + (1-\xi)\mathbf{q}', \quad \xi \in [0,1],$$

or $\mathbf{x} \in \overline{\mathbf{q}\mathbf{q}''}$, i.e.,

$$\mathbf{x} = \eta\mathbf{q} + (1-\eta)\mathbf{q}'', \quad \eta \in [0,1].$$

In particular, by taking the intersection of $\overline{\mathbf{v}_1\mathbf{v}_3}$ with the line through $\mathbf{v}_2$ and $\mathbf{x}$, we have that

$$\mathbf{x} \in \overline{\mathbf{q}\mathbf{q}'} \quad \Rightarrow \quad \mu \in [\lambda_1/(1-\lambda_2), 1],$$

while

$$\mathbf{x} \in \overline{\mathbf{q}\mathbf{q}''} \quad \Rightarrow \quad \mu \in [0, \lambda_1/(1-\lambda_2)].$$

The case $\mathbf{x} \in \overline{\mathbf{q}\mathbf{q}'}$ leads to a result completely analogous to what we have already seen for $\ell = 2$ and to the following one-dimensional function:

$$g_1(\mu) = \frac{\lambda_1}{\mu} f(\mu\mathbf{v}_1 + (1-\mu)\mathbf{v}_3) + \left(1 - \frac{\lambda_1}{\mu}\right) f\left(\frac{\mu\lambda_2}{\mu-\lambda_1}\mathbf{v}_2 + \left(1 - \frac{\mu\lambda_2}{\mu-\lambda_1}\right)\mathbf{v}_3\right).$$

The case $\mathbf{x} \in \overline{\mathbf{q}\mathbf{q}''}$ can be dealt with in a completely similar way. Indeed, by substitution we get

$$\begin{aligned}\mathbf{x} &= \eta(\mu\mathbf{v}_1 + (1-\mu)\mathbf{v}_3) + (1-\eta)(\beta\mathbf{v}_1 + (1-\beta)\mathbf{v}_2)\\ &= [\eta\mu + (1-\eta)\beta]\mathbf{v}_1 + (1-\eta)(1-\beta)\mathbf{v}_2 + (1-\mu)\eta\mathbf{v}_3,\end{aligned}$$

$$\begin{aligned}\eta &= \lambda_3/(1-\mu),\\ \beta &= \frac{\lambda_1 - \mu\lambda_1 - \mu\lambda_3}{1-\mu-\lambda_3},\end{aligned}$$

which leads to the one-dimensional function

$$\begin{array}{rcl} g_2(\mu) & = & \frac{\lambda_3}{1-\mu} f(\mu\mathbf{v}_1 + (1-\mu)\mathbf{v}_3) \\ & + & \left(1 - \frac{\lambda_3}{1-\mu}\right) f\left(\frac{\lambda_1-\mu\lambda_1-\mu\lambda_3}{1-\mu-\lambda_3}\mathbf{v}_1 + \frac{(1-\mu)\lambda_2}{1-\mu-\lambda_3}\mathbf{v}_2\right). \end{array}$$

After defining the one-dimensional function

$$g(\mu) = \begin{cases} g_2(\mu), & \mu \in [0, \lambda_1/(1-\lambda_2)], \\ g_1(\mu), & \mu \in [\lambda_1/(1-\lambda_2), 1], \end{cases}$$

we have that the convex envelope of f at $\mathbf{x}$ is equal to the optimal value of the following one-dimensional problem:

$$\min_{\mu\in[0,1]} g(\mu). \quad \square$$

An example of application of the theorem above follows.

Example 4.21. Let $f(x_1,x_2) = x_1x_2$ and X be the triangle with vertices $\mathbf{v}_1 = (0\ 1)$, $\mathbf{v}_2 = (0\ 0)$, $\mathbf{v}_3 = (2\ 2)$. In this case $\ell = 2$ and the two edges along which the function is strictly convex are $\overline{\mathbf{v}_1\mathbf{v}_3}$ and $\overline{\mathbf{v}_2\mathbf{v}_3}$. As seen in the proof of Theorem 4.20, we need to solve the following one-dimensional problem:

$$\begin{aligned}\min_{\mu\in[\lambda_1/(1-\lambda_2),1]} & \frac{\lambda_1}{\mu} f\left(\mu\mathbf{v}_1 + (1-\mu)\mathbf{v}_3\right) \\ & + \left(1-\frac{\lambda_1}{\mu}\right) f\left(\frac{\mu\lambda_2}{\mu-\lambda_1}\mathbf{v}_2 + \left(1-\frac{\mu\lambda_2}{\mu-\lambda_1}\right)\mathbf{v}_3\right),\end{aligned}$$

where

$$\lambda_1 = x_2 - x_1, \quad \lambda_2 = 1 - x_2 + \frac{x_1}{2}.$$

Elementary but quite tedious computations show that the result is the following function:

$$Conv_{f,X}(\mathbf{x}) = \begin{cases} \frac{x_1^2}{1-x_2+x_1}, & \frac{\sqrt{2}}{2}x_1 + x_2 \le 1, \\ (3-2\sqrt{2})x_1^2 + (6-4\sqrt{2})x_2^2 + (6\sqrt{2}-8)x_1x_2 & \\ -(4\sqrt{2}-6)x_1 + (4\sqrt{2}-6)x_2 & \text{otherwise.} \end{cases} \quad \blacksquare$$

4.2.4 Value and subgradient of a convex envelope at some point

Laraki and Lasserre (Laraki & Lasserre, 2008) (see also (Lasserre, 2009)) proved an interesting result for the computation of the convex envelope and of a subgradient of the convex envelope at some point for a rational function over the convex hull $chull(X)$ of a compact semialgebraic set X. For some continuous function f, the following result is proven in (Laraki & Lasserre, 2008).

Theorem 4.22. *Let $\mathbb{P}(X)$ be the set of probability measures on X. Then, for any $\mathbf{x} \in chull(X)$,*

$$conv_{f,chull(X)}(\mathbf{x}) = \inf\left\{\int f(\mathbf{y})d\mu(\mathbf{y}) \ :\ \int y_i d\mu(\mathbf{y}) = x_i,\ i = 1,\dots,n,\ \mu \in \mathbb{P}(X)\right\}. \tag{4.22}$$

Note that (4.22) is strictly related to (4.5). Indeed, in the latter the points $\mathbf{x}_i$ and the related λ_i values define discrete probability measures over X. Laraki and Lasserre discuss the case of a rational function

$$f(\mathbf{x}) = \frac{p(\mathbf{x})}{q(\mathbf{x})}, \quad q(\mathbf{x}) > 0\ \ \forall\, \mathbf{x} \in X,$$

where p,q are polynomials. The dual of (4.22) when f is the rational function defined above can be proven to be the following problem:

$$\sup_{\gamma\in\mathbb{R},\mathbf{a}\in\mathbb{R}^n}\left\{\gamma+\mathbf{a}^T\mathbf{x}\ :\ \frac{p(\mathbf{y})}{q(\mathbf{y})}\geq\gamma+\mathbf{a}^T\mathbf{y}\ \ \forall\,\mathbf{y}\in X\right\}. \tag{4.23}$$

The dual is strictly related to (4.4): we take the supremum of the value at $\mathbf{x}$ of all the affine functions $\gamma+\mathbf{a}^T\mathbf{y}$ which underestimate f over X. In (Laraki & Lasserre, 2008) it is proven that for rational functions there is no duality gap between (4.22) and (4.23). Moreover, they also prove that the dual is always solvable for any $\mathbf{x}\in chull(X)$, and, given an optimal solution $(\mathbf{a}^\star,\gamma^\star)$ of (4.23), $\mathbf{a}^\star$ is a subgradient of $conv_{f,chull(X)}$ at $\mathbf{x}$. Unfortunately, problems (4.22) and (4.23) are difficult to solve. However, Laraki and Lasserre provide a sequence of convex functions which converge uniformly to $conv_{f,chull(X)}$. The computation of the value and of a subgradient for each of these functions at some point $\mathbf{x}$ requires the solution of a semidefinite program. We refer the reader to the paper for the details but point out that these are strictly related to the material about polynomial programming problems which will be discussed in Section 4.5.

Another result about the computation of the convex envelope at some point has been proven in (Khajavirad & Sahinidis, 2013), where it is shown that, under suitable assumptions, the value of the convex envelope can be obtained by solving a convex problem. More precisely, assume that $G(f,X)=\cup_{i\in I}S_i$, where I is a finite index set and S_i, $i\in I$, are closed convex sets which can be described through a finite set of inequalities, i.e.,

$$S_i=\{\mathbf{x}\in X\ :\ h_{ij}(\mathbf{x})\leq 0,\ j=1,\ldots,r_i\}\ \ \forall\,i\in I,$$

where the functions h_{ij} are convex. Then, it is possible to prove that

$$\begin{array}{lll} conv_{f,X}(\mathbf{x})=\min_{\mathbf{x}^i,\lambda} & \sum_{i\in I}\lambda_i f\left(\frac{\mathbf{x}^i}{\lambda_i}\right) & \\ & \sum_{i\in I}\mathbf{x}^i=\mathbf{x}, & \\ & \sum_{i\in I}\lambda_i=1, & \\ & \lambda_i\geq 0, & i\in I, \\ & \lambda_i h_{ij}\left(\frac{\mathbf{x}^i}{\lambda_i}\right)\leq 0, & i\in I,\ j=1,\ldots,r_i, \end{array} \tag{4.24}$$

which is a convex optimization problem. As pointed out in (Khajavirad & Sahinidis, 2012b), a possible drawback of problem (4.24) which prevents us from using gradient-based convex solvers for its solution is that its objective and/or constraint functions might be nondifferentiable at its optimal solution, particularly when some of the λ_i values are equal to 0. However, under suitable assumptions we are able to overcome this difficulty by deriving the optimal λ_i values analytically. In (Khajavirad & Sahinidis, 2012b) it is shown that this is true when

$$f(\mathbf{x})=h(x_1)g(\mathbf{x}'),\ \text{ where }\ \mathbf{x}'=(x_2,\ldots,x_n),$$

$$X=[a,b]\times[\mathbf{c},\mathbf{d}],\ \ [a,b]\subset\mathbb{R},\ [\mathbf{c},\mathbf{d}]\subset\mathbb{R}^{n-1},$$

and the following conditions are satisfied:

- h is a nonnegative, monotone convex function over $[a,b]$ with one of the following forms:
$$h(x_1) = x_1^t,\ t \in \mathbb{R}\setminus[0,1],\ a \geq 0, \text{ or } h(x_1) = s^{x_1}, s > 0;$$
- g is componentwise concave (i.e., concave with respect to one variable when the values of the other $n-2$ variables are fixed); and
- the restriction of g to $V([\mathbf{c},\mathbf{d}])$ is submodular and nondecreasing (or nonincreasing) with respect to each argument (relaxations of this assumption are possible when g is a bivariate function; see (Khajavirad & Sahinidis, 2012b) for the details).

Note that under the assumptions above, we have that

$$\begin{aligned} G(f,X) = \{&[a,b]\times\{\mathbf{v}_i\},\ \mathbf{v}_i \in V([\mathbf{c},\mathbf{d}])\ :\ g(\mathbf{v}_i) > 0\} \\ &\cup\{\{x_1\}\times\{\mathbf{v}_i\},\ x_1 \in \{a,b\},\ \mathbf{v}_i \in V([\mathbf{c},\mathbf{d}])\ :\ g(\mathbf{v}_i) \leq 0\}. \end{aligned}$$

Therefore, the condition that $G(f,X)$ is the union of finitely many (though exponential with respect to n) closed convex sets is fulfilled. Problem (4.24) can be rewritten as

$$\begin{aligned} \min_{x^i,\lambda_i}\quad & \textstyle\sum_{i\in I\ :\ g(\mathbf{v}_i)<0}\left(\frac{\delta_h}{\delta}x^i + \frac{bh(a)-ah(b)}{\delta}\lambda_i\right)g(\mathbf{v}_i) + \sum_{i\in I\ :\ g(\mathbf{v}_i)>0}\lambda_i h\left(\frac{x^i}{\lambda_i}\right)g(\mathbf{v}_i) \\ & \textstyle\sum_{i\in I} x^i = x_1, \\ & \textstyle\sum_{i\in I}\lambda_i\mathbf{v}_i = \mathbf{x}', \\ & \textstyle\sum_{i\in I}\lambda_i = 1, \\ & \lambda_i a \leq x^i \leq \lambda_i b, \quad i \in I, \\ & \lambda_i \geq 0, \qquad\quad i \in I, \end{aligned}$$

where

- $\delta = b - a$;
- $\delta_h = h(b) - h(a)$;
- $I = \{1,\ldots,2^{n-1}\}$ is the index set of the vertices in $V([\mathbf{c},\mathbf{d}])$.

For some $\mathbf{x} = (x_1\ \mathbf{x}') \in X$, the optimal λ_i values for the problem above are obtained by (i) identifying a simplex Δ_π in the Kuhn's triangulation of $[\mathbf{c},\mathbf{d}]$ (see Definition 4.14) such that $\mathbf{x}' \in \Delta_\pi$; (ii) setting to 0 the λ_i values corresponding to vertices $\mathbf{v}_i \notin V(\Delta_\pi)$; (iii) setting the λ_i values corresponding to vertices $\mathbf{v}_i \in V(\Delta_\pi)$ equal to the coefficients of such vertices in their unique convex combination returning $\mathbf{x}'$. The following example illustrates the results above.

Example 4.23. Let $X = [0,1]^3$ and $f(x_1,x_2,x_3) = x_1^2(1 - x_2x_3)$. Then, the assumptions are fulfilled by taking

- $h(x_1) = x_1^2$ with $[a,b] = [0,1]$;
- $g(x_2,x_3) = (1 - x_2x_3)$ with $[\mathbf{c},\mathbf{d}] = [0,1]^2$.

The vertices of $[0,1]^2$ are

$$\mathbf{v}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \ \mathbf{v}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \ \mathbf{v}_3 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \ \mathbf{v}_4 = \begin{pmatrix} 1 \\ 1 \end{pmatrix},$$

and the two simplices in Kuhn's triangulation are

$$\Delta_1 = chull\{\mathbf{v}_1, \mathbf{v}_3, \mathbf{v}_4\}, \quad \Delta_2 = chull\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_4\}.$$

For some $(x_1\ x_2\ x_3) \in [0,1]^3$ we have

$$\begin{array}{lll} conv_{f,[0,1]^3}(x_1,x_2,x_3) = \min_{y_i,\lambda_i,\ i=1,\dots,4} & \frac{y_1^2}{\lambda_1} + \frac{y_2^2}{\lambda_2} + \frac{y_3^2}{\lambda_3} & \\ & y_1 + y_2 + y_3 + y_4 = x_1, & \\ & \sum_{i=1}^4 \lambda_i \mathbf{v}_i = \begin{pmatrix} x_2 \\ x_3 \end{pmatrix}, & \\ & \sum_{i=1}^4 \lambda_i = 1, & \\ & \lambda_i \geq 0, & i = 1,\dots,4, \\ & 0 \leq y_i \leq \lambda_i, & i = 1,\dots,4. \end{array} \tag{4.25}$$

If $x_2 \geq x_3$ (the case $x_2 \leq x_3$ is completely analogous), in view of the previous discussion the optimal λ_i values are

$$\lambda_1^\star = 1 - x_2, \ \lambda_2^\star = 0, \ \lambda_3^\star = x_2 - x_3, \ \lambda_4^\star = x_3.$$

Thus, problem (4.25) reduces to

$$\begin{array}{ll} conv_{f,[0,1]^3}(x_1,x_2,x_3) = \min & \frac{y_1^2}{1-x_2} + \frac{y_3^2}{x_2-x_3} \\ & y_1 + y_3 + y_4 = x_1, \\ & 0 \leq y_1 \leq 1 - x_2, \\ & 0 \leq y_3 \leq x_2 - x_3, \\ & 0 \leq y_4 \leq x_3. \end{array}$$

If $x_1 \leq x_3$, then we have the optimal solution $y_4^\star = x_1$, $y_1^\star = y_3^\star = 0$ with optimal value equal to 0. If $x_1 > x_3$, then we can set $y_4^\star = x_3$ and we are left with the following problem:

$$\begin{array}{ll} conv_{f,[0,1]^3}(x_1,x_2,x_3) = \min & \frac{y_1^2}{1-x_2} + \frac{y_3^2}{x_2-x_3} \\ & y_1 + y_3 = x_1 - x_3, \\ & 0 \leq y_1 \leq 1 - x_2, \\ & 0 \leq y_3 \leq x_2 - x_3. \end{array}$$

Some easy computations show that the optimal solution of this problem is

$$y_1^\star = \frac{(x_1 - x_3)(1 - x_2)}{1 - x_3}, \quad y_3^\star = \frac{(x_1 - x_3)(x_2 - x_3)}{1 - x_3},$$

with the optimal value

$$\frac{(x_1 - x_3)^2}{1 - x_3}. \quad \blacksquare$$

For the computation of the value of the convex envelope at some point, a result based on the dual formulation (4.23) has been proven in (Locatelli & Schoen, 2010). Let f be a bivariate twice-differentiable function and let X be a polytope P with known vertex set $V(P)$. Assume that f and P satisfy Conditions 1 and 2 stated in Section 4.2.3 (indefiniteness of the Hessian and concavity or strict convexity of f along each edge of P). Then, in (Locatelli & Schoen, 2010) it is shown that, in this case, the dual formulation (4.23) can be solved through the solution of a three-dimensional continuously differentiable convex problem. Moreover, it is also shown that for all indefinite quadratic functions with dimension up to 4 and some polynomial and rational bivariate functions (including the fractional function $\frac{x_2}{x_1}$), the dual problem can be reformulated as a single semidefinite programming problem. In what follows we prove the first result.

Following the dual formulation (4.23), given some point $(x_0\ y_0) \in P$, the value of the convex envelope of f over P is given by the solution of the following optimization problem in the three variables a, b, and c with an infinite number of linear constraints:

$$\begin{aligned} Conv_{f,P}(x_1', x_2') = \quad & \max\ c \\ & f(x_1, x_2) - [a(x_1 - x_1') + b(x_2 - x_2') + c] \geq 0 \quad \forall\ (x_1\ x_2) \in P. \end{aligned}$$

The infinite number of constraints can be substituted by a single one, but it involves a bi-level optimization problem:

$$\begin{aligned} Conv_{f,P}(x_1', x_2') = \quad & \max\ c \\ & \min_{(x_1,x_2)\in P} f(x_1, x_2) - [a(x_1 - x_1') + b(x_2 - x_2') + c] \geq 0. \end{aligned}$$

The optimal solution $(a^\star, b^\star, c^\star)$ of this problem defines a supporting hyperplane for the convex envelope of f over P at point $(x_1'\ x_2')$. In view of Condition 1, the minimum of $f(x_1, x_2) - [a(x_1 - x_1') + b(x_2 - x_2') + c]$ cannot be attained (only) in the interior of P, and is always attained at a vertex of P or along an edge of P such that the restriction of f along the edge is a strictly convex function. Therefore, the constraint

$$\min_{(x_1,x_2)\in P} f(x_1, x_2) - [a(x_1 - x_1') + b(x_2 - x_2') + c] \geq 0$$

can be rewritten as

$$\begin{aligned} & f(x_1^{v_i}, x_2^{v_i}) - [a(x_1^{v_i} - x_1') + b(x_2^{v_i} - x_2') + c] \geq 0 && \forall\ (x_1^{v_i}\ x_2^{v_i}) \in V(P), \\ & \min_{(x_1,x_2)\in e_j} f(x_1, x_2) - [a(x_1 - x_1') + b(x_2 - x_2') + c] \geq 0 && \forall\ e_j \in E'(P), \end{aligned}$$

where $E'(P)$ denotes the set of edges of P along which f is strictly convex. More precisely, we can substitute the requirement for all the vertices in $V(P)$ with that for all the vertices in $V'(P) \subseteq V(P)$, where $V'(P)$ is the set of vertices which do *not* belong to edges in $E'(P)$, i.e., we have

$$\begin{aligned} Conv_{f,P}(x_1', x_2') = \max\ c & \\ & f(x_1^{v_i}, x_2^{v_i}) - [a(x_1^{v_i} - x_1') + b(x_2^{v_i} - x_2') + c] \geq 0 && \forall\ (x_1^{v_i}\ x_2^{v_i}) \in V'(P) \qquad (4.26) \\ & \min_{(x_1,x_2)\in e_j} f(x_1, x_2) - [a(x_1 - x_1') + b(x_2 - x_2') + c] \geq 0 && \forall\ e_j \in E'(P). \end{aligned}$$

The constraints related to the vertices in $V'(P)$ are simple linear ones with respect to the unknowns a,b, and c. Let us now consider the constraints related to the edges in $E'(P)$ that still involve optimization problems. For some $e_j \in E'(P)$, assume that the edge belongs to the line

$$x_2 = m_j x_1 + q_j.$$

Let $f_{e_j}(x_1) = f(x_1, m_j x_1 + q_j)$ denote the restriction of f along the edge $e_j \in E'(P)$ and $z_1^{e_j}, z_2^{e_j}$, $z_1^{e_j} < z_2^{e_j}$, denote the x-coordinates of the two vertices in $V(P)$ defining e_j. To be more precise, we should also consider the case where the edge lies along a line $x_1 = \beta$ for some constant β, but this can be dealt with in a completely analogous way. The constraint related to e_j is then

$$\min_{z_1^{e_j} \le x_1 \le z_2^{e_j}} f_{e_j}(x_1) - [a(x_1 - x_1') + b(m_j x_1 + q_j - x_2')] \ge c. \tag{4.27}$$

Denote by $s_j(a,b)$ the unconstrained minimum point of

$$f_{e_j}(x_1) - (a + bm_j)x_1. \tag{4.28}$$

We also allow for $s_j(a,b) = +\infty$ $(-\infty)$ if the function is decreasing (increasing). In particular, note that if $s_j(a,b) \neq \pm\infty$, then

$$f'_{e_j}(s_j(a,b)) = a + bm_j. \tag{4.29}$$

Therefore, the minimum point in the left-hand side of (4.27) is

$$\begin{cases} z_1^{e_j} & \text{if } s_j(a,b) \le z_1^{e_j}, \\ z_2^{e_j} & \text{if } s_j(a,b) \ge z_2^{e_j}, \\ s_j(a,b) & \text{otherwise,} \end{cases}$$

with the minimum value

$$\begin{cases} f_{e_j}(z_1^{e_j}) - (a+bm_j)z_1^{e_j} + ax_1' + bx_2' - bq_j & \text{if } s_j(a,b) \le z_1^{e_j}, \\ f_{e_j}(z_2^{e_j}) - (a+bm_j)z_2^{e_j} + ax_1' + bx_2' - bq_j & \text{if } s_j(a,b) \ge z_2^{e_j}, \\ f_{e_j}(s_j(a,b)) - (a+bm_j)s_j(a,b) + ax_1' + bx_2' - bq_j & \text{otherwise.} \end{cases} \tag{4.30}$$

Now, set

$$\begin{aligned} l_j^1(a,b) &= f_{e_j}(z_1^{e_j}) - (a+bm_j)z_1^{e_j}, \\ l_j^2(a,b) &= f_{e_j}(z_2^{e_j}) - (a+bm_j)z_2^{e_j}, \\ h_j(a,b) &= f_{e_j}(s_j(a,b)) - (a+bm_j)s_j(a,b). \end{aligned}$$

Taking into account that, in view of the strict convexity assumption, the first derivative of f_{e_j} is increasing along the interval $[z_1^{e_j}, z_2^{e_j}]$, then we can rewrite (4.30) as

$$\begin{cases} l_j^1(a,b) + ax_1' + bx_2' - bq_j & \text{if } f'_{e_j}(z_1^{e_j}) - (a+bm_j) \ge 0, \\ l_j^2(a,b) + ax_1' + bx_2' - bq_j & \text{if } f'_{e_j}(z_2^{e_j}) - (a+bm_j) \le 0, \\ h_j(a,b) + ax_1' + bx_2' - bq_j & \text{otherwise.} \end{cases}$$

If we set

$$g_j(a,b) = \begin{cases} l_j^1(a,b) & \text{if } f'_{e_j}(z_1^{e_j}) - (a+bm_j) \geq 0, \\ l_j^2(a,b) & \text{if } f'_{e_j}(z_2^{e_j}) - (a+bm_j) \leq 0, \\ h_j(a,b) & \text{otherwise,} \end{cases}$$

and

$$r_i(a,b) = f(x_1^{v_i}, x_2^{v_i}) - ax_1^{v_i} - bx_2^{v_i} \quad \forall\, (x_1^{v_i}\ x_2^{v_i}) \in V'(P),$$

we can rewrite (4.26) as

$$\begin{aligned} Conv_{f,P}(x'_1, x'_2) = \max\ & c \\ & r_i(a,b) + ax'_1 + bx'_2 \geq c && \forall\, (x_1^{v_i}\ x_2^{v_i}) \in V'(P), \\ & g_j(a,b) + ax'_1 + bx'_2 - bq_j \geq c && \forall\, e_j \in E'(P). \end{aligned} \tag{4.31}$$

The following theorem proves that this problem is convex and continuously differentiable.

Theorem 4.24. *Under the given conditions, problem* (4.31) *is convex with continuously differentiable constraint functions.*

Proof. We first prove convexity. We need only prove that the functions g_j for all $e_j \in E'(P)$ are concave. We notice that

$$g_j(a,b) = \min_{z_1^{e_j} \leq x_1 \leq z_2^{e_j}} f_{e_j}(x_1) - (a + m_j b)x_1,$$

i.e., g_j is the minimum of an infinite collection of affine functions and is thus a concave function.

Next, we prove that the constraint functions are continuously differentiable. Continuity of the constraint functions is trivially seen. In order to prove that the constraint functions are also continuously differentiable, we need only prove that the functions g_j are continuously differentiable. We notice that the single pieces l_j^1, l_j^2, and h_j, of which the functions g_j are made, are continuously differentiable. Thus, we need only prove continuity of the first derivatives when $s_j(a,b) = z_1^{e_j}$ and when $s_j(a,b) = z_2^{e_j}$. The first partial derivative of h_j with respect to a is

$$\frac{\partial\, h_j}{\partial\, a}(a,b) = f'_{e_j}(s_j(a,b))\frac{\partial\, s_j}{\partial\, a}(a,b) - s_j(a,b) - (a+bm_j)\frac{\partial\, s_j}{\partial\, a}(a,b) = -s_j(a,b),$$

where the last equality follows from (4.29). Since

$$\frac{\partial\, l_j^t}{\partial\, a}(a,b) = -z_t^{e_j}, \quad t = 1,2,$$

we immediately see that the first partial derivative of g_j with respect to a is continuous for $s_j(a,b) = z_1^{e_j}$ and $s_j(a,b) = z_2^{e_j}$. In a completely analogous way we can prove continuity for the first partial derivative of g_j with respect to b, thus concluding the proof. □

While the proposed approach aims at computing the value and a supporting hyperplane of the convex envelope of a function at some point belonging to a polytope, in some cases it can also be employed to derive an analytic formula for the convex envelope. In this case it might be convenient to split each constraint related to an edge $e_j \in E'(P)$ into three different sets of constraints: one pair of linear constraints,

$$\begin{aligned} &l_1^j(a,b) + ax_1' + bx_2' - bq_j \geq c, \\ &f'_{e_j}(z_1^{e_j}) - (a + bm_j) \geq 0; \end{aligned}$$

another pair of linear constraints,

$$\begin{aligned} &l_2^j(a,b) + ax_1' + bx_2' - bq_j \geq c, \\ &f'_{e_j}(z_2^{e_j}) - (a + bm_j) \leq 0; \end{aligned}$$

and a third set with two linear constraints and a convex constraint,

$$\begin{aligned} &h_j(a,b) + ax_1' + bx_2' - bq_j \geq c, \\ &f'_{e_j}(z_1^{e_j}) - (a + bm_j) \leq 0, \\ &f'_{e_j}(z_2^{e_j}) - (a + bm_j) \geq 0. \end{aligned}$$

Taking into account that such splitting into three groups of constraints needs to be done for all edges in $E'(P)$, if we denote by ℓ the cardinality of $E'(P)$, we see that in this case we need to solve (at most) 3^ℓ three-dimensional subproblems in order to compute the value of the convex envelope of f over P at $(x_1'\ x_2')$. We illustrate this with an example.

A domain over which it is reasonable to compute the convex envelope for a bivariate function is a polytope P obtained by intersecting a box with a half-plane $y + mx \leq q$. Indeed, by feasibility- (respectively, optimality-)based domain reduction techniques (see Section 5.5), one might compute the upper bound q for the value of $x_2 + mx_1$ over the feasible region (respectively, over the set of optimal solutions) of the problem at hand, e.g., by maximizing q over a relaxation of the feasible region (respectively, of the set of optimal solutions). Then, P is a better approximation (with respect to the box) of the set of values which can be attained by the variables x_1 and x_2 at feasible (respectively, optimal) solutions of the problem. In view of this observation, we discuss the following example.

Example 4.25. Let $f(x_1,x_2) = \frac{x_2}{x_1}$ and

$$P = \left\{(x_1\ x_2) \in \mathbb{R}^2\ :\ -x_1 + 2x_2 \leq 2,\ 1 \leq x_1 \leq 2,\ 0 \leq x_2 \leq 2\right\}.$$

The polytope P has the four vertices

$$\mathbf{v}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \mathbf{v}_2 = \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \quad \mathbf{v}_3 = \begin{pmatrix} 1 \\ \frac{3}{2} \end{pmatrix}, \quad \mathbf{v}_4 = \begin{pmatrix} 2 \\ 2 \end{pmatrix}.$$

For the following development we subdivide this polytope in two subregions, namely, the two triangles

- T_1 with vertices $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_4$;
- T_2 with vertices $\mathbf{v}_1, \mathbf{v}_3, \mathbf{v}_4$.

After adding the additional vertex $\mathbf{v}_5 = \begin{pmatrix} 2 \\ \frac{1}{2} \end{pmatrix}$, the triangle T_1 is further subdivided into the two triangles

- T_1' with vertices $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_5$;
- T_1'' with vertices $\mathbf{v}_1, \mathbf{v}_5, \mathbf{v}_4$.

There is a single edge, the one joining vertices $\mathbf{v}_3$ and $\mathbf{v}_4$, along which f is strictly convex. Therefore, given $(x_1' \; x_2') \in P$, we need to solve the three subproblems

$$\begin{aligned} \max \;& c \\ & c \leq -a + ax_1' + bx_2', \\ & c \leq -2a + ax_1' + bx_2', \\ & c \leq \frac{1}{2} + 2\sqrt{-(a + \frac{1}{2}b)} - b + ax_1' + bx_2', \\ & \frac{1}{4} \leq -\left(a + \frac{1}{2}b\right) \leq 1, \end{aligned} \tag{4.32}$$

$$\begin{aligned} \max \;& c \\ & c \leq -a + ax_1' + bx_2' \\ & c \leq -2a + ax_1' + bx_2', \\ & c \leq 1 - 2a - 2b + ax_1' + bx_2', \\ & -\left(a + \frac{1}{2}b\right) \leq \frac{1}{4}, \end{aligned} \tag{4.33}$$

$$\begin{aligned} \max \;& c \\ & c \leq -a + ax_1' + bx_2' \\ & c \leq -2a + ax_1' + bx_2', \\ & c \leq \frac{3}{2} - \frac{3}{2}b - a + ax_1' + bx_2', \\ & -\left(a + \frac{1}{2}b\right) \geq 1. \end{aligned} \tag{4.34}$$

Here we only report the optimal values without deriving them. The optimal value for (4.32), which is explicitly derived in (Locatelli & Schoen, 2010), is equal to

$$\begin{cases} -\frac{1}{2}x_2', & (x_1' \; x_2') \in T_1', \\ -\frac{2}{3}x_1' + \frac{5}{6}x_2' + \frac{2}{3}, & (x_1' \; x_2') \in T_1'', \\ \frac{x_2'(1-x_1'+2x_2')}{2(x_1'+x_2'-1)}, & (x_1' \; x_2') \in T_2. \end{cases} \tag{4.35}$$

Similar (but somewhat simpler) computations lead to the optimal value of the subproblem (4.33),

$$\begin{cases} \frac{1}{2}x_2', & (x_1'\ x_2') \in T_1, \\ -\frac{2}{3}x_1' + \frac{5}{6}x_2' + \frac{2}{3}, & (x_1'\ x_2') \in T_2, \end{cases}$$

while the subproblem (4.34) has optimal value

$$\begin{cases} -2x_2', & (x_1'\ x_2') \in T_1', \\ -\frac{3}{2}x_1' + x_2' + \frac{3}{2}, & (x_1'\ x_2') \in T_2 \cup T_1''. \end{cases}$$

Taking the maximum of all these optimal values (and after a comparison of the different functions over the different subregions), we end up with the following convex envelope:

$$Conv_{f,P}(x_1', x_2') = \begin{cases} \frac{1}{2}x_2', & (x_1'\ x_2') \in T_1, \\ \frac{x_2'(1-x_1'+2x_2')}{2(x_1'+x_2'-1)}, & (x_1'\ x_2') \in T_2. \end{cases}$$

Note that, as expected from the theory, the convex envelope is equal to the function over the convex edge. ∎

As previously mentioned, when the function f is an indefinite quadratic function with dimension up to 4, we might alternatively solve a single semidefinite programming problem. Indeed, the nonnegative conditions along convex faces (not necessarily edges) of P can be reformulated as semidefinite conditions. A similar reformulation is also possible for bivariate polynomial and rational functions satisfying Conditions 1 and 2. We refer to (Locatelli & Schoen, 2010) for the details.

4.2.5 Convex envelopes over polytopes and polyhedral subdivisions

In Section 4.2.2 we have seen that a vertex polyhedral convex envelope over some polytope X is characterized by a triangulation of X. Triangulations are a special case of polyhedral subdivisions.

Definition 4.26. *Given a polytope X, a* polyhedral subdivision *for X is a finite collection $\{S_i\ :\ i \in I\}$, where*

- *S_i is a polytope for each $i \in I$;*
- *$\cup_{i\in I} S_i = X$;*
- *$S_i \cap S_j$, $i \neq j$, is a (possibly empty) face both for S_i and for S_j.*

If $V(S_i) \subseteq V(X)$ for each $i \in I$, we say that the polyhedral subdivision does not add vertices.

It turns out that in some cases we can characterize also nonpolyhedral convex envelopes through polyhedral subdivisions which do not add vertices. In (Tawarmalani et al., 2012) functions with the following form are considered:

$$f(\mathbf{x}) = x_1 g(\mathbf{x}'),\ \ \mathbf{x}' = (x_2 \ldots x_n),$$

where g is convex and nonincreasing , while $X = [0,1]^n$. Let $N = \{2,\dots,n\}$ and for some $J \subseteq N$ define the polytope

$$S_J = \{\mathbf{x} \in \mathbb{R}^n \ : \ 0 \le x_i \le x_1 \le x_j \le 1, \ \forall\, i \in J, \ j \in N \setminus J\}.$$

The collection $\{S_J \ : \ J \subseteq N\}$ defines a polyhedral subdivision of $[0,1]^n$ which does not add vertices. Finally, for $\mathbf{x} \in S_J$, with $x_1 > 0$, let $\mathbf{z_x} \in \mathbb{R}^{n-1}$ be the vector whose components are

$$[\mathbf{z_x}]_i = \begin{cases} 1 & \text{if } i \in N \setminus J, \\ \frac{x_i}{x_1} & \text{if } i \in J. \end{cases}$$

Then, in (Tawarmalani et al., 2012) it is proven that for each $\mathbf{x} = (x_1 \ \mathbf{x}') \in S_J$ we have that

$$conv_{f,X}(\mathbf{x}) = \begin{cases} x_1 g(\mathbf{z_x}) & \text{if } x_1 > 0, \\ 0 & \text{if } x_1 = 0. \end{cases}$$

We illustrate this result through a simple example taken from that paper.

Example 4.27. Let $n = 2$ and $g(x_2) = \frac{1}{1+x_2}$. Then, we need only consider the two polytopes

$$S_\emptyset = \{(x_1 \ x_2) \ : \ 0 \le x_1 \le x_2 \le 1\},$$

$$S_{\{2\}} = \{(x_1 \ x_2) \ : \ 0 \le x_2 \le x_1 \le 1\}.$$

Then, for $(x_1 \ x_2) \in S_\emptyset$ we have

$$conv_{f,[0,1]^2}(x_1, x_2) = \begin{cases} \frac{x_1}{2} & \text{if } x_1 > 0, \\ 0 & \text{if } x_1 = 0, \end{cases}$$

while for $(x_1 \ x_2) \in S_{\{2\}}$ we have

$$conv_{f,[0,1]^2}(x_1, x_2) = \begin{cases} \frac{x_1^2}{x_1+x_2} & \text{if } x_1 > 0, \\ 0 & \text{if } x_1 = 0. \end{cases}$$

In a compact form, for $(x_1 \ x_2) \in [0,1]^2$ we have

$$conv_{f,[0,1]^2}(x_1, x_2) = \begin{cases} \frac{x_1^2}{x_1+\min\{x_1,x_2\}} & \text{if } x_1 > 0, \\ 0 & \text{if } x_1 = 0. \end{cases} \quad \blacksquare$$

We finally remark that in (Locatelli & Schoen, 2010) an example is presented for which the convex envelope of the bilinear function over a polytope X is still characterized by a polyhedral subdivision but in that case vertices are added to those in $V(X)$. In fact, in the working paper (Locatelli, 2012b), it is shown that a polyhedral subdivision (possibly adding vertices) always characterizes $conv_{f,X}$ if f and the polytope X satisfy Conditions 1 and 2 stated in Section 4.2.3 (i.e., indefiniteness of the Hessian and concavity or strict convexity of f along each edge of X). In the same paper, it is shown that over each member of the polyhedral subdivision, the convex envelope of the bilinear and fractional functions has one of three distinct possible functional forms: (i) linear; (ii) quadratic; (iii) ratio between quadratic and linear. For other functions and polytopes satisfying the Conditions 1 and 2 stated in Section 4.2.3, the functional form of the convex envelope over each member of the polyhedral subdivision is implicitly defined.

4.2.6 Convex envelopes for the sum of functions

Let f_1 and f_2 be two functions defined over a set $X \subseteq \mathbb{R}^n$. In general we have that

$$conv_{f_1+f_2,X}(\mathbf{x}) \geq conv_{f_1,X}(\mathbf{x}) + conv_{f_2,X}(\mathbf{x}) \quad \forall\, \mathbf{x} \in X. \tag{4.36}$$

This is an immediate consequence of the fact that the sum $conv_{f_1,X} + conv_{f_2,X}$ is a convex underestimator for $f_1 + f_2$ (the sum of convex functions is itself a convex function). Equality in (4.36) would be a desirable property but, unfortunately, we may have strict inequality. Indeed, consider the following simple example:

$$f_1(x_1,x_2) = x_1x_2, \quad f_2(x_1,x_2) = -x_1x_2,$$

and $X = [0,1]^2$. Obviously, $conv_{f_1+f_2,X}(x_1,x_2) = 0$ for all $(x_1\ x_2) \in X$, while

$$conv_{f_1,X}(x_1,x_2) = \max\{0, x_1 + x_2 - 1\}, \quad conv_{f_2,X}(x_1,x_2) = \max\{-x_1,-x_2\},$$

so that, e.g.,

$$conv_{f_1+f_2,X}(1/2,1/2) = 0 > conv_{f_1,X}(1/2,1/2) + conv_{f_2,X}(1/2,1/2) = -1/2.$$

However, there are some cases where equality holds. An easy case where this happens is when one of the two functions is affine, but the same is not true if one of the two functions is convex, as can be seen by taking

$$f_1(x_1,x_2) = x_1^2 + x_2^2, \quad f_2(x_1,x_2) = 2x_1x_2.$$

Another relatively simple case where we have equality is when the function $f_1 + f_2$ is separable, i.e.,

$$f_1 + f_2 : X_1 \times X_2 \to \mathbb{R}, \quad f_1 : X_1 \to \mathbb{R}, \quad f_2 : X_2 \to \mathbb{R}.$$

Indeed, it can be easily proven that the conjugate $(f_1 + f_2)^\star$ and second conjugate $(f_1 + f_2)^{\star\star}$ functions of a separable sum $f_1 + f_2$ are equal to the sum of the conjugate functions $f_1^\star + f_2^\star$ and second conjugate functions $f_1^{\star\star} + f_2^{\star\star}$, respectively. Then, equality in (4.36) follows from (4.6).

Other conditions under which equality holds have been proven in (Tardella, 2008). We first need to introduce some definitions and notation. For some polytope P with vertex set $V(P)$, we consider a function f which is edge-concave (and, thus, admits a vertex polyhedral convex envelope) over P. We denote by $\{f_i \ : \ f_i$ is affine, $i \in I\}$ the *facet representation* of $conv_{f,P}$, i.e., the set of affine functions defining the convex envelope. Next, we denote by

$$F_i = \{\mathbf{x} \in P \ : \ conv_{f,P}(\mathbf{x}) = f_i(\mathbf{x})\}, \quad i \in I,$$

the *linearity domains* of the convex envelope. We have that each set F_i is a polytope and $P = \cup_{i \in I} F_i$. Vertex polyhedrality of $conv_{f,P}$ implies that $V(F_i) \subseteq V(P)$ for each $i \in I$. Now we are ready for the proof of the following theorem.

Theorem 4.28. *Let P be a polytope and let f, g be edge-concave functions over P. Let F_i, $i \in I$, be the linearity domains of f, and G_j, $j \in J$, be the linearity domains of g. Then, the following conditions are equivalent:*

(i) *$conv_{f,P} + conv_{g,P}$ is vertex polyhedral;*

(ii) *$conv_{f,P} + conv_{g,P} = conv_{f+g,P}$;*

(iii) *$V(F_i \cap G_j) \subseteq V(P)$ for all $i \in I$, $j \in J$.*

Proof. **(i) ⇒ (ii)** Let $\{h_k \ : \ k \in K\}$ be the facet representation of the vertex polyhedral function $conv_{f,P} + conv_{g,P}$, and

$$H_k = \{\mathbf{x} \in P \ : \ conv_{f,P}(\mathbf{x}) + conv_{g,P}(\mathbf{x}) = h_k(\mathbf{x})\}, \ \ k \in K,$$

be the corresponding linearity domains. By vertex polyhedrality, we have that $V(H_k) \subseteq V(P)$ for any $k \in K$. Therefore, for all $k \in K$

$$conv_{f,P}(\mathbf{v}) + conv_{g,P}(\mathbf{v}) = f(\mathbf{v}) + g(\mathbf{v}) \quad \forall\, \mathbf{v} \in V(H_k).$$

Indeed, it follows from (4.5) that the convex envelope of some function over a polytope P is always equal to the function itself at the vertices of the polytope. By the same observation we can also see that

$$conv_{f+g,P}(\mathbf{v}) = f(\mathbf{v}) + g(\mathbf{v}) \quad \forall\, \mathbf{v} \in V(P).$$

Then, we can conclude that for each $k \in K$

$$conv_{f,P}(\mathbf{v}) + conv_{g,P}(\mathbf{v}) = conv_{f+g,P}(\mathbf{v}) \quad \forall\, \mathbf{v} \in V(H_k).$$

Moreover, we have that

$$conv_{f,P}(\mathbf{x}) + conv_{g,P}(\mathbf{x}) \leq conv_{f+g,P}(\mathbf{x}) \quad \forall\, \mathbf{x} \in P.$$

Indeed, as already observed, $conv_{f,P} + conv_{g,P}$ is a convex function underestimating $f+g$ over P and, thus, by definition of convex envelope, is not larger than $conv_{f+g,P}$. Therefore, we have that for each $k \in K$, the function $conv_{f,P} + conv_{g,P}$

- is an affine function over H_k;
- underestimates $conv_{f+g,P}$ over H_k;
- is equal to $conv_{f+g,P}$ at the vertices $V(H_k)$.

By convexity, we can conclude that

$$conv_{f,P}(\mathbf{x}) + conv_{g,P}(\mathbf{x}) = conv_{f+g,P}(\mathbf{x}) \quad \forall\, \mathbf{x} \in H_k.$$

Finally, (ii) follows from $\cup_{k\in K} H_k = P$.

(ii) $\Rightarrow$ **(iii)** By contradiction we assume that there exists $\overline{\mathbf{v}} \in V(F_i \cap G_j) \setminus V(P)$. Since $\overline{\mathbf{v}} \in F_i \cap G_j$, we have that

$$conv_{f,P}(\overline{\mathbf{v}}) + conv_{g,P}(\overline{\mathbf{v}}) = f_i(\overline{\mathbf{v}}) + g_j(\overline{\mathbf{v}}).$$

Edge-concavity of f, g over P also implies edge-concavity of $f+g$ over P, so that $f+g$ admits a vertex polyhedral convex envelope. In particular, this implies that there exists $\mathbf{v}_1, \ldots, \mathbf{v}_R \in V(P)$, $R \geq 2$, such that

$$conv_{f+g,P}(\overline{\mathbf{v}}) = \sum_{r=1}^{R} \lambda_r (f(\mathbf{v}_r) + g(\mathbf{v}_r)),$$

where

- $\lambda_r > 0$, $r = 1, \ldots, R$;
- $\sum_{r=1}^{R} \lambda_r = 1$;
- $\sum_{r=1}^{R} \lambda_r \mathbf{v}_r = \overline{\mathbf{v}}$.

Since $\overline{\mathbf{v}}$ is a vertex of $F_i \cap G_j$, there must exist $\mathbf{v}_t$, $t \in \{1, \ldots, R\}$, such that $\mathbf{v}_t \notin F_i \cap G_j$ and, by definition of F_i and G_j, at least one of the inequalities

$$f(\mathbf{v}_t) > f_i(\mathbf{v}_t), \quad g(\mathbf{v}_t) > g_j(\mathbf{v}_t)$$

must hold. Therefore, also recalling that

$$f(\mathbf{v}_r) \geq f_i(\mathbf{v}_r), \quad g(\mathbf{v}_r) \geq g_j(\mathbf{v}_r), \quad r = 1, \ldots, R,$$

and that functions f_i and g_j are affine, we have that

$$\begin{aligned} conv_{f,P}(\overline{\mathbf{v}}) + conv_{g,P}(\overline{\mathbf{v}}) = f_i(\overline{\mathbf{v}}) + g_j(\overline{\mathbf{v}}) &= \textstyle\sum_{r=1}^{R} \lambda_r (f_i(\mathbf{v}_r) + g_j(\mathbf{v}_r)) \\ &< \textstyle\sum_{r=1}^{R} \lambda_r (f(\mathbf{v}_r) + g(\mathbf{v}_r)) = conv_{f+g,P}(\overline{\mathbf{v}}), \end{aligned}$$

which contradicts (ii).

(iii) ⇒ (i) As before, we denote by H_k the linearity domains of $conv_{f,P}+conv_{g,P}$. For each $k\in K$ we have that

$$H_k=\cup_{(i,j)\in A_k}F_i\cap G_j,$$

where $A_k\subseteq I\times J$ and is an equivalence class of the following equivalence relation:

$$(i,j)\equiv(s,t)\quad\Leftrightarrow\quad f_i+g_j=f_s+g_t.$$

Then, every vertex of H_k must be a vertex of $F_i\cap G_j$ for some $(i,j)\in A_k$. By (iii), this implies that $V(H_k)\subseteq V(P)$, so that (i) is true. □

Now, let $P_1\subseteq\mathbb{R}^n$, $P_2\subseteq\mathbb{R}^m$, and $P_3\subseteq\mathbb{R}^p$ be three polytopes. Let

$$f,g:P=P_1\times P_2\times P_3\to\mathbb{R}$$

be such that for all $(\mathbf{x},\mathbf{y},\mathbf{z})\in P_1\times P_2\times P_3$

$$f(\mathbf{x},\mathbf{y},\mathbf{z})=\bar{f}(\mathbf{x},\mathbf{y}),\quad g(\mathbf{x},\mathbf{y},\mathbf{z})=\bar{g}(\mathbf{x},\mathbf{z}).$$

Therefore, the common variables of f and g are only the $\mathbf{x}$ ones. The linearity domains of f and g have the form

$$F_i=F_i^{P_1}\times F_i^{P_2}\times P_3,\quad G_j=G_j^{P_1}\times P_2\times G_j^{P_3}.$$

Then, in (Tardella, 2008) the following corollary of Theorem 4.28 is stated. Such corollary extends previous results proven in (Meyer & Floudas, 2005a; Rikun, 1997).

Corollary 4.29. *Let f,g be defined as above. Then, if $\bar{f},\bar{g}$ are edge-concave functions, respectively, over $P_1\times P_2$ and $P_1\times P_3$, we have that $conv_{f+g,P}=conv_{f,P}+conv_{g,P}$ if and only if for all linearity domains F_i of $conv_{f,P}$ and G_j of $conv_{g,P}$, we have that*

$$V(F_i^{P_1}\cap G_j^{P_1})\subseteq V(P_1).$$

4.2.7 Some specific results

While we have previously given some general results about convex envelopes, here we briefly review some results about specific functions over specific sets. In some cases we report formulae but without giving an explicit proof for them. However, we usually indicate how the theory discussed in the previous subsections can be applied to derive them.

The most widely studied function is the bilinear function x_1x_2. For this function the convex envelope has been derived over different sets.

Rectangles: The result in (4.12) over the unit square can be easily generalized into the following one over the rectangle $X=[a_1,b_1]\times[a_2,b_2]$:

$$conv_{x_1x_2,[a_1,b_1]\times[a_2,b_2]}(x_1,x_2)=\max\{b_2x_1+b_1x_2-b_1b_2,a_2x_1+a_1x_2-a_1a_2\},$$

while for the concave envelope we have

$$conc_{x_1x_2,[a_1,b_1]\times[a_2,b_2]}(x_1,x_2)=\min\{a_2x_1+b_1x_2-b_1a_2,b_2x_1+a_1x_2-a_1b_2\}.$$

These results were first presented in (McCormick, 1976).

D-polytopes: These are polytopes whose edges all have nonpositive slope, so that the bilinear function is concave along them. In this case we can exploit Theorem 4.2 to see that the convex envelope is vertex polyhedral. Such a case has been studied in (Sherali & Alameddine, 1992).

Triangles: In (Linderoth, 2005) the case of triangles with a single edge over which the bilinear function is convex and not concave has been discussed. In (Anstreicher & Burer, 2010) a representation of the convex envelope over general triangles through doubly nonnegative matrices (i.e., matrices which are both semidefinite and nonnegative) is given. This can also be extended to a general polytope after a triangulation of the polytope itself. As already previously seen, in (Locatelli & Schoen, 2010) a general theory for the derivation of convex envelopes of some bivariate functions over triangles is presented, which, in particular, allows us to derive all the formulae for the convex envelopes of the bilinear function over triangles.

Another widely studied case is the rational function x_1/x_2 over rectangles. For rectangles $X = [a_1,b_1] \times [a_2,b_2]$ lying in the nonnegative orthant (in particular, $a_1 \geq 0$, $a_2 > 0$), a formula for the convex envelope of the rational function has been reported in (Zamora & Grossmann, 1999). We can also employ the results discussed in (Tawarmalani & Sahinidis, 2001) and (Jach et al., 2008) and reported in Subsection 4.2.3 to derive such a formula. Indeed, for fixed x_1 value the rational function is a convex one, while for fixed x_2 value it is linear (i.e., both convex and concave). Moreover, the Hessian of the function is indefinite over the box. Thus, both the theory in (Jach et al., 2008) about 1-convex functions and the theory discussed in (Tawarmalani & Sahinidis, 2001) can be applied. The rather complicated formula is the following:

$$\begin{aligned} conv_{f,X}(x_1,x_2) &= \frac{b_1-x_1}{b_1-a_1}\frac{a_1}{\max\left\{a_2,\frac{b_2-x_2}{b_1-x_1}(a_1-x_1)+x_2,\frac{x_2\sqrt{a_1}(b_1-a_1)}{(b_1-x_1)\sqrt{a_1}+(x_1-a_1)\sqrt{b_1}}\right\}} \\ &+ \frac{x_1-a_1}{b_1-a_1}\frac{b_1}{\min\left\{\frac{x_2-a_2}{x_1-a_1}(b_1-x_1)+x_2,b_2,\frac{x_2\sqrt{b_1}(b_1-a_1)}{(b_1-x_1)\sqrt{a_1}+(x_1-a_1)\sqrt{b_1}}\right\}}. \end{aligned}$$

In (Tawarmalani & Sahinidis, 2001) a formula for the case of rectangles not lying in the nonnegative orthant (in particular, with $a_1 < 0$) has also been derived. In (Benson, 2004) the vertex polyhedral envelopes of the rational function over special quadrilaterals such as parallelograms and trapezoids are given.

In (Meyer & Floudas, 2004) the convex envelope of the trilinear function $x_1x_2x_3$ over a hyper-rectangle $X = [a_1,b_1] \times [a_2,b_2] \times [a_3,b_3]$ with $a_ib_i < 0$ for some $i = 1,2,3$ is derived. We do not report here the quite long formulae for the different subcases discussed in the paper, but we observe that Theorem 4.2 shows that the convex envelope is vertex polyhedral, so that, following Theorem 4.12, the derivation of the formulae can be reduced to the detection of a proper triangulation of X.

For monomial functions of odd degree over intervals $[a,b]$, $a < 0$, $b > 0$, the non polyhedral convex envelope has been computed in (Liberti & Pantelides, 2003).

Different results also exist for the convex envelope over the unit hypercube $X = [0,1]^n$ of some special cases of *multilinear functions*, i.e., functions defined as

$$f(x_1,\ldots,x_n) = \sum_{j=1}^{t} f_j(x_1,\ldots,x_n),$$

where

$$f_j(x_1,\ldots,x_n) = \alpha_j \prod_{i\in I_j} x_i$$

and $I_j \subseteq \{1,\ldots,n\}$. It is easily seen by Theorem 4.2 that such functions admit a vertex polyhedral convex envelope over the unit hypercube. Indeed, by definition, once we fix the value of $n-1$ variables, the function is affine with respect to the remaining variable. For the case

$$t = 1 \text{ and } \alpha_1 = \alpha < 0,\ I_1 = \{1,\ldots,n\},$$

in (Crama, 1993) the following formula has been derived:

$$conv_{f,[0,1]^n} = \max_{j=1,\ldots,n} \{\alpha x_j\}.$$

Exploiting the results discussed in Subsection 4.2.6, in (Meyer & Floudas, 2005a) it is shown that for $\alpha_j < 0$, $j = 1,\ldots,t$, we have that

$$conv_{f,[0,1]^n}(x_1,\ldots,x_n) = \sum_{j=1}^{t} conv_{f_j,[0,1]^n}(x_1,\ldots,x_n).$$

For the case

$$\begin{array}{c} \alpha_j = 1, \quad j = 1,\ldots,t, \\ \{I_j\ :\ j = 1,\ldots,t\} \text{ are all possible subsets of } \{1,\ldots,n\} \text{ with cardinality } m \leq n; \end{array} \tag{4.37}$$

in (Rikun, 1997) the convex envelope for $m = 2$ is derived; while in (Sherali, 1997) the result is extended to all $m \leq n$. Exploiting the fact that a multilinear function admits a vertex polyhedral convex envelope over the unit hypercube, we can conclude that its convex envelope can be derived from formulae (4.9)–(4.11). Then, in (Sherali, 1997) the vertices of (4.10) for the case (4.37) are computed and the following formula for the convex envelope is reported:

$$\begin{aligned} &conv_{f,[0,1]^n}(x_1,\ldots,x_n) \\ &\quad = \max\left\{0, \binom{k}{m-1} \textstyle\sum_{i=1}^n x_i - (m-1)\binom{k+1}{m},\ k = m-1,\ldots,n-1\right\}. \end{aligned}$$

In the same paper a similar approach is also employed to derive a formula for the concave envelope of multilinear functions satisfying (4.37) over the unit hypercube.

4.3 Reformulation-linearization technique

In Section 4.2.2 we derived the formula (4.12) for the convex envelope of the bilinear term x_1x_2 over the unit square. By the same approach we could easily derive the same formula for the convex envelope of a bilinear term over a general rectangle $R = [\ell_1,u_1] \times [\ell_2,u_2]$ (see Section 4.2.7). This can also be obtained through a different approach. Over the rectangle R we have that

$$\begin{aligned} (x_1-\ell_1)(x_2-\ell_2) \geq 0 &\quad\Rightarrow\quad x_1x_2 \geq \ell_2x_1 + \ell_1x_2 - \ell_1\ell_2, \\ (u_1-x_1)(u_2-x_2) \geq 0 &\quad\Rightarrow\quad x_1x_2 \geq u_2x_1 + u_1x_2 - u_1u_2. \end{aligned}$$

We can linearize the above inequalities by substituting the bilinear term x_1x_2 with the single variable x_{12}, so that

$$x_{12} \geq \max\{\ell_2x_1 + \ell_1x_2 - \ell_1\ell_2, u_2x_1 + u_1x_2 - u_1u_2\}.$$

The right-hand side of the latter inequality is the convex envelope of the bilinear term over R. This is an example of application of the *reformulation-linearization technique* (RLT) introduced in (Sherali & Tuncbilek, 1992) and later developed in different papers, e.g., (Audet, Hansen, Jaumard, & Savard, 2000; Sherali & Tuncbilek, 1995; Sherali & Fraticelli, 2002). RLT is employed to define, through the addition of variables (like x_{12} in the previous example), linear relaxations for polynomial programming problems

$$\begin{array}{ll} \min & p_0(\mathbf{x}) = \sum_{k=1}^{t} \alpha_{0j} \prod_{i=1}^{n} x_i^{j_{0i}} \\ & p_r(\mathbf{x}) = \sum_{k=1}^{t} \alpha_{rj} \prod_{i=1}^{n} x_i^{j_{ri}} \geq 0, \quad r = 1,\ldots,m, \\ & p_r(\mathbf{x}) = \sum_{k=1}^{t} \alpha_{rj} \prod_{i=1}^{n} x_i^{j_{ri}} = 0, \quad r = m+1,\ldots,m+s, \end{array} \tag{4.38}$$

where the objective and constraint functions are polynomial ones with integer exponents (in fact, RLT is also applied to other problems, such as polynomial problems with rational exponents, and we refer the reader to, e.g., (Sherali, 2002) for a discussion about such problems). In the following subsections we will first present the simplest RLT constraints—those based on bound factors (Subsection 4.3.1). Then, we present some other RLT constraints, distinguishing between those which can be generated a priori (Subsection 4.3.2), and those which can be generated after solving a relaxation of the problem (Subsection 4.3.3). Finally, we conclude the section with some more remarks about RLT (Subsection 4.3.4).

4.3.1 Bound factor based RLT constraints

The simplest application of RLT, already illustrated by the previous example, is based on *bound factors*. Let ℓ_i, u_i, $i = 1,\ldots,n$, denote, respectively, known lower and upper bounds for variables x_i over the feasible region X (assumed to be bounded) of the polynomial programming problem (4.38). Then

$$x_i - \ell_i \geq 0, \quad u_i - x_i \geq 0, \quad i = 1,\ldots,n, \tag{4.39}$$

over X. Terms $(x_i - \ell_i)$ and $(u_i - x_i)$ are called, respectively, lower and upper bound factors. Now, let J_ℓ be a set of indices for lower bound factors, possibly with some repetitions (i.e., the same index i might occur more than once in J_ℓ). Similarly, let J_u be a set of indices for upper bound factors. Then, in view of (4.39) the following constraint is valid over X:

$$\prod_{i \in J_\ell}(x_i - \ell_i) \prod_{i \in J_u}(u_i - x_i) \geq 0. \tag{4.40}$$

The addition of constraints of this type represents the *reformulation phase*. Next, we expand the left-hand side of the above inequality and substitute each product term $x_{i_1}x_{i_2}\cdots x_{i_k}$ with a single variable $x_{i_1i_2\ldots i_k}$, thus linearizing the inequality. This is called the *linearization phase*, and the final linear constraint is called a RLT constraint. The following example will further clarify the technique.

Example 4.30. Let us consider the following polynomial programming problem:

$$\begin{aligned} \min \quad & x_1^2x_2 - 2x_1x_2 - x_1 \\ & x_1^2 + x_1x_2 \le 3, \\ & x_1^2 - x_2^2 = 1, \\ & x_1 + x_2 \le 1, \\ & 0 \le x_1 \le 1, \\ & 1 \le x_2 \le 2. \end{aligned}$$

The lower bound factors are

$$x_1 \ge 0, \quad x_2 - 1 \ge 0,$$

while the upper bound factors are

$$1 - x_1 \ge 0, \quad 2 - x_2 \ge 0.$$

Then, we have, e.g.,

$$\begin{aligned} J_\ell = \{1,1\},\ J_u = \{2\} &\Rightarrow x_1^2(2 - x_2) \ge 0 \\ &\Rightarrow 2x_{11} - x_{112} \ge 0, \\ J_\ell = \{1\},\ J_u = \{1,2\} &\Rightarrow x_1(1 - x_1)(2 - x_2) \ge 0 \\ &\Rightarrow 2x_1 - 2x_{11} - x_{12} + x_{112} \ge 0, \\ J_\ell = \{1,1,2\},\ J_u = \emptyset &\Rightarrow x_1^2(x_2 - 1) \ge 0 \\ &\Rightarrow -x_{11} + x_{112} \ge 0. \quad \blacksquare \end{aligned}$$

We note that in the above example we just considered RLT constraints derived from (4.40) with $|J_\ell \cup J_u| = 3$, where 3 is the maximum degree of the polynomials involved in the definition of the problem. In general, if we denote by δ the maximum degree of the polynomials in (4.38), and we impose $|J_\ell \cup J_u| = \delta$, the overall number of distinct RLT constraints derived from inequalities (4.40) is

$$\binom{2n + \delta - 1}{\delta}$$

(20 in the above example). The linearization phase requires the addition of

$$\binom{n + \delta}{\delta} - n - 1$$

variables (7 in the above example). There is a theoretical reason behind the choice of ignoring the cases where $|J_\ell \cup J_u| < \delta$. Indeed, in (Sherali & Tuncbilek, 1992) it is proven that all linear constraints obtained by linearizing (4.40) when $|J_\ell \cup J_u| < \delta$ are implied by the linear constraints obtained by linearizing (4.40) when $|J_\ell \cup J_u| = \delta$. Instead, one could employ constraints (4.40) with $|J_\ell \cup J_u| > \delta$, but this should be done with some care in order to keep the size of the relaxation manageable. In order to impose a limit on the number of RLT constraints generated, in what follows we will never consider RLT constraints with terms of degree higher than δ, although their inclusion is possible and might, in fact, improve the relaxation.

Once RLT constraints have been generated, they give rise to an RLT relaxation of the original problem, whose solution is denoted here with

$$\bar{x}_{i_1 \dots i_k}, \quad k = 1, \dots, \delta, \quad i_1, \dots, i_k \in \{1, \dots, n\}.$$

The following observation can be proven (in fact, not only for relaxations based on bound factors but for any RLT relaxation).

Observation 4.1. *If*

$$\forall\, k = 1, \dots, \delta, \quad \forall\, i_1, \dots, i_k \in \{1, \dots, n\} \; : \; \bar{x}_{i_1 \dots i_k} = \bar{x}_{i_1} \cdots \bar{x}_{i_k},$$

then $(\bar{x}_1 \dots \bar{x}_n)$ *is an optimal solution of the original problem (4.38).*

4.3.2 Other RLT constraints based on constraints of the original problem

Bound factor based RLT constraints are not the only possible ones. Below we give some further examples of RLT constraints based on original constraints of (4.38), while we refer, e.g., to (Sherali, 2002) for more examples. If linear constraints

$$\mathbf{w}_h \mathbf{x} \geq v_h, \quad h \in I_{lin},$$

are present in the problem description, then we could employ the *linear constraint factors*

$$\prod_{h \in I'} (\mathbf{w}_h \mathbf{x} - v_h) \geq 0,$$

where $I' \subseteq I_{lin}$, possibly with the repetition of some indices. Also in this case $|I'| = \delta$ is imposed. The linearization is performed as for the RLT constraints derived from (4.40).

Example 4.31. For the previous example, we have that

$$(1 - x_1 - x_2)^3 \geq 0,$$

which is linearized into

$$1 + 3x_{11} + 3x_{22} + 6x_{12} - 3x_1 - 3x_2 - x_{111} - x_{222} - 3x_{122} - 3x_{112} \geq 0. \quad \blacksquare$$

Further RLT constraints can be generated by mixing bound and linear constraint factors, i.e.,

$$\prod_{h \in I''} (\mathbf{w}_h \mathbf{x} - v_h) \prod_{i \in J'_\ell} (x_i - \ell_i) \prod_{i \in J'_u} (u_i - x_i) \geq 0.$$

Example 4.32. We have the following constraint for our example:

$$(1 - x_1 - x_2) x_1 (2 - x_2) \geq 0,$$

which is linearized into

$$2x_1 - 2x_{11} - 3x_{12} + x_{122} + x_{112} \geq 0. \quad \blacksquare$$

Also *polynomial constraint factors* can be employed to generate new RLT constraints. Equality constraints can be multiplied by original variables in such a way that the resulting polynomial has degree not larger than δ, while inequality constraints can be multiplied by bound factors (or linear constraint factors, or mixed bound and linear constraint factors) in such a way that the resulting polynomial has degree δ.

Example 4.33. In our example, by multiplying the equality constraint $x_1^2 - x_2^2 - 1 = 0$ by, e.g., x_1, and linearizing the result, we end up with the constraint

$$x_{111} - x_{122} - x_1 = 0.$$

Multiplying the inequality constraint $-x_1^2 - x_1x_2 + 3 \geq 0$ by the upper bound factor $2 - x_2$ leads, after linearization, to the constraint

$$6 - 3x_2 - 2x_{11} - 2x_{12} + x_{112} + x_{122} \geq 0. \quad \blacksquare$$

4.3.3 RLT constraints based on the solution of a relaxation

All the previously discussed RLT constraints can be a priori generated, i.e., generated before any RLT relaxation has been solved. A different approach is that of adding RLT constraints based on the solution of an RLT relaxation, following the typical scheme of cutting algorithms. In such an approach a whole class of RLT constraints is available but initially not employed for the definition of the RLT relaxation. Once the relaxation has been solved, some separation procedure identifies one or more constraints within the class which are violated by the solution of the relaxation, or establishes that none of them is violated. In the former case, the identified constraints are added to the relaxation.

RLT constraints based on a semidefinite condition

An infinite class of RLT constraints has been proposed in (Sherali & Fraticelli, 2002) for quadratic programs. Such a class establishes a connection between RLT constraints and semidefinite programming. If we consider

$$\mathbf{X} = (x_{ij})_{i,j=1,\dots,n},$$

i.e., $\mathbf{X}$ is the $n \times n$ matrix whose entries are the variables x_{ij} representing second-order terms, then

$$x_{ij} = x_i x_j, \ \ i,j = 1,\dots,n \ \Rightarrow \ \mathbf{X} = \mathbf{x}\mathbf{x}^T,$$

i.e., $\mathbf{X}$ is a rank-1 positive semidefinite matrix. The equality $\mathbf{X} = \mathbf{x}\mathbf{x}^T$ can be relaxed into

$$\mathbf{X} - \mathbf{x}\mathbf{x}^T \in \mathcal{P}_n,$$

or, equivalently (in view of Observation A.1),

$$\begin{pmatrix} \mathbf{X} & \mathbf{x} \\ \mathbf{x}^T & 1 \end{pmatrix} \in \mathcal{P}_{n+1}. \tag{4.41}$$

The latter is equivalent to

$$\begin{pmatrix} \boldsymbol{\alpha} \\ \alpha_{n+1} \end{pmatrix}^T \begin{pmatrix} \mathbf{X} & \mathbf{x} \\ \mathbf{x}^T & 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \alpha_{n+1} \end{pmatrix} \geq 0 \quad \forall\, (\boldsymbol{\alpha}\ \alpha_{n+1}) \in \mathbb{R}^{n+1}. \tag{4.42}$$

It turns out that the left-hand side of these inequalities is the linearization of the constraints

$$(\boldsymbol{\alpha}^T \mathbf{x} + \alpha_{n+1})^2 \geq 0.$$

Therefore, (4.42) gives rise to an infinite class of RLT constraints. Now, let us assume that after solving an RLT relaxation, where the variable matrix $\mathbf{X}$ appears but constraint (4.41) is not imposed, we end up with a solution $(\bar{\mathbf{X}}, \bar{\mathbf{x}})$. If

$$\begin{pmatrix} \bar{\mathbf{X}} & \bar{\mathbf{x}} \\ \bar{\mathbf{x}}^T & 1 \end{pmatrix} \notin \mathcal{P}_{n+1},$$

then we can identify $(\bar{\boldsymbol{\alpha}}\ \bar{\alpha}_{n+1}) \in \mathbb{R}^{n+1}$ such that

$$\begin{pmatrix} \bar{\boldsymbol{\alpha}} \\ \bar{\alpha}_{n+1} \end{pmatrix}^T \begin{pmatrix} \bar{\mathbf{X}} & \bar{\mathbf{x}} \\ \bar{\mathbf{x}}^T & 1 \end{pmatrix} \begin{pmatrix} \bar{\boldsymbol{\alpha}} \\ \bar{\alpha}_{n+1} \end{pmatrix} < 0,$$

i.e., we identify an RLT constraint, namely the linearization of

$$(\bar{\boldsymbol{\alpha}}^T \mathbf{x} + \bar{\alpha}_{n+1})^2 \geq 0,$$

which is violated by the solution of the relaxation.

A class of RLT constraints based on paraboloids

Other classes of RLT constraints for quadratic programs, from which we can identify cuts violated by the solution of the current RLT relaxation, have been proposed in (Audet et al., 2000). Here we just present one of such classes, while we refer the reader to (Audet et al., 2000) for other classes. Let $\alpha_i \in [\ell_i, u_i]$, $\alpha_j \in [\ell_j, u_j]$, and $\gamma \in \mathbb{R}$. Then, the following inequality holds:

$$[(\alpha_i - x_i) + \gamma(\alpha_j - x_j)]^2 \geq 0.$$

Its linearization,

$$\gamma^2(x_{jj} - 2\alpha_j x_j + \alpha_j^2) + (x_{ii} - 2\alpha_i x_i + \alpha_i^2) + 2\gamma(x_{ij} - \alpha_i x_j - \alpha_j x_i + \alpha_i \alpha_j) \geq 0, \tag{4.43}$$

is an RLT constraint. We can also rewrite (4.43) as

$$x_{ii} + \gamma^2 x_{jj} + 2\gamma x_{ij} \geq 2(\alpha_i + \gamma\alpha_j)(x_i + \gamma x_j) - (\alpha_i + \gamma\alpha_j)^2, \tag{4.44}$$

where the left-hand side is the linearization of the paraboloid $(x_i + \gamma x_j)^2$, while the right-hand side is the tangent plane to the same paraboloid at point $(\alpha_i\ \alpha_j)$. Now, let $(\bar{\mathbf{X}}, \bar{\mathbf{x}})$ be the solution of an RLT relaxation. Let

$$\begin{aligned}\tau_i &= \bar{x}_{ii} - 2\alpha_i\bar{x}_i + \alpha_i^2,\\ \tau_j &= \bar{x}_{jj} - 2\alpha_j\bar{x}_j + \alpha_j^2,\\ \pi_{ij} &= \bar{x}_{ij} - \alpha_i\bar{x}_j - \alpha_j\bar{x}_i + \alpha_i\alpha_j.\end{aligned}$$

Let us impose the following conditions:

$$0 \le \ell_i \le \bar{x}_i < \sqrt{\bar{x}_{ii}} \le u_i, \quad 0 \le \ell_j \le \bar{x}_j < \sqrt{\bar{x}_{jj}} \le u_j, \quad \tau_i\tau_j < \pi_{ij}^2. \tag{4.45}$$

Then, the following proposition allows us to identify an RLT constraint of type (4.43) most violated by the solution of the current relaxation for fixed α_i, α_j values.

Proposition 4.34. *If* (4.45) *is true, then for any $\alpha_i \in (\bar{x}_i, \sqrt{\bar{x}_{ii}})$ and $\alpha_j \in (\bar{x}_j, \sqrt{\bar{x}_{jj}})$, the value γ for which the left-hand side of* (4.43) *is most negative at the solution of the current relaxation is*

$$\hat{\gamma} = -\frac{\pi_{ij}}{\tau_j}.$$

Proof. At the solution of the current relaxation the left-hand side of (4.43) is equal to

$$\tau_j\gamma^2 + 2\pi_{ij}\gamma + \tau_i. \tag{4.46}$$

Since, in view of (4.45),

$$\tau_j > \bar{x}_j^2 - 2\alpha_j\bar{x}_j + \alpha_j^2 = (\bar{x}_j - \alpha_j)^2 \ge 0,$$

(4.46) is a convex function with respect to γ, and attains its minimum value at

$$\hat{\gamma} = -\frac{\pi_{ij}}{\tau_j}.$$

We have that

$$\tau_j\hat{\gamma}^2 + 2\pi_{ij}\hat{\gamma} + \tau_i = \tau_i - \frac{\pi_{ij}^2}{\tau_j},$$

which, in view of (4.45), is a negative value, i.e., the solution of the current relaxation $(\bar{\mathbf{X}}, \bar{\mathbf{x}})$ violates the inequality. □

In order to choose $(\alpha_i\ \alpha_j)$, Audet et al. suggest minimizing the maximum distance between the paraboloid $(x_i + \gamma x_j)^2$ and its tangent plane at point $(\alpha_i\ \alpha_j)$ (whose formula is given by the right-hand side of (4.44)) over the box

$$\bar{B} = [\bar{x}_i, \sqrt{\bar{x}_{ii}}] \times [\bar{x}_j, \sqrt{\bar{x}_{jj}}].$$

Then, if we set

$$\eta = \alpha_i + \gamma\alpha_j, \tag{4.47}$$

we need to solve

$$\min_{(\alpha_i\ \alpha_j)}\ \max_{(x_i\ x_j)\in\bar{B}} d_{\alpha_i,\alpha_j}(x_i,x_j) = (x_i+\gamma x_j)^2 - 2\eta(x_i+\gamma x_j)+\eta^2.$$

Since the function d_{α_i,α_j} is convex with respect to x_i,x_j for fixed α_i,α_j, its maximum is attained at a vertex of the box $\bar{B}$. If we restrict the attention to the two vertices $(\bar{x}_i\ \sqrt{\bar{x}_{jj}})$ and $(\sqrt{\bar{x}_{ii}}\ \bar{x}_j)$ of $\bar{B}$, the following proposition identifies $(\alpha_i\ \alpha_j)$ values for which the maximum of d_{α_i,α_j} at these two vertices is minimized. In a completely analogous way it can be proven that the same values also minimize the maximum of d_{α_i,α_j} at the other two vertices of $\bar{B}$, i.e., $(\bar{x}_i\ \bar{x}_j)$ and $(\sqrt{\bar{x}_{ii}}\ \sqrt{\bar{x}_{jj}})$, so that these values minimize the maximum of d_{α_i,α_j} over the whole box $\bar{B}$.

Proposition 4.35. *We have that*

$$\min_{\alpha_i,\alpha_j}\max\{d_{\alpha_i,\alpha_j}(\bar{x}_i,\sqrt{\bar{x}_{jj}}),d_{\alpha_i,\alpha_j}(\sqrt{\bar{x}_{ii}},\bar{x}_j)\}$$

is attained for

$$\alpha_i = \frac{\bar{x}_i+\sqrt{\bar{x}_{ii}}}{2},\quad \alpha_j = \frac{\bar{x}_j+\sqrt{\bar{x}_{jj}}}{2}.$$

Proof. Recalling the definition of d_{α_i,α_j}, we need to minimize

$$\max\{(\bar{x}_i+\gamma\sqrt{\bar{x}_{jj}})^2-2\eta(\bar{x}_i+\gamma\sqrt{\bar{x}_{jj}})+\eta^2,(\sqrt{\bar{x}_{ii}}+\gamma\bar{x}_j)^2-2\eta(\sqrt{\bar{x}_{ii}}+\gamma\bar{x}_j)+\eta^2\}.$$

The minimum is attained when the two quantities are equal, and this is true if and only if

$$\eta = \frac{\bar{x}_i+\sqrt{\bar{x}_{ii}}}{2}+\frac{\gamma(\bar{x}_j+\sqrt{\bar{x}_{jj}})}{2}.$$

Recalling the definition (4.47), we have that this is true for

$$\alpha_i = \frac{\bar{x}_i+\sqrt{\bar{x}_{ii}}}{2},\quad \alpha_j = \frac{\bar{x}_j+\sqrt{\bar{x}_{jj}}}{2},$$

as we wanted to prove. □

4.3.4 Further remarks on RLT

We conclude this section about RLT with some more remarks.

- Besides linear RLT constraints, some convex constraints can also be added to an RLT relaxation. For instance, in (Sherali & Tuncbilek, 1995), for each variable x_i and positive integers p,q such that p is prime and $pq \le \delta$ (where we recall that δ denotes the highest degree of the polynomials in (4.38)), the following convex constraints can be added:

$$(x_{\underbrace{i\ldots i}_{q\text{ times}}})^p \le x_{\underbrace{i\ldots i}_{pq\text{ times}}}.$$

So, for instance, if $q = 2$, $p = 3$, we have

$$(x_{11})^3 \leq x_{111111}.$$

Note that in cases where p is not prime we could as well generate a convex constraint, but this would be already implied by those obtained with prime numbers p.

- The number of RLT constraints could be quite high. A *constraint filtering scheme* is described in (Sherali, 2002) in order to remove a subset of RLT constraints, which, presumably, are not active at an optimal solution of the current RLT relaxation.

- Binary problems can be viewed as polynomial ones with the standard substitution of the binary constraints $x_i \in \{0,1\}$ with the quadratic equality constraints $x_i(1-x_i) = 0$ for each $i = 1,\ldots,n$. Through bound factors of degree d, i.e.,

$$\prod_{j\in J_1} x_j \prod_{j\in J_2} (1-x_j),$$

with $J_1 \cap J_2 = \emptyset$, $|J_1 \cup J_2| = d$ (repetitions in $J_1 \cup J_2$ are not allowed since $x_j = x_j^2$ for any $j \in \{1,\ldots,n\}$), a hierarchy of relaxations can be defined. As the degree d increases, the projection of the feasible region of the relaxations over the space of the original variables gets closer and closer to the convex hull of the feasible region of the binary problem, until for $d = n$ it becomes equal to the convex hull itself (we refer to (Sherali & Adams, 1990, 1994) for the details).

- An interesting comparison between RLT and semidefinite relaxations for problems with quadratic objective and constraints has been done in (Anstreicher, 2009). In particular, the effect of adding RLT constraints based on bound factors is compared with that of adding the semidefinite condition (4.41). Restricting to the bilinear term x_1x_2 over the unit box $[0,1]\times[0,1]$, for $0 \leq x_1 \leq x_2 \leq 1/2$, the bound factor based RLT constraints are

$$\begin{array}{ccccc} 0 & \leq & x_{11} & \leq & x_1, \\ 0 & \leq & x_{22} & \leq & x_2, \\ 0 & \leq & x_{12} & \leq & x_1, \end{array} \tag{4.48}$$

while the semidefinite condition is

$$\begin{pmatrix} x_{11} & x_{12} & x_1 \\ x_{12} & x_{22} & x_2 \\ x_1 & x_2 & 1 \end{pmatrix} \in \mathcal{P}_3,$$

which can also be rewritten as

$$\begin{array}{ccl} x_{11} & \geq & x_1^2, \\ x_{22} & \geq & x_2^2, \\ x_{12} & \leq & x_1x_2 + \sqrt{(x_{11}-x_1^2)(x_{22}-x_2^2)}, \\ x_{12} & \geq & x_1x_2 - \sqrt{(x_{11}-x_1^2)(x_{22}-x_2^2)}. \end{array} \tag{4.49}$$

Next, Anstreicher computes for fixed values x_1, x_2 the volume of the set of feasible points $(x_{11}\ x_{22}\ x_{12})$ with respect to the RLT constraints (4.48) alone, which is equal to $x_1^2 x_2$, and the volume of the set of feasible points with respect both to (4.48) and the semidefinite constraints (4.49), whose rather long formula is

$$x_1^2 x_2(1-x_2) - \tfrac{1}{9}x_1^3(6x_2^2 - 6x_2 + 5) + \tfrac{1}{3}x_1^3[(1-x_2)^3 - x_2^3]\log\left(\tfrac{1-x_2}{x_2}\right)$$
$$-\tfrac{1}{3}x_1^3[(1-x_2)^3 + x_2^3]\log\left(\tfrac{1-x_1}{x_1}\right).$$

Then, the ratio of such volumes is compared for different values of x_1 and x_2, observing that the minimum of the ratio is $1/9$ attained for $x_1 = x_2 = 1/2$, while for $x_2 \to 0$ and $x_1/x_2 \to 0$, the ratio converges to 1. Anstreicher also computes the volume (equal to $1/60$) of the set of feasible points $(x_1\ x_2\ x_{11}\ x_{22}\ x_{12})$ with respect to the RLT constraints (4.48) and the volume (equal to $1/240$) of the set of feasible points with respect to both (4.48) and (4.49). As observed in (Anstreicher, 2009), there is no clear relation between the volume of a relaxation and the quality of the corresponding bound. Moreover, these results only apply to two variables of the problem. However, some computational experiments also show that bounds based both on the RLT and on the semidefinite constraints, though computationally expensive, are quite good with respect to those obtained with the RLT constraints alone and with the semidefinite constraints alone.

- In (Liberti, 2005) Liberti observes that equality constraints can be exploited to reduce the nonlinearities before proceeding to the RLT relaxation of a nonconvex problem. We illustrate this through an example. Consider the problem

$$\begin{aligned} \min\quad & -x_1^2 - x_2^2 \\ & x_1 + x_2 = 1, \\ & x_1 x_2 \geq \tfrac{1}{4}, \\ & 0 \leq x_1, x_2 \leq 1. \end{aligned}$$

If we immediately derive the RLT relaxation (based on bound factors) of this problem, we are led to the following:

$$\begin{aligned} \min\quad & -x_{11} - x_{22} \\ & x_1 + x_2 = 1, \\ & x_{12} \geq \tfrac{1}{4}, \\ & \max\{2x_1 - 1, 0\} \leq x_{11} \leq x_1, \\ & \max\{2x_2 - 1, 0\} \leq x_{22} \leq x_2, \\ & x_{12} \leq \min\{x_1, x_2\}, \\ & x_{12} \geq \max\{x_1 + x_2 - 1, 0\}, \\ & 0 \leq x_1, x_2 \leq 1, \end{aligned}$$

with a lower bound equal to -1 obtained by setting all the variables equal to $\frac{1}{2}$. Instead, if we first multiply the equality constraint $x_1+x_2=1$ by x_1 and x_2 we get to

$$\begin{aligned}\min\quad & -x_1^2-x_2^2\\ & x_1+x_2=1,\\ & x_1^2+x_1x_2=x_1,\\ & x_1x_2+x_2^2=x_2,\\ & x_1x_2\geq \tfrac{1}{4},\\ & 0\leq x_1,x_2\leq 1.\end{aligned}$$

The terms x_1^2 and x_2^2 can be eliminated, so that we are left with the single nonlinear term x_1x_2:

$$\begin{aligned}\min\quad & -1+2x_1x_2\\ & x_1+x_2=1,\\ & x_1x_2\geq \tfrac{1}{4},\\ & 0\leq x_1,x_2\leq 1,\end{aligned}$$

which leads to the following relaxation:

$$\begin{aligned}\min\quad & -1+2x_{12}\\ & x_1+x_2=1,\\ & x_{12}\geq \tfrac{1}{4},\\ & x_{12}\leq \min\{x_1,x_2\},\\ & x_{12}\geq \max\{x_1+x_2-1,0\},\\ & 0\leq x_1,x_2\leq 1,\end{aligned}$$

and to the improved bound $-\frac{1}{2}$.

4.4 Quadratic programming problems

GO problems with a quadratic objective function and linear constraints are called *quadratic programming* (QP) problems. More formally, given $\mathbf{Q}\in\mathbb{R}^{n\times n}$, $\mathbf{A}\in\mathbb{R}^{m\times n}$, $\mathbf{c}\in\mathbb{R}^n$, $\mathbf{b}\in\mathbb{R}^m$, a QP problem is

$$\begin{aligned}\min\quad & \tfrac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x}+\mathbf{c}^T\mathbf{x}\\ & \mathbf{A}\mathbf{x}\leq \mathbf{b},\\ & \mathbf{x}\geq \mathbf{0}.\end{aligned}\tag{4.50}$$

Without loss of generality, the matrix $\mathbf{Q}$ can be assumed to be symmetric. We can also assume that $\mathbf{Q}$ is *not* semidefinite positive; otherwise the problem is convex. Such problems are highly structured ones for which different relaxations are possible. In the following subsections we review some results about QPs. More precisely, in Section 4.4.1 we show that all QP problems (possibly with some binary variables) can be reformulated as problems over the cone of completely positive matrices, and we see how to exploit such reformulation to define polynomially solvable relaxations for QPs. In Section 4.4.2 we consider linear and semidefinite relaxations for QPs with box constraints and for general QPs, based on KKT conditions. In Section 4.4.3 we reformulate the problem QP with box constraints as a problem with a linear objective function and a convex feasible region which is usually unknown but can be outer approximated through the identification of valid inequalities. Finally, in Section 4.4.4 we consider problems where also the constraint functions are quadratic ones.

4.4.1 Reformulations as completely positive problems

Recent results have shown that many problems can be reformulated as linear problems over the cone of copositive matrices (see Definition A.21) or over its dual cone of completely positive matrices (see Definition A.22). (We refer the reader to (Bomze, 2011; Bomze, Schachinger, & Uchida, 2012) for a survey about copositive optimization.) Unfortunately, although both cones are convex ones, they are not tractable. We are not able to solve problems over these cones in polynomial time, e.g., by interior point methods, because while they allow self-concordant barrier functions, the computation of these functions can not be performed in polynomial time. Still, such reformulations appear to be of primary importance since they disclose a new point of view from which we can look at some problems and, consequently, a bunch of new ideas to tackle them.

To the authors' knowledge, the first problem for which a reformulation as a completely positive one has been given is a special case of the QP problem, namely the standard quadratic programming (StQP) problem. In (Bomze et al., 2000) it is observed that given a StQP problem

$$\begin{aligned} \min \quad & \mathbf{x}^T\mathbf{Q}\mathbf{x} \\ & \mathbf{e}^T\mathbf{x} = 1, \\ & \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

whose feasible region is the unit simplex, it is possible to reformulate it as a linear one over the n-dimensional cone of completely positive matrices. First, we notice that the StQP problem can be reformulated as

$$\begin{aligned} \min \quad & \mathbf{Q} \bullet \mathbf{x}\mathbf{x}^T \\ & \mathbf{e}^T\mathbf{x}\mathbf{x}^T\mathbf{e} = 1, \\ & \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

A relaxation of this problem can be attained by replacing the rank-1 n-dimensional completely positive matrix $\mathbf{x}\mathbf{x}^T$ with a general completely positive matrix $\mathbf{X}$:

$$\begin{aligned} \min \quad & \mathbf{Q} \bullet \mathbf{X} \\ & \mathbf{E} \bullet \mathbf{X} = 1, \\ & \mathbf{X} \in \mathcal{C}_n^*. \end{aligned} \tag{4.51}$$

In fact, in (Bomze et al., 2000) it is proven that the extreme points of the feasible region for this problem are the rank-1 matrices $\mathbf{x}\mathbf{x}^T$ with $\mathbf{x}$ belonging to the unit simplex. Therefore, as already remarked in Section 2.5.1, the optimal value of the relaxation is equal to that of the StQP problem. Further equivalence results have been later established for other quadratic problems such as a minimum cut graph tri-partitioning problem (Povh & Rendl, 2007) and the quadratic assignment problem (Povh & Rendl, 2009). Burer has finally shown that *all* QPs over linear constraints (possibly also with some binary variables) can be reformulated as linear ones over the cone of completely positive matrices (an extension of this result for the cases where an additional set constraint $\mathbf{x} \in K$ is present can be found in (Eichfelder & Povh, 2012)[2]). Following (Burer, 2009), we will prove this equivalence result. We consider problems with the form

$$\begin{aligned} \min \quad & \mathbf{x}^T\mathbf{Q}\mathbf{x} + 2\mathbf{c}^T\mathbf{x} \\ & \mathbf{a}_i^T\mathbf{x} = b_i, \qquad i = 1,\ldots,m, \\ & \mathbf{x} \geq \mathbf{0}, \\ & x_j \in \{0,1\}, \qquad j \in B \subseteq \{1,\ldots,n\}, \end{aligned} \tag{4.52}$$

which, with respect to problem (4.50), may also include some binary variables. We denote by $\mathcal{F}$ the feasible set of (4.52) and make the following assumptions.

Assumption 4.1. $\mathcal{F} \neq \emptyset$.

Assumption 4.2. *Let*

$$L = \{\mathbf{x} \in \mathbb{R}_+^n \ : \ \mathbf{a}_i^T\mathbf{x} = b_i, \ i = 1,\ldots,m\};$$

then

$$\mathbf{x} \in L \quad \Rightarrow \quad 0 \leq x_j \leq 1 \ \forall\, j \in B.$$

Note that Assumption 4.2 is easily attained by adding the equality constraints $x_j + s_j = 1$ for all $j \in B$, where each $s_j \geq 0$ is a slack variable. In view of Assumption 4.2, we also have that

$$d_j = 0 \ \forall\, j \in B, \ \mathbf{d} \in 0^+(L), \tag{4.53}$$

where

$$0^+(L) = \{\mathbf{d} \geq \mathbf{0} \ : \ \mathbf{a}_i^T\mathbf{d} = 0, \ i = 1,\ldots,m\}$$

is the recession cone of L (see Definition A.13). Problem (4.52) can be rewritten as

$$\begin{aligned} \min \quad & \mathbf{Q} \bullet \mathbf{x}\mathbf{x}^T + 2\mathbf{c}^T\mathbf{x} \\ & \mathbf{a}_i^T\mathbf{x} = b_i, \qquad i = 1,\ldots,m, \\ & \mathbf{a}_i^T\mathbf{x}\mathbf{x}^T\mathbf{a}_i = b_i^2, \qquad i = 1,\ldots,m, \\ & \mathbf{x} \geq \mathbf{0}, \\ & x_j = x_j^2, \qquad j \in B \subseteq \{1,\ldots,n\}. \end{aligned} \tag{4.54}$$

[2] A mistake in this paper has been corrected in (Dickinson, Eichfelder, & Povh, 2012).

By substituting, as before, $\mathbf{x}\mathbf{x}^T$ with $\mathbf{X}$ (i.e., relaxing the rank-1 constraint), we end up with the following relaxation of problem (4.52):

$$\begin{array}{lll} \min & \mathbf{Q} \bullet \mathbf{X} + 2\mathbf{c}^T\mathbf{x} & \\ & \mathbf{a}_i^T\mathbf{x} = b_i, & i = 1,\dots,m, \\ & \mathbf{a}_i^T\mathbf{X}\mathbf{a}_i = b_i^2, & i = 1,\dots,m, \\ & x_j = X_{jj}, & j \in B \subseteq \{1,\dots,n\}, \\ & \begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \mathbf{X} \end{pmatrix} \in \mathcal{C}_{n+1}^*. & \end{array} \tag{4.55}$$

The following theorem states that, in fact, problems (4.52) and (4.55) are equivalent.

Theorem 4.36. *The optimal values of problems* (4.52) *and* (4.55) *are equal. Moreover, if* $(\mathbf{x}^*\ \mathbf{X}^*)$ *is an optimal solution for* (4.55)*, then* $\mathbf{x}^*$ *is in the convex hull of the optimal solutions for* (4.52).

The key result to prove this theorem is the following proposition. Let $\mathcal{G}$ be the feasible region of (4.55) and let us define the following sets of matrices:

$$\begin{array}{rcl} \mathcal{F}^+ & = & \left\{ \begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \mathbf{x}\mathbf{x}^T \end{pmatrix} : \mathbf{x} \in \mathcal{F} \right\}, \\ \mathcal{G}^+ & = & \left\{ \begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \mathbf{X} \end{pmatrix} : (\mathbf{x}\,\mathbf{X}) \in \mathcal{G} \right\}, \\ L_\infty^+ & = & \left\{ \begin{pmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & \mathbf{d}\mathbf{d}^T \end{pmatrix} : \mathbf{d} \in 0^+(L) \right\}. \end{array}$$

Then, Proposition 4.37 follows.

Proposition 4.37.

$$\mathcal{G}^+ = \mathcal{F}^+ + L_\infty^+.$$

Proof. Obviously, $\mathcal{F}^+ \subseteq \mathcal{G}^+$. Now, let us consider the recession cone of $\mathcal{G}^+$

$$\left\{ \begin{pmatrix} 0 & \mathbf{d}^T \\ \mathbf{d} & \mathbf{D} \end{pmatrix} \in \mathcal{C}_{n+1}^* : \begin{array}{ll} \mathbf{a}_i^T\mathbf{d} = 0 & i = 1,\dots,m \\ \mathbf{a}_i^T\mathbf{D}\mathbf{a}_i = 0 & i = 1,\dots,m \\ d_j = D_{jj} & j \in B \end{array} \right\}.$$

We notice that

$$\begin{pmatrix} 0 & \mathbf{d}^T \\ \mathbf{d} & \mathbf{D} \end{pmatrix} \in \mathcal{C}_{n+1}^* \Rightarrow \begin{array}{c} \begin{pmatrix} 0 & \mathbf{d}^T \\ \mathbf{d} & \mathbf{D} \end{pmatrix} = \sum_{k=1}^K \begin{pmatrix} \eta_k^2 & \eta_k\mathbf{z}_k^T \\ \eta_k\mathbf{z}_k & \mathbf{z}_k\mathbf{z}_k^T \end{pmatrix}, \\ \eta_k \geq 0, \mathbf{z}_k \in \mathbb{R}_+^n,\ \forall\, k, \end{array}$$

so that $\eta_k = 0$ for all k. Therefore, the recession cone of $\mathcal{G}^+$ is also equal to

$$\left\{ \begin{pmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & \mathbf{D} \end{pmatrix} \in \mathcal{C}_{n+1}^* : \begin{array}{ll} \mathbf{a}_i^T\mathbf{D}\mathbf{a}_i = 0 & i = 1,\dots,m \\ 0 = D_{jj} & j \in B \end{array} \right\}.$$

In view of (4.53), L_∞^+ is a subset of the recession cone for $\mathcal{G}^+$, which implies

$$\mathcal{F}^+ + L_\infty^+ \subseteq \mathcal{G}^+. \quad \square$$

To prove the converse, we first need the result of the following lemma.

Lemma 4.38. *Let* $(\mathbf{x}\ \mathbf{X}) \in \mathcal{G}$ *and let*

$$\sum_{k=1}^{K} \begin{pmatrix} \eta_k^2 & \eta_k \mathbf{z}_k^T \\ \eta_k \mathbf{z}_k & \mathbf{z}_k \mathbf{z}_k^T \end{pmatrix}, \tag{4.56}$$

with $\eta_k \geq 0, \mathbf{z}_k \in \mathbb{R}_+^n$*, for all k, be the completely positive representation of the corresponding matrix*

$$\begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \mathbf{X} \end{pmatrix} \in \mathcal{G}^+.$$

Let

$$\begin{aligned} K_+ &= \{k \ : \ \eta_k > 0\}, \\ K_0 &= \{k \ : \ \eta_k = 0\}. \end{aligned}$$

Then,

$$\begin{aligned} k \in K_+ &\Rightarrow \mathbf{z}_k/\eta_k \in L, \\ k \in K_0 &\Rightarrow \mathbf{z}_k \in 0^+(L). \end{aligned}$$

Proof. We must have that

$$\sum_{k=1}^{K} \eta_k^2 = 1. \tag{4.57}$$

Moreover,

$$\begin{aligned} \mathbf{a}_i^T \mathbf{x} = b_i &\Rightarrow \textstyle\sum_{k=1}^K \eta_k (\mathbf{a}_i^T \mathbf{z}_k) = b_i, \\ \mathbf{a}_i^T \mathbf{X} \mathbf{a}_i = b_i^2 &\Rightarrow \textstyle\sum_{k=1}^K (\mathbf{a}_i^T \mathbf{z}_k)^2 = b_i^2, \end{aligned} \tag{4.58}$$

so that

$$\left[\sum_{k=1}^{K} \eta_k (\mathbf{a}_i^T \mathbf{z}_k)\right]^2 = \sum_{k=1}^{K} (\mathbf{a}_i^T \mathbf{z}_k)^2.$$

Since we are in a case where Cauchy–Schwarz inequality is, in fact, an equality, then there must exist δ_i such that

$$\delta_i \eta_k = \mathbf{a}_i^T \mathbf{z}_k, \quad \forall\, k,\ i = 1, \ldots, m. \tag{4.59}$$

Then,

$$k \in K_0 \Rightarrow \eta_k = 0 \Rightarrow \mathbf{a}_i^T \mathbf{z}_k = 0 \Rightarrow \mathbf{z}_k \in 0^+(L).$$

Moreover, in view of (4.57)–(4.59),

$$k \in K_+ \;\Rightarrow\; \mathbf{a}_i^T(\mathbf{z}_k/\eta_k) = \delta_i = \sum_{k=1}^{K} \eta_k(\delta_i \eta_k) = \sum_{k=1}^{K} \eta_k(\mathbf{a}_i^T \mathbf{z}_k) = b_i. \quad \square$$

Then, setting $\mathbf{v}_k = \mathbf{z}_k/\eta_k$ and $\lambda_k = \eta_k^2 > 0$ for each $k \in K^+$, we are able to rewrite (4.56) as

$$\sum_{k \in K_+} \lambda_k \begin{pmatrix} 1 & \mathbf{v}_k^T \\ \mathbf{v}_k & \mathbf{v}_k \mathbf{v}_k^T \end{pmatrix} + \sum_{k \in K_0} \begin{pmatrix} 0 & \mathbf{0} \\ \mathbf{0} & \mathbf{z}_k \mathbf{z}_k^T \end{pmatrix}. \tag{4.60}$$

Noting that $\sum_{k \in K_+} \lambda_k = 1$, in order to prove that $\mathcal{G}^+ \subseteq \mathcal{F}^+ + L_\infty^+$, we only need to show that $v_j^k \in \{0,1\}$ for all k and all $j \in B$.

For any $j \in B$ we have, in view of Assumption 4.2 and (4.53), that $v_j^k \in [0,1]$ for all $k \in K_+$, and $z_j^k = 0$ for all $k \in K_0$. Since

$$\begin{aligned} x_j &= \textstyle\sum_{k \in K_+} \lambda_k v_j^k, \\ X_{jj} &= \textstyle\sum_{k \in K_+} \lambda_k (v_j^k)^2 + \sum_{k \in K_0} (z_j^k)^2, \end{aligned}$$

and since the equality constraint $x_j = X_{jj}$ must be satisfied for all $j \in B$, we must have that

$$\sum_{k \in K_+} \lambda_k v_j^k (1 - v_j^k) = 0.$$

Then,

$$\lambda_k > 0,\; v_j^k(1 - v_j^k) \geq 0 \;\; \forall\, k \in K_+ \quad \Rightarrow \quad v_j^k \in \{0,1\}. \quad \square$$

Now we are ready for the proof of Theorem 4.36.

Proof. Problem (4.52) (once rewritten as (4.54) and once $\mathbf{X} = \mathbf{x}\mathbf{x}^T$ has been imposed) and problem (4.55) have the common objective function

$$\begin{pmatrix} 0 & \mathbf{c}^T \\ \mathbf{c} & \mathbf{Q} \end{pmatrix} \bullet \begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \mathbf{X} \end{pmatrix},$$

while the feasible regions are $\mathcal{F}^+$ and $\mathcal{G}^+$ for (4.52) and (4.55), respectively. We also denote by opt_1, opt_2 the optimal values for problems (4.52) and (4.55), respectively. Since $\mathcal{F}^+ \subseteq \mathcal{G}^+$, we must have $opt_1 \geq opt_2$. Therefore, if $opt_1 = -\infty$, equality with opt_2 is obviously true. Therefore, let us assume that $opt_1 > -\infty$. In order to prove that $opt_1 = opt_2$ is still true, we first remark that $opt_1 > -\infty$ implies $\mathbf{d}^T\mathbf{Q}\mathbf{d} \geq 0$ for all $\mathbf{d} \in 0^+(L)$. Then

$$\begin{pmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & \mathbf{d}\mathbf{d}^T \end{pmatrix} \in L_\infty^+ \quad \Rightarrow \quad \begin{pmatrix} 0 & \mathbf{c}^T \\ \mathbf{c} & \mathbf{Q} \end{pmatrix} \bullet \begin{pmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & \mathbf{d}\mathbf{d}^T \end{pmatrix} = \mathbf{d}^T\mathbf{Q}\mathbf{d} \geq 0.$$

Thus, the equality $opt_1 = opt_2$ follows from Proposition 4.37.

Now, let us consider the representation (4.60) for the optimal solution $(\mathbf{x}^* \; \mathbf{X}^*)$ of problem (4.55). Then,

$$\mathbf{x}^* = \sum_{k \in K_+} \lambda_k \mathbf{v}_k, \quad \sum_{k \in K_+} \lambda_k = 1, \quad \lambda_k \geq 0 \; \forall \, k \in K_+.$$

Therefore, in order to prove that $\mathbf{x}^*$ belongs to the convex hull of the optimal solutions for problem (4.52), it is enough to observe that

$$opt_1 = opt_2 \quad \Rightarrow \quad \mathbf{v}_k \text{ optimal for (4.52)} \quad \forall \, k \in K_+$$

(note that we must have $\mathbf{z}_k^T \mathbf{Q} \mathbf{z}_k = 0$ for all $k \in K_0$). □

A limit of the completely positive representation (4.55) is that its feasible region has no interior, while existence of a nonempty interior is usually important both from the theoretical and a practical point of view. For instance, if the problem has an interior point (Slater's condition) and is bounded, then strong duality holds, i.e., for problems over the cone of completely positive matrices, these have the same optimal value of the corresponding dual problems over the dual cone of copositive matrices. To see that the feasible region of (4.55) has an empty interior, let us consider $(\mathbf{x}\,\mathbf{X}) \in \mathcal{G}$ and verify that

$$\begin{pmatrix} b_i \\ -\mathbf{a}_i^T \end{pmatrix}^T \begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \mathbf{X} \end{pmatrix} \begin{pmatrix} b_i \\ -\mathbf{a}_i^T \end{pmatrix} = 0,$$

so that

$$\begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \mathbf{X} \end{pmatrix} \notin int(\mathcal{P}_{n+1}) \supseteq int(\mathcal{C}_{n+1}^*),$$

where $int(\mathcal{P}_{n+1})$ is the interior of the cone of the semidefinite matrices (the latter inclusion follows from $\mathcal{P}_{n+1} \supseteq \mathcal{C}_{n+1}^*$).

Again following (Burer, 2009), we will now show that under a mild assumption, problem (4.52) can also be reformulated as a completely positive problem over $\mathcal{C}_n^*$ instead of $\mathcal{C}_{n+1}^*$ (by eliminating the vector of variables $\mathbf{x}$). The feasible region of such a problem may have a nonempty interior. The mild assumption is the following:

$$\exists \, \mathbf{y} \in \mathbb{R}^m \; : \; \sum_{i=1}^{m} y_i \mathbf{a}_i = \boldsymbol{\alpha} \geq \mathbf{0}, \quad \sum_{i=1}^{m} y_i b_i = 1. \tag{4.61}$$

As for Assumption 4.2, this can be easily attained by taking a binary variable x_j and adding the equality constraint $x_j + s_j = 1$, where $s_j \geq 0$ is a slack variable. This way we may set the variable y_i related to this constraint equal to 1, while all other y variables are set to 0 in order to satisfy (4.61).

If (4.61) is true, we may add the redundant constraint $\boldsymbol{\alpha}^T \mathbf{x} = 1$ to problem (4.52). The corresponding completely positive representation (equivalent to (4.55)) then becomes

$$\begin{aligned} \min \quad & \mathbf{Q} \bullet \mathbf{X} + 2\mathbf{c}^T \mathbf{x} \\ & (\mathbf{x}\,\mathbf{X}) \in \mathcal{G}, \\ & \boldsymbol{\alpha}^T \mathbf{x} = 1, \\ & \boldsymbol{\alpha}^T \mathbf{X} \boldsymbol{\alpha} = 1. \end{aligned} \tag{4.62}$$

Next we observe that the constraint $\mathbf{X}\boldsymbol{\alpha} = \mathbf{x}$ is redundant for such a problem. Indeed, recalling (4.56), let us compute

$$\begin{aligned}\begin{pmatrix} 1-\boldsymbol{\alpha}^T\mathbf{x} \\ \mathbf{x}-\mathbf{X}\boldsymbol{\alpha} \end{pmatrix} &= \begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \mathbf{X} \end{pmatrix}\begin{pmatrix} 1 \\ -\boldsymbol{\alpha} \end{pmatrix} \\ &= \textstyle\sum_{k=1}^K \begin{pmatrix} \eta_k^2 & \eta_k\mathbf{z}_k^T \\ \eta_k\mathbf{z}_k & \mathbf{z}_k\mathbf{z}_k^T \end{pmatrix}\begin{pmatrix} 1 \\ -\boldsymbol{\alpha} \end{pmatrix} \\ &= \textstyle\sum_{k=1}^K \begin{pmatrix} \eta_k^2 - \eta_k(\mathbf{z}_k^T\boldsymbol{\alpha}) \\ \eta_k\mathbf{z}_k - \mathbf{z}_k(\mathbf{z}_k^T\boldsymbol{\alpha}) \end{pmatrix}.\end{aligned}$$

In view of Lemma 4.38, we have that

$$k \in K_+ \ \Rightarrow\ \boldsymbol{\alpha}^T(\mathbf{z}_k/\eta_k) = 1, \quad k \in K_0 \ \Rightarrow\ \boldsymbol{\alpha}^T\mathbf{z}_k = 0,$$

so that

$$\begin{pmatrix} 1-\boldsymbol{\alpha}^T\mathbf{x} \\ \mathbf{x}-\mathbf{X}\boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} 0 \\ \mathbf{0} \end{pmatrix},$$

from which we can conclude that $\mathbf{X}\boldsymbol{\alpha} = \mathbf{x}$.

Then, (4.62) is also equivalent to

$$\begin{aligned} \min \quad & \mathbf{Q}\bullet\mathbf{X} + 2\mathbf{c}^T\mathbf{X}\boldsymbol{\alpha} \\ & (\mathbf{X}\boldsymbol{\alpha}\ \mathbf{X}) \in \mathcal{G}, \\ & \boldsymbol{\alpha}^T\mathbf{X}\boldsymbol{\alpha} = 1. \end{aligned} \tag{4.63}$$

We have that

$$\begin{pmatrix} 1 & \boldsymbol{\alpha}^T\mathbf{X} \\ \mathbf{X}\boldsymbol{\alpha} & \mathbf{X} \end{pmatrix} \in \mathcal{C}_{n+1}^* \quad \Leftrightarrow \quad \mathbf{X} \in \mathcal{C}_n^*. \tag{4.64}$$

The ⇒ implication follows from the fact that principal submatrices of completely positive matrices are still completely positive. The ⇐ implication is also true. Indeed, let

$$\mathbf{X} = \sum_{k=1}^K \mathbf{z}_k\mathbf{z}_k^T, \quad \mathbf{z}_k \in \mathbb{R}_+^n\ \forall\, k.$$

Then,

$$\boldsymbol{\alpha}^T\mathbf{X}\boldsymbol{\alpha} = 1 \quad \Rightarrow \quad \sum_{k=1}^K(\boldsymbol{\alpha}^T\mathbf{z}_k)^2 = 1,$$

which further implies that

$$\begin{pmatrix} 1 & \boldsymbol{\alpha}^T\mathbf{X} \\ \mathbf{X}\boldsymbol{\alpha} & \mathbf{X} \end{pmatrix} = \sum_{k=1}^K \begin{pmatrix} (\boldsymbol{\alpha}^T\mathbf{z}_k)^2 & (\boldsymbol{\alpha}^T\mathbf{z}_k)\mathbf{z}_k^T \\ (\boldsymbol{\alpha}^T\mathbf{z}_k)\mathbf{z}_k & \mathbf{z}_k\mathbf{z}_k^T \end{pmatrix} \in \mathcal{C}_{n+1}^*,$$

where $\boldsymbol{\alpha}^T \mathbf{z}_k \geq 0$ since $\boldsymbol{\alpha}, \mathbf{z}_k \geq \mathbf{0}$. Therefore, we can finally rewrite (4.63) as

$$\begin{aligned} \min \quad & \mathbf{Q} \bullet \mathbf{X} + 2\mathbf{c}^T \mathbf{X}\boldsymbol{\alpha} & \\ & \mathbf{a}_i^T \mathbf{X}\boldsymbol{\alpha} = b_i, & i = 1,\ldots,m, \\ & \mathbf{a}_i^T \mathbf{X}\mathbf{a}_i = b_i^2, & i = 1,\ldots,m, \\ & [\mathbf{X}\boldsymbol{\alpha}]_j = X_{jj}, & j \in B, \\ & \mathbf{X} \in \mathcal{C}_n^*, & \\ & \boldsymbol{\alpha}^T \mathbf{X}\boldsymbol{\alpha} = 1. & \end{aligned} \tag{4.65}$$

In view of the chain of equivalences stated above, we can conclude, similar to Theorem 4.36, that problem (4.52) and problem (4.65) have the same optimal value, and if $\mathbf{X}^*$ is an optimal solution of (4.65), then $\mathbf{X}^*\boldsymbol{\alpha}$ is in the convex hull of the optimal solutions for (4.52).

A possible question is whether the results proven above can also be extended to the case of quadratic constraints

$$\mathbf{x}^T \mathbf{A}_i \mathbf{x} + \mathbf{a}_i^T \mathbf{x} \leq (=, \geq)\, a_{0i}.$$

In a way completely analogous to what we have previously seen, these constraints can be linearized as follows if we impose $\mathbf{X} = \mathbf{x}\mathbf{x}^T$,

$$\mathbf{A}_i \bullet \mathbf{X} + \mathbf{a}_i^T \mathbf{x} \leq (=, \geq)\, a_{0i}.$$

A relaxation is obtained through the introduction of the completely positive condition

$$\begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \mathbf{X} \end{pmatrix} \in \mathcal{C}_{n+1}^*,$$

but in the case of quadratic constraints we can not guarantee that the optimal value of the relaxation equals that of the original problem with the quadratic constraints. However, we can observe that if the cone of completely positive matrices is substituted with any tractable cone containing it (such as the semidefinite cone), a relaxation solvable in polynomial time is obtained (we refer to Section 4.4.4 for a more detailed discussion about the case of quadratic constraints). Moreover, the following remarks hold.

Remark 4.1. *Burer (Burer,* 2009*) extends the equivalence result to some cases where quadratic equality constraints are present. In fact, the binary constraint over a variable can be seen as a special case of quadratic equality constraint. Besides this case, in (Burer,* 2009*) a completely positive representation is also given for the case of complementarity constraints $x_j x_k = 0$ on bounded variables. The conditions under which the equivalence holds are also discussed in (Peña, Vera, & Zuluaga,* 2011*), a paper where copositive reformulations are extended to problems involving polynomial functions.*

Remark 4.2. *Burer and Dong (Burer & Dong,* 2012*) show that, under the assumption that the quadratic constraints define a nonempty and bounded feasible region, a quadratic* *problem with quadratic constraints can be reformulated as a generalized copositive program, i.e., a linear conic program over a cone of the form*

$$C(\mathcal{K}) = cl(chull\{\mathbf{X} \,:\, \mathbf{x} \in \mathcal{K},\, \mathbf{X} = \mathbf{x}\mathbf{x}^T\}),$$

which is called a generalized completely positive cone over $\mathcal{K}$ (the cone of completely positive matrices is a special case where $\mathcal{K} = \mathbb{R}^n_+$). In particular, in (Burer & Dong, 2012*) two reformulations are given. In both reformulations we have*

$$\mathcal{K} = \mathbb{R}_+ \times \mathcal{K}'.$$

In the first reformulation $\mathcal{K}'$ is a direct product of second-order cones (see Definition A.23*), while in the second one*

$$\mathcal{K}' = \mathcal{P}_{n+1} \times \mathbb{R}^r_+,$$

where r is the number of quadratic constraints.

Finally, we also make the following remark about Assumption 4.2.

Remark 4.3. *The relevance of Assumption* 4.2 *has been discussed in (Bomze & Jarre,* 2010*; Jarre,* 2012*). It is observed that if Assumption* 4.2 *is not satisfied, the optimal value of* (4.55) *may be strictly lower than that of* (4.52)*, so that enforcing Assumption* 4.2 *through the addition of slack variables may increase the optimal value of* (4.55)*. Instead, Jarre (Jarre,* 2012*) observes the following in spite of the addition of the slack variables:*

- *The optimal value of* (4.55) *does not change if the cone $\mathcal{C}^*_{n+1}$ is replaced by the semidefinite cone.*

- *If* (4.55) *is augmented with some valid linear inequalities, its optimal value does not change if the cone $\mathcal{C}^*_{n+1}$ is replaced by the intersection of the semidefinite and the nonnegative cone, i.e., the cone $\mathcal{DNN}_{n+1}$ of doubly nonnegative matrices (see Definition* A.20*).*

In particular, the linear inequalities to be added to (4.55) *are*

$$\mathbf{X}_B \leq \mathbf{x}_B\mathbf{e}^T, \quad \mathbf{X}_B \geq \mathbf{x}_B\mathbf{e}^T - \mathbf{e}\mathbf{x}_B^T - \mathbf{e}\mathbf{e}^T, \quad \mathbf{X}_{B,C} \leq \mathbf{e}\mathbf{x}_C^T,$$

where $\mathbf{x}_B$ and $\mathbf{X}_B$ denote, respectively, the subvector of binary variables and the corresponding portion of matrix $\mathbf{X}$, $\mathbf{x}_C$ denotes the subvector of continuous variables, and $\mathbf{X}_{B,C}$ denotes the portion of the matrix $\mathbf{X}$ related to products of binary and continuous variables.

How to exploit these reformulations

As previously remarked, if the representation (4.65) has a nonempty interior and is bounded, then strong duality holds, i.e., (4.65) has the same optimal value of its dual problem over the cone of copositive matrices. In the literature different hierarchies of (tractable) cones have been proposed which define increasingly sharper inner approximations of the copositive cone. One such hierarchy, introduced in (de Klerk & Pasechnik, 2002) and denoted here as $\{\mathcal{C}_n^r\}_{r\in\mathbb{N}_0}$, has been already presented in Section 2.5.1: $\mathcal{C}_n^r$ is made up by the symmetric matrices $\mathbf{M}$ for which the polynomial

$$P^{(r)}(\mathbf{x}) = \sum_{i,j=1}^{n} M_{ij} x_i^2 x_j^2 \left(\sum_{k=1}^{n} x_k^2\right)^r$$

has nonnegative coefficients. We have that

$$\mathcal{C}_n^r \subseteq \mathcal{C}_n^{r+1} \subseteq \mathcal{C}_n \quad \forall\, r.$$

The cone $\mathcal{C}_n^0$ is equal to the cone $\mathcal{N}_n$ of the symmetric matrices with nonnegative entries.

In (Parrillo, 2000) another, sharper, hierarchy, denoted by $\{\mathcal{K}_n^r\}_{r\in\mathbb{N}_0}$, has been proposed: $\mathcal{K}_n^r$ is made up by the symmetric matrices $\mathbf{M}$ for which the polynomial function $P^{(r)}(\mathbf{x})$ can be decomposed into a sum of squares. Obviously $\mathcal{C}_n^r \subseteq \mathcal{K}_n^r$, and

$$\mathcal{K}_n^r \subseteq \mathcal{K}_n^{r+1} \subseteq \mathcal{C}_n \quad \forall\, r.$$

The cone $\mathcal{K}_n^0$ is equal to $\mathcal{P}_n + \mathcal{N}_n$. A well-known theorem by Polya (Polya, 1974), states that for every homogeneous polynomial p positive over the unit simplex,

$$\left(\sum_{i=1}^{n} x_i\right)^r p(x_1,\ldots,x_n)$$

is a polynomial with positive coefficients for r sufficiently large. Therefore, every strictly copositive matrix lies in $\mathcal{C}_n^r$ (and, consequently, also in $\mathcal{K}_n^r$) for some sufficiently large integer r.

In (Peña, Vera, & Zuluaga, 2007) a further hierarchy of cones has been proposed. Noting that $\mathbf{M} \in \mathcal{K}_n^r$ if and only if

$$\left(\sum_{i=1}^{n} x_i\right)^r \mathbf{x}^T\mathbf{M}\mathbf{x} = \sum_{i_1,\ldots,i_n \,:\, \sum_{j=1}^n i_j \le r+2} g_{i_1,\ldots,i_n}(\mathbf{x}) \prod_{j=1}^{n} x_j^{i_j},$$

where each $g_{i_1,\ldots,i_n}$ can be written as a sum of squares, in (Peña et al., 2007) the cones $\mathcal{Q}_r^n$ have been defined as follows: $\mathbf{M} \in \mathcal{Q}_n^r$ if and only if

$$\left(\sum_{i=1}^{n} x_i\right)^r \mathbf{x}^T\mathbf{M}\mathbf{x} = \sum_{i_1,\ldots,i_n \,:\, \sum_{j=1}^n i_j = r} q_{i_1,\ldots,i_n}(\mathbf{x}) \prod_{j=1}^{n} x_j^{i_j},$$

where $q_{i_1,\ldots,i_n}(\mathbf{x})$ can be written as $\mathbf{x}^T\mathbf{Q}\mathbf{x}$, with $\mathbf{Q} \in \mathcal{P}_n + \mathcal{N}_n$. We have that $\mathcal{Q}_r^n \subseteq \mathcal{K}_r^n$ for all r (equality holds for $r = 0, 1$). Also, this hierarchy of cones is increasing with increasing r, i.e., $\mathcal{Q}_r^n \subset \mathcal{Q}_{r+1}^n$.

Bounds computed through these hierarchies of cones tend to be sharp. For instance, when applied to the computation of the stability number $\alpha(G)$ of a graph G, in (de Klerk & Pasechnik, 2002) it is conjectured that the hierarchy based on $\mathcal{K}_n^r$ cones delivers the exact number for $r \ge \alpha(G) - 1$, and this conjecture is proven to be true for graphs with $\alpha(G) \le 8$ in (Gvozdenovic & Laurent, 2007). However, although problems over these hierarchies of cones can be reformulated as semidefinite ($\mathcal{K}_r^n, \mathcal{Q}_r^n$) or linear programming problems ($\mathcal{C}_r^n$), the main drawback is that the size of these problems tends to increase quite rapidly with r, involving $n^{r+1} \times n^{r+1}$ matrix variables or a comparable number of $n \times n$ matrices.

In (Bundfuss & Duer, 2009) it is observed that all these hierarchies approximate the copositive cone uniformly, without taking into account any information based on the objective function. In the same work the authors provide a technique to define a *polyhedral* inner approximation (and also an outer one) with a constant number of variables but a number of constraints that grows as the approximation is refined. Moreover, the refining is guided by the observed objective function values. The technique (see also (Bundfuss & Duer, 2008)) is based on simplicial partitions of the unit simplex

$$\Delta = \left\{ \mathbf{x} \in \mathbb{R}^n \ : \ \sum_{i=1}^{n} x_i = 1,\ x_i \geq 0,\ i = 1,\ldots,n \right\}.$$

A simplicial partition $\mathcal{SP}_m = \{\Delta_1,\ldots,\Delta_m\}$ is such that

$$\Delta_i = chull\{\mathbf{v}_1^i,\ldots,\mathbf{v}_n^i\}, \quad \mathbf{v}_1^i,\ldots,\mathbf{v}_n^i \in \Delta,$$

and

$$\cup_{i=1}^m \Delta_i = \Delta, \quad int(\Delta_i) \cap int(\Delta_j) = \emptyset\ \forall\, i \neq j.$$

A possible characterization of copositive matrices is that matrix $\mathbf{A} \in \mathcal{C}_n$ if and only if

$$\mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0 \quad \forall\, \mathbf{x} \in \Delta.$$

By definition, $\mathbf{x} \in \Delta$ implies that $\mathbf{x} \in \Delta_i$ for some $i \in \{1,\ldots,m\}$. Therefore, we also have that $\mathbf{A} \in \mathcal{C}_n$ if and only if

$$\mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0 \quad \forall\, \mathbf{x} \in \Delta_i,\ i = 1,\ldots,m.$$

Now, let

$$\mathbf{V}_i = [\mathbf{v}_1^i \cdots \mathbf{v}_n^i]$$

be the matrix whose columns are the vertices of Δ_i. Then, we can also state that $\mathbf{A} \in \mathcal{C}_n$ if and only if

$$\mathbf{x}^T \mathbf{V}_i^T \mathbf{A} \mathbf{V}_i \mathbf{x} \geq 0 \quad \forall\, \mathbf{x} \in \Delta,\ i = 1,\ldots,m,$$

i.e., if and only if

$$\mathbf{V}_i^T \mathbf{A} \mathbf{V}_i \in \mathcal{C}_n \quad i = 1,\ldots,m. \tag{4.66}$$

Based on this result, in (Bundfuss & Duer, 2009) the copositive cone is approximated by the following cone:

$$\mathcal{I}^m = \{\mathbf{A} \ : \ \mathbf{V}_i^T \mathbf{A} \mathbf{V}_i \in \mathcal{N}_n,\ i = 1,\ldots,m\}.$$

Since $\mathcal{N}_n \subset \mathcal{C}_n$, it turns out that $\mathcal{I}^m \subseteq \mathcal{C}_n$. Moreover, the cone $\mathcal{I}^m$ is a polyhedral one. In fact, we remark here that a valid inner approximation can be attained by replacing $\mathcal{C}_n$ in (4.66) not only with $\mathcal{N}_n$ but with any tractable cone which inner approximates the copositive one, such as those of the previously presented hierarchies.

In (Bundfuss & Duer, 2009) the polyhedral cone

$$\mathcal{O}^m = \{\mathbf{A} \,:\, diag(\mathbf{V}_i^T \mathbf{A} \mathbf{V}_i) \geq \mathbf{0},\ i = 1,\ldots,m\}$$

was also introduced. Such cone outer approximates the copositive cone. It is also shown that, if we denote by

$$\delta(\mathcal{SP}_m) = \max_{k,h=1,\ldots,n,\ i=1,\ldots,m} \|\mathbf{v}_k^i - \mathbf{v}_h^i\|$$

the diameter of the partition, and if we keep on refining the partition so that $\delta(\mathcal{SP}_m) \to 0$ as $m \to \infty$, then under suitable assumptions (boundedness of the feasible region of the copositive problem, existence of strictly feasible points), the sequences of optimal solutions of the problems where the copositive cone is replaced by $\mathcal{I}^m$ and $\mathcal{O}^m$ have accumulation points which are all optimal solutions for the copositive problem.

A further idea to exploit the completely positive representation of some problems has been first investigated in (Bomze, Frommlet, & Locatelli, 2010) for the completely positive representation of the max-clique problem, and later extended to more general problems in (H. Dong & Anstreicher, 2010). Let

$$\mathcal{DNN}_n = \mathcal{P}_n \cap \mathcal{N}_n$$

be the cone of doubly nonnegative matrices (see Definition A.20) whose dual cone is $\mathcal{K}_n^0 = \mathcal{P}_n + \mathcal{N}_n$. We have that $\mathcal{C}_n^* \subseteq \mathcal{DNN}_n$. Therefore, $\mathcal{DNN}_n$ outer approximates $\mathcal{C}_n^*$. Actually, equality holds for $n \leq 4$, but for $n = 5$ strict inequality holds as the following matrix shows (see (Diananda, 1967)):

$$Q = \begin{bmatrix} 1 & -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 & -1 \\ -1 & 1 & 1 & -1 & 1 \end{bmatrix} \in \mathcal{DNN}_5 \setminus \mathcal{C}_5^*.$$

Then, following the same principle of cutting algorithms in integer programming, if we solve the relaxation of the completely positive problem over $\mathcal{DNN}_n$ and the solution $\mathbf{X}^*$ of the relaxation is such that

$$\mathbf{X}^* \in \mathcal{DNN}_n \setminus \mathcal{C}_n^*,$$

we should look for a matrix $\mathbf{V} \in \mathcal{C}_n$ such that

$$\mathbf{V} \bullet \mathbf{X}^* < 0.$$

Recalling that $\mathcal{C}_n$ is the dual cone of $\mathcal{C}_n^*$, by definition

$$\mathbf{V} \bullet \mathbf{Y} \geq 0 \quad \forall\, \mathbf{Y} \in \mathcal{C}_n^*.$$

Therefore, the linear inequality $\mathbf{V} \bullet \mathbf{X} \geq 0$ separates $\mathbf{X}^*$ from $\mathcal{C}_n^*$, which can be added to the relaxation to improve the quality of the bound.

The main difficulty with this approach is the detection of proper copositive matrices $\mathbf{V}$. In (Bomze et al., 2010) such matrices are derived from graphs with known clique number. In (H. Dong & Anstreicher, 2010) a separation procedure is first proposed for matrices $\mathbf{X} \in \mathcal{DNN}_5 \setminus \mathcal{C}_5^*$ with at least one off-diagonal zero. By permutations and diagonal scaling such matrices $\mathbf{X}$ can be rewritten as

$$\mathbf{X} = \begin{pmatrix} \mathbf{X}_{11} & \boldsymbol{\alpha}_1 & \boldsymbol{\alpha}_2 \\ \boldsymbol{\alpha}_1^T & 1 & 0 \\ \boldsymbol{\alpha}_2^T & 0 & 1 \end{pmatrix}, \tag{4.67}$$

with $\mathbf{X}_{11} \in \mathcal{DNN}_3$. In (Berman & Xu, 2004) it is proven that any matrix $\mathbf{X}$ with the form (4.67) belongs to $\mathcal{C}_5^*$ if and only if there exist matrices $\mathbf{A}_{11}, \mathbf{A}_{22}$ such that $\mathbf{X}_{11} = \mathbf{A}_{11} + \mathbf{A}_{22}$ and

$$\begin{pmatrix} \mathbf{A}_{ii} & \boldsymbol{\alpha}_i \\ \boldsymbol{\alpha}_i^T & 1 \end{pmatrix} \in \mathcal{DNN}_4 \quad i = 1,2.$$

Next, the following theorem is proven.

Theorem 4.39. *If* $\mathbf{X}$ *has the form* (4.67)*, then* $\mathbf{X} \in \mathcal{DNN}_5 \setminus \mathcal{C}_5^*$ *if and only if there exists*

$$\mathbf{V} = \begin{pmatrix} \mathbf{V}_{11} & \boldsymbol{\beta}_1 & \boldsymbol{\beta}_2 \\ \boldsymbol{\beta}_1^T & \gamma_1 & 0 \\ \boldsymbol{\beta}_2^T & 0 & \gamma_2 \end{pmatrix},$$

such that

$$\begin{pmatrix} \mathbf{V}_{11} & \boldsymbol{\beta}_i \\ \boldsymbol{\beta}_i^T & \gamma_i \end{pmatrix} \in \mathcal{DNN}_4^* \quad i = 1,2,$$

and $\mathbf{V} \bullet \mathbf{X} < 0$.

Proof. Consider the problem

$$\begin{aligned} \min \quad & 2\theta \\ & \begin{pmatrix} \mathbf{A}_{ii} & \boldsymbol{\alpha}_i \\ \boldsymbol{\alpha}_i^T & 1 \end{pmatrix} + \theta(\mathbf{I} + \mathbf{E}) \in \mathcal{DNN}_4, \quad i = 1,2, \\ & \mathbf{X}_{11} = \mathbf{A}_{11} + \mathbf{A}_{22}, \\ & \theta \geq 0. \end{aligned}$$

By the previously mentioned result proven in (Berman & Xu, 2004), the problem has optimal value equal to 0 if and only if $\mathbf{X} \in \mathcal{C}_5^*$. The dual of the above problem is

$$\begin{aligned} \max \quad & -(\mathbf{V}_{11} \bullet \mathbf{X}_{11} + 2\boldsymbol{\alpha}_1^T \boldsymbol{\beta}_1 + 2\boldsymbol{\alpha}_2^T \boldsymbol{\beta}_2 + \gamma_1 + \gamma_2) \\ & \begin{pmatrix} \mathbf{V}_{11} & \boldsymbol{\beta}_i \\ \boldsymbol{\beta}_i^T & \gamma_i \end{pmatrix} \in \mathcal{DNN}_4^*, \qquad i = 1,2, \\ & (\mathbf{I} + \mathbf{E}) \bullet \mathbf{V}_{11} + \mathbf{e}^T \boldsymbol{\beta}_1 + \mathbf{e}^T \boldsymbol{\beta}_2 + \gamma_1 + \gamma_2 \leq 1. \end{aligned}$$

Both problems are strictly feasible, so that strong duality holds. The proof is completed by observing that the objective function turns out to be equal to $-\mathbf{V} \bullet \mathbf{X}$. □

The matrix $\mathbf{V}$ defined in the theorem still does not do the job because $\mathbf{V} \in \mathcal{C}_5$ is not necessarily true. However, in (H. Dong & Anstreicher, 2010) it is shown that the following modification of $\mathbf{V}$,

$$\mathbf{V}(s) = \begin{pmatrix} \mathbf{V}_{11} & \boldsymbol{\beta}_1 & \boldsymbol{\beta}_2 \\ \boldsymbol{\beta}_1^T & \gamma_1 & s \\ \boldsymbol{\beta}_2^T & s & \gamma_2 \end{pmatrix},$$

still satisfies $\mathbf{V}(s) \bullet \mathbf{X} < 0$, and for any $s \geq \sqrt{\gamma_1 \gamma_2}$ we also have $\mathbf{V}(s) \in \mathcal{C}_5$.

A similar approach has also been proposed in (H. Dong & Anstreicher, 2010) for the separation of matrices $\mathbf{X} \in \mathcal{DNN}_n \setminus \mathcal{C}_n^*$ for $n > 5$, when

- $\mathbf{X}$ has the form

$$\mathbf{X} = \begin{pmatrix} \mathbf{X}_{11} & \mathbf{X}_{12} & \mathbf{X}_{13} & \cdots & \mathbf{X}_{1k} \\ \mathbf{X}_{12}^T & \mathbf{X}_{22} & \mathbf{O} & \cdots & \mathbf{O} \\ \mathbf{X}_{13}^T & \mathbf{O} & \mathbf{X}_{33} & \cdots & \mathbf{O} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{X}_{1k}^T & \mathbf{O} & \mathbf{O} & \cdots & \mathbf{X}_{kk} \end{pmatrix},$$

  with $k \geq 3$;

- given

$$\mathbf{X}^i = \begin{pmatrix} \mathbf{X}_{11} & \mathbf{X}_{1i} \\ \mathbf{X}_{1i}^T & \mathbf{X}_{ii} \end{pmatrix}, \quad i = 2, \ldots, k,$$

  we denote by $G(\mathbf{X}^i)$ the graph with edge set the support of $\mathbf{X}^i$

$$\{(r\ s)\ :\ X_{rs}^i \neq 0\},$$

  and require that each $G(\mathbf{X}^i)$ be a completely positive graph or, equivalently, contain no odd cycles of length greater than or equal to 5.

We refer the reader to (H. Dong & Anstreicher, 2010) for the details and also for the description of a further class of matrices for which a separation procedure is available.

Recently, the first separation algorithm for matrices in $\mathcal{C}_5^*$ has been proposed by Burer and Dong (Burer & Dong, 2013). For a given pointed polyhedral cone

$$P = \{\mathbf{x} \in \mathbb{R}_n\ :\ \mathbf{Ax} = \mathbf{0},\ \mathbf{Bx} \geq \mathbf{0}\},$$

such that

$$P \cap \{\mathbf{x} \,:\, x_1 = 1\} \neq \emptyset,$$

they consider the cone

$$\mathcal{W}_n = \mathcal{W}_n(P) = \{\mathbf{x}\mathbf{x}^T \,:\, \mathbf{x} \in P\}.$$

A semidefinite relaxation for this cone is

$$\mathcal{V}_n = \mathcal{V}_n(P) = \{\mathbf{X} \in \mathcal{P}_n \,:\, \mathbf{A}\mathbf{X}\mathbf{A}^T = \mathbf{O},\ \mathbf{B}\mathbf{X}\mathbf{B}^T \geq \mathbf{O}\}.$$

Burer and Dong propose a separation algorithm, based on the solution of a suitable optimization problem over a cone related to the boundary of P, which for some $\mathbf{X} \in \mathcal{V}_n$ establishes whether $\mathbf{X} \in \mathcal{W}_n$ or not, and in the latter case returns a matrix $\mathbf{Q}$ in the dual cone $\mathcal{W}_n^*$ such that $\mathbf{Q} \bullet \mathbf{X} < 0$. In particular, if $P = \mathbb{R}^n_+$, i.e., if $\mathbf{A}$ is empty and $\mathbf{B} = \mathbf{I}_n$, then $\mathcal{W}_n = \mathcal{C}_n^*$ and $\mathcal{V}_n = \mathcal{DNN}_n$. In this case the separation algorithm solves an optimization problem over n copies of the cone $\mathcal{C}_{n-1}$. In particular, for $n = 5$ we need to deal with the cone $\mathcal{C}_4$, which is a tractable cone.

4.4.2 QP and KKT conditions

In this subsection we will consider linear and semidefinite relaxations of QP problems based on the addition of KKT conditions (see Definition 2.19). We will first deal with the special case of QPs with box constraints (BoxQPs), and then move to general QPs.

Linear relaxations for BoxQPs

BoxQPs are a special case of (4.50) where $\mathbf{A}$ is the identity matrix $\mathbf{I}$ with the same dimension as the vector of variables $\mathbf{x}$, and $\mathbf{b}$ is a strictly positive vector. Without loss of generality, we can always assume that $\mathbf{b} = \mathbf{e}$. Therefore, BoxQP problems can be written as

$$\begin{aligned} \max \quad & \tfrac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x} \\ & \mathbf{0} \leq \mathbf{x} \leq \mathbf{e}. \end{aligned} \tag{4.68}$$

We remark that from here till the end of the section we will refer to *maximization* problems, following (Vandenbussche & Nemhauser, 2005b, 2005a; Burer & Vandenbussche, 2008, 2009). Of course, all the results can also be reformulated for minimization problems through some changes of sign. However, we decided to keep maximization in order to make much more straightforward the comparison for those readers who are willing to refer to the previously mentioned papers. We also remark that we will anticipate here some concepts (such as branching rules, branch-and-bound nodes, valid cuts) which will be more thoroughly discussed in Chapter 5, about the branch-and-bound methods.

Obviously, any global optimal solution of (4.68) is also a KKT point for the same problem. Therefore, Vandenbussche and Nemhauser in (Vandenbussche & Nemhauser, 2005b) (the work is also closely related to a previous work by Hansen et al. (Hansen, Jaumard, Ruiz, & Xiong, 1993)) observed that problem (4.68) is equivalent to the following:

$$\max \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x} \tag{4.69}$$

$$\mathbf{0} \leq \mathbf{x} \leq \mathbf{e}, \tag{4.70}$$

$$\mathbf{y} - \mathbf{Q}\mathbf{x} - \mathbf{z} = \mathbf{c}, \tag{4.71}$$

$$\mathbf{y}^T(\mathbf{e} - \mathbf{x}) = \mathbf{0}, \tag{4.72}$$

$$\mathbf{z}^T\mathbf{x} = \mathbf{0}, \tag{4.73}$$

$$\mathbf{y}, \mathbf{z} \geq \mathbf{0}, \tag{4.74}$$

where $\mathbf{y}, \mathbf{z}$ are the Lagrange multipliers for the constraints $\mathbf{x} \leq \mathbf{e}$ and $\mathbf{x} \geq \mathbf{0}$, respectively, while the constraints (4.70)–(4.74) represent the KKT conditions (an extension to a case with binary variables and fixed costs appearing in the objective function can be found in (T. C. Lin & Vandenbussche, 2008)). From (4.71)–(4.73), it follows that

$$\frac{1}{2}\mathbf{x}^T[\mathbf{Q}\mathbf{x} + \mathbf{c}] = \frac{1}{2}(\mathbf{y} - \mathbf{z})^T\mathbf{x} = \frac{1}{2}\mathbf{e}^T\mathbf{y}.$$

Therefore, the objective (4.69) can also be written as the linear function

$$\frac{1}{2}\mathbf{c}^T\mathbf{x} + \frac{1}{2}\mathbf{e}^T\mathbf{y} \tag{4.75}$$

(see also (Giannessi & Tomasin, 1973)). Such reformulation of BoxQP leads to the following linear relaxation:

$$\begin{aligned} \max\ & \frac{1}{2}\mathbf{c}^T\mathbf{x} + \frac{1}{2}\mathbf{e}^T\mathbf{y} \\ & \mathbf{0} \leq \mathbf{x} \leq \mathbf{e}, \\ & \mathbf{y} - \mathbf{Q}\mathbf{x} - \mathbf{z} = \mathbf{c}, \\ & \mathbf{y}, \mathbf{z} \geq \mathbf{0}, \end{aligned} \tag{4.76}$$

where the nonlinear complementarity conditions (4.72)–(4.73) have been dropped. The feasible region of (4.76) is unbounded. Indeed, given a feasible solution $(\bar{\mathbf{x}}, \bar{\mathbf{y}}, \bar{\mathbf{z}})$, any common positive vector can be added to both $\bar{\mathbf{y}}$ and to $\bar{\mathbf{z}}$ without losing feasibility. However, simple bounds for the variables $\mathbf{y}, \mathbf{z}$ can be derived. These are special cases of a class of valid inequalities for the problem (4.69)–(4.74), proposed in (Vandenbussche & Nemhauser, 2005b). Such a class is based on the *one row relaxation*. For each $i \in \{1, \ldots, n\}$, one can define the set

$$\begin{aligned} T_i = \{(\mathbf{x}, \mathbf{y}, \mathbf{z}) :\ & y_i - \textstyle\sum_{j=1}^n Q_{ij}x_j - z_i = c_i, \\ & y_i(1 - x_i) = 0,\ \ z_i x_i = 0, \\ & \mathbf{0} \leq \mathbf{x} \leq \mathbf{e},\ \ y_i, z_i \geq 0\}, \end{aligned}$$

i.e., the set of all points for which only the KKT conditions related to the index i are required to be satisfied. Obviously, for each i the set T_i contains the feasible region of the problem (4.69)–(4.74), and, consequently, each valid inequality for T_i is also a valid inequality for that problem. Each set T_i is the union of a finite number of polytopes. Therefore, its convex hull $chull(T_i)$ is a polytope. In (Vandenbussche & Nemhauser, 2005b) all the inequalities inducing nontrivial facets (i.e., those not induced by bounds on the variables) of $chull(T_i)$ have been characterized. First, the following quantities have been defined:

$$\bar{y}_i = c_i + Q_{ii} + \sum_{j \in N_i^+} Q_{ij},$$
$$\bar{z}_i = -c_i - \sum_{j \in N_i^-} Q_{ij},$$

where $N_i^+ = \{j \neq i \ : \ Q_{ij} \geq 0\}$ and $N_i^- = \{1,\ldots,n\} \setminus (\{i\} \cup N_i^+)$. We have that, for each $(\mathbf{x}\ \mathbf{y}\ \mathbf{z}) \in T_i$,

$$y_i > 0 \ \Rightarrow \ x_i = 1 \ \Rightarrow \ z_i = 0 \ \Rightarrow \ y_i \leq \bar{y}_i,$$

while

$$z_i > 0 \ \Rightarrow \ x_i = 0 \ \Rightarrow \ y_i = 0 \ \Rightarrow \ z_i \leq \bar{z}_i.$$

Note that, as a consequence of the above implications, we have that the valid bounds

$$y_i \leq \max\{0, \bar{y}_i\}, \quad z_i \leq \max\{0, \bar{z}_i\}, \tag{4.77}$$

could be added to (4.76) in order to make its feasible region bounded. In fact, as we will see, stronger inequalities could be added. The following theorem, whose rather long proof can be found in (Vandenbussche & Nemhauser, 2005b), characterizes all nontrivial facets of $chull(T_i)$ when $\bar{y}_i, \bar{z}_i > 0$ (if one or both of these quantities are nonpositive, similar results are reported in (Vandenbussche & Nemhauser, 2005b)).

Theorem 4.40. *If $\bar{y}_i, \bar{z}_i > 0$, then each nontrivial facet of $chull(T_i)$ is induced by an inequality belonging to one of the following two classes.*

Class I.

$$y_i + \sum_{j=1}^{n} \alpha_j x_j \leq \sum_{j \in N_i^-} \alpha_j,$$

with

- $\sum_{j=1}^{n} |\alpha_j| = \bar{y}_i$;
- $\alpha_i \leq 0$;
- $-Q_{ij} \leq \alpha_j \leq 0 \quad \forall\, j \in N_i^+$;
- $0 \leq \alpha_j \leq -Q_{ij} \quad \forall\, j \in N_i^-$.

Class II.

$$z_i + \sum_{j=1}^{n} \alpha_j x_j \leq \sum_{j \in N_i^+} \alpha_j + \alpha_i,$$

with

- $\sum_{j=1}^{n} |\alpha_j| = \bar{z}_i$;
- $\alpha_i \geq 0$;
- $Q_{ij} \leq \alpha_j \leq 0 \quad \forall\, j \in N_i^-$;
- $0 \leq \alpha_j \leq Q_{ij} \quad \forall\, j \in N_i^+$.

We illustrate the result through a simple example.

Example 4.41. Consider the BoxQP problem

$$\begin{aligned} \max \quad & \tfrac{1}{2}x_1^2 - \tfrac{1}{4}x_1 \\ & 0 \leq x_1 \leq 1. \end{aligned}$$

Notice that $\bar{y}_1 = \frac{3}{4} > 0$, $\bar{z}_1 = \frac{1}{4} > 0$. The set T_1 is defined as

$$\begin{aligned} T_1 = \{(x_1, y_1, z_1) \in \mathbb{R}^3 : \quad & y_1 - x_1 - z_1 = -\tfrac{1}{4}, \\ & y_1(1 - x_1) = 0, \;\; z_1 x_1 = 0, \\ & 0 \leq x_1 \leq 1, \;\; y_1, z_1 \geq 0\}, \end{aligned}$$

and is also equal to

$$\left\{\left(\frac{1}{4}\,0\,0\right)\right\} \cup \left\{\left(0\,0\,\frac{1}{4}\right)\right\} \cup \left\{\left(1\,\frac{3}{4}\,0\right)\right\}.$$

The set $chull(T_1)$ is then the following polytope:

$$\begin{aligned} chull(T_1) = \quad & \{(x_1\; y_1\; z_1) \; : \\ & y_1, z_1 \geq 0, \; 0 \leq x_1 \leq 1, \\ & x_1 + 4z_1 \leq 1, \; -3x_1 + 4y_1 \leq 0, \\ & 4x_1 - 4y_1 + 4z_1 = 1\}. \end{aligned}$$

The two inequalities $x_1 + 4z_1 \leq 1$ and $-3x_1 + 4y_1 \leq 0$, which are indeed those obtained via Theorem 4.40, both induce the nontrivial facet corresponding to the edge joining the two vertices $(0\;0\;\frac{1}{4})$ and $(1\;\frac{3}{4}\;0)$. The other two facets, namely, the edge joining the two vertices $(0\;0\;\frac{1}{4})$ and $(\frac{1}{4}\;0\;0)$ and the edge joining the two vertices $(\frac{1}{4}\;0\;0)$ and $(1\;\frac{3}{4}\;0)$, are trivial ones, induced, respectively, by $y_1 \geq 0$ and $z_1 \geq 0$. ■

Note that by taking

$$\alpha_i = -\bar{y}_i, \;\; \alpha_j = 0 \;\; \forall\, j \neq i$$

in the first class and

$$\alpha_i = \bar{z}_i, \;\; \alpha_j = 0 \;\; \forall\, j \neq i$$

in the second class, we are led to the inequalities

$$y_i \leq \bar{y}_i x_i, \quad z_i + \bar{z}_i x_i \leq \bar{z}_i,$$

which can be extended to

$$y_i \leq \max\{0, \bar{y}_i\} x_i, \quad z_i + \max\{0, \bar{z}_i\} x_i \leq \max\{0, \bar{z}_i\} \tag{4.78}$$

to cover also the cases where $\bar{y}_i \leq 0$ and/or $\bar{z}_i \leq 0$. Note that the inequalities (4.78) strengthen the bounds (4.77). Now, let us define the following polyhedra:

$$\begin{aligned} P_y^i = \{\boldsymbol{\alpha} \ : \quad & \textstyle\sum_{j\in N_i^-} \alpha_j - \sum_{j\in N_i^+} \alpha_j - \alpha_i = \bar{y}_i, \\ & -Q_{ij} \leq \alpha_j \leq 0 \quad \forall\, j \in N_i^+, \\ & 0 \leq \alpha_j \leq -Q_{ij} \quad \forall\, j \in N_i^-, \\ & \alpha_i \leq 0\}, \end{aligned}$$

and

$$\begin{aligned} P_z^i = \{\boldsymbol{\alpha} \ : \quad & -\textstyle\sum_{j\in N_i^-} \alpha_j + \sum_{j\in N_i^+} \alpha_j + \alpha_i = \bar{z}_i, \\ & Q_{ij} \leq \alpha_j \leq 0 \quad \forall\, j \in N_i^-, \\ & 0 \leq \alpha_j \leq Q_{ij} \quad \forall\, j \in N_i^+, \\ & \alpha_i \geq 0\}. \end{aligned}$$

Theorem 4.40 shows that the coefficients of each nontrivial facet of $chull(T_i)$ of class I are the coordinates of a vertex of the polyhedron P_y^i, while the coefficients of each nontrivial facet of $chull(T_i)$ of class II are the coordinates of a vertex of the polyhedron P_z^i. Such observation leads to the definition in (Vandenbussche & Nemhauser, 2005a) of a branch-and-cut approach for BoxQP. Instead of adding to (4.76) all the nontrivial facets of each set $chull(T_i)$ at the same time, some separation problems are solved in order to establish whether a given solution $(\tilde{\mathbf{x}}\ \tilde{\mathbf{y}}\ \tilde{\mathbf{z}})$ violates a nontrivial facet or not. More precisely, for each i, the following two problems are solved:

$$\begin{aligned} \max \quad & \tilde{y}_i + \textstyle\sum_{j=1}^n \alpha_j \tilde{x}_j - \sum_{j\in N_i^-} \alpha_j \\ & \boldsymbol{\alpha} \in P_y^i \end{aligned}$$

and

$$\begin{aligned} \max \quad & \tilde{z}_i + \textstyle\sum_{j=1}^n \alpha_j \tilde{x}_j - \sum_{j\in N_i^+} \alpha_j - \alpha_i \\ & \boldsymbol{\alpha} \in P_z^i. \end{aligned}$$

Both problems can be reformulated as continuous knapsack problems. For instance, after neglecting constant terms in the objective function, the latter can be rewritten as follows after the change of variables $\alpha_j = Q_{ij}\alpha'_j$ and the elimination of the variable α_i:

$$\begin{aligned} \max \quad & \textstyle\sum_{j\in N_i^+} Q_{ij}(\tilde{x}_j - \tilde{x}_i)\alpha'_j + \sum_{j\in N_i^-} Q_{ij}(\tilde{x}_j + \tilde{x}_i - 1)\alpha'_j \\ & \textstyle\sum_{j=1,\ j\neq i}^n |Q_{ij}|\alpha'_j \leq \bar{z}_i, \\ & 0 \leq \alpha'_j \leq 1, \qquad\qquad\qquad\qquad j = 1,\ldots,n,\ j \neq i. \end{aligned}$$

A nontrivial facet of $chull(T_i)$ is violated if and only if one of the problems above has optimal value larger than 0. In such a case we are also able to identify the most violated inequality, which can then be added to the current relaxation in order to strengthen it. In the branch-and-cut approach discussed in (Vandenbussche & Nemhauser, 2005a) branching rules are also defined. If the solution $(\mathbf{x}^*\ \mathbf{y}^*\ \mathbf{z}^*)$ of the current relaxation violates one of the complementarity conditions, (4.72) or (4.73), e.g., if $x_i^* z_i^* > 0$, branching is performed by fixing $x_i^* = 0$ in one branch and $z_i^* = 0$ in the other branch, so that the complementarity condition $x_i z_i = 0$ will certainly be satisfied in the two child nodes. Rules for the selection of the complementarity condition used for the branching are given in (Vandenbussche & Nemhauser, 2005a) and will be discussed in Section 5.3.6. At each node of the branch-and-bound tree some variables x_i are fixed to 0, some others to 1. That will imply some slight changes in the separation problems which have to be solved at each node. Let us denote by $L_{\mathcal{U}}$ the index set of the x variables whose value has *not* been fixed at the current node $\mathcal{U}$. First, we note that for each index $i \notin L_{\mathcal{U}}$, the inequalities (4.78) imply that the corresponding complementarity conditions are satisfied, so that for these indices no separation problem has to be solved. For $i \in L_{\mathcal{U}}$, the set T_i is now defined as follows:

$$\begin{aligned} T_i^{\mathcal{U}} = \{(\mathbf{x},\mathbf{y},\mathbf{z}) :\ & y_i - \textstyle\sum_{j\in L_{\mathcal{U}}} Q_{ij}x_j - z_i = c_i + \textstyle\sum_{j\notin L_{\mathcal{U}}\,:\,x_j=1} Q_{ij}, \\ & y_i(1-x_i) = 0,\ z_i x_i = 0, \\ & 0 \le x_j \le 1\ \ \forall\, j \in L_{\mathcal{U}},\ \ y_i, z_i \ge 0\}. \end{aligned}$$

The values $\bar{y}_i, \bar{z}_i$ are changed into

$$\begin{aligned} \bar{y}_i^{\mathcal{U}} &= \bar{y}_i + \textstyle\sum_{j\in N_i^-\setminus L_{\mathcal{U}}\,:\,x_j=1} Q_{ij} - \textstyle\sum_{j\in N_i^+\setminus L_{\mathcal{U}}\,:\,x_j=0} Q_{ij}, \\ \bar{z}_i^{\mathcal{U}} &= \bar{z}_i + \textstyle\sum_{j\in N_i^-\setminus L_{\mathcal{U}}\,:\,x_j=0} Q_{ij} - \textstyle\sum_{j\in N_i^+\setminus L_{\mathcal{U}}\,:\,x_j=1} Q_{ij}. \end{aligned}$$

Now, let us assume that $\bar{y}_i^{\mathcal{U}}, \bar{z}_i^{\mathcal{U}} > 0$. Then, Theorem 4.40 extends so that the nontrivial facets of $chull(T_i^{\mathcal{U}})$ are defined by the vertices of the polyhedra

$$\begin{aligned} P_{y,\mathcal{U}}^i = \{\boldsymbol{\alpha}\ :\ \ & \textstyle\sum_{j\in L_{\mathcal{U}}\cap N_i^-} \alpha_j - \textstyle\sum_{j\in L_{\mathcal{U}}\cap N_i^+} \alpha_j - \alpha_i = \bar{y}_i^{\mathcal{U}}, \\ & -Q_{ij} \le \alpha_j \le 0\ \ \forall\, j \in L_{\mathcal{U}} \cap N_i^+, \\ & 0 \le \alpha_j \le -Q_{ij}\ \ \forall\, j \in L_{\mathcal{U}} \cap N_i^-, \\ & \alpha_i \le 0\} \end{aligned}$$

and

$$\begin{aligned} P_{z,\mathcal{U}}^i = \{\boldsymbol{\alpha}\ :\ \ & -\textstyle\sum_{j\in L_{\mathcal{U}}\cap N_i^-} \alpha_j + \textstyle\sum_{j\in L_{\mathcal{U}}\cap N_i^+} \alpha_j + \alpha_i = \bar{z}_i^{\mathcal{U}}, \\ & Q_{ij} \le \alpha_j \le 0\ \ \forall\, j \in L_{\mathcal{U}} \cap N_i^-, \\ & 0 \le \alpha_j \le Q_{ij}\ \ \forall\, j \in L_{\mathcal{U}} \cap N_i^+, \\ & \alpha_i \ge 0\}. \end{aligned}$$

For a given vertex $\bar{\boldsymbol{\alpha}}$ of $P_{z,\mathcal{U}}^i$, which delivers a nontrivial facet for $chull(T_i^{\mathcal{U}})$, we define

$$\alpha_j = \begin{cases} Q_{ij} & \text{if } (j \in N_i^- \setminus L_{\mathcal{U}} \text{ and } x_j = 0) \text{ or } (j \in N_i^+ \setminus L_{\mathcal{U}} \text{ and } x_j = 1), \\ 0 & \text{if } (j \in N_i^+ \setminus L_{\mathcal{U}} \text{ and } x_j = 0) \text{ or } (j \in N_i^- \setminus L_{\mathcal{U}} \text{ and } x_j = 1), \\ \bar{\alpha}_j, & j \in L_{\mathcal{U}}. \end{cases}$$

This turns out to be a vertex of P_z^i that defines a nontrivial facet of $chull(T_i)$ (note that $\bar{y}_i^{\mathcal{U}}, \bar{z}_i^{\mathcal{U}} > 0$ implies $\bar{y}_i, \bar{z}_i > 0$). Given the solution $(\bar{\mathbf{x}}\ \bar{\mathbf{y}}\ \bar{\mathbf{z}})$ of the current relaxation at node $\mathcal{U}$, the equality

$$\bar{\alpha_i} + \sum_{j \in N_i^+ \cap L_{\mathcal{U}}} \bar{\alpha_j} - \bar{z}_i - \sum_{j \in L_{\mathcal{U}}} \bar{\alpha_j} \bar{x}_j = \alpha_i + \sum_{j \in N_i^+} \alpha_j - \bar{z}_i - \sum_{j \in L_{\mathcal{U}}} \alpha_j \bar{x}_j - \sum_{j \in N_i^+ \setminus L_{\mathcal{U}} \,:\, x_j = 1} \alpha_j$$

shows that the violation of the facet of $chull(T_i^{\mathcal{U}})$,

$$z_i + \sum_{j \in L_{\mathcal{U}}} \bar{\alpha}_j x_j \le \sum_{j \in N_i^+ \cap L_{\mathcal{U}}} \bar{\alpha}_j + \bar{\alpha}_i,$$

is exactly equal to the violation of the facet of $chull(T_i)$,

$$z_i + \sum_{j=1}^{n} \alpha_j x_j \le \sum_{j \in N_i^+} \alpha_j + \alpha_i.$$

However, while the former is a valid inequality only for node $\mathcal{U}$, the latter is valid at each node of the branch-and-bound tree. Of course, a completely analogous result can be derived for nontrivial facets of $chull(T_i^{\mathcal{U}})$ derived by vertices of $P_{y,\mathcal{U}}^i$.

Semidefinite programming (SDP) relaxations for general QPs

In (Burer & Vandenbussche, 2008) the relaxation (4.76) for BoxQP is extended to the general QP problem (4.50) (again we replace min with max). We will assume that the feasible region of the QP problem,

$$P = \{\mathbf{x} \in \mathbb{R}^n \;:\; \mathbf{A}\mathbf{x} \le \mathbf{b},\ \mathbf{x} \ge \mathbf{0}\},$$

is bounded and has a nonempty interior. Without loss of generality, we will also assume that P is contained in the unit box $\{\mathbf{x} \in \mathbb{R}^n \;:\; \mathbf{0} \le \mathbf{x} \le \mathbf{e}\}$. By imposing the KKT conditions, the QP problem can be rewritten as

$$\max \quad \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x} \tag{4.79}$$

$$\mathbf{A}\mathbf{x} \le \mathbf{b}, \tag{4.80}$$

$$\mathbf{y}^T\mathbf{A} - \mathbf{Q}\mathbf{x} - \mathbf{z} = \mathbf{c}, \tag{4.81}$$

$$\mathbf{y}^T(\mathbf{b} - \mathbf{A}\mathbf{x}) = \mathbf{0}, \tag{4.82}$$

$$\mathbf{z}^T\mathbf{x} = \mathbf{0}, \tag{4.83}$$

$$\mathbf{y}, \mathbf{z}, \mathbf{x} \ge \mathbf{0}, \tag{4.84}$$

where $\mathbf{y}, \mathbf{z}$ are the Lagrange multipliers for the constraints $\mathbf{A}\mathbf{x} \le \mathbf{b}$ and $\mathbf{x} \ge \mathbf{0}$, respectively. Under the given constraints, the objective function can be rewritten as

$$\frac{1}{2}\mathbf{c}^T\mathbf{x} + \frac{1}{2}\mathbf{b}^T\mathbf{y}.$$

By removing the nonlinear complementarity conditions (4.82)–(4.83), we end up with the following linear relaxation for QP:

$$\begin{aligned} \max \quad & \frac{1}{2}\mathbf{c}^T\mathbf{x} + \frac{1}{2}\mathbf{b}^T\mathbf{y} \\ & \mathbf{A}\mathbf{x} \le \mathbf{b}, \\ & \mathbf{y}^T\mathbf{A} - \mathbf{Q}\mathbf{x} - \mathbf{z} = \mathbf{c}, \\ & \mathbf{y}, \mathbf{z}, \mathbf{x} \ge \mathbf{0}. \end{aligned} \tag{4.85}$$

The following proposition, proven in (Burer & Vandenbussche, 2008), shows a drawback of the above relaxation.

Proposition 4.42. *Under the given assumptions on P (boundedness and nonempty interior), it is possible to prove that*

- *the set*
$$W = \{(\boldsymbol{\delta}_y\ \boldsymbol{\delta}_z) \ge (\mathbf{0}\ \mathbf{0}) : \ \boldsymbol{\delta}_y^T A - \boldsymbol{\delta}_z = 0\} \tag{4.86}$$
contains nonzero points;
- $\mathbf{b}^T\boldsymbol{\delta}_y \ge 0 \quad \forall\, (\boldsymbol{\delta}_y\ \boldsymbol{\delta}_z) \in W$*;*
- *if P has a nonempty interior, then*
$$\mathbf{b}^T\boldsymbol{\delta}_y > 0 \quad \forall\, (\boldsymbol{\delta}_y\ \boldsymbol{\delta}_z) \in W \setminus \{(\mathbf{0}\ \mathbf{0})\}.$$

Proof. The proof is based on the analysis of the dual of

$$\max_{\mathbf{x}\in P} \mathbf{d}^T\mathbf{x}$$

for two distinct choices of **d**. For any **d**, the dual problem is

$$\begin{aligned} \min \quad & \mathbf{b}^T\mathbf{y} \\ & \mathbf{y}^T\mathbf{A} - \mathbf{z} = \mathbf{d}, \\ & \mathbf{y}, \mathbf{z} \ge \mathbf{0}. \end{aligned}$$

We denote by $D_{\mathbf{d}}$ the feasible set of the dual. Since P is assumed to be nonempty and bounded, $D_{\mathbf{d}} \neq \emptyset$. Now, let us set $\mathbf{d} = \mathbf{e}$ and let $(\bar{\mathbf{y}}_0\ \bar{\mathbf{z}}_0) \in D_{\mathbf{e}}$. Then it turns out that

$$\bar{\mathbf{y}}_0^T\mathbf{A} - \bar{\mathbf{z}}_0 - \mathbf{e} = \mathbf{0},$$

i.e.,

$$(\mathbf{0}\ \mathbf{0}) \neq (\bar{\mathbf{y}}_0\ \bar{\mathbf{z}}_0 + \mathbf{e}) \in W.$$

Next, let us set $\mathbf{d} = \mathbf{0}$. In this case $D_{\mathbf{0}} = W$ and weak duality immediately implies that $\mathbf{b}^T\boldsymbol{\delta}_y \ge 0$ for all $(\boldsymbol{\delta}_y\ \boldsymbol{\delta}_z) \in W$. Moreover, if $\mathbf{x}_0$ is an interior point of P, it is also an optimal solution for the primal when $\mathbf{d}=\mathbf{0}$, so that the complementarity conditions imply that $(\mathbf{0}\ \mathbf{0})$ is the unique optimal solution of the dual. □

From this proposition it immediately follows that if P is bounded and has a nonempty interior, then (4.85) has an unbounded objective value. Indeed, given a point $(\bar{\mathbf{x}}\ \bar{\mathbf{y}}\ \bar{\mathbf{z}})$ feasible for (4.85), from Proposition 4.42 it follows that there exists $(\boldsymbol{\delta}_y\ \boldsymbol{\delta}_z) \neq (\mathbf{0}\ \mathbf{0})$ such that $(\bar{\mathbf{x}}\ \bar{\mathbf{y}}+\lambda\boldsymbol{\delta}_y\ \bar{\mathbf{z}}+\lambda\boldsymbol{\delta}_z)$ is also feasible for (4.85) for any $\lambda \geq 0$, and its objective function value diverges to infinity as $\lambda \to \infty$. Of course, the feasible region of BoxQP is bounded and has a nonempty interior, so that the issue of unbounded objective value arises also in that special case. However, it has previously been shown that for BoxQP it is possible to add the inequalities (4.78) to the linear relaxation (4.76), thus making the feasible region and, consequently, also the objective value of this problem bounded. The same is not possible for general QP problems. Burer and Vandenbussche (Burer & Vandenbussche, 2008) solved this difficulty through an SDP relaxation. Setting

$$\mathbf{Y}_{\mathbf{X},\mathbf{x}} = \begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \mathbf{x}\mathbf{x}^T \end{pmatrix}, \tag{4.87}$$

one can observe the following.

- For each $\mathbf{x} \in \mathbb{R}^n$,

  $$\frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x}+\mathbf{c}^T\mathbf{x} = \frac{1}{2}\tilde{\mathbf{Q}} \bullet \mathbf{Y}_{\mathbf{X},\mathbf{x}},$$

  where

  $$\tilde{\mathbf{Q}} = \begin{pmatrix} 0 & \mathbf{c}^T \\ \mathbf{c} & \mathbf{Q} \end{pmatrix}.$$

- If we multiply the constraints $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ by some $x_i \geq 0$, we obtain the valid inequalities $x_i\mathbf{b} - x_i\mathbf{A}\mathbf{x} \geq \mathbf{0}$, which can also be written as

  $$\mathbf{Y}_{\mathbf{X},\mathbf{x}}\mathbf{e}_i \in K = \{(x_0\ \mathbf{x}) \in \mathbb{R}_+^{n+1}\ :\ \mathbf{A}\mathbf{x} \leq x_0\mathbf{b}\}, \quad i=1,\dots,n, \tag{4.88}$$

  where $\mathbf{e}_i$ denotes the vector defining the direction of the ith axis so that $\mathbf{Y}_{\mathbf{X},\mathbf{x}}\mathbf{e}_i$ is the ith column of $\mathbf{Y}_{\mathbf{X},\mathbf{x}}$ (in particular, $\mathbf{Y}_{\mathbf{X},\mathbf{x}}\mathbf{e}_0$, i.e., the 0th column of $\mathbf{Y}_{\mathbf{X},\mathbf{x}}$, is the vector $(1\ \mathbf{x})$).

Noting that $\mathbf{Y}_{\mathbf{X},\mathbf{x}} \in \mathcal{P}_{n+1}$, a first SDP relaxation for the QP problem is the following:

$$\begin{aligned} \max\quad & \tfrac{1}{2}\tilde{\mathbf{Q}} \bullet \mathbf{Y}_{\mathbf{X},\mathbf{x}} \\ & \mathbf{Y}_{\mathbf{X},\mathbf{x}}\mathbf{e}_i \in K, \qquad i=1,\dots,n, \\ & \mathbf{Y}_{\mathbf{X},\mathbf{x}}\mathbf{e}_0 = (1\ \mathbf{x}), \\ & \mathbf{x} \in P, \\ & \mathbf{Y}_{\mathbf{X},\mathbf{x}} \in \mathcal{P}_{n+1}. \end{aligned} \tag{4.89}$$

Taking into account the KKT conditions, we can strengthen the above SDP relaxation as follows:

$$\begin{array}{lll} \max & \frac{1}{2}\tilde{\mathbf{Q}} \bullet \mathbf{Y}_{\mathbf{X},\mathbf{x}} & \\ & \mathbf{Y}_{\mathbf{X},\mathbf{x}}\mathbf{e}_i \in K, & i = 1,\dots,n, \\ & \mathbf{Y}_{\mathbf{X},\mathbf{x}}\mathbf{e}_0 = (1\ \mathbf{x}), & \\ & \mathbf{y}^T\mathbf{A} - \mathbf{Q}\mathbf{x} - \mathbf{z} = \mathbf{c}, & \\ & \mathbf{x} \in P, & \\ & \mathbf{Y}_{\mathbf{X},\mathbf{x}} \in \mathcal{P}_{n+1}, & \\ & \frac{1}{2}\tilde{\mathbf{Q}} \bullet \mathbf{Y}_{\mathbf{X},\mathbf{x}} = \frac{1}{2}(\mathbf{b}^T\mathbf{y} + \mathbf{c}^T\mathbf{x}), & \\ & \mathbf{y},\mathbf{z} \geq \mathbf{0}. & \end{array} \tag{4.90}$$

Opposite to the linear relaxation (4.85), the SDP relaxation (4.90) has a bounded feasible set and, consequently, a finite optimal value.

Proposition 4.43. *The feasible sets of (4.89) and (4.90) are bounded.*

Proof. By assumption $\mathbf{x}$ is bounded. Then, $\mathbf{Y}_{\mathbf{X},\mathbf{x}}\mathbf{e}_0$ is also bounded and, by symmetry of $\mathbf{Y}_{\mathbf{X},\mathbf{x}}$, the 0th row of $\mathbf{Y}_{\mathbf{X},\mathbf{x}}$ is also bounded. The recession cone for the constraint $\mathbf{Y}_{\mathbf{X},\mathbf{x}}\mathbf{e}_i \in K$ is

$$\{(r_0\ \mathbf{r}) \in \mathbb{R}_+^{n+1}\ :\ \mathbf{A}\mathbf{r} - r_0\mathbf{b} \leq \mathbf{0}\}.$$

In fact, in view of the boundedness of the 0th component of $\mathbf{Y}_{\mathbf{X},\mathbf{x}}\mathbf{e}_i$, we can fix $r_0 = 0$, i.e., the recession cone is

$$\{(0\ \mathbf{r})\ :\ \mathbf{A}\mathbf{r} \leq \mathbf{0}\}.$$

Therefore, $\mathbf{r}$ must belong to the recession cone of P, i.e., $\mathbf{r} = \mathbf{0}$ must be true in view of the boundedness of P. Then, $\mathbf{Y}_{\mathbf{X},\mathbf{x}}\mathbf{e}_i$, $i = 1,\dots,n$, is bounded, so that also the whole matrix $\mathbf{Y}_{\mathbf{X},\mathbf{x}}$ is bounded. This proves the boundedness of the feasible region of (4.89).

For what concerns (4.90), in view of the boundedness of $\mathbf{x}$ by assumption and of $\mathbf{Y}_{\mathbf{X},\mathbf{x}}$ by the proof above, and in view of the definition (4.86) for W, the recession cone of its feasible region is

$$\{(\boldsymbol{\delta}_{\mathbf{Y}_{\mathbf{X},\mathbf{x}}}\ \boldsymbol{\delta}_{\mathbf{x}}\ \boldsymbol{\delta}_{\mathbf{y}}\ \boldsymbol{\delta}_{\mathbf{z}})\ :\ (\boldsymbol{\delta}_{\mathbf{Y}_{\mathbf{X},\mathbf{x}}}\ \boldsymbol{\delta}_{\mathbf{x}}) = (\mathbf{O}\ \mathbf{0}),\ (\boldsymbol{\delta}_{\mathbf{y}}\ \boldsymbol{\delta}_{\mathbf{z}}) \in W,\ \mathbf{b}^T\boldsymbol{\delta}_{\mathbf{y}} = 0\}.$$

However, Proposition 4.42 proves that $(\boldsymbol{\delta}_{\mathbf{y}}\ \boldsymbol{\delta}_{\mathbf{z}}) = (\mathbf{0},\mathbf{0})$ must be true, so that the recession cone is only made up by the null vector, i.e., the feasible set is bounded. □

It also turns out that any optimal solution of (4.90) which satisfies the complementarity conditions also solves the original QP problem.

Proposition 4.44. *If* $(\mathbf{Y}_*\ \mathbf{x}_*\ \mathbf{y}_*\ \mathbf{z}_*)$ *is an optimal solution of* (4.90) *which satisfies the complementarity conditions*

$$\mathbf{y}_*^T(\mathbf{b} - \mathbf{A}\mathbf{x}_*) = \mathbf{0},\quad \mathbf{z}_*^T\mathbf{x}_* = \mathbf{0},$$

then $\mathbf{x}_*$ *solves the QP problem* (4.50).

Proof. By feasibility, we have that

$$\frac{1}{2}\tilde{\mathbf{Q}} \bullet \mathbf{Y}_* = \frac{1}{2}(\mathbf{b}^T\mathbf{y}_* + \mathbf{c}^T\mathbf{x}_*).$$

Note that the above quantity is also an upper bound for the optimal value of the QP problem. Since the complementarity conditions are satisfied, we also have that

$$\frac{1}{2}(\mathbf{b}^T\mathbf{y}_* + \mathbf{c}^T\mathbf{x}_*) = \frac{1}{2}\mathbf{x}_*^T\mathbf{Q}\mathbf{x}_* + \mathbf{c}^T\mathbf{x}_*,$$

i.e., $\mathbf{x}_* \in P$ has an objective function value equal to the upper bound of the QP problem and, consequently, is also an optimal solution of the QP problem. □

In (Burer & Vandenbussche, 2008) the above relaxation is incorporated in a branch-and-bound scheme, where branching is performed in the same way as for BoxQP: if the solution $(\mathbf{Y}_*\ \mathbf{x}_*\ \mathbf{y}_*\ \mathbf{z}_*)$ of a relaxation does not satisfy a complementarity condition, i.e., $z_i^* x_i^* > 0$ for some $i \in \{1,\ldots,n\}$, or $y_j^*(b_j - [\mathbf{A}\mathbf{x}_*]_j) > 0$ for some $j \in \{1,\ldots,m\}$, then branching is performed by setting $x_i = 0$ in one branch and $z_i = 0$ in the other branch in the former case, or by setting $y_j = 0$ in one branch and $b_j - [\mathbf{A}\mathbf{x}]_j = 0$ in the other branch in the latter case. Therefore, four sets,

$$F_{\mathbf{x}}, F_{\mathbf{z}} \subseteq \{1,\ldots,n\}, F_{\mathbf{y}}, F_{\mathbf{b}-\mathbf{A}\mathbf{x}} \subseteq \{1,\ldots,m\},$$

such that

$$F_{\mathbf{x}} \cap F_{\mathbf{z}} = \emptyset, \quad F_{\mathbf{y}} \cap F_{\mathbf{b}-\mathbf{A}\mathbf{x}} = \emptyset, \tag{4.91}$$

are associated to each node $\mathcal{U}$ of the branch-and-bound tree, and in the SDP relaxation for node $\mathcal{U}$ the following constraints are added to those for (4.90):

$$\begin{aligned} x_i &= 0, & i &\in F_{\mathbf{x}}, \\ z_i &= 0, & i &\in F_{\mathbf{z}}, \\ y_j &= 0, & j &\in F_{\mathbf{y}}, \\ b_j - [\mathbf{A}\mathbf{x}]_j &= 0, & j &\in F_{\mathbf{b}-\mathbf{A}\mathbf{x}}. \end{aligned} \tag{4.92}$$

Also, the definition of K is modified as follows:

$$K^{\mathcal{U}} = \{(x_0\ \mathbf{x}) \in \mathbb{R}_+^{n+1}\ :\ \mathbf{A}\mathbf{x} \le x_0\mathbf{b},\ [\mathbf{A}\mathbf{x}]_i = x_0 b_i,\ i \in F_{\mathbf{b}-\mathbf{A}\mathbf{x}},\ x_j = 0,\ j \in F_{\mathbf{x}}\}.$$

The results stated in Propositions 4.43 and 4.44 can be extended to the SDP relaxations at each node. In particular, the extension of Proposition 4.44 shows that each leaf node, i.e., a node such that

$$F_{\mathbf{x}} \cup F_{\mathbf{z}} = \{1,\ldots,n\}, \quad F_{\mathbf{y}} \cup F_{\mathbf{b}-\mathbf{A}\mathbf{x}} = \{1,\ldots,m\},$$

where the complementarity conditions are certainly satisfied, will certainly be fathomed either by infeasibility or because it delivers a solution of the QP problem with the same objective function value of the upper bound given by the SDP relaxation. As a consequence, the branch-and-bound approach is a finite and correct one.

Note that the unboundedness result for the linear relaxation (4.85) has only been proven for the root node, but may still be true at other nodes of the BB tree, especially those at the first levels of the tree where only few complementarity conditions are fixed, thus making the use of such relaxation not advisable. However, it has to be remarked that at a leaf node the linear relaxation, if feasible, returns an optimal value equal to the objective function value of (4.50) computed at the $\mathbf{x}$ part of the optimal solution of the relaxation, so that the leaf node is certainly fathomed.

In (Burer & Vandenbussche, 2008) another SDP relaxation, tighter than (4.90) but at the same time also significantly more computational demanding, is presented. We refer the reader to the paper for the details on such relaxation, as well as for some other details about the implementation, such as the inclusion of the constraint

$$\frac{1}{2}\tilde{\mathbf{Q}} \bullet \mathbf{Y}_{\mathbf{X},\mathbf{x}} = \frac{1}{2}(\mathbf{b}^T\mathbf{y} + \mathbf{c}^T\mathbf{x}),$$

which appears to slow down the solution of the SDP relaxations, only in a preprocessing phase, where bounds on the Lagrange multipliers $\mathbf{y}, \mathbf{z}$ are computed.

In (J. Chen & Burer, 2011) it is observed that, since a QP problem can be reformulated as a problem with linear constraints plus complementarity constraints which can always be rewritten in the form $x_j x_k = 0$, one can exploit the result proven in (Burer, 2009), and already commented on in Remark 4.1, to derive a completely positive representation of the problem, from which it is then possible to derive a semidefinite relaxation.

Remark 4.4. *The idea of branching based on KKT conditions was already exploited in (Audet, Hansen, Jaumard, & Savard,* 1999*) for a special QP case, namely, the case of disjoint bilinear programming:*

$$\begin{aligned} \max \quad & \mathbf{c}^T\mathbf{x} - \mathbf{y}^T\mathbf{Q}\mathbf{x} + \mathbf{y}^T\mathbf{d} \\ & \mathbf{A}\mathbf{x} \leq \mathbf{a}, \\ & \mathbf{B}\mathbf{y} \leq \mathbf{b}, \\ & \mathbf{x}, \mathbf{y} \geq \mathbf{0}. \end{aligned}$$

Any local optimum $(\bar{\mathbf{x}}\ \bar{\mathbf{y}})$ *of this problem is such that* $\bar{\mathbf{x}}$ *is an optimal solution of the linear program*

$$\begin{aligned} \max \quad & (\mathbf{c}^T - \bar{\mathbf{y}}^T\mathbf{Q})\mathbf{x} \\ & \mathbf{A}\mathbf{x} \leq \mathbf{a}, \\ & \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

whose dual is

$$\begin{aligned} \min \quad & \mathbf{v}^T\mathbf{a} \\ & \mathbf{v}^T\mathbf{A} \geq \mathbf{c}^T - \bar{\mathbf{y}}^T\mathbf{Q}, \\ & \mathbf{v} \geq \mathbf{0}, \end{aligned}$$

while $\bar{\mathbf{y}}$ is an optimal solution of the linear program

$$\begin{aligned} \max \quad & \mathbf{y}^T(\mathbf{d}-\mathbf{Q}\bar{\mathbf{x}}) \\ & \mathbf{B}\mathbf{y} \le \mathbf{b}, \\ & \mathbf{y} \ge \mathbf{0}, \end{aligned}$$

whose dual is

$$\begin{aligned} \min \quad & \mathbf{u}^T\mathbf{b} \\ & \mathbf{u}^T\mathbf{B} \ge \mathbf{d}-\mathbf{Q}\bar{\mathbf{x}}, \\ & \mathbf{u} \ge \mathbf{0}. \end{aligned}$$

Therefore, any local optimal solution satisfies the complementary slackness conditions for the two above pairs of primal-dual problems, i.e.,

$$\begin{aligned} \mathbf{v}^T(\mathbf{A}\bar{\mathbf{x}}-\mathbf{a}) = \mathbf{0}, \quad & (\mathbf{v}^T\mathbf{A}+\bar{\mathbf{y}}^T\mathbf{Q}-\mathbf{c}^T)\bar{\mathbf{x}} = \mathbf{0}, \\ \mathbf{u}^T(\mathbf{B}\bar{\mathbf{y}}-\mathbf{b}) = \mathbf{0}, \quad & \bar{\mathbf{y}}^T(\mathbf{d}-\mathbf{Q}\bar{\mathbf{x}}-\mathbf{u}^T\mathbf{B}) = \mathbf{0}. \end{aligned}$$

Then, in (Audet et al., 1999*) branching is performed by imposing that one of the above conditions, violated at the solution of the relaxation of the problem, is satisfied.*

Remark 4.5. *Hu et al. (Hu, Mitchell, & Pang,* 2012*) observe that problem* (4.79)–(4.84) *is a linear problem with complementarity constraints (LPCC), which can thus be reformulated as the following mixed integer problem:*

$$\begin{aligned} \max \quad & \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x}+\mathbf{c}^T\mathbf{x} \\ & \mathbf{A}\mathbf{x} \le \mathbf{b}, \\ & \mathbf{y}^T\mathbf{A}-\mathbf{Q}\mathbf{x}-\mathbf{z} = \mathbf{c}, \\ & \mathbf{0} \le \mathbf{b}-\mathbf{A}\mathbf{x} \le M\mathbf{u}, \\ & \mathbf{0} \le \mathbf{y} \le M(\mathbf{e}-\mathbf{u}), \\ & \mathbf{0} \le \mathbf{x} \le M\mathbf{w}, \\ & \mathbf{0} \le \mathbf{z} \le M(\mathbf{e}-\mathbf{w}), \\ & \mathbf{y},\mathbf{z},\mathbf{x} \ge \mathbf{0}, \\ & \mathbf{u} \in \{0,1\}^m, \quad \mathbf{w} \in \{0,1\}^n, \end{aligned}$$

where M is a large constant value. Then, they employ a method developed in (Hu, Mitchell, Pang, Bennett, & Kunapuli, 2008*) to solve the LPCC reformulated as a mixed integer problem. To be precise, some care is needed here because M is not known in advance and* *may not even exist. We refer to (Hu et al.,* 2008, 2012*) to see how to deal with this difficulty. In (Hu et al.,* 2012*) it is actually proposed to solve two LPCCs. The first one establishes whether the QP problem is bounded from above or not over the feasible region. In the latter case a feasible ray along which the objective function is unbounded is returned; in the former case a second LPCC is solved returning an optimal solution of the QP problem.*

SDP relaxations for BoxQPs

Obviously, the SDP relaxation proposed in (Burer & Vandenbussche, 2008) is defined also for the BoxQP subcase. However, the special structure of BoxQP suggests a different SDP relaxation, similar to (4.89), where the Lagrange multipliers $\mathbf{y}, \mathbf{z}$ are handled implicitly. Such relaxation has been presented in (Burer & Vandenbussche, 2009). The main observation is that, in the BoxQP subcase, given any point $\bar{\mathbf{x}} \in P$, we might set

$$\bar{\mathbf{y}} = \max\{\mathbf{0}, \mathbf{Q}\bar{\mathbf{x}} + \mathbf{c}\}, \quad \bar{\mathbf{z}} = -\min\{\mathbf{0}, \mathbf{Q}\bar{\mathbf{x}} + \mathbf{c}\}, \tag{4.93}$$

where max and min are taken componentwise. As a consequence, imposing in a node $\mathcal{U}$ that $y_j = 0$ is equivalent to the linear constraint, only depending on $\mathbf{x}$,

$$[\mathbf{Qx}]_j + c_j \leq 0,$$

while imposing $z_j = 0$ is equivalent to the linear constraint

$$[\mathbf{Qx}]_j + c_j \geq 0.$$

Therefore, at some node $\mathcal{U}$ the following SDP relaxation is solved:

$$\begin{array}{lll}
\max & \frac{1}{2}\tilde{\mathbf{Q}} \bullet \mathbf{Y}_{\mathbf{X},\mathbf{x}} & \\
& \mathbf{Y}_{\mathbf{X},\mathbf{x}}\mathbf{e}_i \in K^{\mathcal{U}}, & i = 1, \ldots, n, \\
& \mathbf{Y}_{\mathbf{X},\mathbf{x}}\mathbf{e}_0 = (1\ \mathbf{x}), & \\
& \mathbf{0} \leq \mathbf{x} \leq \mathbf{e}, & \\
& \mathbf{Y}_{\mathbf{X},\mathbf{x}} \in \mathcal{P}_{n+1}, & \\
& x_j = 0, & j \in F_{\mathbf{x}}, \\
& [\mathbf{Qx}]_j + c_j \geq 0, & j \in F_{\mathbf{z}}, \\
& [\mathbf{Qx}]_j + c_j \leq 0, & j \in F_{\mathbf{y}}, \\
& x_j = 1, & j \in F_{\mathbf{e}-\mathbf{x}},
\end{array}$$

where additional complementarity constraints are added to the feasible region of (4.89) and $K^{\mathcal{U}}$ is defined as either

$$\begin{aligned}
K^{\mathcal{U}} = \{(x_0\ \mathbf{x}) \in K \ : \quad & x_j = 0,\ j \in F_{\mathbf{x}},\ x_j = x_0,\ j \in F_{\mathbf{e}-\mathbf{x}}, \\
& [\mathbf{Qx}]_j + x_0 c_j \leq 0,\ j \in F_{\mathbf{y}},\ [\mathbf{Qx}]_j + x_0 c_j \geq 0,\ j \in F_{\mathbf{z}}\}
\end{aligned}$$

or in the simpler form

$$K^{\mathcal{U}} = \{(x_0\ \mathbf{x}) \in K\ :\ [\mathbf{Qx}]_j + x_0 c_j \le 0,\ j \in F_{\mathbf{y}},\ [\mathbf{Qx}]_j + x_0 c_j \ge 0,\ j \in F_{\mathbf{z}}\}.$$

Note that the Lagrange multipliers $\mathbf{y}, \mathbf{z}$ do not appear in this relaxation. Based on a result similar to that of Proposition 4.44, in (Burer & Vandenbussche, 2009) it is proven that the corresponding branch-and-bound approach is finite and correct. The computational results reported in the paper show that in this approach the number of nodes of the branch-and-bound tree is much smaller than the number of nodes for the branch-and-cut approach presented in (Vandenbussche & Nemhauser, 2005a), although the time required to obtain a bound at each node is considerably larger. Overall, according to the experiments, the branch-and-cut approach is faster over the smaller instances but becomes slower for larger instances with respect to the SDP-based branch-and-bound approach, which, thus, seems to scale better.

In (Burer & Chen, 2011) a further development has been discussed with semidefinite relaxations which incorporate (also) second-order necessary optimality conditions. Such relaxations are computationally compared with the most basic semidefinite relaxation for BoxQP due to (N. Shor, 1987), namely,

$$\begin{aligned} \max \quad & \tfrac{1}{2}\tilde{\mathbf{Q}} \bullet \mathbf{Y}_{\mathbf{X},\mathbf{x}} \\ & \mathbf{0} \le \mathbf{x} \le \mathbf{e}, \\ & \mathbf{Y}_{\mathbf{X},\mathbf{x}} \in \mathcal{P}_{n+1}, \\ & diag(\mathbf{Y}_{\mathbf{X},\mathbf{x}}) \le \begin{pmatrix} 1 \\ \mathbf{x} \end{pmatrix}. \end{aligned}$$

It turns out that the use of more advanced semidefinite relaxations allows us to reduce the number of nodes explored by a BB approach. However, on the other hand, this does not result in an overall reduction of the computation times.

4.4.3 Valid inequalities for BoxQPs

The BoxQP problem (4.68) can be reformulated as follows by adding $\frac{1}{2}n(n+1)$ variables y_{ij}, $i, j = 1, \dots, n$, $i \le j$, and moving all nonlinearities from the objective function to the constraints:

$$\begin{aligned} \max \quad & \tfrac{1}{2}\textstyle\sum_{i=1}^n Q_{ii} y_{ii} + \sum_{i=1}^n \sum_{j=i+1}^n Q_{ij} y_{ij} + \mathbf{c}^T \mathbf{x} \\ & \mathbf{0} \le \mathbf{x} \le \mathbf{e}, \\ & y_{ij} = x_i x_j, \qquad\qquad 1 \le i \le j \le n. \end{aligned}$$

If we denote by

$$S_n = \{(\mathbf{x}\ \mathbf{y}) \in [0,1]^{\frac{1}{2}(n^2+3n)}\ :\ y_{ij} = x_i x_j,\ 1 \le i \le j \le n\}$$

the feasible set of the above problem, then the BoxQP problem is also equivalent to the following problem:

$$\begin{aligned} \max \quad & \tfrac{1}{2}\textstyle\sum_{i=1}^n Q_{ii} y_{ii} + \sum_{i=1}^n \sum_{j=i+1}^n Q_{ij} y_{ij} + \mathbf{c}^T \mathbf{x} \\ & (\mathbf{x}\ \mathbf{y}) \in chull(S_n). \end{aligned}$$

Therefore, any outer approximation of $chull(S_n)$ by valid inequalities corresponds to a relaxation of BoxQP. Note that $chull(S_n)$ is not a polyhedral region. This can be easily seen even for the case $n = 1$, where

$$chull(S_1) = \{(x_1\ y_{11}) \in [0,1]^2\ :\ y_{11} \le x_1,\ y_{11} \ge x_1^2\}. \tag{4.94}$$

In what follows, for a given $\mathbf{x} \in [0,1]^n$, we will denote by $\mathbf{y_x}$ the corresponding point such that $(\mathbf{x}\ \mathbf{y_x}) \in S_n$.

Burer and Letchford in (Burer & Letchford, 2009) performed a very detailed analysis of $chull(S_n)$, also extending some previously known results. First, they proved the results collected in the following theorem. We need to recall here the definitions of dimension of a convex set X (see Definition A.11), of extreme points of X (see Definition A.7), and that of vertices of X (see Definition A.9). For instance, if we consider the set $chull(S_1)$, defined in (4.94), we see that its dimension is 2, its extreme points are those lying along the curve $y_{11} = x_1^2$, $x_1 \in [0,1]$, and its vertices are the two binary points (0 0) and (1 1) along this curve.

Theorem 4.45. *We have that*

1. *the set $chull(S_n)$ is full-dimensional (i.e., its dimension is $\frac{1}{2}(n^2+3n)$);*
2. *its extreme points are the points in S_n;*
3. *its vertices are all the binary points in S_n, i.e., points for which $x_i \in \{0,1\}$ for all $i = 1,\dots,n$.*

Proof. In order to prove that $chull(S_n)$ is full-dimensional we need to find $\frac{1}{2}(n^2+3n)+1$ affinely independent points belonging to this set. It turns out that the following points $(\mathbf{x}\ \mathbf{y_x}) \in S_n$ do the job:

- the origin;
- the n points with $x_i = 1$, $x_j = 0$ for $i = 1,\dots,n$, $j \neq i$;
- the n points with $x_i = \frac{1}{2}$, $x_j = 0$ for $i = 1,\dots,n$, $j \neq i$;
- the $\frac{1}{2}(n^2-n)$ points with $x_i = x_j = 1$, $x_k = 0$ for $1 \le i < j \le n$, $k \neq i,j$.

By definition, every extreme point of $chull(S_n)$ belongs to S_n. What we need to show is that each point in S_n is an extreme point. Let $(\bar{\mathbf{x}}\ \mathbf{y}_{\bar{\mathbf{x}}})$ be a generic point in S_n. Consider the BoxQP where we minimize the objective function

$$\sum_{i=1}^{n}(x_i^2 - 2\bar{x}_i x_i) = \sum_{i=1}^{n}(x_i - \bar{x}_i)^2 - \sum_{i=1}^{n}\bar{x}_i^2.$$

Obviously, $\bar{\mathbf{x}}$ is the *unique* minimum point of the problem with this objective function and, consequently, $(\bar{\mathbf{x}}\ \mathbf{y}_{\bar{\mathbf{x}}})$ is the unique minimizer of the linear function $\sum_{i=1}^{n}(y_{ii} - 2\bar{x}_i x_i)$ over $chull(S_n)$. Then, $(\bar{\mathbf{x}}\ \mathbf{y}_{\bar{\mathbf{x}}})$ is an extreme point of $chull(S_n)$.

To prove the last statement, we first show that any binary point $(\bar{\mathbf{x}}\ \mathbf{y}_{\bar{\mathbf{x}}})$ is a vertex. Without loss of generality, we can assume that the variables equal to 0 in $\bar{\mathbf{x}}$ are the first t

ones. We need to show that $\bar{\mathbf{x}}$ satisfies as an equality $\frac{1}{2}(n^2+3n)$ linearly independent valid inequalities. These are

- $x_i \geq 0$ for all i such that $\bar{x}_i = 0$;
- $x_i \leq 1$ for all i such that $\bar{x}_i = 1$;
- $y_{ij} \geq 0$ for all i such that $\bar{x}_i = 0$, and all $i \leq j \leq n$;
- $y_{ij} \leq 1$ for all $i \leq j$ such that $\bar{x}_i = 1$ (recall that $i \leq j$ and $\bar{x}_i = 1$ imply $\bar{x}_j = 1$).

To prove the reverse, let us assume by contradiction that the extreme point $(\bar{\mathbf{x}}\ \mathbf{y}_{\bar{\mathbf{x}}})$ is a vertex with $\bar{x}_k \in (0,1)$ for some $k \in \{1,\dots,n\}$. For some small $\varepsilon > 0$, let $(\bar{\mathbf{x}}_0(\varepsilon)\ \mathbf{y}_{\bar{\mathbf{x}}_0(\varepsilon)}) \in S_n$ be obtained from $(\bar{\mathbf{x}}\ \mathbf{y}_{\bar{\mathbf{x}}})$ by subtracting ε from $\bar{x}_k$, and let $(\bar{\mathbf{x}}_1(\varepsilon)\ \mathbf{y}_{\bar{\mathbf{x}}_1(\varepsilon)}) \in S_n$ be obtained from $(\bar{\mathbf{x}}\ \mathbf{y}_{\bar{\mathbf{x}}})$ by adding ε to $\bar{x}_k$. Now, let us consider an arbitrary supporting hyperplane $\mathbf{v}^T\mathbf{x}+\mathbf{w}^T\mathbf{y}$ for $chull(S_n)$ at the point $(\bar{\mathbf{x}}\ \mathbf{y}_{\bar{\mathbf{x}}})$. Then,

$$\mathbf{v}^T\bar{\mathbf{x}}+\mathbf{w}^T\mathbf{y}_{\bar{\mathbf{x}}} \geq \mathbf{v}^T\bar{\mathbf{x}}_0(\varepsilon)+\mathbf{w}^T\mathbf{y}_{\bar{\mathbf{x}}_0(\varepsilon)},$$
$$\mathbf{v}^T\bar{\mathbf{x}}+\mathbf{w}^T\mathbf{y}_{\bar{\mathbf{x}}} \geq \mathbf{v}^T\bar{\mathbf{x}}_1(\varepsilon)+\mathbf{w}^T\mathbf{y}_{\bar{\mathbf{x}}_1(\varepsilon)},$$

which, on the other hand, are equivalent to

$$v_k + \textstyle\sum_{i\neq k} w_{ik}\bar{x}_i + (2\bar{x}_k+\varepsilon)w_{kk} \leq 0,$$
$$-v_k - \textstyle\sum_{i\neq k} w_{ik}\bar{x}_i - (2\bar{x}_k-\varepsilon)w_{kk} \leq 0.$$

Since the two inequalities above are true for any (sufficiently small) $\varepsilon > 0$, we can conclude that for any vector $(\mathbf{v}\ \mathbf{w})$ defining a supporting hyperplane at $(\bar{\mathbf{x}}\ \mathbf{y}_{\bar{\mathbf{x}}})$, we have the following equation:

$$v_k + \sum_{i\neq k} w_{ik}\bar{x}_i + 2\bar{x}_k w_{kk} = 0.$$

But this means that there can not exist $\frac{1}{2}(n^2+3n)$ linearly independent supporting hyperplanes, thus contradicting the fact that $(\bar{\mathbf{x}}\ \mathbf{y}_{\bar{\mathbf{x}}})$ is a vertex. □

Next, Burer and Letchford studied different valid inequalities for $chull(S_n)$. Some of them have already been introduced.

RLT inequalities. The four inequalities

$$y_{ij} \geq 0,\ \ y_{ij} \leq x_i,\ \ y_{ij} \leq x_j,\ \ y_{ij} \geq x_i + x_j - 1 \tag{4.95}$$

are RLT inequalities already discussed in Section 4.3. Burer and Letchford proved that the inequalities $y_{ii} \leq x_i$, $i = 1,\dots,n$, and all the inequalities (4.95) for $i \neq j$ induce facets of $chull(S_n)$.

Psd inequalities. Positive semidefinite (psd) inequalities, which can also be viewed as RLT inequalities, were introduced in Section 4.3: for each vector $(\boldsymbol{\alpha}\ \alpha_{n+1}) \in \mathbb{R}^{n+1}$, substitute each term x_ix_j with y_{ij} into the inequality

$$(\boldsymbol{\alpha}^T\mathbf{x}+\alpha_{n+1})^2 \geq 0$$

in order to end up with the psd inequality

$$2\alpha_{n+1}\boldsymbol{\alpha}^T\mathbf{x}+\sum_{i=1}^{n}\alpha_i^2 y_{ii}+2\sum_{i=1}^{n}\sum_{j=i+1}^{n}\alpha_i\alpha_j y_{ij}+\alpha_{n+1}^2\geq 0. \tag{4.96}$$

Note that for all $(\mathbf{x}\ \mathbf{y})\in S_n$,

$$\begin{array}{l} 2\alpha_{n+1}\boldsymbol{\alpha}^T\mathbf{x}+\sum_{i=1}^{n}\alpha_i^2 y_{ii}+2\sum_{i=1}^{n}\sum_{j=i+1}^{n}\alpha_i\alpha_j y_{ij}+\alpha_{n+1}^2=0\\ \Updownarrow\\ \boldsymbol{\alpha}^T\mathbf{x}+\alpha_{n+1}=0. \end{array} \tag{4.97}$$

From (4.97) it follows that the face of $chull(S_n)$ induced by the psd inequality (4.96) is given by the following set:

$$\{\mathbf{x}\in chull(S_n)\ :\ \boldsymbol{\alpha}^T\mathbf{x}+\alpha_{n+1}=0\}. \tag{4.98}$$

Burer and Letchford showed that a key role is played by the dimension of the set

$$K(\boldsymbol{\alpha},\alpha_{n+1})=\{\mathbf{x}\in[0,1]^n\ :\ \boldsymbol{\alpha}^T\mathbf{x}+\alpha_{n+1}=0\}.$$

If this set has dimension lower than $n-1$, then the corresponding psd inequality is dominated by the inequalities (4.95) (i.e., it can be obtained as a convex combination of these inequalities), while if the dimension is equal to $n-1$, the face of $chull(S_n)$ induced by the inequality has dimension $\frac{1}{2}(n^2+n-2)$. In the latter case, either (i) $K(\boldsymbol{\alpha},\alpha_{n+1})$ contains a point in the interior of the unit hypercube $[0,1]^n$, in which case the face is a maximal one for $chull(S_n)$ (i.e., it is not contained in some other face of $chull(S_n)$) and the psd inequality is not dominated; or (ii) $K(\boldsymbol{\alpha},\alpha_{n+1})$ is a facet of the unit hypercube. In the latter case the psd inequality reduces to an RLT inequality $y_{ii}\geq 0$ or $y_{ii}\geq 2x_i-1$. Indeed, if $K(\boldsymbol{\alpha},\alpha_{n+1})$ is a facet of the unit hypercube, then $\boldsymbol{\alpha}^T\mathbf{x}+\alpha_{n+1}\equiv x_i=0$ or $\boldsymbol{\alpha}^T\mathbf{x}+\alpha_{n+1}\equiv x_i=1$ for some $i\in\{1,\ldots,n\}$. That is, in the psd inequality (4.96) we have either $\alpha_i=1$, $\alpha_j=0$, $j\neq i$, $\alpha_{n+1}=0$, so that the psd inequality reduces to $y_{ii}\geq 0$, or $\alpha_i=1$, $\alpha_j=0$, $j\neq i$, $\alpha_{n+1}=-1$, so that the psd inequality reduces to $y_{ii}\geq 2x_i-1$. Such faces are contained in the facet induced by the RLT inequality $y_{ii}\leq x_i$ but, still, are not dominated. This counterintuitive nondominance result is illustrated in (Burer & Letchford, 2009) through the simple $n=1$ case. Let us consider the psd inequalities with $(\alpha_1\ \alpha_2)$, respectively, equal to $(1\ 0)$ and $(1\ -1)$. In the first case, we have that $K(\alpha_1,\alpha_2)=\{0\}$, lying along the facet $x_1=0$ of the unit hypercube; the psd inequality is $y_{11}\geq 0$; and the corresponding face of $chull(S_1)$ only contains the origin. This is contained in the facet induced by $y_{11}\leq x_1$ but the psd inequality $y_{11}\geq 0$ can not be obtained as a convex combination of other valid linear inequalities and is, thus, not dominated. Similarly, in the second case, we have that $K(\alpha_1,\alpha_2)=\{1\}$, lying along the facet $x_1=1$; the psd inequality is $y_{11}\geq 2x_1-1$; and the corresponding face of $chull(S_1)$ only contains point $(1\ 1)$. This is also contained in the facet induced by $y_{11}\leq x_1$ but, again, the psd inequality $y_{11}\geq 2x_1-1$ can not be obtained as a convex combination of other valid linear inequalities.

Valid inequalities for the Boolean quadric polytope Yajima and Fujie in (Yajima & Fujie, 1998) extend to $chull(S_n)$ some results previously obtained by Padberg (Padberg, 1989) for so-called *Boolean quadric polytopes*:

$$BQP_n = chull\{(\mathbf{x},\mathbf{y}) \in \{0,1\}^{\frac{1}{2}(n^2+n)} \ : \ y_{ij} = x_i x_j, \ 1 \leq i < j \leq n\}.$$

Padberg introduced different valid inequalities for BQP_n such as, e.g., the *triangle inequalities*, defined for each triple i, j, k as

$$x_i + x_j + x_k \leq y_{ij} + y_{ik} + y_{jk} + 1,$$
$$y_{ij} + y_{ik} \leq x_i + y_{jk},$$

or the *clique inequalities*, defined for each $U \subseteq \{1, \ldots, n\}$ and $\beta \leq |U|$ as

$$\beta \sum_{i \in U} x_i - \sum_{i,j \in U \ : \ i<j} y_{ij} \leq \frac{1}{2}\beta(\beta+1).$$

Besides proving the validity of these inequalities for $chull(S_n)$ also, Yajima and Fujie also proved the validity for $chull(S_n)$ of so-called *cut-type inequalities*, first introduced in (Sherali, Lee, & Adams, 1995). These are defined as follows. Let $U, W \subseteq \{1, \ldots, n\}$ be such that $U \cap W = \emptyset$ and β be an integer. Then,

$$\begin{array}{l} \sum_{i,j \in U \ : \ i<j} y_{ij} + \sum_{i,j \in W \ : \ i<j} y_{ij} - \sum_{i,j: \ i<j, \ ((i \in U, j \in W) \text{ or } (i \in W, j \in U))} y_{ij} \\ -\beta \sum_{i \in U} x_i + (1+\beta) \sum_{i \in W} x_i + \frac{1}{2}\beta(\beta+1) \geq 0 \end{array} \tag{4.99}$$

are called cut-type inequalities (notice that the clique inequalities are a special case of the cut-type ones, where $W = \emptyset$). Based on these valid inequalities, in (Yajima & Fujie, 1998) a cutting plane algorithm is proposed with some techniques employed to identify valid inequalities violated by the solution of the current relaxation. The proof of the validity for $chull(S_n)$ of the cut-type inequalities proposed in (Yajima & Fujie, 1998) is not straightforward. Burer and Letchford could prove an even more general result in a simpler way. The result is based on the following theorem.

Theorem 4.46. *We have that*

$$BQP_n = chull\{(\mathbf{x}\ \mathbf{y}) \in [0,1]^{\frac{1}{2}(n^2+n)} \ : \ y_{ij} = x_i x_j, \ 1 \leq i < j \leq n\},$$

where the set on the right-hand side is the projection of $chull(S_n)$ onto $\mathbb{R}^{\frac{1}{2}(n^2+n)}$.

Proof. First we notice that $(\mathbf{x}\ \mathbf{y}) \in [0,1]^{\frac{1}{2}(n^2+n)}$ can be an extreme point for the projection of $chull(S_n)$ only if $y_{ij} = x_i x_j$ for all $i \neq j$. Otherwise, $(\mathbf{x}\ \mathbf{y})$ is the projection of a point in $chull(S_n)$ which is *not* an extreme point of $chull(S_n)$ and can, thus, be obtained as a convex combination of at least two distinct extreme points in $chull(S_n)$. Consequently, the projections of these extreme points deliver a convex combination of points returning the point $(\mathbf{x}\ \mathbf{y})$ which, thus, can not be an extreme point of the projection of $chull(S_n)$. Therefore, we can restrict the attention to points $(\mathbf{x}\ \mathbf{y})$ for which $y_{ij} = x_i x_j$ for all $i \neq j$. Let us assume that $(\mathbf{x}\ \mathbf{y})$ has at least one fractional component, say $x_k \in (0,1)$. Then, $(\mathbf{x}\ \mathbf{y})$ is not an extreme point of the projection because it can be obtained as the convex combination

$$(1-\lambda)(\mathbf{x}^0\ \mathbf{y}^0) + \lambda(\mathbf{x}^1\ \mathbf{y}^1),$$

where $\lambda = x_k$, $\mathbf{x}^\delta$, $\delta \in \{0,1\}$, is obtained from $\mathbf{x}$ by setting $x_k^\delta = \delta$, $x_i^\delta = x_i$ for all $i \neq k$, and $\mathbf{y}_{ij}^\delta = x_i^\delta x_j^\delta$ for all $1 \leq i < j \leq n$. Therefore, all extreme points in the projection must be binary points, i.e., the projection is exactly equal to BQP_n. □

The result of the theorem shows that *all* valid inequalities for BQP_n are also valid for $chull(S_n)$,[3] thus generalizing the result of Yajima and Fujie about the possibility of extending to $chull(S_n)$ the validity of some inequalities for BQP_n. Next, Burer and Letchford face the question of which facets of BQP_n are also facet of $chull(S_n)$. They prove the following result.

Theorem 4.47. *Let an inequality inducing a facet of BQP_n be given. Then, it also induces a facet of $chull(S_n)$ if and only if there exist n points $(\mathbf{x}^1\ \mathbf{y}^1),\dots,(\mathbf{x}^n\ \mathbf{y}^n) \in S_n$ such that*

- *they satisfy the inequality as an equality;*
- $x_i^i \in (0,1)$*,* $i = 1,\dots,n$*;*
- $x_i^j \in \{0,1\}$*,* $i \neq j$*;*

or, equivalently, if and only if there exist $2n$ vertices of BQP_n $(\bar{\mathbf{x}}^1\ \bar{\mathbf{y}}^1),\dots,(\bar{\mathbf{x}}^n\ \bar{\mathbf{y}}^n)$ and $(\tilde{\mathbf{x}}^1\ \tilde{\mathbf{y}}^1),\dots,(\tilde{\mathbf{x}}^n\ \tilde{\mathbf{y}}^n)$ (not necessarily distinct ones) such that

- *they satisfy the inequality as an equality;*
- $\bar{x}_i^i = 1 - \tilde{x}_i^i$*,* $i = 1,\dots,n$*;*
- $\bar{x}_i^j = \tilde{x}_i^j$*,* $i \neq j$*.*

Proof. $\Rightarrow$ For each $i \in \{1,\dots,n\}$, there must exist some extreme point $(\mathbf{x}^i\ \mathbf{y}^i)$ of $chull(S_n)$ lying on the face and with x_i fractional. If this were not the case, the face could not be a facet of $chull(S_n)$. Indeed, the extreme points of $chull(S_n)$ lying within the face would also lie within the facet induced by the RLT inequality $y_{ii} \leq x_i$, so that the whole face would be contained in such facet and, in fact, strictly contained, since the inequality inducing the face is valid for BQP_n and does not contain the variable y_{ii}. Now, if all the other coordinates of $\mathbf{x}^i$ are binary ones, then we are done. Otherwise, if for some $k \neq i$, $x_k^i \in (0,1)$, we can proceed as in the proof of Theorem 4.46. First we take $(\mathbf{x}^i\ \mathbf{y}^i)$ as a convex combination of two points $(\mathbf{x}^{0i}\ \mathbf{y}^{0i})$ and $(\mathbf{x}^{1i}\ \mathbf{y}^{1i})$, both belonging to S_n, with the following properties: (i) $x_h^{0i} = x_h^{1i} = x_h^i$ for all $h \neq k$; (ii) $x_k^{0i} = 0$, $x_k^{1i} = 1$; (iii) they both satisfy the linear inequality. Moreover, since the slack of the inequality at $(\mathbf{x}^i\ \mathbf{y}^i)$ is equal to the convex combination of the slacks of the inequality at the two points $(\mathbf{x}^{0i}\ \mathbf{y}^{0i})$ and $(\mathbf{x}^{1i}\ \mathbf{y}^{1i})$, the fact that $(\mathbf{x}^i\ \mathbf{y}^i)$ satisfies the inequality as an equality implies that the same is true for $(\mathbf{x}^{0i}\ \mathbf{y}^{0i})$ and $(\mathbf{x}^{1i}\ \mathbf{y}^{1i})$. Then, we redefine $(\mathbf{x}^i\ \mathbf{y}^i)$ as either $(\mathbf{x}^{0i}\ \mathbf{y}^{0i})$ or $(\mathbf{x}^{1i}\ \mathbf{y}^{1i})$. In this way, the number of fractional components different from x_i in $(\mathbf{x}^i\ \mathbf{y}^i)$ is decreased by one. By repeating this procedure until x_i is the unique fractional component in $(\mathbf{x}^i\ \mathbf{y}^i)$, we end up with the desired $(\mathbf{x}^i\ \mathbf{y}^i)$ point.

[3] To be more precise, taking into account the different dimensions of BQP_n and $chull(S_n)$, we can convert valid inequalities for BQP_n into valid inequalities for $chull(S_n)$ after adding the variables y_{ii}, $i = 1,\dots,n$, with coefficients all equal to 0.

$\Leftarrow$ By assumption the inequality induces a facet of BQP_n. Then, it contains $\frac{1}{2}(n^2+n)$ affinely independent binary extreme points of $chull(S_n)$. In order to have a facet of $chull(S_n)$ we need n more affinely independent points. It turns out that the n points $(\mathbf{x}^i\ \mathbf{y}^i)$, $i=1,\dots,n$, do the job. Indeed, each point $(\mathbf{x}^i\ \mathbf{y}^i)$ is the only point in the complete collection of $\frac{1}{2}(n^2+3n)$ points which does not satisfy the equation $y_{ii}=x_i$.

The equivalent statement of the theorem is easily proven by taking each one of the n extreme points $(\mathbf{x}^i\ \mathbf{y}^i)$ of $chull(S_n)$ detected above, rewriting it as the convex combination of two binary points $(\mathbf{x}^{0i}\ \mathbf{y}^{0i})$ and $(\mathbf{x}^{1i}\ \mathbf{y}^{1i})$ as done in the proof of Theorem 4.46, and finally projecting these two binary points onto BQP_n to end up with the two desired vertices of BQP_n $(\bar{\mathbf{x}}^i\ \bar{\mathbf{y}}^i)$ and $(\tilde{\mathbf{x}}^i\ \tilde{\mathbf{y}}^i)$. □

Boros and Hammer (Boros & Hammer, 1993) proved that for all $\mathbf{v}\in\mathbb{Z}^n$ and $s\in\mathbb{Z}$, all extreme points of BQP_n satisfy

$$(\mathbf{v}^T\mathbf{x}+s)(\mathbf{v}^T\mathbf{x}+s-1)\geq 0$$

(this is relatively easy to see, since the two integer factors differ by one, so that they are always either both nonnegative or both nonpositive for each binary point). Therefore, the inequalities

$$\sum_{i=1}^{n} v_i(v_i+2s-1)x_i+2\sum_{i=1}^{n}\sum_{j=i+1}^{n} v_iv_jy_{ij}\geq s(1-s) \tag{4.100}$$

are valid for BQP_n and, consequently, in view of Theorem 4.46, they are also valid for $chull(S_n)$. The cut-type inequalities are a special case of the inequalities (4.100) obtained when $\mathbf{v}\in\{-1,0,1\}^n$, which can be seen by taking

$$U=\{i\ :\ v_i=-1\},\ \ W=\{i\ :\ v_i=1\},\ \ \beta=s-1$$

in (4.99). In fact, exploiting Theorem 4.47, Burer and Letchford could prove a further strong result about cut-type inequalities.

Theorem 4.48. *Assume that an inequality* (4.100) *induces a facet of BQP_n. It induces a facet of $chull(S_n)$ if and only if $\mathbf{v}\in\{-1,0,1\}^n$, i.e., it is a cut-type inequality.*

Proof. $\Rightarrow$ For a given vertex of BQP_n equality in (4.100) holds if and only if $(\mathbf{v}^T\mathbf{x}+s)\in\{0,1\}$. Moreover, in view of the second part of Theorem 4.47, we have $2n$ extreme points of BQP_n, $(\bar{\mathbf{x}}^1\ \bar{\mathbf{y}}^1),\dots,(\bar{\mathbf{x}}^n\ \bar{\mathbf{y}}^n)$ and $(\tilde{\mathbf{x}}^1\ \tilde{\mathbf{y}}^1),\dots,(\tilde{\mathbf{x}}^n\ \tilde{\mathbf{y}}^n)$, with the properties described in the statement of the theorem. Then, one of the following must be true for each $i\in\{1,\dots,n\}$:

$$\mathbf{v}^T\bar{\mathbf{x}}^i=\mathbf{v}^T\tilde{\mathbf{x}}^i \quad\Rightarrow\quad v_i=0;$$

$$\mathbf{v}^T\bar{\mathbf{x}}^i+s=0,\ \ \mathbf{v}^T\tilde{\mathbf{x}}^i+s=1 \quad\Rightarrow\quad v_i=1;$$

$$\mathbf{v}^T\bar{\mathbf{x}}^i+s=1,\ \ \mathbf{v}^T\tilde{\mathbf{x}}^i+s=0 \quad\Rightarrow\quad v_i=-1.$$

Therefore, $v_i\in\{-1,0,1\}$ for all $i=1,\dots,n$.

$\Leftarrow$ For some given $\mathbf{v},s$ such that $v_i\in\{-1,0,1\}$ for all $i=1,\dots,n$, one can relatively easily build the $2n$ extreme points of BQP_n required by Theorem 4.47 by a constructive procedure, which we do not detail but just illustrate below through an example. □

Example 4.49. Let $n = 3$, $v_1 = 0$, $v_2 = -1$, $v_3 = 1$, $s = 0$ so that the inequality is $2x_2 - 2y_{23} \geq 0$. Consider the vertex (1 1 1) which satisfies the inequality as an equality. Then, one can take

$$(\bar{x}_1^1\ \bar{x}_2^1\ \bar{x}_3^1) = (1\ 1\ 1), \quad (\tilde{x}_1^1\ \tilde{x}_2^1\ \tilde{x}_3^1) = (0\ 1\ 1),$$

$$(\bar{x}_1^2\ \bar{x}_2^2\ \bar{x}_3^2) = (1\ 1\ 1), \quad (\tilde{x}_1^2\ \tilde{x}_2^2\ \tilde{x}_3^2) = (1\ 0\ 1),$$

while for the third pair of vertices one can take

$$(\bar{x}_1^3\ \bar{x}_2^3\ \bar{x}_3^3) = (1\ 0\ 1), \quad (\tilde{x}_1^3\ \tilde{x}_2^3\ \tilde{x}_3^3) = (1\ 0\ 0)$$

(note that the $2n = 6$ vertices are not distinct). ■

Convex, concave, indefinite: A classification of valid inequalities Burer and Letchford introduced a classification of valid inequalities for $chull(S_n)$. Each valid inequality can be written as

$$\sum_{i=1}^{n} \alpha_i x_i + \sum_{1 \leq i \leq j \leq n} \beta_{ij} y_{ij} \leq \gamma. \tag{4.101}$$

Let us define the quadratic function

$$q_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i x_i + \sum_{1 \leq i \leq j \leq n} \beta_{ij} x_i x_j.$$

Based on this function, valid inequalities are classified into the three categories "concave," "convex," or "indefinite," respectively, if $q_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ is a concave, convex, or indefinite quadratic function. For what concerns the concave inequalities, it turns out that the only nonredundant ones are the psd inequalities. In particular, it is possible to prove the following theorem.

Theorem 4.50. *If* (4.101) *is a concave inequality, then it is valid for the set*

$$\left\{ (\mathbf{x}\ \mathbf{y}) \in [0,1]^n \times \mathbb{R}^{\frac{1}{2}(n^2+n)} \ : \ \begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \mathbf{Y} \end{pmatrix} \in \mathcal{P}_{n+1} \right\}, \tag{4.102}$$

where $Y_{ij} = y_{ij}$ for all $i, j \in \{1, \ldots, n\}$.

Proof. Let $(\mathbf{x}\ \mathbf{y})$ be a point within the set (4.102). By assumption, we have that

$$q_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\mathbf{x}) = \boldsymbol{\alpha}^T \mathbf{x} + \mathbf{x}^T \mathbf{B} \mathbf{x},$$

for some symmetric matrix $\mathbf{B}$ such that $-\mathbf{B} \in \mathcal{P}_n$. We can write the left-hand side of (4.101) as

$$\boldsymbol{\alpha}^T \mathbf{x} + \mathbf{B} \bullet \mathbf{Y}.$$

Therefore, recalling that $\mathbf{x}^T \mathbf{B} \mathbf{x} = \mathbf{B} \bullet \mathbf{x}\mathbf{x}^T$, and remarking that

$$-\mathbf{B}, (\mathbf{Y} - \mathbf{x}\mathbf{x}^T) \in \mathcal{P}_n \ \Rightarrow \ \mathbf{B} \bullet (\mathbf{Y} - \mathbf{x}\mathbf{x}^T) \leq 0$$

(recall that $(\mathbf{Y}-\mathbf{x}\mathbf{x}^T) \in \mathcal{P}_n$ follows from Observation A.1), we have that

$$\begin{aligned}
&\boldsymbol{\alpha}^T\mathbf{x}+\mathbf{B}\bullet\mathbf{Y}\\
&=\boldsymbol{\alpha}^T\mathbf{x}+\mathbf{B}\bullet(\mathbf{Y}-\mathbf{x}\mathbf{x}^T)+\mathbf{x}^T\mathbf{B}\mathbf{x}\\
&\leq \boldsymbol{\alpha}^T\mathbf{x}+\mathbf{x}^T\mathbf{B}\mathbf{x}=q_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\mathbf{x}).
\end{aligned}$$

Since the maximum value over $chull(S_n)$ of the left-hand side of (4.101) is equal to the maximum value of $q_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\mathbf{x})$ over $[0,1]^n$, it turns out that the validity of (4.101) for $chull(S_n)$ implies $q_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\mathbf{x}) \leq \gamma$ over $[0,1]^n$ and, thus, the result is proven. □

About convex inequalities, the following theorem is proven.

Theorem 4.51. *If* (4.101) *is a convex inequality, then it is valid for the polytope made up by all points* $(\mathbf{x}\ \mathbf{y})$ *whose projection onto* $\mathbb{R}^{\frac{1}{2}(n^2+n)}$ *(simply obtained by dropping all components* y_{ii}, $i=1,\ldots,n$*) belongs to* BQP_n*, and, moreover, the points* $(\mathbf{x}\ \mathbf{y})$ *satisfy the RLT inequalities* $y_{ii} \leq x_i$, $i=1,\ldots,n$.

Proof. Convexity of $q_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ implies that its maximum over $[0,1]^n$ is attained at a binary point. The validity of (4.101) for $chull(S_n)$ implies that the maximum is not larger than γ. Therefore,

$$\begin{aligned}
\gamma\ &\geq \max\nolimits_{\mathbf{x}\in\{0,1\}^n} \textstyle\sum_{i=1}^n \alpha_i x_i+\sum_{i\leq j}\beta_{ij}x_i x_j\\
&=\max\nolimits_{\mathbf{x}\in\{0,1\}^n} \textstyle\sum_{i=1}^n(\alpha_i+\beta_{ii})x_i+\sum_{i<j}\beta_{ij}x_i x_j.
\end{aligned}$$

Then, the inequality

$$\sum_{i=1}^n(\alpha_i+\beta_{ii})x_i+\sum_{i<j}\beta_{ij}y_{ij}\leq\gamma \tag{4.103}$$

is valid for BQP_n. Now, let us consider a point $(\mathbf{x}\ \mathbf{y})$ whose projection onto $\mathbb{R}^{\frac{1}{2}(n^2+n)}$ belongs to BQP_n and such that $y_{ii} \leq x_i$ for all $i \in \{1,\ldots,n\}$. We would like to prove that (4.101) is true for all such points. Noticing that convexity of $q_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ implies $\beta_{ii} \geq 0$ for all $i \in \{1,\ldots,n\}$, we have that

$$\begin{aligned}
&\textstyle\sum_{i=1}^n\alpha_i x_i+\sum_{i\leq j}\beta_{ij}y_{ij}\\
&=\textstyle\sum_{i=1}^n\alpha_i x_i+\sum_{i<j}\beta_{ij}y_{ij}+\sum_{i=1}^n\beta_{ii}(x_i-x_i+y_{ii})\\
&\leq\textstyle\sum_{i=1}^n\alpha_i x_i+\sum_{i<j}\beta_{ij}y_{ij}+\sum_{i=1}^n\beta_{ii}x_i\\
&=\textstyle\sum_{i=1}^n(\alpha_i+\beta_{ii})x_i+\sum_{i<j}\beta_{ij}y_{ij}.
\end{aligned}$$

Then, the validity of (4.103) for BQP_n concludes the proof. □

For the indefinite case, in (Burer & Letchford, 2009) a result is presented which allows us to check the validity of the inequality by checking the validity of $2n$ inequalities for $chull(S_{n-1})$. The idea is that of fixing each variable x_k in turn to 0 and 1, thus ending up with the following $2n$ inequalities for all $k \in \{1,\ldots,n\}$, $\delta \in \{0,1\}$:

$$\sum_{i=1,\, i\neq k}^{n} (\alpha_i + \delta\beta_{ik})x_i + \sum_{1\leq i\leq j\leq n,\, i,j\neq k} \beta_{ij}y_{ij} \leq \gamma - (\alpha_k + \beta_{kk})\delta.$$

These turn out to be valid for $chull(S_{n-1})$ if and only if the original indefinite inequality is valid for $chull(S_n)$. In (Burer & Letchford, 2009) it has been observed that indefinite valid inequalities are redundant for $chull(S_2)$ (see also (Anstreicher & Burer, 2010)) but get nonredundant already for $chull(S_3)$.

McCormick under- and overestimators If we consider the quadratic function

$$f(\mathbf{x}) = \sum_{(i,j)\in E} Q_{ij}x_ix_j,$$

where E is a set of pairs of *different* indices belonging to $\{1,\ldots,n\}$, the convex underestimator and the concave overestimator for this function over the unit box $X = [\mathbf{0},\mathbf{e}]$ obtained by introducing the RLT inequalities are also called McCormick under- and overestimators (see also Section 4.12) and are, respectively, equal to

$$f_\ell^{MC}(\mathbf{x}) = \min\left\{\sum_{(i,j)\in E} Q_{ij}y_{ij},\ y_{ij} \geq x_i + x_j - 1,\ y_{ij} \leq x_i,\ y_{ij} \leq x_j,\ y_{ij} \geq 0\ \forall\,(i,j)\in E\right\},$$

$$f_u^{MC}(\mathbf{x}) = \max\left\{\sum_{(i,j)\in E} Q_{ij}y_{ij},\ y_{ij} \geq x_i + x_j - 1,\ y_{ij} \leq x_i,\ y_{ij} \leq x_j,\ y_{ij} \geq 0\ \forall\,(i,j)\in E\right\}.$$

An interesting analysis has been carried out in (Luedtke, Namazifar, & Linderoth, 2010) to compare the quality of these under- and overestimators with that of the convex and concave envelope of the function f over the unit hypercube. Consider the graph G, whose vertex set is $\{1,\ldots,n\}$ and whose edge set is E. Assume that G has a coloring of size ξ. Then, in (Luedtke et al., 2010) the following result has been proven.

Theorem 4.52. *If $Q_{ij} > 0$ for all $(i,j)\in E$, then, if ξ is even,*

$$f_u^{MC}(\mathbf{x}) - f_\ell^{MC}(\mathbf{x}) \leq \left(2 - \frac{2}{\xi}\right)(conc_{f,X}(\mathbf{x}) - conv_{f,X}(\mathbf{x}))\quad \forall\,\mathbf{x}\in X.$$

If ξ is odd,

$$f_u^{MC}(\mathbf{x}) - f_\ell^{MC}(\mathbf{x}) \leq \left(2 - \frac{2}{(\xi+1)}\right)(conc_{f,X}(\mathbf{x}) - conv_{f,X}(\mathbf{x}))\quad \forall\,\mathbf{x}\in X.$$

If the coefficients Q_{ij} might also be negative, a less strong result has been proven. In particular, if ξ is even, then

$$f_u^{MC}(\mathbf{x}) - f_\ell^{MC}(\mathbf{x}) \leq 2(\xi - 1)(conc_{f,X}(\mathbf{x}) - conv_{f,X}(\mathbf{x}))\quad \forall\,\mathbf{x}\in X,$$

while if ξ is odd, then

$$f_u^{MC}(\mathbf{x}) - f_\ell^{MC}(\mathbf{x}) \leq 2\xi(conc_{f,X}(\mathbf{x}) - conv_{f,X}(\mathbf{x})) \quad \forall\, \mathbf{x} \in X.$$

Since the gap between McCormick's underestimator and the convex envelope over the unit box might be quite large, use of the latter might be convenient. Basically, instead of considering a term-by-term relaxation as done by McCormick's underestimator, it might be more convenient to consider the multiterm relaxation corresponding to the convex envelope over the box. Such an envelope is polyhedral, i.e., it is defined by a finite number of facets/ affine functions. However, the number of these facets might be very high and it would be unfeasible to generate all of them at once. Therefore, in (Bao, Sahinidis, & Tawarmalani, 2009) a cutting plane strategy is proposed where facets are generated one by one and not all of them need to be generated to get to a bound with the same quality as the one obtained by including all the facets at the same time. We illustrate the difference between the bound obtained by the multiterm relaxation and the bound obtained through McCormick's underestimator through an example taken from (Bao et al., 2009).

Example 4.53. Let

$$f(x_1,x_2,x_3) = x_1x_2 + x_1x_3 + x_2x_3, \quad x_1,x_2,x_3 \in [0,1].$$

If we underestimate each single term separately, we end up with the underestimator

$$\max\{0, x_1+x_2-1\} + \max\{0, x_1+x_3-1\} + \max\{0, x_2+x_3-1\},$$

while with the multiterm relaxation we end up with

$$\max\{0, x_1+x_2+x_3-1, 2(x_1+x_2+x_3)-3\},$$

which is better than the single term relaxation, e.g., at point $\left(\frac{1}{2}\,\frac{1}{2}\,\frac{1}{2}\right)$. ∎

4.4.4 Quadratically constrained problems

In this section we briefly discuss *quadratically constrained quadratic programming* (QCQP) problems, in which a quadratic function is minimized over a feasible region defined (also) by quadratic constraints

$$\begin{aligned} \min \quad & q_0(\mathbf{x}) := \mathbf{x}^T\mathbf{Q}_0\mathbf{x} + \mathbf{c}_0^T\mathbf{x} \\ & q_i(\mathbf{x}) := \mathbf{x}^T\mathbf{Q}_i\mathbf{x} + \mathbf{c}_i^T\mathbf{x} + d_i \leq 0, \quad i \in I, \\ & q_i(\mathbf{x}) := \mathbf{x}^T\mathbf{Q}_i\mathbf{x} + \mathbf{c}_i^T\mathbf{x} + d_i = 0, \quad i \in E, \\ & \mathbf{x} \in P, \end{aligned} \tag{4.104}$$

where

$$P = \{\mathbf{x} \in \mathbb{R}^n \ : \ \mathbf{A}\mathbf{x} \leq \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\}$$

is a polytope with $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^m$. We denote by $S \subseteq P$ the feasible region of the problem. BB approaches (see Chapter 5) for QCQP based on linear relaxations have been proposed, e.g., in (Audet et al., 2000; Linderoth, 2005; Raber, 1998). Recently, a lot of attention has been dedicated to semidefinite relaxations for this problem. The most basic semidefinite relaxation for QCQP is Shor's relaxation

$$\begin{array}{ll} \min & \mathbf{Q}_0 \bullet \mathbf{X} + \mathbf{c}_0^T \mathbf{x} \\ & \mathbf{Q}_i \bullet \mathbf{X} + \mathbf{c}_i^T \mathbf{x} + d_i \leq 0, \quad i \in I, \\ & \mathbf{Q}_i \bullet \mathbf{X} + \mathbf{c}_i^T \mathbf{x} + d_i = 0, \quad i \in E, \\ & \mathbf{x} \in P, \\ & \mathbf{Y}_{\mathbf{X},\mathbf{x}} \in \mathcal{P}_{n+1}, \end{array} \tag{4.105}$$

where $\mathbf{Y}_{\mathbf{X},\mathbf{x}}$ is derived from (4.87). If P contains the box constraints $\mathbf{x} \in [0,1]^n$, in (Anstreicher, 2009) the RLT constraints

$$X_{ij} \leq \min\{x_i, x_j\}, \;\; X_{ij} \geq \max\{0, x_i + x_j - 1\}, \;\; i, j = 1, \ldots, n, \tag{4.106}$$

are introduced to strengthen the relaxation (4.105). An interesting theoretical result has been proven by Anstreicher (Anstreicher, 2012). Let $E = \emptyset$. A possible convex relaxation for (4.104) is obtained by substituting each quadratic function in the left-hand side of the quadratic constraints as well as the objective function with its convex envelope over P, i.e., each function q_i is replaced by $conv_{q_i,P}$:

$$\begin{array}{ll} \min & conv_{q_0,P}(\mathbf{x}) \\ & conv_{q_i,P}(\mathbf{x}) \leq 0, \quad i \in I, \\ & \mathbf{x} \in P. \end{array} \tag{4.107}$$

Another possible convex relaxation is obtained by first introducing the convex hull

$$W = chull \left\{ \begin{pmatrix} 1 \\ \mathbf{x} \end{pmatrix} \begin{pmatrix} 1 \\ \mathbf{x} \end{pmatrix}^T \; : \; \mathbf{x} \in P \right\},$$

and then by solving the problem

$$\begin{array}{ll} \min & \mathbf{Q}_0 \bullet \mathbf{X} + \mathbf{c}_0^T \mathbf{x} \\ & \mathbf{Q}_i \bullet \mathbf{X} + \mathbf{c}_i^T \mathbf{x} + d_i \leq 0, \quad i \in I, \\ & \mathbf{x} \in P, \\ & \mathbf{Y}_{\mathbf{X},\mathbf{x}} \in W. \end{array} \tag{4.108}$$

It turns out that the problem (4.107) is equivalent to

$$\begin{array}{lll} \min & \mathbf{Q}_0 \bullet \mathbf{X}_0 + \mathbf{c}_0^T \mathbf{x} & \\ & \mathbf{Q}_i \bullet \mathbf{X}_i + \mathbf{c}_i^T \mathbf{x} + d_i \leq 0, \quad i \in I, & \\ & \mathbf{x} \in P, & \\ & \mathbf{Y}_{\mathbf{X}_i,\mathbf{x}} \in W, & i \in \{0\} \cup I. \end{array}$$

Therefore, problem (4.108) is equal to the problem above with the additional constraints $\mathbf{X}_i = \mathbf{X}_j$ for all $i, j \in \{0\} \cup I$, which allows us to conclude that the bound returned by the problem (4.108) is always at least as good as (i.e., as large as) the bound returned by (4.107). Unfortunately, the two relaxations (4.107) and (4.108) share a common drawback: both the problem of computing the convex envelopes of the quadratic functions over P, and that of characterizing the convex hull W are intractable. However, for what concerns the characterization of W, we have that this can be represented through the cone of completely positive matrices. In particular, if we denote

$$\mathbf{Y}'_{\mathbf{X},\mathbf{x}} = \begin{pmatrix} 1 & \mathbf{x}^T & \mathbf{s}_{\mathbf{x}}^T \\ \mathbf{x} & \mathbf{X} & \mathbf{Z}_{\mathbf{X},\mathbf{x}} \\ \mathbf{s}_{\mathbf{x}} & \mathbf{Z}_{\mathbf{X},\mathbf{x}}^T & \mathbf{S}_{\mathbf{X},\mathbf{x}} \end{pmatrix},$$

where

$$\begin{aligned} &\mathbf{s}_{\mathbf{x}} = \mathbf{b} - \mathbf{A}\mathbf{x}, \\ &\mathbf{S}_{\mathbf{X},\mathbf{x}} = \mathbf{b}\mathbf{b}^T - \mathbf{A}\mathbf{x}\mathbf{b}^T - \mathbf{b}\mathbf{x}^T\mathbf{A}^T + \mathbf{A}\mathbf{X}\mathbf{A}^T, \\ &\mathbf{Z}_{\mathbf{X},\mathbf{x}} = \mathbf{x}\mathbf{b}^T - \mathbf{X}\mathbf{A}^T, \end{aligned}$$

it is proven in (Burer, 2009) that

$$W = \{\mathbf{Y}_{\mathbf{X},\mathbf{x}} \,:\, \mathbf{Y}'_{\mathbf{X},\mathbf{x}} \in \mathcal{C}^*_{m+n+1}\}.$$

We can obtain a computable relaxation if we substitute the condition $\mathbf{Y}_{\mathbf{X},\mathbf{x}} \in W$ with some valid constraints for W. In particular, in (Anstreicher, 2012) when $P = [0,1]^n$, a relaxation is suggested where $\mathbf{Y}_{\mathbf{X},\mathbf{x}} \in W$ is substituted with the semidefinite condition $\mathbf{Y}_{\mathbf{X},\mathbf{x}} \in \mathcal{P}_{n+1}$ and with the RLT constraints $diag(\mathbf{X}) \leq \mathbf{x}$ (it is also proven that the bound obtained with such relaxation is always at least as good as the one obtained via the classical α-BB approach, which will be discussed in Section 4.6), while for a general polytope P a relaxation is obtained by substituting $\mathbf{Y}_{\mathbf{X},\mathbf{x}} \in W$ with $\mathbf{Y}'_{\mathbf{X},\mathbf{x}} \in \mathcal{DNN}_{m+n+1}$, i.e., we impose that the matrix $\mathbf{Y}'_{\mathbf{X},\mathbf{x}}$ is doubly nonnegative (see Definition A.20). It turns out that the resulting relaxation is equivalent to imposing the semidefinite condition $\mathbf{Y}_{\mathbf{X},\mathbf{x}} \in \mathcal{P}_{n+1}$ plus the RLT constraints which can be obtained from the linear constraints in P. In (Anstreicher, 2012) it is also shown that the bound returned by this relaxation is always at least as good as a bound based on difference-of-convex (DC) decompositions of the quadratic functions proposed in (Zheng, Sun, & Li, 2011b) (we refer the reader to Section 4.7 both for the definition of DC decompositions and for a further discussion about the bound presented in (Zheng et al., 2011b)).

In (Zheng, Sun, & Li, 2011a) a rather general framework, which allows us to include many existing semidefinite relaxations for QCQP, is proposed. The approach is based on a decomposition of the quadratic functions and on a piecewise linear underestimation of the "difficult" part of the decomposition. The approach is described in detail in what follows. First a matrix cone $\mathcal{U}$ is introduced with the following property: for each matrix $\mathbf{M} \in \mathcal{U}$

$$-\mathbf{x}^T\mathbf{M}\mathbf{x} \geq \max\{\boldsymbol{\alpha}_1\mathbf{x} + \beta_1, \boldsymbol{\alpha}_2\mathbf{x} + \beta_2\} \quad \forall \mathbf{x} \in S, \tag{4.109}$$

where $\boldsymbol{\alpha}_i, \beta_i$, $i = 1,2$, can be expressed through linear functions with respect to the entries of $\mathbf{M}$. Then, we consider the following decomposition of the objective function:

$$\mathbf{x}^T\mathbf{Q}_0\mathbf{x} + \mathbf{c}_0^T\mathbf{x} = \mathbf{x}^T(\mathbf{Q}_0 + \mathbf{M})\mathbf{x} + \mathbf{c}_0^T\mathbf{x} - \mathbf{x}^T\mathbf{M}\mathbf{x},$$

with $\mathbf{Q}_0 + \mathbf{M} \in \mathcal{P}_n$. For any fixed matrix $\mathbf{M} \in \mathcal{U}$ we have that the optimal value $r(\mathbf{M})$ of the following problem is a lower bound for the problem (4.104):

$$\begin{aligned} r(\mathbf{M}) = \min \quad & \mathbf{x}^T(\mathbf{Q}_0 + \mathbf{M})\mathbf{x} + \mathbf{c}_0^T\mathbf{x} + \max\{\boldsymbol{\alpha}_1\mathbf{x} + \beta_1, \boldsymbol{\alpha}_2\mathbf{x} + \beta_2\}, \\ & \mathbf{x} \in S. \end{aligned}$$

The best possible of these bounds is obtained by solving the following problem:

$$\begin{aligned} r^* = \sup \quad & r(\mathbf{M}), \\ & \mathbf{Q}_0 + \mathbf{M} \in \mathcal{P}_n, \\ & \mathbf{M} \in \mathcal{U}. \end{aligned}$$

In (Zheng et al., 2011a) it is proven that the conic program

$$\begin{aligned} c^* = \sup \quad & \tau \\ & D(\boldsymbol{\lambda}_I, \boldsymbol{\lambda}_E, \pi_1, \pi_2, \tau) \in \mathcal{P}_{n+1}, \\ & \boldsymbol{\lambda}_I \geq \mathbf{0}, \\ & \pi_1, \pi_2 \geq 0, \\ & \pi_1 + \pi_2 = 1, \\ & \mathbf{Q}_0 + \mathbf{M} \in \mathcal{P}_n, \\ & \mathbf{M} \in \mathcal{U}, \end{aligned} \tag{4.110}$$

where

$$D(\boldsymbol{\lambda}_I, \boldsymbol{\lambda}_E, \pi_1, \pi_2, \tau)$$
$$= \begin{pmatrix} \sum_{i \in I \cup E} \lambda_i d_i + \pi_1\beta_1 + \pi_2\beta_2 - \tau & \frac{1}{2}\left[\mathbf{b}_0 + \sum_{i \in I \cup E} \lambda_i \mathbf{b}_i + \pi_1\boldsymbol{\alpha}_1 + \pi_2\boldsymbol{\alpha}_2\right]^T \\ \frac{1}{2}\left[\mathbf{b}_0 + \sum_{i \in I \cup E} \lambda_i \mathbf{b}_i + \pi_1\boldsymbol{\alpha}_1 + \pi_2\boldsymbol{\alpha}_2\right] & \mathbf{Q}_0 + \mathbf{M} + \sum_{i \in I \cup E} \lambda_i \mathbf{Q}_i \end{pmatrix},$$

is such that $c^* \leq r^*$ and equality is true if the feasible set S is convex.

In order to derive possible relaxations, we need to introduce some cones of matrices for which the condition (4.109) is satisfied. In (Zheng et al., 2011a) such different cones are introduced. Some examples follow.

- For some fixed vectors $\mathbf{v}_i \in \mathbb{R}^n$, $i = 1, \ldots, s$, let $\mathbf{V}_{ij} = \frac{1}{2}(\mathbf{v}_i\mathbf{v}_j^T + \mathbf{v}_j\mathbf{v}_i^T)$, $i, j \in \{1, \ldots, s\}$. Consider the matrix cone

$$\mathcal{W}_1 = \left\{ \sum_{i,j=1}^{s} (p_{ij} + n_{ij})\mathbf{V}_{ij} \;:\; p_{ij} = p_{ji} \geq 0,\ n_{ij} = n_{ji} \leq 0,\ i, j = 1, \ldots, s \right\}.$$

Let l_i and u_i be, respectively, known as the lower and upper bounds for $\mathbf{v}_i^T\mathbf{x}$ over S, $i=1,\ldots,s$. Then, it is proven that for each

$$\mathbf{M}=\sum_{i,j=1}^{s}(p_{ij}+n_{ij})\mathbf{V}_{ij}\in\mathcal{W}_1,$$

the following is true for all $\mathbf{x}\in S$:

$$-\mathbf{x}^T\mathbf{M}\mathbf{x}\geq\sum_{i,j=1}^{s}[p_{ij}\max\{-\ell_{ij}^3(\mathbf{x}),-\ell_{ji}^3(\mathbf{x})\}-n_{ij}\max\{\ell_{ij}^1(\mathbf{x}),\ell_{ji}^2(\mathbf{x})\}],$$

where

$$\begin{aligned}\ell_{ij}^1(\mathbf{x})&=(u_i\mathbf{v}_j^T+u_j\mathbf{v}_i^T)\mathbf{x}-u_iu_j,\\ \ell_{ij}^2(\mathbf{x})&=(l_i\mathbf{v}_j^T+l_j\mathbf{v}_i^T)\mathbf{x}-l_il_j,\\ \ell_{ij}^3(\mathbf{x})&=(l_i\mathbf{v}_j^T+u_j\mathbf{v}_i^T)\mathbf{x}-l_iu_j.\end{aligned}$$

In particular, for $\mathbf{v}_i=\mathbf{e}_i$, $i=1,\ldots,n$, and $S\subseteq[0,1]^n$, it turns out that $\mathcal{W}_1$ is equivalent to the cone $\mathcal{N}_n+(-\mathcal{N}_n)$, the sum of the cones of nonnegative and nonpositive matrices, while, by taking $l_i=0$, $u_i=1$ for all i, we have

$$\ell_{ij}^1(\mathbf{x})=x_i+x_j-1,\quad \ell_{ij}^2(\mathbf{x})=0,\quad \ell_{ij}^3(\mathbf{x})=x_i.$$

- For each $i\in I$, if $rank(\mathbf{Q}_i)=r_i$, we have the spectral decomposition

$$\mathbf{Q}_i=\sum_{j=1}^{p_i}\mathbf{u}_{ij}\mathbf{u}_{ij}^T-\sum_{j=p_i+1}^{r_i}\mathbf{u}_{ij}\mathbf{u}_{ij}^T.$$

Let l_{ij} and u_{ij} be, respectively, a lower and an upper bound for $\mathbf{u}_{ij}^T\mathbf{x}$ over S, and define

$$\ell_{ij}(\mathbf{x})=-(l_{ij}+u_{ij})\mathbf{u}_{ij}^T\mathbf{x}+l_{ij}u_{ij}.$$

Finally, let

$$\mathcal{W}_2=\left\{\sum_{i\in I\cup E}\lambda_i\mathbf{Q}_i\ :\ \lambda_i\geq 0,\ i\in I\right\}.$$

Then, for any $\mathbf{M}\in\mathcal{W}_2$, we have that for all $\mathbf{x}\in S$

$$-\mathbf{x}^T\mathbf{M}\mathbf{x}\geq\sum_{i\in I}\lambda_i\max\left\{\mathbf{c}_i^T\mathbf{x}+d_i,\sum_{j=1}^{p_i}\ell_{ij}(\mathbf{x})\right\}+\sum_{i\in E}\lambda_i(\mathbf{c}_i^T\mathbf{x}+d_i).$$

- Let $\mathbf{w}>\mathbf{0}$, $\mathbf{w}\in\mathbb{R}^n$, and let $u_{\mathbf{w}}$ be a positive upper bound for $\mathbf{w}^T\mathbf{x}$ over S. It can be proven that for any $\mathbf{M}\in\mathcal{P}_n$ and any $\mathbf{x}\in S$

$$-\mathbf{x}^T\mathbf{M}\mathbf{x}\geq -u_{\mathbf{w}}diag(\mathbf{M})^T Diag(\mathbf{w})^{-1}\mathbf{x}.$$

Now, we consider the cone decomposition

$$\mathcal{W} = \mathcal{W}_1 + \mathcal{W}_2 + \mathcal{P}_n.$$

Then, if $\mathbf{M} \in \mathcal{W}$ we have

$$\mathbf{M} = \sum_{i,j=1}^{s} (p_{ij} + n_{ij})\mathbf{V}_{ij} + \sum_{i \in I \cup E} \lambda_i \mathbf{Q}_i + \mathbf{M}',$$

with

$$\begin{aligned} &\textstyle\sum_{i,j=1}^{s} (p_{ij} + n_{ij})\mathbf{V}_{ij} \in \mathcal{W}_1, \\ &\textstyle\sum_{i \in I \cup E} \lambda_i \mathbf{Q}_i \in \mathcal{W}_2, \\ &\mathbf{M}' \in \mathcal{P}_n. \end{aligned}$$

As in (Zheng et al., 2011a), we consider the simpler inequality

$$-\mathbf{x}^T \mathbf{M} \mathbf{x} \geq \sum_{i \in I \cup E} \lambda_i (\mathbf{c}_i^T \mathbf{x} + d_i)$$

for any matrix $\mathbf{M} \in \mathcal{W}_2$. Then, for any $\mathbf{M} \in \mathcal{W}$ and any $\mathbf{x} \in S$ we have the following:

$$\begin{aligned} -\mathbf{x}^T \mathbf{M} \mathbf{x} \geq &\textstyle\sum_{i \in I \cup E} \lambda_i (\mathbf{c}_i^T \mathbf{x} + d_i) + \sum_{i,j=1}^{s} p_{ij} \max\{-\ell_{ij}^3(\mathbf{x}), -\ell_{ji}^3(\mathbf{x})\} \\ &- \textstyle\sum_{i,j=1}^{s} n_{ij} \max\{\ell_{ij}^1(\mathbf{x}), \ell_{ji}^2(\mathbf{x})\} - u_{\mathbf{w}} diag(\mathbf{M}')^T Diag(\mathbf{w})^{-1} \mathbf{x}. \end{aligned}$$

Taking into account the above result, in (Zheng et al., 2011a) the conic problem (4.110) is specialized to the case $\mathcal{U} = \mathcal{W}$. It is shown that in this case the conic dual of (4.110) is the following problem:

$$\begin{array}{lll} \min & \mathbf{Q}_0 \bullet \mathbf{X} + \mathbf{c}_0^T \mathbf{x} & \\ & \mathbf{Q}_i \bullet \mathbf{X} + \mathbf{c}_i^T \mathbf{x} + d_i \leq 0, & i \in I, \\ & \mathbf{Q}_i \bullet \mathbf{X} + \mathbf{c}_i^T \mathbf{x} + d_i = 0, & i \in E, \\ & \mathbf{V}_{ij} \bullet \mathbf{X} \leq \ell_{ij}^3(\mathbf{x}), & i,j = 1,\ldots,s, \\ & \mathbf{V}_{ij} \bullet \mathbf{X} \leq \ell_{ji}^3(\mathbf{x}), & i,j = 1,\ldots,s, \\ & \mathbf{V}_{ij} \bullet \mathbf{X} \geq \ell_{ij}^1(\mathbf{x}), & i,j = 1,\ldots,s, \\ & \mathbf{V}_{ij} \bullet \mathbf{X} \geq \ell_{ij}^2(\mathbf{x}), & i,j = 1,\ldots,s, \\ & u_{\mathbf{w}} Diag(\mathbf{w})^{-1} Diag(\mathbf{x}) - \mathbf{X} \in \mathcal{P}_n, & \\ & \mathbf{x} \in P, & \\ & \mathbf{Y}_{\mathbf{X},\mathbf{x}} \in \mathcal{P}_{n+1}, & \end{array} \tag{4.111}$$

whose optimal value is a lower bound for the optimal value of the problem (4.104) and, by duality, we have that such optimal value is always at least as large as the optimal value of the conic problem (4.110) for $\mathcal{U} = \mathcal{W}$. Moreover, equality is true if one of the two problems is bounded and strictly feasible. We notice that the problem (4.111) adds some constraints to the basic Shor's relaxation (4.105). When $S \subseteq [0,1]^n$, as previously mentioned, we can choose $\mathbf{v}_i = \mathbf{e}_i$, $i = 1,\ldots,n$, so that the constraints involving the products $\mathbf{V}_{ij} \bullet \mathbf{X}$ reduce exactly to (4.106), which proves that under these choices for the vectors $\mathbf{v}_i$, the lower bound for QCQP returned by (4.111) is always at least as good as the lower bound returned by the relaxation obtained with the addition of the RLT constraints (4.106). Further matrix cones based on convex quadratic and/or linear inequalities $q_i(\mathbf{x}) \leq 0$, $i \in I$, are introduced in (Zheng et al., 2011a). By including also these cones in the matrix decomposition, it is possible to strengthen (4.111) with additional constraints (including a second-order cone inequality also proposed in (Burer & Saxena, 2009)). We refer to (Zheng et al., 2011a) for details about this further development.

We finally remark that many semidefinite relaxations are discussed in (Bao, Sahinidis, & Tawarmalani, 2011), where the quality of the corresponding bounds is also compared. Such relaxations also refer to reformulations of QCQP including the KKT conditions, as seen in Section 4.4.2 for QP and BoxQP problems.

4.5 Polynomial programming

Recently *polynomial programming* (PP) problems have received a great deal of attention. These are defined as

$$\begin{aligned} \inf \quad & f(\mathbf{x}) \\ & p_j(\mathbf{x}) = 0, \quad j = 1,\ldots,m, \\ & p_j(\mathbf{x}) \geq 0, \quad i = m+1,\ldots,m+s, \end{aligned} \tag{4.112}$$

where $f, p_j \in \mathbb{R}[\mathbf{x}]$, $j = 1,\ldots,m+s$ ($\mathbb{R}[\mathbf{x}]$ denotes the ring of all polynomials in n variables $\mathbf{x} = (x_1 \ldots x_n)$ with real coefficients, while $\mathbb{R}[\mathbf{x}]_d$ will denote the set of polynomials of degree not larger than d). The RLT technique discussed in Section 4.3 can be used to define relaxations of these problems. However, many other techniques have been proposed to derive bounds for such problems, in particular based on the solution of semidefinite problems. A detailed presentation of all these techniques is beyond the scope of this book. We will give an overview about unconstrained PP (UPP) problems, while for more details about PP problems we refer to the extensive survey by Laurent (Laurent, 2009) for which updates are also available at the web site `http://homepages.cwi.nl/~monique/`. UPP problems are defined as

$$\begin{aligned} \inf \quad & f(\mathbf{x}) \\ & \mathbf{x} \in \mathbb{R}^n, \end{aligned} \tag{4.113}$$

where $f \in \mathbb{R}[\mathbf{x}]$. We will denote by f_* the infimum value of the problem ($f_* = -\infty$ might occur). Before discussing lower bounding techniques for the UPP problem, we need to introduce some notation and definitions.

4.5.1 Notation and definitions

For some polynomial function $f \in \mathbb{R}[\mathbf{x}]_d$ defined as

$$f(x_1,\dots,x_n) = \sum_{\alpha_1,\dots,\alpha_n \,:\, \sum_{j=1}^n \alpha_j \le d} f_{\alpha_1,\dots,\alpha_n} x_1^{\alpha_1} \cdots x_n^{\alpha_n},$$

we denote by

- $\mathbf{x} = (x_1 \dots x_n)$ the vector of variables;
- $\boldsymbol{\alpha} = (\alpha_1 \dots \alpha_n)$ the vector of exponents of a monomial;
- $\mathbf{x}^{\boldsymbol{\alpha}}$ the monomial $x_1^{\alpha_1} \cdots x_n^{\alpha_n}$;
- $f_{\boldsymbol{\alpha}}$ its corresponding coefficient $f_{\alpha_1,\dots,\alpha_n}$.

Then,

$$f(\mathbf{x}) = \sum_{\boldsymbol{\alpha} \,:\, \sum_{j=1}^n \alpha_j \le d} f_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}}.$$

Now, let

$$\tau(n,d) = \binom{n+d}{d}.$$

We note that the $\tau(n,d)$-dimensional vector $[\mathbf{x}]_d = (\mathbf{x}^{\boldsymbol{\alpha}})_{\boldsymbol{\alpha} \,:\, \sum_{i=1}^n \alpha_i \le d}$ forms a basis for $\mathbb{R}[\mathbf{x}]_d$. We also denote by $\mathbf{f}_d$ the $\tau(n,d)$-dimensional vector whose components are the coefficients $f_{\boldsymbol{\alpha}}$. Note that $f(\mathbf{x}) = \mathbf{f}_d^T [\mathbf{x}]_d$.

Definition 4.54. *A polynomial f such that $f_* \ge 0$ is called nonnegative. We denote by $\mathcal{Q}_n, \mathcal{Q}_{n,d}$ the sets of* nonnegative polynomials *of n variables, respectively, of any degree and of degree not larger than d. A polynomial f of degree $2d$ is said to be a* sum-of-squares *(SOS) polynomial if there exists a finite number t of polynomials $p_j \in \mathbb{R}[\mathbf{x}]_d$ such that*

$$f(\mathbf{x}) = \sum_{j=1}^{t} [p_j(\mathbf{x})]^2.$$

We denote by $\Sigma_n, \Sigma_{n,d}$ the sets of SOS polynomials of n variables, respectively, of any degree and of degree not larger than d.

Obviously $\Sigma_n \subseteq \mathcal{Q}_n$ and $\Sigma_{n,d} \subseteq \mathcal{Q}_{n,d}$. Hilbert (Hilbert, 1888) proved the following theorem.

Theorem 4.55. *The equality $\Sigma_{n,d} = \mathcal{Q}_{n,d}$ is satisfied if*

- *$n = 1$;*
- *$d = 2$;*
- *$n = 2$, $d = 4$.*

An example of a nonnegative polynomial which is not SOS is the Motzkin polynomial (with $n = 3$ and $d = 6$)

$$f(x_1,x_2,x_3) = x_1^4x_2^2 + x_1^2x_2^4 - 3x_1^2x_2^2x_3^2 + x_3^6. \tag{4.114}$$

Its infimum value is attained at $x_1 = x_2 = x_3 = 1$ and is equal to 0, so that it is nonnegative, but it is not SOS.

Definition 4.56. *A set $\mathcal{I} \subset \mathbb{R}[\mathbf{x}]$ is an* ideal *if*

$$p \in \mathcal{I},\ \ q \in \mathbb{R}[\mathbf{x}] \ \ \Rightarrow\ \ pq \in \mathcal{I}.$$

Given m polynomials $p_1,\ldots,p_m \in \mathbb{R}[\mathbf{x}]$, we denote by $\langle p_1,\ldots,p_m\rangle$ the smallest ideal containing these m polynomials. It turns out that such an ideal is the set of all polynomials which can be obtained by a linear combination of $p_1,\ldots,p_m$ whose coefficients are polynomials in $\mathbb{R}[\mathbf{x}]$, i.e.,

$$\begin{aligned}\langle p_1,\ldots,p_m\rangle = \{h \in \mathbb{R}[\mathbf{x}] \ :\ \ & h(\mathbf{x}) = h_1(\mathbf{x})p_1(\mathbf{x}) + \cdots + h_m(\mathbf{x})p_m(\mathbf{x}),\\ & h_1,\ldots,h_m \in \mathbb{R}[\mathbf{x}]\}.\end{aligned}$$

Moreover, every ideal $\mathcal{I}$ has a finite generating set, i.e., there exists $p_1,\ldots,p_m$ such that $\mathcal{I} = \langle p_1,\ldots,p_m\rangle$.

Definition 4.57. *For a given ideal $\mathcal{I}$, we denote by $\mathbb{R}[\mathbf{x}]/\mathcal{I}$ the* quotient ring*, whose elements are the equivalence classes defined by the following congruence relation $\tilde{R}_\mathcal{I}$ defined over $\mathbb{R}[\mathbf{x}]$:*

$$p\ \tilde{R}_\mathcal{I}\ q \quad \Leftrightarrow \quad p - q \in \mathcal{I}.$$

Definition 4.58. *Given an ideal $\mathcal{I}$, we define the* variety *of $\mathcal{I}$, denoted by $V(\mathcal{I})$, as the set of all common complex zeros of the polynomials in $\mathcal{I}$, i.e.,*

$$V(\mathcal{I}) = \{\mathbf{x} \in \mathbb{C}^n \ :\ p(\mathbf{x}) = 0 \ \ \forall\ p \in \mathcal{I}\}.$$

If $\mathcal{I} = \langle p_1,\ldots,p_m\rangle$, then we have

$$V(\mathcal{I}) = \{\mathbf{x} \in \mathbb{C}^n \ :\ p_i(\mathbf{x}) = 0 \ \ \ i = 1,\ldots,m\}.$$

If $V(\mathcal{I})$ is a finite set, then the ideal $\mathcal{I}$ is said to be *zero-dimensional*. The subset of real points in $V(\mathcal{I})$ is denoted by $V^{\mathbb{R}}(\mathcal{I})$ and is called *real variety* of $\mathcal{I}$. The following observation can be proven.

Observation 4.2. *An ideal $\mathcal{I}$ is zero-dimensional if and only if the quotient ring $\mathbb{R}[\mathbf{x}]/\mathcal{I}$ is a finite-dimensional $\mathbb{R}$-vector space.*

Definition 4.59. *Given an ideal $\mathcal{I}$, its* radical *is the ideal*

$$\sqrt{\mathcal{I}} = \{p \in \mathbb{R}[\mathbf{x}] \ :\ p^t \in \mathcal{I} \textit{ for some } \ t \in \mathbb{N}\}.$$

Obviously $\mathcal{I} \subseteq \sqrt{\mathcal{I}}$, and if we have an equality, then $\mathcal{I}$ is called a radical ideal.

Definition 4.60. *The* gradient ideal *with respect to a polynomial f is denoted by $\mathcal{I}^f_{grad}$ and is defined as the ideal generated by the partial derivatives of f, i.e.,*

$$\mathcal{I}^f_{grad} = \left\langle \frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right\rangle.$$

4.5.2 The SOS relaxation

The following theorem states the important result that an SOS polynomial can be recognized by a semidefinite problem.

Theorem 4.61. *A polynomial $f \in \mathbb{R}[\mathbf{x}]_{2d}$ is an SOS polynomial if and only if the following semidefinite problem has a feasible solution:*

$$\begin{aligned} &\mathbf{X} \in \mathcal{P}_{\tau(n,d)}, \\ &\textstyle\sum_{\boldsymbol{\alpha},\boldsymbol{\beta}\in\mathbb{N}^n_d \,:\, \boldsymbol{\alpha}+\boldsymbol{\beta}=\boldsymbol{\gamma}} \mathbf{X}_{\boldsymbol{\alpha},\boldsymbol{\beta}} = f_{\boldsymbol{\gamma}}, \quad \boldsymbol{\gamma} \in \mathbb{N}^n_{2d}, \end{aligned} \tag{4.115}$$

where

$$\mathbb{N}^n_{2d} = \left\{ \boldsymbol{\alpha} \in \mathbb{N}^n \;:\; \sum_{i=1}^n \alpha_i \le 2d \right\}.$$

Proof. Let us assume that $f \in \Sigma_{n,2d}$. Then, by definition

$$f(\mathbf{x}) = \sum_{i=1}^t [p^i(\mathbf{x})]^2,$$

for some polynomials $p^i \in \mathbb{R}[\mathbf{x}]_d$. This can also be written as

$$f(\mathbf{x}) = \sum_{i=1}^t [\mathbf{x}]_d^T \{\mathbf{p}^i_d(\mathbf{p}^i_d)^T\} [\mathbf{x}]_d.$$

Now let

$$\mathbf{Q} = \sum_{i=1}^t \mathbf{p}^i_d(\mathbf{p}^i_d)^T$$

be a $\tau(n,d)$-dimensional square matrix, obtained as a finite sum of the rank-1 positive semidefinite matrices $\mathbf{p}^i_d(\mathbf{p}^i_d)^T$, $i = 1,\dots,t$. The matrix $\mathbf{Q}$ is positive semidefinite. In fact, all positive semidefinite matrices can be written as a finite sum of rank-1 positive semidefinite matrices (the so-called Gram decomposition). The matrix $\mathbf{Q}$ is indexed over $\mathbb{N}^n_d$ and its $(\boldsymbol{\alpha},\boldsymbol{\beta})$-entry is denoted by $Q_{\boldsymbol{\alpha},\boldsymbol{\beta}}$. Then, we can also write

$$f(\mathbf{x}) = \sum_{\boldsymbol{\alpha},\boldsymbol{\beta}\in\mathbb{N}^n_d} \mathbf{x}^{\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\beta}} Q_{\boldsymbol{\alpha},\boldsymbol{\beta}} = \sum_{\boldsymbol{\gamma}\in\mathbb{N}^n_{2d}} \mathbf{x}^{\boldsymbol{\gamma}} \left\{ \sum_{\boldsymbol{\alpha},\boldsymbol{\beta}\in\mathbb{N}^n_d \,:\, \boldsymbol{\alpha}+\boldsymbol{\beta}=\boldsymbol{\gamma}} Q_{\boldsymbol{\alpha},\boldsymbol{\beta}} \right\}.$$

Then, we can conclude that $\mathbf{Q}$ is a feasible solution for (4.115).

The converse is also true. Given a feasible solution $\mathbf{Q}$ for (4.115), its Gram decomposition immediately delivers an SOS representation for f. □

A very simple application of the above result is given in the following example.

Example 4.62. It is well known that the one-dimensional polynomial (quadratic) function $x^2 + bx + c$ is nonnegative (equivalent to SOS since $n = 1$; see Theorem 4.55) if and only if $b^2 \leq 4c$. We would like to derive this result from Theorem 4.61. We need to check when the following semidefinite problem is feasible:

$$\begin{aligned} &\begin{pmatrix} X_{00} & X_{01} \\ X_{01} & X_{11} \end{pmatrix} \in \mathcal{P}_2, \\ &X_{00} = 1, \\ &2X_{01} = b, \\ &X_{11} = c, \end{aligned}$$

from which we derive immediately the condition $b^2 \leq 4c$. ■

The UPP problem for some $f \in \mathbb{R}[\mathbf{x}]_{2d}$ can be rewritten as

$$\begin{aligned} \sup \quad & \alpha \\ & f(\mathbf{x}) - \alpha \in \mathcal{Q}_{n,2d}. \end{aligned}$$

In view of $\Sigma_{n,2d} \subseteq \mathcal{Q}_{n,2d}$ the problem

$$\begin{aligned} \sup \quad & \alpha \\ & f(\mathbf{x}) - \alpha \in \Sigma_{n,2d} \end{aligned} \tag{4.116}$$

delivers a lower bound for UPP. Problem (4.116) is called the *SOS relaxation* of the UPP problem. In view of Theorem 4.61, the SOS relaxation can be solved as a semidefinite problem with dimension $\tau(n,d)$. While in some cases (e.g., those presented in Theorem 4.55), the SOS relaxation delivers the optimal value of the UPP problem, this can not always be guaranteed. For instance, if we consider the Motzkin polynomial (4.114), it can be seen that the optimal value of the UPP is 0, but the SOS relaxation returns an optimal value equal to $-\infty$. In fact, Blekherman (Blekherman, 2006) has proven that the ratio between the volume of homogeneous (all monomials have the same degree) nonnegative polynomials of degree $2d$ and the volume of homogeneous SOS polynomials of the same degree tends to infinity as n diverges to infinity.

As we have seen, the nice feature of the SOS relaxation is that it can be reformulated as a semidefinite problem. On the other hand, it can deliver a poor approximation of the optimal value of UPP. Therefore, many approaches have been proposed to keep the good part of the SOS relaxation (the possibility of computing the corresponding bound through a semidefinite problem), while the bounds, computed through the solution of semidefinite problems of increasing size, define a hierarchy converging to the optimal value of UPP. In what follows we will discuss some of these approaches.

4.5.3 Hierarchies of bounds

In (Lasserre, 2001) an approach has been proposed, based on the following theorem proven in (Cassier, 1984).

Theorem 4.63. *Given $f \in \mathbb{R}[\mathbf{x}]$, f is nonnegative over the ball centered at the origin and of radius R if and only if for all $\varepsilon > 0$ there exists $p,q \in \Sigma_n$ such that*

$$f(\mathbf{x}) + \varepsilon = p(\mathbf{x}) + q(\mathbf{x})(R^2 - \|\mathbf{x}\|_2^2).$$

The problem

$$\begin{aligned} \max \quad & \alpha \\ & f(\mathbf{x}) - \alpha = p(\mathbf{x}) + q(\mathbf{x})(R^2 - \|\mathbf{x}\|_2^2), \\ & p,q \in \Sigma_{n,2k}, \end{aligned}$$

where we restrict to sums of squares of degree at most $2k$, can be solved as a semidefinite one, and the optimal value of these problems converges to the optimal value of f over the ball centered at the origin and with radius R as $k \to \infty$. If the infimum of f is attained over $\mathbb{R}^n$ and the global minimizer lies within the ball centered at the origin with radius R, then the above approach also allows us to solve the unconstrained problem.

In (Laurent, 2007) the following observation is proven.

Observation 4.3. *Consider the PP problem* (4.112). *Then, if the ideal generated by the polynomial functions defining the equality constraints, i.e., $\langle p_1, \ldots, p_m \rangle$, has a finite variety (or, equivalently, it is zero-dimensional), then the problem can be reformulated as a semidefinite programming problem.*

Now we notice that if f attains its infimum value over $\mathbb{R}^n$ (i.e., there exists $\mathbf{x}^* \in \mathbb{R}^n$ such that $f(\mathbf{x}^*) = f_*$), then we can reformulate the UPP problem as the following constrained one by imposing the stationarity condition

$$\begin{aligned} \inf \quad & f(\mathbf{x}) \\ & \tfrac{\partial f}{\partial x_j}(\mathbf{x}) = 0, \quad j = 1, \ldots, n. \end{aligned}$$

Then, if the gradient variety is finite, Observation 4.3 allows us to conclude that the optimal value of the problem can be computed by a single semidefinite problem.

Since the objective function f might not attain its infimum value and, even in this case, might not have a finite gradient variety, Jibetean and Laurent (Jibetean & Laurent, 2005) (see also (Hanzon & Jibetean, 2003)) suggest to add a perturbation term to f. For some small $\varepsilon > 0$, the resulting perturbed function is

$$f_\varepsilon(\mathbf{x}) = f(\mathbf{x}) + \varepsilon \|\mathbf{x}\|_2^{2d+2}.$$

Then, it can be proven that

- f_ε attains its infimum value for any $\varepsilon > 0$;
- the gradient variety of f_ε is finite, i.e., the gradient ideal is zero-dimensional, and, in view of Observation 4.2, the corresponding quotient ring is a finite-dimensional $\mathbb{R}$-vector space.

Then, Observation 4.3 guarantees that the optimal value f_ε^* of f_ε can be computed by solving a semidefinite problem. Moreover,

$$f_\varepsilon^* \downarrow f_* \quad \text{as} \quad \varepsilon \downarrow 0,$$

i.e., the optimal values converge to f_* as ε decreases to 0.

Another approach based on a perturbation of f has been presented in (Lasserre, 2006). In that paper the following result has been proven.

Theorem 4.64. *Let $f \in \mathbb{R}[\mathbf{x}]$. We have that $f \in \mathcal{Q}_n$ if and only if for all $\varepsilon > 0$ there exists a nonnegative integer s such that*

$$f_\varepsilon(\mathbf{x}) = f(\mathbf{x}) + \varepsilon \sum_{i=1}^{n} \sum_{j=0}^{s} \frac{x_i^{2j}}{j!} \in \Sigma_n.$$

The value s depends on ε and on f (more precisely, in (Lasserre & Netzer, 2006) it is proven that it depends on ε, n, the degree of f, and on some bound on the size of the coefficients of f). We note that the perturbation term is the sum of the Taylor series for the exponential functions $\exp(x_i^2)$ truncated at the term of order $2s$. We also note that while the result by Blekherman cited in Section 4.5.2 suggests that the SOS polynomials are few when compared with the nonnegative polynomials, this result shows that the SOS polynomials form at least a dense set within the set of nonnegative polynomials. In view of Theorem 4.64, the infimum f_*^ε of f_ε over $\mathbb{R}^n$ can be computed by solving the corresponding SOS relaxation.

The main drawback of the two approaches above based on a perturbation f_ε of the objective function f is that small ε values lead to polynomials with small coefficients which, in turn, lead to semidefinite problems whose solution is made quite complicated by numerical instability.

Such drawback has been avoided in the approach described in (Nie, Demmel, & Sturmfels, 2006). In that paper the following theorem has been proven.

Theorem 4.65. *If the gradient ideal I_{grad}^f is radical and $f \geq 0$ over the real variety $V^{\mathbb{R}}(\mathcal{I}_{grad}^f)$ (the set of stationary points of f), then*

$$f(\mathbf{x}) = \sum_{i=1}^{t} p_i(\mathbf{x})^2 + \sum_{j=1}^{n} q_j(\mathbf{x}) \frac{\partial f}{\partial x_j}(\mathbf{x}),$$

for some $p_i, q_j \in \mathbb{R}[\mathbf{x}]$, $i = 1, \ldots, s$, $j = 1, \ldots, n$.

In other words, the theorem states that f is an SOS in the quotient ring $\mathbb{R}[\mathbf{x}]/\mathcal{I}^f_{grad}$. While Theorem 4.65 is valid when the gradient ideal I^f_{grad} is radical, the same result is also true when f is strictly positive over the gradient real variety, as stated in the following theorem.

Theorem 4.66. *If $f > 0$ over the real variety $V^{\mathbb{R}}(\mathcal{I}^f_{grad})$, then f is an SOS in the quotient ring $\mathbb{R}[\mathbf{x}]/\mathcal{I}^f_{grad}$.*

Let us consider the SOS relaxation

$$\begin{aligned} f_*^k = \max \quad & \alpha \\ & f(\mathbf{x}) - \alpha - \textstyle\sum_{j=1}^n q_j(\mathbf{x}) \frac{\partial f}{\partial x_j}(\mathbf{x}) \in \Sigma_{n,2k}, \\ & q_j \in \mathbb{R}[\mathbf{x}]_{2k-d+1}, \qquad\qquad j = 1,\dots,n, \end{aligned} \tag{4.117}$$

where the q_j's are constrained to belong to the set of polynomials with degree not larger than $2k-d+1$ (this is a finite-dimensional vector space, defined by the coefficients of the monomials, whose dimension is $\tau(n, 2k-d+1)$). This relaxation can be rewritten as a semidefinite problem where the coefficients of the polynomials q_j's are variables.

The following theorem has been proven in (Nie et al., 2006).

Theorem 4.67. *Assume that f attains its infimum value f_* at some point in $\mathbb{R}^n$. Then*

$$f_*^k \uparrow f_* \quad \text{as} \quad k \to \infty,$$

and if $\mathcal{I}^f_{grad}$ is radical, then there exists some integer K such that $f_*^K = f_*$.

Proof. First we notice that the infimum value is certainly attained at a stationary point, i.e., a point in $V^{\mathbb{R}}(\mathcal{I}^f_{grad})$. The optimal value f_*^k of (4.117) is certainly a lower bound for f_*. Indeed, for any feasible α we have that

$$f(\mathbf{x}) - \alpha - \sum_{j=1}^n q_j(\mathbf{x}) \frac{\partial f}{\partial x_j}(\mathbf{x}) \geq 0 \quad \forall\, \mathbf{x} \in \mathbb{R}^n.$$

In particular,

$$f(\mathbf{x}) - \alpha \geq 0 \quad \mathbf{x} \in V^{\mathbb{R}}(\mathcal{I}^f_{grad}),$$

which implies $\alpha \leq f_*$. Moreover, the sequence $\{f_*^k\}$ is nondecreasing (as k increases, the polynomials q_j's can be chosen in a larger set of polynomials), and, consequently, the sequence converges. Now we can remark that for an arbitrary $\varepsilon > 0$, we have that $f(\mathbf{x}) - f_* + \varepsilon$ is strictly positive over its real gradient variety, which is exactly $V^{\mathbb{R}}(\mathcal{I}^f_{grad})$. Therefore, we can apply Theorem 4.66, and for some $k = k(\varepsilon)$, $f_*^k \geq f_* - \varepsilon$. Then, the sequence $\{f_*^k\}$ converges to f_*.

The proof of the second statement is the following. If the gradient ideal of f (and, thus, also of $f - f_*$) is radical, Theorem 4.65 ensures that $f - f_*$ is SOS in the quotient ring $\mathbb{R}[\mathbf{x}]/\mathcal{I}^f_{grad}$. Therefore, there exists some integer K such that $f_*^K = f_*$. □

Unfortunately, the above approach strongly relies on the assumption that f attains its infimum value. If this is not the case, then the approach does not necessarily converge to f_*, as the following example shows.

Example 4.68. The simplest example of polynomial with a finite infimum value which is not attained in $\mathbb{R}^n$ is the polynomial

$$(1-x_1x_2)^2+x_2^2,$$

whose unique stationary point is the origin, where the objective function value is equal to 1, but whose infimum value is 0, at which the function converges along the curve $(x_1\ \frac{1}{x_1})$ as x_1 diverges to ∞. It can be seen that the above approach converges to the infimum of f over the gradient real variety (i.e., to 1) and not to the infimum value 0 over $\mathbb{R}^n$. ∎

Some attempts have been made to remove the assumption that f attains its infimum. We note that the approach in (Nie et al., 2006) relies on the fact that, if the infimum is attained, then the infimum value over $\mathbb{R}^n$ is equal to the infimum value over the gradient real variety $V^{\mathbb{R}}(\mathcal{I}_{grad})$. Therefore, a possible idea is that of finding new sets (other than the gradient real variety) with the following two properties:

- the infimum of f over $\mathbb{R}^n$ and over the set are equal;
- nonnegativity over the set can be certified via SOS.

We would like to include all points at which the gradient may not vanish but reduces to the null vector as the norm of the points diverges to infinity. Schweighofer (Schweighofer, 2006) employs the notion of generalized critical value.

Definition 4.69. *A* critical value *of f is a value attained by f at a stationary point. A* generalized critical value *is a value c such that there exists a sequence $\{\mathbf{x}_k\}$ such that*

$$\|\nabla f(\mathbf{x}_k)\|_2(1+\|\mathbf{x}_k\|_2)\to\ 0,\quad f(\mathbf{x}_k)\to c,\quad \textit{as}\ \ k\to\infty.$$

Then, Schweighofer introduces the following set, called *principal gradient tentacle*:

$$S(\nabla f)=\{\mathbf{x}\in\mathbb{R}^n\ :\ 1-\|\nabla f\|_2^2\|\mathbf{x}\|_2^2\geq 0\}.$$

Note that this set contains the gradient real variety, but also sequences of points which define generalized critical values. In (Schweighofer, 2006) the following result has been proven.

Theorem 4.70. *If f is bounded from below, then its infimum value f_* is a generalized critical value and is also equal to the infimum value of f over $S(\nabla f)$.*

Next, the notion of isolated singularities at infinity is introduced.

Definition 4.71. *Let $\mathbb{P}^{n-1}(\mathbb{C})$ denote the $(n-1)$-dimensional projective space over $\mathbb{C}$ (the projective space of lines in $\mathbb{C}^n$, where each $\mathbf{z}\neq\mathbf{0}$, $\mathbf{z}\in\mathbb{C}^n$, is equated to each $\lambda\mathbf{z}$, $\lambda\in\mathbb{C}$, $\lambda\neq 0$). We write a polynomial f of degree d as a sum of its homogeneous components, i.e., $f(\mathbf{x})=\sum_{i=0}^d f^i(\mathbf{x})$, where each f^i is zero or homogeneous of degree i. Then, f has* *only* isolated singularities at infinity *either if it is constant, or if there are only finitely many* $\mathbf{z} \in \mathbb{P}^{n-1}(\mathbb{C})$ *such that*

$$\frac{\partial f^d}{\partial x_1}(\mathbf{z}) = \cdots = \frac{\partial f^d}{\partial x_n}(\mathbf{z}) = f^{d-1}(\mathbf{z}) = 0.$$

We remark that for $n = 2$ all polynomials have only isolated singularities at infinity. Then, the following theorem is proven.

Theorem 4.72. *Let us assume that f is bounded from below and that either it has only isolated singularities at infinity or $S(\nabla f)$ is compact. Then, f is nonnegative over $\mathbb{R}^n$ if and only if for all $\varepsilon > 0$,*

$$f(\mathbf{x}) + \varepsilon = p(\mathbf{x}) + q(\mathbf{x})(1 - \|\nabla f(\mathbf{x})\|_2^2 \|\mathbf{x}\|_2^2)$$

for $p, q \in \Sigma_n$.

Then, the following subproblems are defined:

$$\begin{aligned} \max \quad & \alpha \\ & f(\mathbf{x}) - \alpha = p(\mathbf{x}) + q(\mathbf{x})(1 - \|\nabla f\|_2^2 \|\mathbf{x}\|_2^2), \\ & q \in \Sigma_{n,2k}, \ p \in \Sigma_n. \end{aligned}$$

These can be reformulated as semidefinite problems, and their optimal values f_*^k are proven to converge to f_*. Note that the degree of p is not restricted, but the restriction imposed on the degree of q automatically implies that the degree of p can not be larger than $2(k+d)$. Schweighofer leaves as an open problem if the above results are still true if the assumption that f has only isolated singularities at infinity is removed.

Vui and So'n (Vui & So'n, 2008) have shown that such assumption can be removed if a set different from the principal gradient tentacle is considered. For each $1 \le i < j \le n$, they define the polynomials

$$g_{ij}(\mathbf{x}) = x_j \frac{\partial f}{\partial x_i}(\mathbf{x}) - x_i \frac{\partial f}{\partial x_j}(\mathbf{x}).$$

Let M be a value attained by f over $\mathbb{R}^n$, such as $M = f(\mathbf{0})$. Then, they define the *truncated tangency variety* as

$$\Gamma_M(f) = \{\mathbf{x} \in \mathbb{R}^n \ : \ f(\mathbf{x}) \le M, \ g_{ij}(\mathbf{x}) = 0, \ 1 \le i < j \le n\}.$$

In the definition *tangency* refers to the fact that the points satisfying $g_{ij}(\mathbf{x}) = 0$, $1 \le i < j \le n$, are those where the level sets of f are tangent to the sphere centered at the origin and with radius $\|\mathbf{x}\|_2$, while *truncated* refers to the intersection with the nonempty level set $\{\mathbf{x} \ : \ f(\mathbf{x}) \le M\}$. In (Vui & So'n, 2008) the following results (related to Theorems 4.70–4.72) are proven.

Theorem 4.73. *The infimum value f_* of f over $\mathbb{R}^n$ is equal to the infimum value of f over $\Gamma_M(f)$.*

Theorem 4.74. *The polynomial f is nonnegative over $\mathbb{R}^n$ if and only if for all $\varepsilon > 0$,*

$$f(\mathbf{x}) + \varepsilon = p(\mathbf{x}) + q(\mathbf{x})(M - f(\mathbf{x})) + \sum_{1 \le i < j \le n} \phi_{ij}(\mathbf{x}) g_{ij}(\mathbf{x}),$$

where $p, q \in \Sigma_n$, $\phi_{ij} \in \mathbb{R}[\mathbf{x}]$.

Then, the following subproblems, where we restrict the degrees of the polynomials p, q, and ϕ_{ij}, are defined:

$$\begin{aligned}
\max \quad & \alpha \\
& f(\mathbf{x}) - \alpha = p(\mathbf{x}) + q(\mathbf{x})(M - f(\mathbf{x})) + \textstyle\sum_{1 \le i < j \le n} \phi_{ij}(\mathbf{x}) g_{ij}(\mathbf{x}), \\
& p, q \in \Sigma_{n,2k}, \\
& \phi_{ij} \in \mathbb{R}[\mathbf{x}]_{2k}, \qquad\qquad 1 \le i < j \le n.
\end{aligned}$$

Also, these problems can be reformulated as semidefinite ones, and the optimal values f_*^k are proven to converge to f_*.

4.5.4 The moment relaxation

For the relaxations defined above it is possible to define dual counterparts. Following (Lasserre, 2001), in what follows we derive the dual counterpart of the SOS relaxation (4.116), but similar results can also be derived for other relaxations. The dual problem is based on moments, whose definition is the following.

Definition 4.75. *Let μ be a Borel measure on $\mathbb{R}^n$. For some monomial $\mathbf{x}^{\boldsymbol{\alpha}}$, we denote by*

$$y_{\boldsymbol{\alpha}} = \int \mathbf{x}^{\boldsymbol{\alpha}} \mu(d\mathbf{x})$$

the moment *of order $\boldsymbol{\alpha}$ for μ. The corresponding sequence of moments $\mathbf{y} = (y_{\boldsymbol{\alpha}})_{\boldsymbol{\alpha} \in \mathbb{N}^n}$ is said to be* represented *by the measure μ. If we restrict the attention to monomials of order not larger than some nonnegative integer d, then $\mathbf{y}_d = (y_{\boldsymbol{\alpha}})_{\boldsymbol{\alpha} \in \mathbb{N}^n_d}$ is called the sequence of moments of μ up to order d.*

Note that $y_0 = 1$. Also note that for some polynomial f with coefficient vector $\mathbf{f}$, and some measure μ, we have that

$$\int f(\mathbf{x}) \mu(d\mathbf{x}) = \sum_{\boldsymbol{\alpha}} f_{\boldsymbol{\alpha}} \int \mathbf{x}^{\boldsymbol{\alpha}} \mu(d\mathbf{x}) = \sum_{\boldsymbol{\alpha}} f_{\boldsymbol{\alpha}} y_{\boldsymbol{\alpha}} = \mathbf{y}^T \mathbf{f}. \tag{4.118}$$

Now, let

$$\mathcal{M} = \{\mathbf{y} \in \mathbb{R}^{\mathbb{N}^n} \ : \ \mathbf{y} \text{ has some representing measure } \mu\}.$$

Then, the following observation shows that minimizing f over $\mathbb{R}^n$ is equivalent to minimizing a linear function over $\mathcal{M}$.

Observation 4.4. *We have that*

$$\inf_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) = \inf_{\mu} \int f(\mathbf{x})\mu(d\mathbf{x}) = \inf_{\mathbf{y}\in\mathcal{M}} \mathbf{y}^T\mathbf{f}.$$

Proof. Let

$$f_* = \inf_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}).$$

Obviously, for any measure μ

$$\int f(\mathbf{x})\mu(d\mathbf{x}) \geq \int f_*\mu(d\mathbf{x}) = f_*.$$

Conversely, let $\{\mathbf{x}_k\}$ be a sequence of points such that

$$f(\mathbf{x}_k) \downarrow f_*.$$

Let

$$\mu_k = \delta_{\mathbf{x}_k}$$

be the Dirac measure at $\mathbf{x}_k$ (i.e., the measure whose mass is all concentrated at the point $\mathbf{x}_k$). Then,

$$\int f(\mathbf{x})\mu_k(d\mathbf{x}) = f(\mathbf{x_k}),$$

so that

$$\inf_{\mu} \int f(\mathbf{x})\mu(d\mathbf{x}) \leq \inf_{\delta_{\mathbf{x}_k}} \int f(\mathbf{x})\delta_{\mathbf{x}_k}(d\mathbf{x}) = \inf_{\mathbf{x}_k} f(\mathbf{x}_k) = f_*.$$

The equality

$$\inf_{\mu} \int f(\mathbf{x})\mu(d\mathbf{x}) = \inf_{\mathbf{y}\in\mathcal{M}} \mathbf{y}^T\mathbf{f}$$

follows from (4.118). □

Now, for some infinite-dimensional vector $\mathbf{y} \in \mathbb{N}^n$, let $M(\mathbf{y})$ be the matrix indexed by $\mathbb{N}^n$ and whose $(\boldsymbol{\alpha},\boldsymbol{\beta})$-entry is $y_{\boldsymbol{\alpha}+\boldsymbol{\beta}}$. Similarly, for a sequence $\mathbf{y}_{2t} \in \mathbb{N}^n$, let $M_t(\mathbf{y})$ be the matrix indexed by $\mathbb{N}^n_t$ and whose $(\boldsymbol{\alpha},\boldsymbol{\beta})$-entry is still $y_{\boldsymbol{\alpha}+\boldsymbol{\beta}}$. We can prove the following observation.

Observation 4.5. *If* $\mathbf{y}$ *has a representing measure* μ*, then for each* $t \in \mathbb{N}$*,* $M_t(\mathbf{y})$ *is positive semidefinite.*

Proof. Let $f \in \mathbb{R}[\mathbf{x}]_t$ and $\mathbf{f}_t$ be the coefficient vector of its monomials of order up to t, $t \in \mathbb{N}$. Then,

$$\mathbf{f}_t^T M_t(\mathbf{y})\mathbf{f}_t = \sum_{\boldsymbol{\alpha},\boldsymbol{\beta} \in \mathbb{N}_t^n} f_{\boldsymbol{\alpha}} f_{\boldsymbol{\beta}} y_{\boldsymbol{\alpha}+\boldsymbol{\beta}} = \sum_{\boldsymbol{\alpha},\boldsymbol{\beta} \in \mathbb{N}_t^n} f_{\boldsymbol{\alpha}} f_{\boldsymbol{\beta}} \int \mathbf{x}^{\boldsymbol{\alpha}+\boldsymbol{\beta}} \mu(d\mathbf{x}) = \int [f(\mathbf{x})]^2 \mu(d\mathbf{x}) \geq 0, \tag{4.119}$$

from which the result follows. □

Unfortunately, the converse result is not true in general, i.e., there might exist $\mathbf{y}$ not having a representing measure μ but such that $M_t(\mathbf{y})$ is positive semidefinite. Given the above development, we have that if the degree of f is not larger than $2t$, the semidefinite problem

$$\begin{aligned} \inf \quad & \mathbf{y}_{2t}^T \mathbf{f}_{2t} \\ & y_{\mathbf{0}} = 1, \\ & M_t(\mathbf{y}) \in \mathcal{P}_{\tau(n,t)} \end{aligned} \tag{4.120}$$

returns a lower bound for f_*. Now, if we consider the SOS relaxation (4.116), it turns out that it is the dual semidefinite problem of the moment relaxation (4.120). Moreover, strong duality holds. Indeed, Slater's condition (i.e., a point lies in the relative interior of the feasible region) is satisfied for the moment relaxation. A strict feasible solution for (4.120) can be obtained by considering a measure with a strict positive density g over $\mathbb{R}^n$. In this case the left-hand side in (4.119) is also equal to

$$\int [f(\mathbf{x})]^2 g(\mathbf{x}) d\mathbf{x},$$

which, in view of the strict positivity of g, is > 0 for any $\mathbf{f} \neq \mathbf{0}$. Then, we can conclude that the two bounds of the SOS and moment relaxations are equal (see also (Lasserre, 2001)).

4.5.5 Further remarks about PP

Although we have mainly discussed relaxations based on semidefinite problems, further relaxations are possible for PP problems.

- In (Lasserre, 2005) hierarchies of linear programming (LP) relaxations for PP problems are also proposed. Their convergence, under suitable conditions, to the optimal value of PP problems is proven. However, it is also underlined that such relaxations suffer by numerical instability because of the presence of very large binomial coefficients appearing in the constraints.

- We have already mentioned RLT relaxations for PP problems. In this respect, in (Lasserre, 2002) an interesting relation has been established between the RLT constraints discussed in Section 4.3 and the moment reformulation of a PP problem. If we consider a problem with bounds $x_i \in [0,1]$, $i = 1,\ldots,n$, in (Lasserre, 2002) it is observed, for instance, that the bound-factor RLT constraints are a finite subset of the infinitely many Hausdorff moment conditions which are necessary and sufficient for the variables $x_{i_1 i_2 \ldots i_k}$ (corresponding to the variables $y_{\boldsymbol{\alpha}}$ for suitable vectors $\boldsymbol{\alpha}$) to be moments of a probability measure μ over $[0,1]^n$.

- As in (Jibetean & Laurent, 2005) and (Lasserre, 2006) (see Section 4.5.3), in (Zeng & Xiao, 2012) a perturbation of the objective function is proposed where the perturbing function is a quadratic one but its coefficients belong to a field containing $\mathbb{R}$.
- Further ways to obtain bounds for polynomial functions over a box have been proposed in (J. F. Zhang & Kwong, 2005), based on the evaluation of the polynomial function values at the set of Chebychev points over the box, and in (Nataraj & Arounassalame, 2011), based on the Bernstein form of a polynomial.

4.5.6 Rational programming problems

We finally remark that PP approaches can also be employed to deal with problems where the objective function is a *rational* one, i.e.,

$$f(\mathbf{x}) = \frac{p(\mathbf{x})}{q(\mathbf{x})},$$

where p, q are relatively prime polynomials. Indeed, if q changes sign over $\mathbb{R}^n$, then

$$\inf_{\mathbf{x}\in\mathbb{R}^n,\ q(\mathbf{x})\neq 0} f(\mathbf{x})$$

is equal to $-\infty$, while if q does not change sign over $\mathbb{R}^n$ (say, it is always positive), then the problem is equivalent to

$$\sup\{\alpha\ :\ p(\mathbf{x}) - \alpha q(\mathbf{x}) \geq 0\quad \forall\, \mathbf{x} \in \mathbb{R}^n\},$$

where we are back to PP problems, since for each fixed α, the constraints of the problem above are equivalent to the requirement of nonnegativity for the polynomial $p - \alpha q$. For more details we refer, e.g., to (Jibetean & de Klerk, 2006; Nie, Demmel, & Gu, 2008).

4.6 The α-BB approaches

While for the approaches discussed up to now we imposed that the objective and the constraint functions have to be of special forms (quadratic or polynomial), in the *α-BB* (BB stands for branch-and-bound) approaches (see (Adjiman, Androulakis, Maranas, & Floudas, 1996; Adjiman, Dallwig, Floudas, & Neumaier, 1998; Adjiman, Androulakis, & Floudas, 1998; Akrotirianakis & Floudas, 2004; Maranas & Floudas, 1994; Meyer & Floudas, 2005b)) the requirements on the functions are loosen. We only assume that a function f (which could be an objective or constraint function) belongs to $\mathcal{C}^2$, i.e., it is twice-continuously differentiable. The search for a convex underestimator of f over a box $X = \prod_{i=1}^n [\ell_i, u_i]$ can be viewed as the problem of finding a function q such that

$$\nabla^2(f(\mathbf{x}) - q(\mathbf{x})) \in \mathcal{P}_n, \qquad \forall \mathbf{x} \in X, \tag{4.121}$$

$$q(\mathbf{x}) \geq 0, \qquad \forall\, \mathbf{x} \in X, \tag{4.122}$$

where the constraint (4.121) imposes that $f - q$ is a convex function over the box, while the constraint (4.122) imposes that $f - q$ underestimates f over the box. Of course, we also would like to impose that q is as small as possible with respect to some measure, but optimization with respect to a given measure under the given constraints might be a rather complicated task. In the classical α-BB approach the attention is restricted to quadratic functions with a diagonal Hessian, defined as

$$q(\mathbf{x};\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i (x_i - \ell_i)(u_i - x_i),$$

with $\alpha_i \geq 0$, $i = 1,\ldots,n$, so that (4.122) is satisfied over X (we also refer to (Zhu & Kuno, 2005) for a similar development with boxes replaced by simplices). We notice that

$$\nabla^2(f(\mathbf{x}) - q(\mathbf{x})) = \nabla^2 f(\mathbf{x}) + 2Diag(\boldsymbol{\alpha}),$$

where $Diag(\boldsymbol{\alpha})$ is the diagonal matrix whose diagonal elements are the α_i's. Therefore, (4.121) can be satisfied if large enough α_i values are chosen. Suitable values are computed through interval arithmetic, e.g., by exploiting Gerschgorin theorem. Such theorem states that, given a symmetric $n \times n$ matrix $\mathbf{A}$, every eigenvalue of $\mathbf{A}$ lies within one of the intervals

$$\left[A_{ii} - \sum_{j \neq i} |A_{ij}|, A_{ii} + \sum_{j \neq i} |A_{ij}| \right], \quad i = 1,\ldots,n.$$

Therefore, each eigenvalue of $\nabla^2 f(\mathbf{x}) + 2Diag(\boldsymbol{\alpha})$ lies in an interval

$$\left[2\alpha_i + [\nabla^2 f(\mathbf{x})]_{ii} - \sum_{j \neq i} |[\nabla^2 f(\mathbf{x})]_{ij}|, 2\alpha_i + [\nabla^2 f(\mathbf{x})]_{ii} + \sum_{j \neq i} |[\nabla^2 f(\mathbf{x})]_{ij}| \right]$$

for some $i \in \{1,\ldots,n\}$. Now, consider the intervals $H_{ij}^X = [\underline{H}_{ij}^X, \overline{H}_{ij}^X]$ such that

$$[\nabla^2 f(\mathbf{x})]_{ij} \in H_{ij}^X \quad \forall\, \mathbf{x} \in X, \ \ \forall\, i,j = 1,\ldots,n.$$

Then, the eigenvalues of $\nabla^2(f(\mathbf{x}) - q(\mathbf{x}))$ are nonnegative (i.e., (4.121) is true) if the following is satisfied for any $i = 1,\ldots,n$:

$$\alpha_i \geq \frac{1}{2} \max_{i=1,\ldots,n} \left\{ 0, -\underline{H}_{ii}^X + \sum_{j \neq i} \max\{|\underline{H}_{ij}^X|, |\overline{H}_{ij}^X|\} \right\}. \tag{4.123}$$

Other possible choices are based on the scaled Gerschgorin theorem: if we consider any strictly positive vector $\mathbf{d}$, then $f - q$ is a convex function if for each $i \in \{1,\ldots,n\}$

$$\alpha_i \geq \frac{1}{2} \max \left\{ 0, -\underline{H}_{ii}^X + \sum_{j \neq i} \max\{|\underline{H}_{ij}^X|, |\overline{H}_{ij}^X|\} \frac{d_j}{d_i} \right\}. \tag{4.124}$$

Typical choices for $\mathbf{d}$ are (i) $d_i = 1, i = 1,\ldots,n$, strictly related to (4.123), whose right-hand side is the maximum of the right-hand sides in (4.124); (ii) $d_i = u_i - \ell_i$, which allows us to take into account the different scalings of the variables.

It is rather simple to compute the maximum distance between f and its convex underestimator. Such distance is attained at the midpoint of the box and is equal to

$$\sum_{i=1}^{n} \alpha_i \left(\frac{u_i - \ell_i}{2} \right)^2 .$$

From this formula we notice that the maximum distance decreases to 0 if we reduce the box to a single point. However, the overestimations made through interval arithmetic may result in α_i values much larger than needed, thus lowering the quality of the underestimator. For this reason in (Meyer & Floudas, 2005b) the spline variant of the classical α-BB approach has been proposed. The basic idea of spline α-BB is that of subdividing the original box into many different subboxes over which sharper estimates of the α values (and, thus, sharper underestimators) can be found. The underestimators over the different subboxes are put together in such a way that convexity, continuity, and smoothness of the overall underestimator are guaranteed. More formally, in spline α-BB, first each interval $[\ell_i, u_i]$ is subdivided into N_i subintervals $[a_i^k, a_i^{k+1}]$, $k = 0, \ldots, N_i - 1$, with

$$a_i^0 = \ell_i < a_i^1 < \cdots < a_i^{N_i - 1} < a_i^{N_i} = u_i .$$

Next, q is defined as

$$q(\mathbf{x}; \boldsymbol{\alpha}) = \sum_{i=1}^{n} q_i^{k_i}(x_i; \alpha_i^{k_i}) \quad \text{for } x_i \in [a_i^{k_i - 1}, a_i^{k_i}],$$

where

$$\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1 \ldots \boldsymbol{\alpha}_n), \quad \boldsymbol{\alpha}_i = (\alpha_i^1 \ldots \alpha_i^{N_i}), \; i = 1, \ldots, n,$$

and

$$q_i^{k_i}(x_i; \alpha_i^{k_i}) = \alpha_i^{k_i}(x_i - a_i^{k_i - 1})(a_i^{k_i} - x_i) + \beta_i^{k_i} x_i + \gamma_i^{k_i}$$

for each $k_i = 1, \ldots, N_i$. The values $\alpha_i^{k_i} \geq 0$ are chosen in such a way to guarantee that $f - q$ is convex over $\prod_{i=1}^{n} [a_i^{k_i - 1}, a_i^{k_i}]$, i.e., in such a way that $\nabla^2 (f - q)$ is positive semidefinite over this box. The other parameters are chosen in such a way that q is guaranteed to be continuous and smooth. This is accomplished as in the definition of spline functions. For each $i = 1, \ldots, n$ we need to impose

$$q_i^1(\ell_i; \alpha_i^1) = q_i^{N_i}(u_i; \alpha_i^{N_i}) = 0$$

(so that $q(\mathbf{x}; \boldsymbol{\alpha}) = 0$ at the vertices of the box X);

$$q_i^k(a_i^k; \alpha_i^{k_i}) = q_i^{k+1}(a_i^k; \alpha_i^{k_i + 1}), \quad k = 1, \ldots, N_i - 1$$

(continuity at the extremes of each subinterval); and

$$\frac{d\, q_i^k}{d\, x_i}(a_i^k; \alpha_i^{k_i}) = \frac{d\, q_i^{k+1}}{d\, x_i}(a_i^k; \alpha_i^{k_i + 1}), \quad k = 1, \ldots, N_i - 1$$

(continuity of the derivatives at the extremes of each subinterval). All this results in a linear system with the $2N_i$ variables $\beta_i^{k_i}, \gamma_i^{k_i}$, $k_i = 1,\dots,N_i$, and the $2N_i$ equations

$$\begin{aligned}
\beta_i^1 a_i^0 + \gamma_i^1 &= 0, \\
\beta_i^{k_i} a_i^{k_i} + \gamma_i^{k_i} &= \beta_i^{k_i+1} a_i^{k_i} + \gamma_i^{k_i+1}, && k_i = 1,\dots,N_i-1, \\
\beta_i^{N_i} a_i^{N_i} + \gamma_i^{N_i} &= 0, \\
-\alpha_i^{k_i}(a_i^{k_i} - a_i^{k_i-1}) + \beta_i^{k_i} &= \alpha_i^{k_i+1}(a_i^{k_i+1} - a_i^{k_i}) + \beta_i^{k_i+1}, && k_i = 1,\dots,N_i-1.
\end{aligned}$$

The following example, taken from (Meyer & Floudas, 2005b), illustrates the difference between classical and spline α-BB.

Example 4.76. Let

$$f(x) = -2x + 10x^2 - 3x^3 - 5x^4,$$

with $X = [0,1]$. In classical α-BB we take

$$q(x;\alpha) = \alpha x(1-x),$$

and we need to choose α in such a way that the second derivative of $f - q$ over $[0,1]$ is always nonnegative. Therefore, the smallest possible value for α is 29. In Figure 4.3 we show the graph of $f(x)$ and of $f(x) - q(x)$.

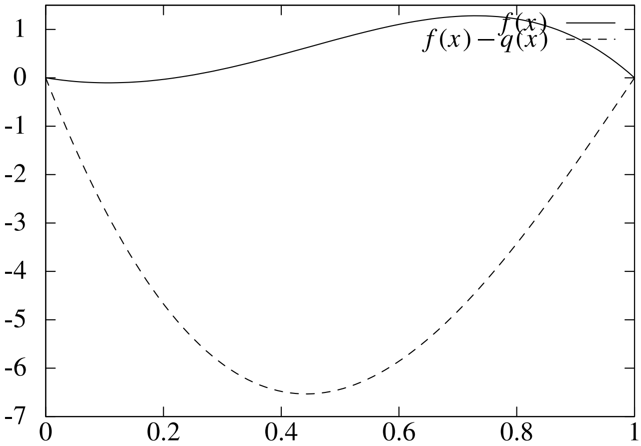


Figure 4.3. *Classical α-BB underestimator*

In spline α-BB we can subdivide $[0,1]$ into the three subintervals $[0,\frac{1}{3}]$, $[\frac{1}{3},\frac{2}{3}]$, and $[\frac{2}{3},1]$. Then, we define q as

$$q(x;\alpha_1,\alpha_2,\alpha_3) = \begin{cases} \alpha_1 x\left(\frac{1}{3} - x\right) + \beta_1 x + \gamma_1, & x \in \left[0,\frac{1}{3}\right], \\ \alpha_2\left(x - \frac{1}{3}\right)\left(\frac{2}{3} - x\right) + \beta_2 x + \gamma_2, & x \in \left[\frac{1}{3},\frac{2}{3}\right], \\ \alpha_3\left(x - \frac{2}{3}\right)(1-x) + \beta_3 x + \gamma_3, & x \in \left[\frac{2}{3},1\right], \end{cases}$$

where convexity is guaranteed by setting

$$\alpha_1 = 0, \quad \alpha_2 = \frac{28}{3}, \quad \alpha_3 = 29,$$

while continuity and smoothness are guaranteed by the solution of the system

$$\begin{aligned}
\gamma_1 &= 0,\\
\tfrac{1}{3}\beta_1 + \gamma_1 &= \tfrac{1}{3}\beta_2 + \gamma_2,\\
\tfrac{2}{3}\beta_2 + \gamma_2 &= \tfrac{2}{3}\beta_3 + \gamma_3,\\
\beta_3 + \gamma_3 &= 0,\\
-\tfrac{1}{3}\alpha_1 + \beta_1 &= \tfrac{1}{3}\alpha_2 + \beta_2,\\
-\tfrac{1}{3}\alpha_2 + \beta_2 &= \tfrac{1}{3}\alpha_3 + \beta_3,
\end{aligned} \tag{4.125}$$

from which we end up with

$$q(x) = \begin{cases} \tfrac{19}{3}x, & x \in \left[0, \tfrac{1}{3}\right],\\ \tfrac{28}{3}\left(x - \tfrac{1}{3}\right)\left(\tfrac{2}{3} - x\right) + \tfrac{29}{9}x + \tfrac{28}{27}, & x \in \left[\tfrac{1}{3}, \tfrac{2}{3}\right],\\ 29\left(x - \tfrac{2}{3}\right)(1 - x) - \tfrac{86}{9}x + \tfrac{86}{9}, & x \in [\tfrac{2}{3}, 1]. \end{cases}$$

In Figure 4.4 the graph of $f(x)$ and of $f(x) - q(x)$ are reported and compared with that of the classical α-BB underestimator.

We remark that such function q is always not larger than the one computed by classical α-BB, thus defining a better underestimator for f. ∎

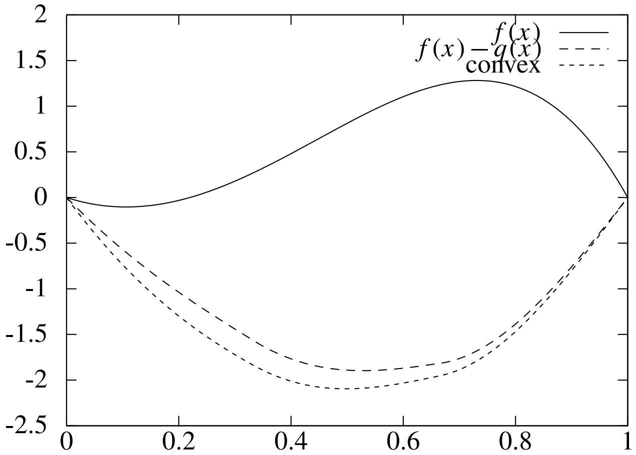


Figure 4.4. *Spline α-BB underestimator*

In order to prove that $f-q$ is a convex underestimator of f two theorems are needed. The first one states that q is nonnegative over X, i.e., $f-q$ underestimates f.

Theorem 4.77. *If $\boldsymbol{\alpha} \geq \mathbf{0}$, then*

$$q(\mathbf{x};\boldsymbol{\alpha}) \geq 0 \quad \forall\, \mathbf{x} \in X.$$

Proof. First observe that each function q_i is concave over $[\ell_i, u_i]$. Indeed, since $\alpha_i^{k_i} \geq 0$, q_i is concave over each subinterval $[a_i^{k_i}, a_i^{k_i+1}]$, $k_i = 1,\dots,N_i-1$. Since the first derivative of q_i is continuous by construction, it is also nonincreasing over $[\ell_i, u_i]$, so that q_i is concave over this interval. As $q(\mathbf{x};\boldsymbol{\alpha}) = \sum_{i=1}^n q_i(x_i;\boldsymbol{\alpha}_i)$, function q is also concave. Next, observe that for each vertex $\mathbf{v}$ of X, $q(\mathbf{v}) = 0$. Since the minimum of a concave function over a polytope is always attained at a vertex of the polytope, the minimum of q over X is equal to 0, and the result of the theorem immediately follows. □

The second theorem states that, if the $\alpha_i^{k_i}$ values are properly chosen, then convexity of $f-q$ is guaranteed.

Theorem 4.78. *If the $\alpha_i^{k_i}$ values, $i=1,\dots,n$, $k_i = 1,\dots,N_i$, are large enough, then $f-q$ is a convex function.*

Proof. If $\alpha_i^{k_i}$ are chosen large enough (e.g., using again the interval version of Gerschgorin theorem), then $g = f-q$ is convex over each subbox

$$Y = \prod_{i=1}^n [a_i^{k_i-1}, a_i^{k_i}], \quad k_i \in \{1,\dots,N_i\}.$$

From a result by Hiriart-Urruty and Lemaréchal (J. Hiriart-Urruty & Lemaréchal, 1993) we have that a differentiable function g is convex over the box Y if and only if its gradient is monotone over Y, i.e., if and only if

$$[\nabla\, g(\mathbf{x}) - \nabla\, g(\mathbf{x}')]^T(\mathbf{x}-\mathbf{x}') \geq 0 \quad \forall\, \mathbf{x},\mathbf{x}' \in Y. \tag{4.126}$$

Now, consider two neighbor subboxes Y and Y', i.e., two distinct subboxes with a common facet. Let $\mathbf{x} \in Y$, $\mathbf{x}' \in Y'$, and let

$$\mathbf{y} = \beta\mathbf{x} + (1-\beta)\mathbf{x}', \quad \beta \in (0,1),$$

be the point belonging to the intersection of the common facet with the line between $\mathbf{x}$ and $\mathbf{x}'$. Then,

$$\begin{aligned} &[\nabla\, g(\mathbf{x}) - \nabla\, g(\mathbf{x}')]^T(\mathbf{x}-\mathbf{x}') \\ &= [\nabla\, g(\mathbf{x}) - \nabla\, g(\mathbf{y})]^T(\mathbf{x}-\mathbf{x}') + [\nabla\, g(\mathbf{y}) - \nabla\, g(\mathbf{x}')]^T(\mathbf{x}-\mathbf{x}') \\ &= \tfrac{1}{1-\beta}[\nabla\, g(\mathbf{x}) - \nabla\, g(\mathbf{y})]^T(\mathbf{x}-\mathbf{y}) + \tfrac{1}{\beta}[\nabla\, g(\mathbf{y}) - \nabla\, g(\mathbf{x}')]^T(\mathbf{y}-\mathbf{x}'). \end{aligned} \tag{4.127}$$

Since both the addends on the right-hand side of the last expression are nonnegative in view of (4.126), we can conclude that

$$[\nabla\, g(\mathbf{x}) - \nabla\, g(\mathbf{x}')]^T(\mathbf{x}-\mathbf{x}') \geq 0 \quad \forall\, \mathbf{x} \in Y,\ \mathbf{x}' \in Y'.$$

If Y and Y' are not neighbor subboxes, we can prove the same result (and thus convexity of g over X) by iterating the proof above. Let $Y_1,\dots,Y_t$ be the intermediate subboxes which are crossed by the line between $\mathbf{x}$ and $\mathbf{x}'$. Then we define $\mathbf{y} \in Y \cap Y_1$ as the point belonging to the intersection of the border of Y and Y_1 with the line between $\mathbf{x}$ and $\mathbf{x}'$. Then, (4.127) is still true and

$$[\nabla\, g(\mathbf{x}) - \nabla\, g(\mathbf{y})]^T(\mathbf{x}-\mathbf{y}) \geq 0.$$

In order to also prove that

$$[\nabla\, g(\mathbf{y}) - \nabla\, g(\mathbf{x}')]^T(\mathbf{y}-\mathbf{x}'),$$

we can iterate the procedure by substituting $\mathbf{x}$ with $\mathbf{y}$ and $\mathbf{y}$ with $\mathbf{y}' \in Y_1 \cap Y_2$, the point belonging to the intersection of the border of Y_1 and Y_2 with the line between $\mathbf{x}$ and $\mathbf{x}'$. We proceed in this way until, after crossing all the intermediate subboxes $Y_1,\dots,Y_t$, we reach subbox Y'. □

A further improvement is proposed in (Meyer & Floudas, 2005b). In the above development we always imposed nonnegativity of the $\alpha_i^{k_i}$ values and, thus, concavity of q. In fact, over regions where f is strictly convex, q does not need to be concave in order to have $f-q$ convex. In other words, we can also choose some negative $\alpha_i^{k_i}$ values. Some care is needed in doing that. Theorem 4.77 is based on the fact that each function q_i is nonnegative over each interval $[\ell_i,u_i]$. This was previously attained by imposing that (i) the function q_i is equal to 0 at the extremes of the interval $[\ell_i,u_i]$; (ii) all the $\alpha_i^{k_i}$ values are nonnegative, so that the function q_i is concave over $[\ell_i,u_i]$. Since we are removing (ii), we need some other way to guarantee the same result. Meyer and Floudas observe that nonnegativity of q_i is also attained by requiring, besides (i), that each function q_i has no stationary point within the subintervals $[a_i^{k_i},a_i^{k_i+1}]$ over which q_i is convex (i.e., those where $\alpha_i^{k_i} < 0$). This way the minimum value of q (equal to 0) is still attained at a vertex of the box X. We refer to (Meyer & Floudas, 2005b) for the details about proper choices of negative α values, together with the related modifications of the β and γ values in order to preserve continuity and smoothness of q. Here we just illustrate the idea through the previous example.

Example 4.79. We notice that the function f is strictly convex over $[0,\frac{1}{3}]$. The minimum value of the second derivative of f over $[0,\frac{1}{3}]$ is equal to $\frac{11}{3}$. Therefore, $f-q$ is convex by taking any value $\alpha_1 \geq -\frac{11}{3}$. In order to guarantee continuity and smoothness, we need to solve the system (4.125) where we set $\alpha_2 = \frac{28}{3}$, $\alpha_3 = 29$, while we leave α_1 as a parameter. The solution of the system, parameterized with respect to α_1, is the following:

$$\gamma_1 = 0,$$

$$\gamma_2 = \frac{28}{27} + \frac{1}{9}\alpha_1,$$

$$\begin{aligned}
\gamma_3 &= \frac{86}{9} + \frac{1}{9}\alpha_1,\\
\beta_1 &= \frac{19}{3} + \frac{2}{9}\alpha_1,\\
\beta_2 &= \frac{29}{9} + \frac{2}{9}\alpha_1,\\
\beta_3 &= -\frac{86}{9} - \frac{1}{9}\alpha_1.
\end{aligned}$$

Finally, we need to impose that q has no stationary point in $[0, \frac{1}{3}]$. In this interval, the function q is equal to

$$\alpha_1 x\left(\frac{1}{3} - x\right) + \left(\frac{19}{3} + \frac{2}{9}\alpha_1\right)x,$$

whose stationary point is

$$\frac{19}{6\alpha_1} + \frac{2}{9},$$

which does not belong to $[0, \frac{1}{3}]$ for $\alpha_1 > -\frac{57}{4}$. Therefore, all requirements (continuity, smoothness, and nonnegativity of q, convexity of $f - q$) are fulfilled for all $\alpha_1 \geq -\frac{11}{3}$. In particular, if we consider the limit value $\alpha_1 = -\frac{11}{3}$, we end up with the following function q

$$q(x) = \begin{cases} -\frac{11}{3}x\left(\frac{1}{3} - x\right) + \frac{149}{27}x, & x \in \left[0, \frac{1}{3}\right],\\ \frac{28}{3}\left(x - \frac{1}{3}\right)\left(\frac{2}{3} - x\right) + \frac{98}{27}x + \frac{17}{27}, & x \in \left[\frac{1}{3}, \frac{2}{3}\right],\\ 29\left(x - \frac{2}{3}\right)(1 - x) - \frac{247}{27}x + \frac{247}{27}, & x \in \left[\frac{2}{3}, 1\right]. \end{cases}$$

Such function q is always not larger than the one computed by imposing $\alpha_1 = 0$. In Figure 4.5 we plot functions $q(x)$ as obtained through the classical and the spline α-BB approach and the one obtained in this example. ∎

Another α-BB approach based on a domain subdivision has been discussed in (Gounaris & Floudas, 2008a, 2008b). In (Gounaris & Floudas, 2008b) the case of univariate functions is studied. A given interval is subdivided into N subintervals of equal length. A different α value is computed over each subinterval, giving rise to a piecewise convex underestimating function. Convexification is obtained by two procedures. The first one identifies, for each pair of subintervals, the tightest line underestimating both convex pieces over the corresponding subintervals; the second one identifies those lines which are needed to define a convex underestimator U_N over the whole interval. The resulting underestimator is composed by some of the line segments identified by this procedure and by parts of the convex pieces. Alternatively, possibly including also the supporting lines at the two extremes of the interval, it is possible to define a piecewise linear convex underestimator V_N by considering only the line segments. The two procedures are illustrated in Figures 4.6–4.7. In (Gounaris & Floudas, 2008b) it is also proven that for N large enough, the function U_N is equal to the convex envelope of the function over the interval, while V_N can be made arbitrarily close to the convex envelope by increasing N. In (Gounaris &

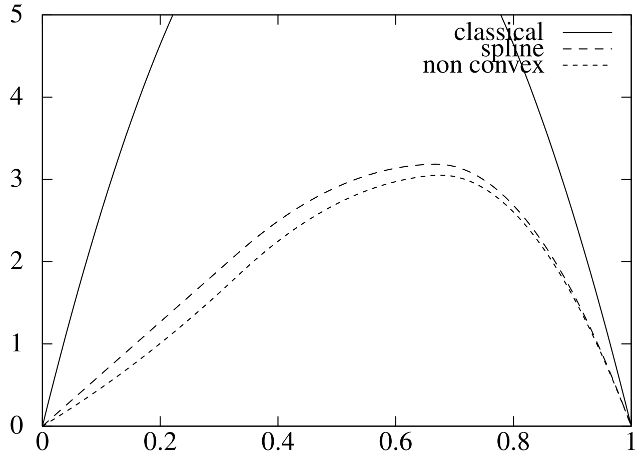


Figure 4.5. *Plot of function $q(x)$ for classical α-BB, spline α-BB, and the nonconvex $q(x)$ obtained in this example*

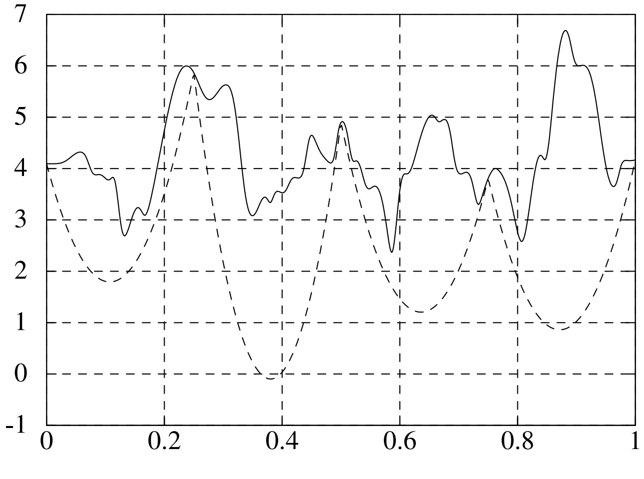


Figure 4.6. *A nonconvex objective function (solid line) and a piecewise quadratic underestimation (NB: the underestimators displayed are for illustration purpose only and where not derived using the formulae of α-BB)*

Floudas, 2008a) the technique is extended to multivariate functions. The extension defines for each variable x_i a convex underestimator V_i which is valid over the whole domain X but only depends on x_i. The computation of these underestimators relies on the techniques developed for the univariate case. Finally, the convex underestimator V over X is computed as the maximum of all the V_i's, i.e.,

$$V(x_1,\ldots,x_n) = \max_{i=1,\ldots,n} V_i(x_i). \tag{4.128}$$

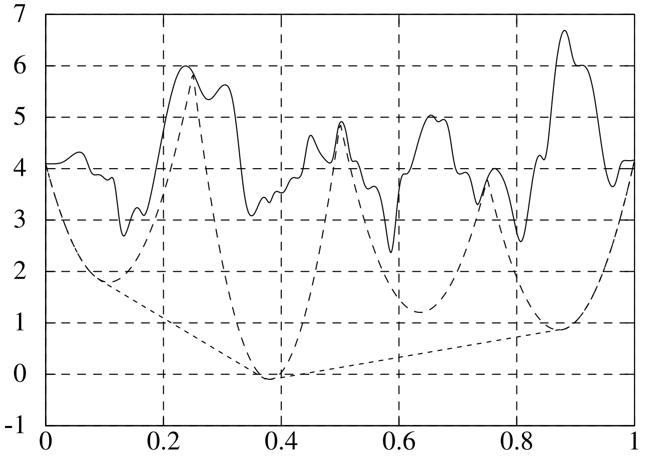


Figure 4.7. *Convex piecewise linear/quadratic underestimation (dotted)*

In (Gounaris & Floudas, 2008a) it is also proposed to improve the quality of the underestimator through rotations of the domain. An orthonormal transformation of the variables' space

$$\mathbf{y} = \mathbf{Q}\mathbf{x} \quad \text{with} \quad \mathbf{Q}^{-1} = \mathbf{Q}^T \quad \text{and} \quad det(\mathbf{Q}) = 1,$$

is performed, and we have that

$$f(\mathbf{x}) = f(\mathbf{Q}^T\mathbf{y}) = f_{\mathbf{Q}}(\mathbf{y})$$

over a domain $X' \supseteq X$ (see Figure 4.8). Then, convex understimators $V_i^{\mathbf{Q}}(y_i)$ with respect to the new variables y_i, which are valid over X' (and, thus, over X), are computed, and finally, the new underestimators are added to the original ones V_i's in (4.128).

Another variant of the classical α-BB approach is the generalized α-BB approach (see (Akrotirianakis & Floudas, 2004)). In the classical α-BB approach the function q is

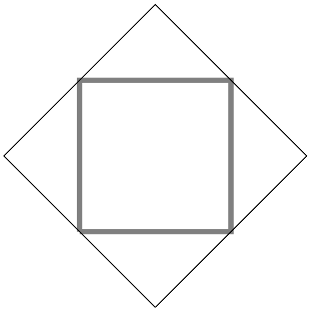


Figure 4.8. *The original feasible set X (in gray) and the domain X' obtained after an orthonormal transformation*

quadratic. In fact, q could be any function provided that $f-q$ is a convex underestimator of f. In (Akrotirianakis & Floudas, 2004) the following function q has been proposed:

$$q(\mathbf{x};\boldsymbol{\gamma})=\sum_{i=1}^{n}(1-\exp(\gamma_i(x_i-\ell_i)))(1-\exp(\gamma_i(u_i-x_i))),$$

where $\gamma_i \geq 0, i=1,\dots,n$. Such function q is separable and its Hessian is a diagonal matrix whose ith diagonal element is

$$-\gamma_i^2(\exp(\gamma_i(x_i-\ell_i))+\exp(\gamma_i(u_i-x_i))).$$

It is immediately seen that this q function is concave. Moreover, q is equal to 0 at the vertices of the box X, and thus q is nonnegative over X. If a large enough $\boldsymbol{\gamma}$ is chosen, the Hessian of $f-q$ is a positive semidefinite matrix over X, i.e., $f-q$ is convex. In (Akrotirianakis & Floudas, 2004) the choice of appropriate $\boldsymbol{\gamma}$ values is performed through a systematic procedure. Such procedure is based on the strict relation between the parameters $\boldsymbol{\gamma}$ and the parameters $\boldsymbol{\alpha}$ in the classical α-BB (it is shown that for some choices of $\boldsymbol{\gamma}$ there are choices for $\boldsymbol{\alpha}$ for which the corresponding underestimators, $f(\mathbf{x})-q(\mathbf{x};\boldsymbol{\gamma})$ and $f(\mathbf{x})-q(\mathbf{x};\boldsymbol{\alpha})$, respectively, have the same maximum separation distance from f), and ends up with an understimator that is at least as good as the one obtained in the classical α-BB with $\boldsymbol{\alpha}$ chosen as in (4.124).

4.7 Difference-of-convex problems

Convex relaxations for some GO problems can also be derived through *difference-of-convex* decompositions of the objective and/or constraint functions.

Definition 4.80. *A function f is called a* DC function *over a convex set X if there exist two convex functions g,h over X such that*

$$f(\mathbf{x})=g(\mathbf{x})-h(\mathbf{x}) \quad \forall\, \mathbf{x}\in X.$$

Note that the class of DC functions includes a large number of functions, as proven by the following theorem.

Theorem 4.81. *Let X be convex and compact, and let $f\in\mathcal{C}^2$. Then, f is a DC function over X.*

Proof. Let

$$\rho \geq \left|\min_{\mathbf{x}\in X}\lambda_{\min}(\nabla^2 f(\mathbf{x}))\right|,$$

where $\lambda_{\min}(\mathbf{A})$ denotes the minimum eigenvalue for the matrix $\mathbf{A}$. Then, the function $f(\mathbf{x})+\frac{1}{2}\rho\|\mathbf{x}\|_2^2$ is convex over X. Indeed, in view of the definition of ρ, its Hessian $\nabla^2 f(\mathbf{x})+\rho\mathbf{I}$ is such that

$$\lambda_{\min}(\nabla^2 f(\mathbf{x})+\rho\mathbf{I})\geq 0 \quad \forall\, \mathbf{x}\in X,$$

i.e., the function is convex (see Observation A.3). Therefore,

$$g(\mathbf{x}) = f(\mathbf{x}) + \frac{1}{2}\rho\|\mathbf{x}\|_2^2, \quad h(\mathbf{x}) = \frac{1}{2}\rho\|\mathbf{x}\|_2^2$$

define a DC decomposition for f. □

Of course, while always stating the *existence* of a DC decomposition, Theorem 4.81 is also useful to build it only if there is some way to bound from below the lowest eigenvalue of the Hessian of f over X (note the close relation of this result to that already seen for the α-BB approach).

The result of some operations on DC functions is still a DC function. For instance, it can be easily seen that any linear combination of DC functions is a DC function. The following proposition states that the DC property is preserved by the min and max operators.

Proposition 4.82. *Let f_i, $i = 1,\ldots,m$, be DC functions over X. Then,*

$$\max\{f_1(\mathbf{x}),\ldots,f_m(\mathbf{x})\}, \quad \min\{f_1(\mathbf{x}),\ldots,f_m(\mathbf{x})\}$$

are also DC functions over X.

Proof. By assumption

$$f_i(\mathbf{x}) = g_i(\mathbf{x}) - h_i(\mathbf{x}) \quad \forall\, \mathbf{x} \in X, \;\; i = 1,\ldots,m,$$

with g_i, h_i convex over X. Since for each $i \in \{1,\ldots,m\}$

$$f_i(\mathbf{x}) = g_i(\mathbf{x}) + \sum_{j=1\,:\,j\neq i}^{m} h_j(\mathbf{x}) - \sum_{j=1}^{m} h_j(\mathbf{x}),$$

we also have that

$$\max\{f_1(\mathbf{x}),\ldots,f_m(\mathbf{x})\} = \max_{i=1,\ldots,m}\left[g_i(\mathbf{x}) + \sum_{j=1\,:\,j\neq i}^{m} h_j(\mathbf{x})\right] - \sum_{j=1}^{m} h_j(\mathbf{x}),$$

where

$$\max_{i=1,\ldots,m}\left[g_i(\mathbf{x}) + \sum_{j=1\,:\,j\neq i}^{m} h_j(\mathbf{x})\right]$$

is a maximum of a finite number of convex functions and is, thus, convex (see Observation A.5), while $\sum_{j=1}^{m} h_j(\mathbf{x})$ is a sum of convex functions and, thus, also convex (see Observation A.4). The proof for the min operator is completely similar. □

Once a DC decomposition for a function f is available, we can exploit it to define convex underestimators for f. Indeed, if f admits a DC decomposition $f = g - h$ over X, then any convex underestimator q over X for the concave function $-h$ immediately delivers the convex underestimator $g + q$ for f over X.

A *DC optimization problem* is a problem where the objective and constraint functions are all DC, i.e.,

$$\begin{aligned} \min \quad & g_0(\mathbf{x}) - h_0(\mathbf{x}) \\ & g_i(\mathbf{x}) - h_i(\mathbf{x}) \leq 0, \quad i = 1,\ldots,m, \end{aligned} \tag{4.129}$$

where all functions g_i, h_i, $i = 0,\ldots,m$, are convex. A convex relaxation for this problem is obtained by substituting each concave function $-h_i$, $i = 0,\ldots,m$, with a convex underestimator q_i, $i = 0,\ldots,m$. In fact, the following proposition shows that through suitable transformations of problem (4.129), it is possible to define a convex relaxation by deriving a convex underestimator for a single function.

Proposition 4.83. *Any DC optimization problem (4.129) can be written in the* canonical DC form

$$\begin{aligned} \min \quad & \mathbf{c}^T \mathbf{y} \\ & \bar{g}(\mathbf{y}) \leq 0, \\ & \bar{h}(\mathbf{y}) \geq 0, \end{aligned} \tag{4.130}$$

where the objective function is linear and the two functions $\bar{g}, \bar{h}$ are convex.

Proof. Problem (4.129) can be rewritten as

$$\begin{aligned} \min \quad & t \\ & g_0(\mathbf{x}) - h_0(\mathbf{x}) - t \leq 0, \\ & g_i(\mathbf{x}) - h_i(\mathbf{x}) \leq 0, \qquad i = 1,\ldots,m. \end{aligned}$$

Next, the $m+1$ DC constraints can be rewritten as the single DC constraint

$$\max\{g_0(\mathbf{x}) - h_0(\mathbf{x}) - t, g_1(\mathbf{x}) - h_1(\mathbf{x}), \ldots, g_m(\mathbf{x}) - h_m(\mathbf{x})\} \leq 0.$$

Let $\tilde{g} - \tilde{h}$ be a DC decomposition of this DC function. Then, the single DC constraint can be decomposed into the two constraints

$$\tilde{g}(\mathbf{x}, t) - z \leq 0, \quad \tilde{h}(\mathbf{x}, t) - z \geq 0.$$

Now the result immediately follows by taking $\mathbf{y} = (\mathbf{x}\ t\ z)$ and

$$\bar{g}(\mathbf{x}, t, z) = \tilde{g}(\mathbf{x}, t) - z, \quad \bar{h}(\mathbf{x}, t, z) = \tilde{h}(\mathbf{x}, t) - z. \qquad \square$$

The constraint $\bar{h}(\mathbf{x}) \geq 0$, with $\bar{h}$ convex, is called a *reverse convex constraint*. Note that in the canonical DC representation all the difficulty has been concentrated into such a constraint, the other constraint being a convex one, while the objective function is even linear. Therefore, any technique which allows us to enclose the region defined by the reverse convex constraint into a convex region allows us to define a convex relaxation of the canonical DC problem. Details about these techniques can be found in the book by Horst and Tuy (Horst & Tuy, 1993) and in Tuy's survey (Tuy, 1995), where this subject has been extensively discussed. In terms of sets a canonical DC problem can also be written as

$$\min_{\mathbf{x}\in S\setminus int(C)} \mathbf{c}^T\mathbf{x}, \tag{4.131}$$

where

$$S = \{\mathbf{x}\in\mathbb{R}^n \ : \ \bar{g}(\mathbf{x})\leq 0\}, \quad C = \{\mathbf{x}\in\mathbb{R}^n \ : \ \bar{h}(\mathbf{x})\leq 0\}$$

are convex sets. Now, let us assume that the regularity condition

$$\min_{\mathbf{x}\in S\setminus int(C)} \mathbf{c}^T\mathbf{x} = \inf_{\mathbf{x}\in S\setminus C} \mathbf{c}^T\mathbf{x}$$

is satisfied (so that (4.131) has an optimal solution lying at the border of $S\setminus C$). In (Yamada, Tanino, & Inuiguchi, 2000) it is observed that, when C is a polytope, then the feasible region of the problem (4.131) is the union of a finite number of convex sets (one for each half-space defining C), so that the problem can be solved by minimizing $\mathbf{c}^T\mathbf{x}$ over each one of these convex sets. When C is not a polytope, in (Yamada et al., 2000) a method is proposed based on polyhedral inner approximations of the set C. In (Bigi, Frangioni, & Zhang, 2010) it is observed that, by introducing the polar set of C

$$C^* = \{\mathbf{z}\in\mathbb{R}^n \ : \ \mathbf{z}^T\mathbf{x}\leq 1 \quad \forall\, \mathbf{x}\in C\},$$

the problem (4.131) can be further rewritten as

$$\begin{aligned} \min \quad & \mathbf{c}^T\mathbf{x} \\ & \mathbf{x}\in S,\ \mathbf{z}\in C^*, \\ & \mathbf{z}^T\mathbf{x}\geq 1. \end{aligned}$$

Based on the observation that γ is an optimal value for the problem if and only if the problem

$$\begin{aligned} \max \quad & \mathbf{z}^T\mathbf{x}-1 \\ & \mathbf{x}\in S,\ \mathbf{z}\in C^*, \\ & \mathbf{c}^T\mathbf{x}\leq \gamma \end{aligned}$$

has nonpositive optimal value, in (Bigi et al., 2010) different algorithms are proposed based on polyhedral outer approximations of the sets S and C^*.

Given a function f, a DC decomposition is not unique: if $f = g-h$, g,h convex, then for any convex function p also $(g+p),(h+p)$ define a DC decomposition for f. We might be interested in finding “optimal” (in some sense) DC decompositions. For polynomial functions, this issue has been dealt with in (Ferrer & Martinez-Legaz, 2009). Here we follow (Bomze & Locatelli, 2004) and introduce the following definition.

Definition 4.84. *We say that $f = g-h$ is an* undominated DC *decomposition for f if there is no other DC decomposition $\bar{g},\bar{h}$ for f such that*

$$g = \bar{g}+p, \quad h = \bar{h}+p,$$

for some nonconstant convex function p.

In order to clarify the importance of undominated DC decompositions, consider the following simple example.

Example 4.85. Let

$$f(x) = x^3 - x^2, \quad X = [0,1].$$

Then,

$$g_t(x) = x^3 + tx^2, \; h_t(x) = (t+1)x^2, \quad t \geq 0,$$

define an infinite class of DC decompositions for f over X, all dominated by the decomposition with $t = 0$. Assume that we want to find a convex underestimator for f over X. Then, we can take

$$g_t(x) - (t+1)x,$$

which has been obtained by replacing the concave function $-h_t$ with its convex envelope over $[0,1]$. Now, if we consider the maximum distance between f and its convex underestimator over X we have that this is always attained at $x = \frac{1}{2}$ and is equal to

$$\max_{x\in[0,1]} f(x) - [g_t(x) - (t+1)x] = \frac{1}{4}(1+t).$$

Therefore, the maximum distance is minimized for $t = 0$. ■

The above simple example shows that the interest for undominated DC decompositions lies in the fact that they allow us to deliver better bounds. For general DC functions it might be difficult to recognize undominated DC decompositions. However, such undominated decompositions can be found in the case of quadratic functions as shown in (Bomze & Locatelli, 2004). Consider the quadratic form $f(\mathbf{x}) = \mathbf{x}^T\mathbf{A}\mathbf{x}$. In this case a DC decomposition is the following:

$$g(\mathbf{x}) = \mathbf{x}^T\mathbf{B}\mathbf{x}, \quad h(\mathbf{x}) = \mathbf{x}^T(\mathbf{B}-\mathbf{A})\mathbf{x}, \quad \mathbf{B}, \; \mathbf{B}-\mathbf{A} \in \mathcal{P}_n.$$

Then, we have the following definition.

Definition 4.86. *Let* $\mathbf{B}$ *and* $\mathbf{B}'$ *define two different DC decompositions of* $\mathbf{A}$*, i.e.,*

$$\mathbf{B}, \mathbf{B}', \mathbf{B}-\mathbf{A}, \mathbf{B}'-\mathbf{A} \in \mathcal{P}_n.$$

Then, $\mathbf{B}$ *is said to* dominate $\mathbf{B}'$ *if* $\mathbf{B}'-\mathbf{B} \in \mathcal{P}_n$*. A DC decomposition* $\mathbf{B}$ *of* $\mathbf{A}$ *is said to be* undominated *if no different DC decomposition of* $\mathbf{A}$ *dominates* $\mathbf{B}$.

The following theorem characterizes undominated DC decompositions.

Theorem 4.87. $\mathbf{B}$ *is an undominated DC decomposition of* $\mathbf{A}$ *if and only if*

$$\text{kernel}(\mathbf{B}) + \text{kernel}(\mathbf{B}-\mathbf{A}) = \mathbb{R}^n, \tag{4.132}$$

where

$$\text{kernel}(\mathbf{C}) = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{C}\mathbf{x} = \mathbf{0}\}$$

denotes the kernel of an $n \times n$ *matrix* $\mathbf{C}$.

Proof. Let $\mathbf{B}$ define an undominated DC decomposition and assume, by contradiction, that kernel($\mathbf{B}$) $\cup$ kernel($\mathbf{B}-\mathbf{A}$) does not generate the whole space $\mathbb{R}^n$. Let $\mathcal{H}$ be the space generated by kernel($\mathbf{B}$) $\cup$ kernel($\mathbf{B}-\mathbf{A}$) and $\mathcal{H}^\perp$ be its orthogonal complement. Let $\mathbf{y} \in \mathcal{H}^\perp$. Let us consider the matrix

$$\mathbf{B}-\varepsilon\mathbf{y}\mathbf{y}^T$$

for $\varepsilon > 0$. First we notice that for all $\mathbf{w} \in$ kernel($\mathbf{B}$)

$$(\mathbf{B}-\varepsilon\mathbf{y}\mathbf{y}^T)\mathbf{w} = \mathbf{B}\mathbf{w} - \varepsilon\mathbf{y}(\mathbf{y}^T\mathbf{w}) = \mathbf{0}$$

for all $\varepsilon \in \mathbb{R}$, i.e., the vectors $\mathbf{w} \in$ kernel($\mathbf{B}$) are still eigenvectors of $\mathbf{B}-\varepsilon\mathbf{y}\mathbf{y}^T$ related to null eigenvalues. All the other eigenvectors of $\mathbf{B}-\varepsilon\mathbf{y}\mathbf{y}^T$ may be perturbed with respect to those of $\mathbf{B}$, but since they are perturbations of eigenvectors of $\mathbf{B}$ related to positive eigenvalues, they are also, for ε small enough, still related to positive eigenvalues. Therefore, for ε small enough

$$\mathbf{B}-\varepsilon\mathbf{y}\mathbf{y}^T \in \mathcal{P}_n.$$

In a completely analogous way it can be seen that, for ε small enough

$$\mathbf{B}-\varepsilon\mathbf{y}\mathbf{y}^T - \mathbf{A} \in \mathcal{P}_n.$$

But then we have that for ε small enough $\mathbf{B}-\varepsilon\mathbf{y}\mathbf{y}^T$ is a DC decomposition which dominates $\mathbf{B}$, which is a contradiction.

To prove the reverse, we use the following simple fact: if $\mathbf{B}-\mathbf{B}' \in \mathcal{P}_n$, then kernel($\mathbf{B}$) $\subseteq$ kernel($\mathbf{B}'$). This is obvious by looking at the quadratic form $\mathbf{x}^T\mathbf{B}'\mathbf{x}$ which vanishes if and only if $\mathbf{B}'\mathbf{x} = \mathbf{0}$. Now assume that $\mathbf{B}'$ is a DC decomposition of $\mathbf{A}$ with $\mathbf{B}-\mathbf{B}' \in \mathcal{P}_n$. Then we know that kernel($\mathbf{B}$) $\subseteq$ kernel($\mathbf{B}'$) but also kernel($\mathbf{B}-\mathbf{A}$) $\subseteq$ kernel($\mathbf{B}'-\mathbf{A}$) by the same argument, so that the given assumption implies kernel($\mathbf{B}$) = kernel($\mathbf{B}'$) and also kernel($\mathbf{B}-\mathbf{A}$) = kernel($\mathbf{B}'-\mathbf{A}$). Now decompose an arbitrary $\mathbf{x} \in \mathbb{R}^n$ into $\mathbf{x} = \mathbf{v}+\mathbf{w}$ with $\mathbf{v} \in$ kernel($\mathbf{B}$) = kernel($\mathbf{B}'$) and $\mathbf{w} \in$ kernel($\mathbf{B}-\mathbf{A}$) = kernel($\mathbf{B}'-\mathbf{A}$). We conclude $(\mathbf{B}-\mathbf{B}')\mathbf{x} = \mathbf{0} + (\mathbf{B}-\mathbf{B}')\mathbf{w} = \mathbf{A}\mathbf{w} - \mathbf{A}\mathbf{w} = \mathbf{0}$. Therefore, $\mathbf{B} = \mathbf{B}'$, from which the result follows. □

The above theorem suggests a (polynomial) procedure to obtain an undominated matrix starting from a generic DC decomposition $\mathbf{B}$. The procedure is as follows:

Step 0. Let $\mathbf{B}$ be a symmetric $n \times n$ matrix such that $\mathbf{B}-\mathbf{A} \in \mathcal{P}_n$ and $\mathbf{B} \in \mathcal{P}_n$.

Step 1. Compute kernel($\mathbf{B}$) and kernel($\mathbf{B}-\mathbf{A}$).

Step 2. Check whether kernel($\mathbf{B}$) $\cup$ kernel($\mathbf{B}-\mathbf{A}$) generates the whole space $\mathbb{R}^n$. If yes, then return $\mathbf{B}$. Otherwise, let $\mathbf{y}$ be a nonzero vector in the orthogonal complement of kernel($\mathbf{B}$) $\cup$ kernel($\mathbf{B}-\mathbf{A}$).

Step 3. Set

$$\mathbf{B} = \mathbf{B} - \gamma\mathbf{y}\mathbf{y}^T,$$

where

$$\gamma = \max\left\{\varepsilon : \ \mathbf{B}-\varepsilon\mathbf{y}\mathbf{y}^T - \mathbf{A},\ \mathbf{B}-\varepsilon\mathbf{y}\mathbf{y}^T \in \mathcal{P}_n\right\},$$

and go back to Step 1.

The correctness of the procedure is proven in the following theorem.

Theorem 4.88. *At each iteration, the choice of γ in Step* 3 *above is such that the dimension of the space generated by* kernel($\mathbf{B}$) $\cup$ kernel($\mathbf{B}-\mathbf{A}$) *increases by at least one. Then, in at most n iterations the procedure delivers an undominated matrix.*

Proof. Denote by $\lambda_i(\varepsilon) > 0$ ($i \in \{n-s,\dots,n\}$) the eigenvalues related to the eigenvectors of $\mathbf{B}-\varepsilon\mathbf{y}\mathbf{y}^T$ orthogonal to kernel($\mathbf{B}$). Similarly, denote by $\mu_i(\varepsilon) > 0$ ($i \in \{n-t,\dots,n\}$) the eigenvalues related to the eigenvectors of $(\mathbf{B}-\mathbf{A})-\varepsilon\mathbf{y}\mathbf{y}^T$ orthogonal to kernel($\mathbf{B}-\mathbf{A}$). Next, denote by

$$\rho(\varepsilon) = \min\{\min\{\lambda_i(\varepsilon) : n-s \le i \le n\}, \min\{\mu_i(\varepsilon) : n-t \le i \le n\}\}$$

the minimum of all the previously mentioned eigenvalues. Now, continuous dependence of λ_i and μ_i upon ε implies that $\rho(\varepsilon)$ is a continuous function of ε. Moreover, ρ is a nonincreasing function because the smallest eigenvalue of $\mathbf{B}-\varepsilon\mathbf{y}\mathbf{y}^T$ over the orthogonal complement of kernel($\mathbf{B}$) and the smallest eigenvalue of $(\mathbf{B}-\mathbf{A})-\varepsilon\mathbf{y}\mathbf{y}^T$ over the orthogonal complement of kernel($\mathbf{B}-\mathbf{A}$) are both nonincreasing functions of ε. Finally, we have that $\rho(\varepsilon) \to -\infty$ as $\varepsilon \to \infty$, so that the value γ in Step 3 is well defined and at least one additional eigenvalue (counting multiplicities) of either $\mathbf{B}-\gamma\mathbf{y}\mathbf{y}^T$ or $(\mathbf{B}-\mathbf{A})-\gamma\mathbf{y}\mathbf{y}^T$ must be zero, with an eigenvector orthogonal to kernel($\mathbf{B}$) or kernel($\mathbf{B}-\mathbf{A}$). Thus the dimension of the space generated by the kernels increases, and the result follows. □

Up to now we have stayed within the quadratic world during our search for undominated DC decompositions for a quadratic form. In fact, this is justified by the following proposition.

Proposition 4.89. *Suppose that* $f(\mathbf{x}) = \mathbf{x}^T\mathbf{A}\mathbf{x}$ *is a quadratic form having the DC decomposition* $f = g - h$ *with two convex, twice-continuously differentiable functions g and h. If this DC decomposition is undominated, then necessarily g and h must be quadratic functions themselves.*

Proof. Take any undominated DC decomposition $\mathbf{B}$ into quadratic functions and suppose that g and h are undominated, too. Then, by definition, $\mathbf{x}^T\mathbf{B}\mathbf{x} - g(\mathbf{x}) = \mathbf{x}^T(\mathbf{B}-\mathbf{A})\mathbf{x} - h(\mathbf{x})$ is a convex function, which means $\mathbf{B} - \nabla^2 g(\mathbf{x}) \in \mathcal{P}_n$ and also $\mathbf{B}-\mathbf{A}-\nabla^2 h(\mathbf{x}) \in \mathcal{P}_n$ for all $\mathbf{x}$. For any fixed $\mathbf{x}$, Definition 4.86 gives therefore $\mathbf{B} = \nabla^2 g(\mathbf{x})$ and $\mathbf{B}-\mathbf{A} = \nabla^2 h(\mathbf{x})$, so that g and h differ from the quadratic forms given by $\mathbf{B}$ and $\mathbf{B}-\mathbf{A}$, respectively, only by affine functions. □

As an example of undominated DC decomposition we can consider the *spectral DC decomposition*. For a symmetric $n \times n$ matrix $\mathbf{A}$, define the spectral DC decomposition $\mathbf{A}_+$ as follows. Suppose $\mathbf{A} = \mathbf{W}\mathbf{L}\mathbf{W}^T$, where $\mathbf{W}$ contains an orthonormal basis of eigenvectors of $\mathbf{A}$ as its columns, and $\mathbf{L}$ is a diagonal matrix formed of the respective ordered eigenvalues $\lambda_i(\mathbf{A})$. Then form $\mathbf{L}_+$ as a diagonal matrix whose diagonal elements are $\max\{\lambda_i(\mathbf{A}), 0\}$, and put $\mathbf{A}_+ = \mathbf{W}\mathbf{L}_+\mathbf{W}^T$. Then $\mathbf{A}_+$ is an undominated DC decomposition for $\mathbf{A}$ which dominates all DC decompositions $\mathbf{B}$ with $\mathbf{B}\mathbf{A} = \mathbf{A}\mathbf{B}$ (see also (Bomze, 2002)).

In (Bomze & Locatelli, 2004) it has been proven that a matrix $\mathbf{A}$ is indefinite if and only if it admits an infinite number of undominated DC decompositions. Therefore, even after restricting to undominated DC decompositions, we are still left with the question about the best possible decomposition, i.e., the decomposition which delivers the best possible bound. This question is in general a difficult one. However, Anstreicher and Burer (Anstreicher & Burer, 2005) gave an answer for StQPs, i.e., for QP problems whose feasible region is the unit simplex Δ_n (see Section 2.5). For a given DC decomposition $\mathbf{B}$ of some matrix $\mathbf{A}$, we have

$$\min_{\mathbf{x}\in\Delta_n} \mathbf{x}^T\mathbf{A}\mathbf{x} \geq \min_{\mathbf{x}\in\Delta_n} \mathbf{x}^T\mathbf{B}\mathbf{x} + \min_{\mathbf{x}\in\Delta_n} -\mathbf{x}^T(\mathbf{B}-\mathbf{A})\mathbf{x} = \min_{\mathbf{x}\in\Delta_n} \mathbf{x}^T\mathbf{B}\mathbf{x} - \max_{i=1,\ldots,n} (B_{ii}-A_{ii}). \quad (4.133)$$

Since $\mathbf{B} \in \mathcal{P}_n$, we have that

$$\begin{aligned} \min\nolimits_{\mathbf{x}\in\Delta_n} \mathbf{x}^T\mathbf{B}\mathbf{x} = \min \quad & \mathbf{B}\bullet\mathbf{X} \\ & \begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \mathbf{X} \end{pmatrix} \in \mathcal{P}_{n+1}, \\ & \mathbf{x}\in\Delta_n, \end{aligned}$$

where the right-hand side is the Shor relaxation. Taking the dual of the problem on the right-hand side, we end up with

$$\begin{aligned} \max \quad & \mu-\sigma \\ & \begin{pmatrix} \sigma & \mathbf{s}^T \\ \mathbf{s} & \mathbf{B} \end{pmatrix} \in \mathcal{P}_{n+1}, \\ & 2\mathbf{s}+\mu\mathbf{e} \leq \mathbf{0}. \end{aligned}$$

Then, taking into account (4.133), the optimal DC bound can be obtained by solving the following SDP problem:

$$\begin{aligned} \sup \quad & \mu-\sigma-\theta \\ & \begin{pmatrix} \sigma & \mathbf{s}^T \\ \mathbf{s} & \mathbf{B} \end{pmatrix} \in \mathcal{P}_{n+1}, \\ & 2\mathbf{s}+\mu\mathbf{e} \leq \mathbf{0}, \\ & \mathbf{B}-\mathbf{A} \in \mathcal{P}_n, \\ & \theta\mathbf{e} \geq diag(\mathbf{B}-\mathbf{A}), \end{aligned}$$

where the last constraint imposes that $\theta \geq \max_{i=1,\ldots,n} B_{ii}-A_{ii}$. The dual of this problem is

$$\begin{aligned} \min \quad & \mathbf{A}\bullet\mathbf{X} \\ & \begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \mathbf{X} \end{pmatrix} \in \mathcal{P}_{n+1}, \\ & Diag(\mathbf{y})-\mathbf{X} \in \mathcal{P}_n, \\ & \mathbf{x},\mathbf{y}\in\Delta_n. \end{aligned}$$

Note that for feasible solutions ($\mathbf{X}$ $\mathbf{x}$ $\mathbf{y}$) we have

$$1 = \mathbf{e}^T(\mathbf{x}\mathbf{x}^T)\mathbf{e} \leq \mathbf{e}^T\mathbf{X}\mathbf{e} \leq \mathbf{e}^T Diag(\mathbf{y})\mathbf{e} = 1,$$

so that

$$\mathbf{e}^T(\mathbf{X}-\mathbf{x}\mathbf{x}^T)\mathbf{e} = \mathbf{e}^T(Diag(\mathbf{y})-\mathbf{X})\mathbf{e} = 0, \tag{4.134}$$

and, consequently, $\mathbf{Xe} = \mathbf{x} = \mathbf{y}$ (recall that $\mathbf{A} \in \mathcal{P}_n$ and $\mathbf{y}^T\mathbf{A}\mathbf{y} = 0$ imply that $\mathbf{A}\mathbf{y} = \mathbf{0}$). Then, the dual problem can be further simplified to

$$\begin{aligned} \min \quad & \mathbf{A} \bullet \mathbf{X} \\ & \begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \mathbf{X} \end{pmatrix} \in \mathcal{P}_{n+1}, \\ & Diag(\mathbf{x}) - \mathbf{X} \in \mathcal{P}_n, \\ & \mathbf{x} \in \Delta_n. \end{aligned} \tag{4.135}$$

In (Anstreicher & Burer, 2005) it is further proven that the problem above has the same optimal value as

$$\begin{aligned} \min \quad & \mathbf{A} \bullet \mathbf{X} \\ & \mathbf{E} \bullet \mathbf{X} = 1, \\ & \mathbf{Xe} \geq \mathbf{0}, \\ & Diag(\mathbf{Xe}) - \mathbf{X} \in \mathcal{P}_n, \\ & \mathbf{X} \in \mathcal{P}_n. \end{aligned} \tag{4.136}$$

Indeed, the following proposition is proven.

Proposition 4.90. $(\mathbf{X}\ \mathbf{x})$ *is feasible for* (4.135) *if and only if* $\mathbf{x} = \mathbf{Xe}$ *and* $\mathbf{X}$ *is feasible for* (4.136).

Proof. If $(\mathbf{X}\ \mathbf{x})$ is feasible for (4.135), then $\mathbf{Xe} = \mathbf{x}$ follows from (4.134), so that $\mathbf{X}$ is feasible for (4.136). Conversely, if $\mathbf{X}$ is feasible for (4.136) and $\mathbf{Xe} = \mathbf{x}$, then $\mathbf{x} \in \Delta_n$ and $Diag(\mathbf{x}) - \mathbf{X} \in \mathcal{P}_n$. Moreover,

$$\begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \mathbf{X} \end{pmatrix} = \begin{pmatrix} \mathbf{e}^T \\ \mathbf{I} \end{pmatrix} \mathbf{X}(\mathbf{e}\ \ \mathbf{I}) \in \mathcal{P}_{n+1},$$

in view of $\mathbf{X} \in \mathcal{P}_n$, which concludes the proof. □

The reformulation (4.136) allows an easy comparison with another bound for StQP, the copositive bound:

$$\begin{aligned} \min \quad & \mathbf{A} \bullet \mathbf{X} \\ & \mathbf{E} \bullet \mathbf{X} = 1, \\ & \mathbf{X} \geq \mathbf{O}, \\ & \mathbf{X} \in \mathcal{P}_n. \end{aligned} \tag{4.137}$$

This is obtained by substituting the constraint $\mathbf{X} \in \mathcal{C}_n^*$ in (4.51) with the constraints $\mathbf{X} \geq \mathbf{O}$, $\mathbf{X} \in \mathcal{P}_n$, or, equivalently, $\mathbf{X} \in \mathcal{DNN}_n$. Indeed, each matrix satisfying these constraints is obviously copositive (in fact, these constraints generate the dual cone of $\mathcal{K}_n^0 = \mathcal{P}_n + \mathcal{N}_n$ which is the first cone of the hierarchy $\{\mathcal{K}_n^r\}_{r\in\mathbb{N}_0}$ presented in Section 4.4.1). The superiority of the copositive bound follows from the fact that each feasible solution of (4.137) is also feasible for (4.136). Strict superiority holds. Indeed, Anstreicher and Burer show that for the matrix

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{4.138}$$

the best DC bound is equal to $-\frac{1}{8}$, while the copositive bound is equal to 0 (equal to the optimal value of the StQP problem in this case). An alternative proof of this dominance result can also be found in (Bomze, Locatelli, & Tardella, 2008), where the class of decompositions

$$\mathbf{x}^T\mathbf{A}\mathbf{x} = \mathbf{x}^T\mathbf{B}\mathbf{x} + \mathbf{x}^T(\mathbf{A}-\mathbf{B})\mathbf{x}$$

is considered with

- $\mathbf{B} \in \mathcal{P}_n$;
- $\mathbf{x}^T(\mathbf{A}-\mathbf{B})\mathbf{x}$ attaining its minimum value at a vertex of the unit simplex, or, equivalently,

$$\min_{i,j=1,\dots,n} (A_{ij} - B_{ij}) = \min_{i=1,\dots,n} (A_{ii} - B_{ii}).$$

Obviously, this class contains the class of DC decompositions (strictly, as shown by the decomposition obtained by taking $\mathbf{B} = \mathbf{O}$ in the example (4.138)), and in (Bomze et al., 2008) it is shown that the best possible bound for this class of decompositions is exactly the copositive bound.

A way to detect the best possible DC decomposition within a class of DC decompositions for QCQP problems (4.104) with $E = \emptyset$ and $\mathbf{Q}_i \in \mathcal{P}_n$, $i \in I$, has been discussed in (Zheng et al., 2011b). The results discussed in that paper can be viewed as a special case of those presented in (Zheng et al., 2011a) (and already discussed in Section 4.4.4) but are briefly reviewed here. The following DC decompositions for the quadratic objective function of (4.104) are considered:

$$\mathbf{x}^T\mathbf{Q}_0\mathbf{x} = \mathbf{x}^T\left(\mathbf{Q}_0 + \sum_{i=1}^{s}\lambda_i\mathbf{v}_i\mathbf{v}_i^T + \sum_{i\in I}\mu_i\mathbf{Q}_i\right)\mathbf{x} - \sum_{i=1}^{s}\lambda_i(\mathbf{v}_i^T\mathbf{x})^2 - \sum_{i\in I}\mu_i\mathbf{x}^T\mathbf{Q}_i\mathbf{x}, \tag{4.139}$$

with $\lambda_i \geq 0$, $i = 1,\dots,s$, $\mu_i \geq 0$, $i \in I$, and

$$\mathbf{Q}_0 + \sum_{i=1}^{s}\lambda_i\mathbf{v}_i\mathbf{v}_i^T + \sum_{i\in I}\mu_i\mathbf{Q}_i \in \mathcal{P}_n.$$

Given the lower bound l_i and the upper bound u_i for $\mathbf{v}_i^T\mathbf{x}$ over the feasible region S, we have that for any $\mathbf{x} \in S$

$$-(\mathbf{v}_i^T\mathbf{x})^2 \geq \ell_i^1(\mathbf{x}) := l_iu_i - (l_i + u_i)\mathbf{v}_i^T\mathbf{x}.$$

Next, if $rank(\mathbf{Q}_i) = r_i$, consider the decomposition

$$\mathbf{Q}_i = \sum_{j=1}^{r_i} \mathbf{w}_{ij}\mathbf{w}_{ij}^T.$$

Let l_{ij} and u_{ij} be, respectively, a lower and an upper bound for $\mathbf{w}_{ij}^T\mathbf{x}$ over S, and let

$$\ell_{ij}^2(\mathbf{x}) := l_{ij}u_{ij} - (l_{ij} + u_{ij})\mathbf{w}_{ij}^T\mathbf{x}.$$

Then, for any $\mathbf{x} \in S$

$$-\mathbf{x}^T\mathbf{Q}_i\mathbf{x} \geq \max\left\{\mathbf{c}_i^T\mathbf{x} + d_i, \sum_{j=1}^{r_i} \ell_{ij}^2(\mathbf{x})\right\}.$$

Therefore, the problem

$$\begin{aligned} v(\boldsymbol{\lambda},\boldsymbol{\mu}) = \min \quad & \mathbf{x}^T(\mathbf{Q}_0 + \textstyle\sum_{i=1}^s \lambda_i\mathbf{v}_i\mathbf{v}_i^T + \sum_{i\in I}\mu_i\mathbf{Q}_i)\mathbf{x} + \sum_{i=1}^s \lambda_i\ell_i^1(\mathbf{x}) \\ & + \textstyle\sum_{i\in I}\mu_i \max\left\{\mathbf{c}_i^T\mathbf{x} + d_i, \sum_{j=1}^{r_i}\ell_{ij}^2(\mathbf{x})\right\}, \\ & \mathbf{x}^T\mathbf{Q}_i\mathbf{x} + \mathbf{c}_i^T\mathbf{x} + d_i \leq 0, \qquad\qquad i \in I, \\ & \mathbf{x} \in P, \end{aligned}$$

is a convex relaxation for (4.104) when $E = \emptyset$ and $\mathbf{Q}_i \in \mathcal{P}_n$, $i \in I$. If we want to find the best possible DC decomposition among those in (4.139), we need to solve the following problem:

$$\begin{aligned} \max \quad & v(\boldsymbol{\lambda},\boldsymbol{\mu}) \\ & \mathbf{Q}_0 + \textstyle\sum_{i=1}^s \lambda_i\mathbf{v}_i\mathbf{v}_i^T + \sum_{i\in I}\mu_i\mathbf{Q}_i \in \mathcal{P}_n, \\ & \boldsymbol{\lambda} \geq \mathbf{0},\ \boldsymbol{\mu} \geq \mathbf{0}. \end{aligned}$$

In (Zheng et al., 2011b) it is proven that such a problem can be reformulated as a semidefinite one (see (4.110)).

4.8 Monotonic optimization

In this section we consider *increasing functions*, whose definition is the following.

Definition 4.91. *A function f is* increasing *over a box* $[\mathbf{a},\mathbf{b}] \subset \mathbb{R}_+^n$ *if*

$$\mathbf{x},\mathbf{x}' \in [\mathbf{a},\mathbf{b}],\ \mathbf{x}' \geq \mathbf{x} \quad \Rightarrow \quad f(\mathbf{x}') \geq f(\mathbf{x}).$$

It is easy to see that any nonnegative linear combination of increasing functions is an increasing function and that the pointwise supremum and infimum of a family of increasing functions is an increasing function. The notion of increasing function is strictly related to that of *normal set*.

Definition 4.92. *A set $X \subseteq \mathbb{R}^n_+$ is* normal *if*

$$\mathbf{x}' \geq \mathbf{x},\ \ \mathbf{x}' \in X \quad \Rightarrow \quad \mathbf{x} \in X.$$

We also introduce the related notion of *reverse normal set.*

Definition 4.93. *A set $X \subseteq \mathbb{R}^n_+$ is* reverse normal *if*

$$\mathbf{x}' \geq \mathbf{x},\ \ \mathbf{x} \in X \quad \Rightarrow \quad \mathbf{x}' \in X.$$

It can be seen that a normal set X is basically the lower level set of an increasing function, i.e., there exists some increasing function g such that $X = \{\mathbf{x} \in \mathbb{R}^n_+ \ :\ g(\mathbf{x}) \leq 0\}$, while any reverse normal set is the upper level set of an increasing function. Next, we introduce the notion of *difference-of-monotonic function.*

Definition 4.94. *Any function f which can be written as the difference of two increasing functions is called a* difference-of-monotonic *(DM) function.*

A general DM optimization problem is a GO problem where all the functions involved are DM, i.e.,

- the feasible region is contained in a box $X \subset \mathbb{R}^n_+$;
- the objective function is $f(\mathbf{x}) = f^1(\mathbf{x}) - f^2(\mathbf{x})$, f^1, f^2 increasing over X;
- the constraint functions are $g_i(\mathbf{x}) = g_i^1(\mathbf{x}) - g_i^2(\mathbf{x})$, g_i^1, g_i^2, $i = 1, \ldots, m$, increasing over X.

By introducing additional variables, the problem can be rewritten as

$$\begin{aligned} \min \quad & f^1(\mathbf{x}) + z \\ & z + f^2(\mathbf{x}) \geq 0, \\ & g_i^1(\mathbf{x}) + t_i \leq 0, \quad i = 1, \ldots, m, \\ & t_i + g_i^2(\mathbf{x}) \geq 0, \end{aligned}$$

and finally as

$$\min\{f(\mathbf{y}) \ :\ \mathbf{y} \in G \cap H\}, \tag{4.140}$$

where $\mathbf{y} = (\mathbf{x}\ z\ \mathbf{t})$ and

- $f(\mathbf{y}) = f^1(\mathbf{x}) + z$ is an increasing function;
- $G = \{\mathbf{y} \ :\ \max_{i=1,\ldots,m}\ g_i^1(\mathbf{x}) + t_i \leq 0\}$ is a normal set;
- $H = \{\mathbf{y} \ :\ \min\{z + f^2(\mathbf{x}), \min_{i=1,\ldots,m}\ g_i^1(\mathbf{x}) + t_i\} \geq 0\}$ is a reverse normal set.

We remark that, if it exists, an optimal solution of problem (4.140) always lies at $G \cap bd(H)$ (recall that $bd(H)$ is the border of H). In order to consider a suitable relaxation and, thus, a proper bound for DM optimization problems, we need to introduce the notion of reverse polyblock.

Definition 4.95. *Given a box* $[\mathbf{a},\mathbf{b}] \subset \mathbb{R}^n_+$ *and a finite set* $T \subset [\mathbf{a},\mathbf{b}]$*, the* reverse polyblock $P(T)$ *generated by* T *is the set*

$$\bigcup_{\mathbf{z}\in T}[\mathbf{z},\mathbf{b}];$$

i.e., it is the union of a finite set of boxes, each one defined by $\mathbf{b}$ *and by a point* $\mathbf{z} \in [\mathbf{a},\mathbf{b}]$.

Note that in $P(T)$ we can keep *proper* vertices of the reverse polyblock, i.e., points $\mathbf{z} \in T$, such that there does not exist $\mathbf{z}' \in T \setminus \{\mathbf{z}\}$ for which $\mathbf{z}' \leq \mathbf{z}$. If a reverse polyblock $P(T) \supset G \cap H$ is given, then a relaxation of (4.140) is readily obtained by substituting $G \cap H$ with $P(T)$, so that an immediate lower bound for problem (4.140) is given by

$$\min_{\mathbf{z}\in T} f(\mathbf{z}). \tag{4.141}$$

For the case where in (4.140) minimization is replaced by maximization, a similar development can be done with *polyblocks*, i.e., the finite union of boxes $[\mathbf{a},\mathbf{z}]$, $\mathbf{z} \in [\mathbf{a},\mathbf{b}]$. For more details we refer, e.g., to (Tuy, 2000; Tuy & Luc, 2000).

4.9 Lipschitz functions

The definition of *Lipschitz function* is the following.

Definition 4.96. *A function* f *is a* Lipschitz function *over a region* X *with* Lipschitz constant L *if*

$$|f(\mathbf{x}) - f(\mathbf{y})| \leq L\|\mathbf{x}-\mathbf{y}\| \;\; \forall\, \mathbf{x},\mathbf{y} \in X.$$

Continuously differentiable functions f over some compact convex region X are examples of Lipschitz functions. Indeed, by the mean value theorem, we have that for all $\mathbf{x},\mathbf{y} \in X$

$$f(\mathbf{x}) - f(\mathbf{y}) = \nabla f(\lambda\mathbf{x} + (1-\lambda)\mathbf{y})(\mathbf{x}-\mathbf{y})$$

for some $\lambda \in [0,1]$. Therefore,

$$|f(\mathbf{x}) - f(\mathbf{y})| \leq \|\nabla f(\lambda\mathbf{x} + (1-\lambda)\mathbf{y})\|\,\|(\mathbf{x}-\mathbf{y})\|,$$

and we can take the Lipschitz constant

$$L = \max_{\mathbf{x}\in X}\|\nabla f(\mathbf{x})\|.$$

The following result, which leads to the definition of a (nonconvex) underestimator for a function f over a region X, can be easily proven.

Proposition 4.97. *If a Lipschitz constant L for the function* f *is known and the function has been evaluated at some points* $\mathbf{x}_1,\ldots,\mathbf{x}_t \in X$*, then the function*

$$h_t(\mathbf{x}) = \max_{i=1,\ldots,t}\{f(\mathbf{x}_i) - L\|\mathbf{x}_i - \mathbf{x}\|\} \tag{4.142}$$

underestimates f *over* X.

Proof. By definition, for each $\mathbf{x} \in X$

$$f(\mathbf{x}) \geq f(\mathbf{x}_i) - L\|\mathbf{x}_i - \mathbf{x}\|, \quad i = 1,\ldots,t,$$

so that the result follows by taking the maximum of the right-hand sides of the above inequalities. □

In the one-dimensional case, where $X = [a,b]$ is an interval, the function h_t defined in (4.142) is a saw-tooth function whose global minimum can be relatively easily detected. This leads to the well-known Piyavskii's algorithm (Piyavskii, 1972) (independently proposed also in (Shubert, 1972)) for one-dimensional Lipschitz functions, where at the tth iteration the minimum of the function h_t over $[a,b]$ is computed and f is evaluated at this point, thus leading to an updated function h_{t+1}. In Figures 4.9–4.10 a sample run of the algorithm is shown. The figures report the objective function and the piecewise linear underestimator after 10 and 20 function evaluations, respectively. According to the algorithm, the 21st observation should be placed at the global minimum of the underestimator, i.e., at point $x = 0.594869$. Further approaches for one-dimensional functions can be found, e.g., in (Evtushenko, 1971; Hansen, Jaumard, & Lu, 1992; Schoen, 1982; Sergeyev, Famularo, & Pugliese, 2001; Timonov, 1977). In the multidimensional case, the minimum of the function h_t, even over a box X, is not easy to detect. Indeed, the function h_t has a large number of local minimizers (the same is true for the one-dimensional case, but in this case there is an efficient way to detect the global minimizer by inspection of subintervals). Different ways to extend Piyavskii's algorithm to the multidimensional case have been proposed (see, e.g., (Mayne & Polak, 1984; Mladineo, 1986)). In (Strongin, 1973) it was proposed to

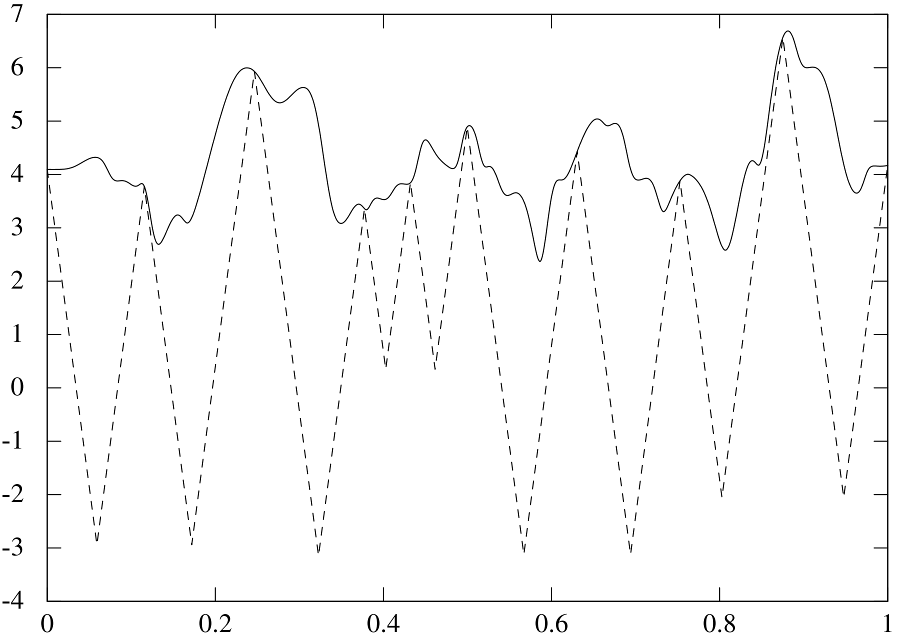


Figure 4.9. *An example of Piyavskii–Shubert algorithm, after* 10 *function evaluations*

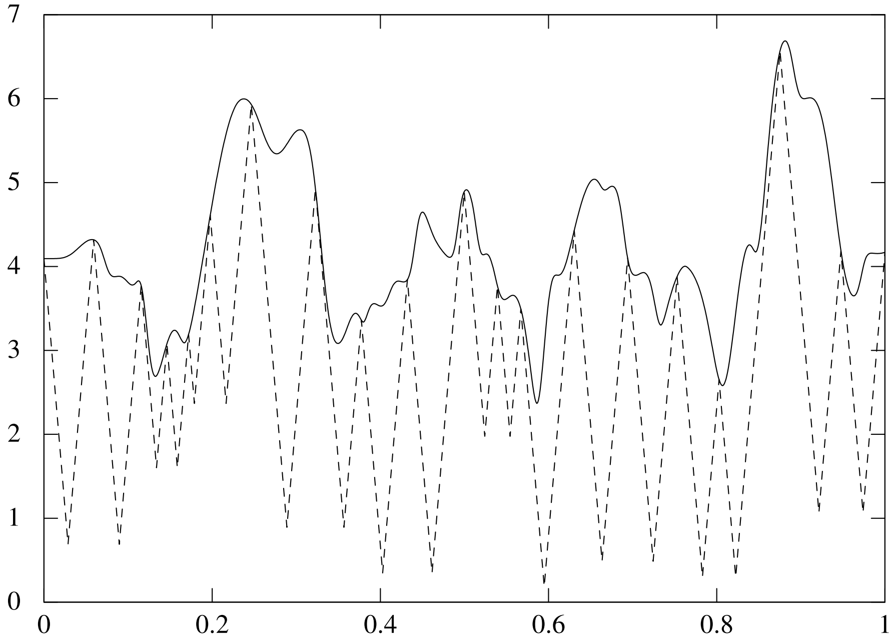


Figure 4.10. *An example of Piyavskii–Shubert algorithm, after* 20 *function evaluations*

reformulate multidimensional problems as one-dimensional ones by using a space-filling curve to approximate X. In particular, the Peano curve is employed. However, this approach has the relevant drawback that points which are close within X become far along the curve, so that the resulting one-dimensional function is quite oscillating with a large number of local minimizers. In (Wood, 1991) an algorithm based on simplicial subdivisions (see Section 5.3.2) is proposed. The approach is closely related to BB methods (see Chapter 5). A further approach based on simplicial partitions can be found in (Clausen & Žilinskas, 2002). BB algorithms based on rectangular partitions (see Section 5.3.1) are given, e.g., in (Galperin, 1985, 1988; Gourdin, Hansen, & Jaumard, 1994; Pinter, 1986). It is easily seen that, for a given box B with diameter $diam(B)$, if a point $\mathbf{y} \in B$ is available, a lower bound for f over B is $f(\mathbf{y}) - L diam(B)$, which can be modified to $f(\mathbf{y}) - \frac{L}{2} diam(B)$ if $\mathbf{y}$ is the center of the box. If f has been evaluated at all the 2^n vertices $\mathbf{v}^i$ of the box, then in (Meewella & Mayne, 1988) a lower bound over the box $B = \prod_{j=1}^{n}[a_j, b_j]$ is obtained by solving the following linear program:

$$\begin{aligned} \min \quad & z \\ & z \geq f(\mathbf{v}^i) - L\left[\textstyle\sum_{j:\ v^i_j = b_j}(b_j - x_j) + \sum_{j:\ v^i_j = a_j}(x_j - a_j)\right], \quad i = 1, \ldots, 2^n, \\ & \mathbf{x} \in B. \end{aligned}$$

This follows from the fact that for $\mathbf{x} \in B$

$$\sum_{j:\ v^i_j = b_j}(b_j - x_j) + \sum_{j:\ v^i_j = a_j}(x_j - a_j) \geq \|\mathbf{x} - \mathbf{v}^i\|.$$

A relevant point, observed in (Sergeyev, 1995) and then further developed in subsequent works, e.g., (Molinaro, Pizzuti, & Sergeyev, 2001; Sergeyev & Kvasov, 2006), is that the use of *local* Lipschitz constants may considerably improve the performance. If we just consider a portion X' of the initial feasible region X, then using a *global* (e.g., valid over all X) Lipschitz constant L is usually quite inefficient with respect to using a *local* (i.e., only valid over X') Lipschitz constant $L' \leq L$. This is an obvious consequence of the fact that the lower the Lipschitz constant is, the closer the underestimating function is to the original function f. In the approach proposed in (Sergeyev & Kvasov, 2006), at each iteration the original box is subdivided into subboxes. For each subbox, different lower bounds, corresponding to different estimates of the Lipschitz constant, are computed considering only the diagonals of the subboxes. Then, only undominated subboxes, i.e., subboxes with the lowest lower bound for at least one estimate of the Lipschitz constant, are taken under consideration for a further partition into subboxes.

4.10 Interval arithmetic

Interval arithmetic is a further source of lower bounds for nonconvex problems. Typical ingredients of interval arithmetic are the usual operations ($+$, $-$, $*$, $/$) and so-called inclusion functions.

Definition 4.98. *Let $B_j = \prod_{i=1}^{n}[\ell_i^j, u_i^j]$, $j = 1,2$, be two given boxes. Then, the following operations are defined.*

Sum:

$$B_1 + B_2 = \prod_{i=1}^{n}[\ell_i^1 + \ell_i^2, u_i^1 + u_i^2].$$

Difference:

$$B_1 - B_2 = \prod_{i=1}^{n}[\ell_i^1 - u_i^2, u_i^1 - \ell_i^2].$$

Product:

$$B_1 * B_2 = \prod_{i=1}^{n}[\min\{\ell_i^1\ell_i^2, \ell_i^1 u_i^2, u_i^1\ell_i^2, u_i^1 u_i^2\}, \max\{\ell_i^1\ell_i^2, \ell_i^1 u_i^2, u_i^1\ell_i^2, u_i^1 u_i^2\}].$$

Division:

$$B_1/B_2 = \prod_{i=1}^{n}\left[\min\left\{\ell_i^1/\ell_i^2, \ell_i^1/u_i^2, u_i^1/\ell_i^2, u_i^1/u_i^2\right\}, \max\left\{\ell_i^1/\ell_i^2, \ell_i^1/u_i^2, u_i^1/\ell_i^2, u_i^1/u_i^2\right\}\right]$$

(defined only if $\mathbf{0} \notin B_2$).

Note that difference and division are not simply the inverse operations, respectively, of sum and product. For instance,

$$[0,2] + [-1,1] = [-1,3] \quad \text{but} \quad [-1,3] - [-1,1] = [-2,4] \neq [0,2].$$

Also note that some properties which are valid for the real number arithmetic extend to the interval arithmetic. In particular, for three given boxes B_1, B_2, B_3 we have that

$$\begin{aligned} &B_1 + B_2 = B_2 + B_1, \\ &B_1 * B_2 = B_2 * B_1, \\ &B_1 + (B_2 + B_3) = (B_1 + B_2) + B_3, \\ &B_1 * (B_2 * B_3) = (B_1 * B_2) * B_3. \end{aligned}$$

However, some other properties are not satisfied. In particular, the distributive law is not true in general, and it has to be substituted with the subdistributive law

$$B_1 * (B_2 + B_3) \subseteq B_1 * B_2 + B_1 * B_3,$$

with equality holding only in some special cases, e.g., when B_1 reduces to a single point. Now we introduce the notion of inclusion function.

Definition 4.99. *Let $\mathcal{B}_n$ denote the set of all n-dimensional boxes (in particular, $\mathcal{B}_1$ is the set of all one-dimensional intervals). A function*

$$F^f : \mathcal{B}_n \to \mathcal{B}_1$$

is called an inclusion function *for an n-dimensional function f if*

$$\forall\ B \in \mathcal{B}_n,\ \{f(\mathbf{x})\ :\ \mathbf{x} \in B\} \subseteq F^f(B).$$

The lower limit of $F^f(B)$ is denoted by $F^f_\ell(B)$, while the upper limit is denoted by $F^f_{\mathbf{u}}(B)$. The inclusion function is called isotone *if*

$$B_1 \subseteq B_2 \quad \Rightarrow \quad F^f(B_1) \subseteq F^f(B_2).$$

Good inclusion functions are available for many well-known functions, such as the trigonometric, exponential, and logarithmic functions (of course, for the latter it is given as understood that only boxes with a strictly positive lower limit are considered). Once the standard operations have been defined and a library of inclusion functions for some standard functions is available, all these can be combined to derive lower and upper bounds for more complicated functions over given boxes, as shown in the following example.

Example 4.100. Let

$$f(x) = x^2e^x - 2xe^x + e^x$$

and $B = [0,2]$. Exploiting the strict monotonicity of the exponential function, an inclusion function for the exponential one is

$$F^{e^x}([a,b]) = [e^a, e^b],$$

while for the quadratic function x^2 it is

$$F^{x^2}([a,b]) = \begin{cases} [\min\{a^2,b^2\}, \max\{a^2,b^2\}] & \text{if } ab \geq 0, \\ [0, \max\{a^2,b^2\}] & \text{otherwise.} \end{cases}$$

Given these inclusion functions, and using the definitions of the sum, difference, and product between intervals, we have that an interval whose lower and upper limits are, respectively, a lower and upper bound for f over $[0,2]$ is the following:

$$\begin{aligned}&[0,4]*[e^0,e^2]-2*[0,2]*[e^0,e^2]+[e^0,e^2]\\&\quad=[0,4e^2]-[0,4e^2]+[1,e^2]\\&\quad=[1-4e^2,5e^2]. \quad \blacksquare\end{aligned}$$

It is important to remark that different but mathematically equivalent expressions for a function may lead to different bounds. This is shown in the following example.

Example 4.101. The function f of the previous example can also be written as

$$(x-1)^2e^x.$$

Inclusion functions for the exponential and quadratic functions have already been given. Then, an interval whose lower and upper limits are, respectively, a lower and upper bound for f over $[0,2]$ is

$$[0,1]*[e^0,e^2]=[0,e^2],$$

which is much tighter than the one obtained before. $\blacksquare$

In the previous examples we obtained an inclusion function for the function f by substituting each occurrence of the variable with the corresponding interval, and each occurrence of a function within a library with its corresponding inclusion function. This is one way to derive an inclusion function for some given function f and is called *natural interval extension* of f over a given box B (Moore, 1966). An alternative is represented by *centered forms*. If f is a differentiable function, then the mean value theorem states that

$$f(\mathbf{x})=f(\mathbf{x}_0)+\nabla f(\lambda\mathbf{x}_0+(1-\lambda)\mathbf{x})(\mathbf{x}-\mathbf{x}_0)$$

for all $\mathbf{x},\mathbf{x}_0\in B$ and for some $\lambda\in[0,1]$. Consequently, if we denote by $F^{\nabla f}$ an inclusion function for the gradient ∇f, we end up with so-called *mean value form function* (Moore, 1966)

$$f(\mathbf{x}_0)+F^{\nabla f}(B)(B-\mathbf{x}_0),$$

which is an inclusion function for f.

Example 4.102. The first derivative of the example is

$$f'(x)=(x^2-1)e^x.$$

Then, taking $x_0=1$ (the midpoint of the interval), the mean value form function is equal to

$$0+([-1,3]*[e^0,e^2])*[-1,1]=[-3e^2,3e^2]. \quad \blacksquare$$

Note that the choice of the point $\mathbf{x}_0\in B$ is arbitrary and may lead to different results. In the example, if we take $x_0=0$ we end up with

$$1+([-1,3]*[e^0,e^2])*[0,2]=[1-2e^2,1+6e^2].$$

The midpoint of the box is a usual choice, but some strategy to select an optimal point has been proposed, e.g., in (Baumann, 1988). Strictly related to the mean value form is the *linear boundary value form* (see, e.g., (Neumaier, 1990)): for a one-dimensional function f, an inclusion function over an interval $[a,b]$ is

$$[\min_{x\in[a,b]} \max\{f(a)+F_\ell^{f'}(x-a), f(b)+F_u^{f'}(x-b)\},$$

$$\max_{x\in[a,b]} \min\{f(a)+F_u^{f'}(x-a), f(b)+F_\ell^{f'}(x-b)\}].$$

In (Lagouanelle, Csendes, & Vinkó, 2004) the optimal form proposed in (Baumann, 1988) and the linear boundary value form are simultaneously employed, thus deriving an inclusion function for one-dimensional functions which is always at least as good as both forms. For twice-differentiable functions one can also define the so-called *Taylor form function* based on the equality

$$f(\mathbf{x}) = f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0)^T(\mathbf{x}-\mathbf{x}_0) + \frac{1}{2}(\mathbf{x}-\mathbf{x}_0)^T\nabla^2 f(\lambda\mathbf{x}_0 + (1-\lambda)\mathbf{x})(\mathbf{x}-\mathbf{x}_0)$$

for all $\mathbf{x},\mathbf{x}_0 \in B$ and for some $\lambda \in [0,1]$. In this case we need an inclusion function $F^{\nabla^2 f}$ for the Hessian of f (we also recall here the role played by inclusion functions for the Hessian of f within the α-BB approach discussed in Section 4.6). It turns out that for large boxes the natural interval extension is often the best option, while the mean value and Taylor form become superior for small boxes. We remark that for functions which are r times differentiable ($r > 2$), inclusion functions of order higher than two can also be defined (see, e.g., (Q. Lin & Rokne, 1996; Nataraj & Kotecha, 2002, 2004, 2005)). An analysis of the rate of convergence for some inclusion functions can be found in (Scholz, 2012).

In (Carrizosa, Hansen, & Messine, 2004) it is observed that better inclusion functions can be obtained through simple translations. In particular, in that paper polynomial functions are considered. We illustrate this through a simple example.

Example 4.103. Let $f(x) = x^2 - x$ with $x \in [0,2]$. Then, the natural interval extension of f over $[0,2]$ is $[-2,4]$. Now, let us rewrite f as follows:

$$f(x) = [(x-1)+1]^2 - [(x-1)+1].$$

Next, consider the translation $y = x-1$, so that as a function of y we can rewrite f as $y^2 + y$ with $y \in [-1,1]$. Now, the natural interval extension is $[-1,2]$, clearly, better than the former one. ∎

As observed in (Carrizosa et al., 2004), by solving two optimization problems, one might even find optimal inclusion functions with respect to all possible translations. However, since these optimization problems might be nonconvex and nondifferentiable, their exact solution might be quite computationally demanding, and thus local searches, less sharp but also much less computationally demanding, are advised.

Interval arithmetic is also used to deal with constrained problems (see, e.g., (Markót, Fernandez, Casado, & Csendes, 2006)). Given the set of constraints

$$g_i(\mathbf{x}) \le 0, \quad i = 1,\dots,m,$$

of the nonconvex problem, interval analysis applied to the constraint functions g_i with the corresponding inclusion functions F^{g_i} allows us to classify a box B into one of the following classes:

- *Certainly feasible*: If $F_{\mathbf{u}}^{g_i}(B) \le 0$ for all $i = 1,\dots,m$ (in particular, we define the box *strictly certainly feasible* if $F_{\mathbf{u}}^{g_i}(B) < 0$ for all $i = 1,\dots,m$).

- *Certainly unfeasible*: If $F_{\ell}^{g_i}(B) > 0$ for some $i \in \{1,\ldots,m\}$ (in this case the box can be discarded from further consideration).
- *Undetermined*: In all the other cases.

As a final comment about interval arithmetic, we remark that while some of the strategies proposed in the previous sections for deriving bounds of nonconvex problems have a quite limited applicability (think about all the bounds which are specific for quadratic problems), a positive aspect of interval arithmetic is its wide applicability. On the other hand, for those highly structured problems for which specific bounds can be derived, the quality of such bounds is usually much higher than interval arithmetic bounds. We also remark that in this section we have just given some introductory material about interval arithmetic. Something more will be said in Section 5.6, but for an extensive presentation we refer the reader, e.g., to (Hansen, 1992; Kearfott, 1996; Neumaier, 2004; Ratschek & Rokne, 1995).

4.11 Dual bounds

We have already met some dual bounds in the previous sections, such as the moment relaxation for PP problems in Section 4.5.4. In this section we will discuss some results about a well-known dual bound, the Lagrangian one, based on relaxations where some constraints (usually the "difficult" ones) are moved into the objective function. However, we point out that other dual bounds can be defined, such as the surrogate one based on relaxations where some constraints are substituted, through a linear combination, by a single one. This will not be discussed in detail here but we refer, e.g., to (Bricker, 1980; Locatelli & Schoen, 2012b) for a discussion about the computation of the surrogate dual bound for concave minimization problems over polytopes.

4.11.1 Lagrangian duality

Here we consider GO problems with the following form:

$$\begin{aligned} f^* = \quad & \min f(\mathbf{x}) \\ & h_i(\mathbf{x}) \leq 0, \quad i = 1,\ldots,m, \\ & \mathbf{x} \in X, \end{aligned} \tag{4.143}$$

where $X \neq \emptyset$ is a convex and compact set,[4] while f, h_i, $i = 1,\ldots,m$, are lower semicontinuous functions over X. We also assume that the feasible region is nonempty, i.e.,

$$S = \{\mathbf{x} \in X \ : \ h_i(\mathbf{x}) \leq 0, \ i = 1,\ldots,m\} \neq \emptyset.$$

For $\boldsymbol{\lambda} \in \mathbb{R}_+^m$, the *Lagrangian function* is defined as

$$L(\mathbf{x},\boldsymbol{\lambda}) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i h_i(\mathbf{x}).$$

[4] In fact, we could also remove the compactness assumption. In this case the min problems in what follows have to be substituted by inf problems.

Obviously, for any $\boldsymbol{\lambda} \geq \mathbf{0}$,

$$g(\boldsymbol{\lambda}) = \min_{\mathbf{x}\in X} L(\mathbf{x},\boldsymbol{\lambda})$$

is a lower bound for (4.143). Indeed, for each $\mathbf{x} \in S \subseteq X$ and each $\boldsymbol{\lambda} \geq \mathbf{0}$, we have that $L(\mathbf{x},\boldsymbol{\lambda}) \leq f(\mathbf{x})$. Note that g is a concave function since it is the pointwise infimum of an infinite collection of affine functions (see, e.g., (Rockafellar, 1970)). The supremum of the bounds $g(\boldsymbol{\lambda})$, $\boldsymbol{\lambda} \geq \mathbf{0}$, is also a lower bound for (4.143) and its computation gives rise to so-called *Lagrangian dual*

$$\begin{aligned} g^* = \quad & \sup\ g(\boldsymbol{\lambda}) \\ & \boldsymbol{\lambda} \geq \mathbf{0}. \end{aligned} \tag{4.144}$$

Weak duality holds, i.e., $g^* \leq f^*$. A case where equality holds is reported in the following theorem (see, e.g., (Geoffrion, 1971)).

Theorem 4.104. *If f, h_i, $i = 1,\dots,m$, are convex functions over S and some constraint qualification is satisfied, then strong duality holds, i.e., $f^* = g^*$.*

Possible constraint qualifications are (i) there exists $\bar{\mathbf{x}} \in X$ such that $h_i(\bar{\mathbf{x}}) < 0$ for each $i = 1,\dots,m$ (Slater's condition); (ii) f is convex on an open set containing S and functions h_i's are affine.

If some nonconvexity appears, then, usually, $f^* > g^*$. Let us first consider the case where f is nonconvex and functions h_i's are affine. In this case we can establish an interesting relation between the Lagrangian dual bound, and the bound

$$\begin{aligned} c^* = \quad & \min\ conv_{f,X}(\mathbf{x}) \\ & h_i(\mathbf{x}) \leq 0, \qquad i = 1,\dots,m, \\ & \mathbf{x} \in X, \end{aligned} \tag{4.145}$$

obtained by substituting the objective function f with its convex envelope over X. If the functions h_i's are affine, then in (Falk, 1969) it has been proved that $c^* = g^*$, i.e., the dual Lagrangian bound is exactly equal to the convex envelope bound. Here we report the proof of this result presented in (Duer & Horst, 1997).

Theorem 4.105. *If*

$$h_i(\mathbf{x}) = \mathbf{a}_i\mathbf{x} - b_i, \quad i = 1,\dots,m,$$

i.e., the functions h_i's are affine and a constraint qualification is satisfied for problem (4.145)*, then $c^* = g^*$.*

Proof. For $\boldsymbol{\lambda} \geq \mathbf{0}$, let us denote by

$$\bar{L}(\mathbf{x},\boldsymbol{\lambda}) = conv_{f,X}(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i(\mathbf{a}_i\mathbf{x} - b_i)$$

the Lagrangian function for the convex envelope $conv_{f,X}$. As discussed in Section 4.2.6, the convex envelope of the sum of a function f with an affine one is equal to the sum of the convex envelope for f with the affine function. Therefore, for all $\boldsymbol{\lambda} \geq \mathbf{0}$ we have that

$$conv_{L,X}(\mathbf{x}) = conv_{f,X}(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i(\mathbf{a}_i\mathbf{x} - b_i) = \bar{L}(\mathbf{x},\boldsymbol{\lambda}) \quad \forall\, \mathbf{x} \in X.$$

As observed in Section 4.2, the minimum value of a function over a region X is equal to the minimum value of the convex envelope over the same region. Then, for each $\boldsymbol{\lambda} \geq \mathbf{0}$ we have that

$$\bar{g}(\boldsymbol{\lambda}) = \min_{\mathbf{x}\in X}\ \bar{L}(\mathbf{x},\boldsymbol{\lambda}) = \min_{\mathbf{x}\in X}\ conv_{L,X}(\mathbf{x}) = \min_{\mathbf{x}\in X}\ L(\mathbf{x},\boldsymbol{\lambda}) = g(\boldsymbol{\lambda}).$$

Therefore,

$$\sup_{\boldsymbol{\lambda}\geq\mathbf{0}}\ \bar{g}(\boldsymbol{\lambda}) = \sup_{\boldsymbol{\lambda}\geq\mathbf{0}}\ g(\boldsymbol{\lambda}) = g^*. \tag{4.146}$$

Since a constraint qualification for the problem (4.145) is satisfied, then Theorem 4.104 allows us to conclude that

$$c^* = \sup_{\boldsymbol{\lambda}\geq\mathbf{0}}\ \bar{g}(\boldsymbol{\lambda}), \tag{4.147}$$

which, combined with (4.146), proves the result. □

More generally, if the functions h_i are convex and some constraint qualification is satisfied, so that (4.147) is satisfied, then

$$\begin{aligned} c^* &= \sup\nolimits_{\boldsymbol{\lambda}\geq\mathbf{0}} \min\nolimits_{\mathbf{x}\in X}\ \left[conv_{f,X}(\mathbf{x}) + \textstyle\sum_{i=1}^{m} \lambda_i h_i(\mathbf{x})\right] \\ &\leq \sup\nolimits_{\boldsymbol{\lambda}\geq\mathbf{0}} \min\nolimits_{\mathbf{x}\in X}\ \left[f(\mathbf{x}) + \textstyle\sum_{i=1}^{m} \lambda_i h_i(\mathbf{x})\right] \\ &= g^*. \end{aligned}$$

Therefore, we might wonder under which conditions $c^* < g^*$, i.e., the Lagrangian dual bound is strictly better than the convex envelope one. The following theorem has been proven in (Duer, 2002).

Theorem 4.106. *If*

- *f is strictly concave;*
- *h_i, $i = 1,\ldots,m$, are strictly convex and continuous differentiable functions;*
- *Slater's condition*

  $$\exists\, \bar{\mathbf{x}} \in X\ :\ h_i(\bar{\mathbf{x}}) < 0,\ i = 1,\ldots,m,$$

  is fulfilled;
- *$f^* > g^*$;*
- *$conv_{f,X}$ is not constant over any interval contained in X;*

then $g^* > c^*$.

We do not report the proof of this result, but we report an example from (Duer, 2002) to illustrate it.

Example 4.107. Consider the one-dimensional problem

$$\begin{aligned}\min \quad & -x^2\\ & x^2-x-2 \le 0,\\ & x \in X=[-2,3].\end{aligned}$$

Its optimal value is equal to -4, attained at $x=2$. The problem (4.145) is

$$\begin{aligned}\min \quad & -x-6\\ & x^2-x-2 \le 0\\ & x \in [-2,3],\end{aligned}$$

with optimal value -8 again attained at $x=2$, while the Lagrangian dual is

$$\max_{\lambda \ge 0} \min_{x\in[-2,3]} \quad -x^2+\lambda(x^2-x-2),$$

with optimal value -4.2 attained when $\lambda=\frac{6}{5}$ and $x=3$. We remark that all the assumptions of Theorem 4.106 are fulfilled. Indeed, $-x^2$ is strictly concave; x^2-x-2 is strictly convex and continuous differentiable; $\bar{x}=0$ shows that Slater's condition is satisfied; $f^*=-4>-4.2=g^*$; and $-x-6$ is not constant over any interval. ■

Ways to improve Lagrangian bounds

There are some ways to improve the quality of the Lagrangian dual bound. In this regard, a relevant observation has been made by Shor (N. Z. Shor, 1992). He observed that the Lagrangian dual bound can be improved if we add redundant constraints to the original formulation of a problem. We illustrate this fact with a very simple example.

Example 4.108. Consider the problem

$$\begin{aligned}\min \quad & -x^2\\ & x=1,\\ & x\ge 0.\end{aligned}$$

Its Lagrangian dual bound is

$$\sup_{\lambda} \inf_{x\ge 0} -x^2+\lambda(x-1),$$

which is equal to $-\infty$. Now, let us add the redundant constraint $x^2=x$, obtained by multiplying the constraint $x=1$ by x, to the original problem. Then, the dual Lagrangian bound for the modified problem is

$$\sup_{\lambda,\mu} \inf_{x\ge 0} -x^2+\lambda(x-1)+\mu(x^2-x),$$

whose optimal value (attained at $\lambda=\mu=1$ and for any $x\ge 0$) is equal to -1 and coincides with the optimal value of the original problem. ■

Ben-Tal, Eiger, and Gershovitz consider in (Ben-Tal, Eiger, & Gershovitz, 1994) the following problem:

$$\begin{aligned} \min \quad & f(\mathbf{x},\mathbf{y}) \\ & \mathbf{y} \in Y, \\ & \mathbf{x} \in X(\mathbf{y}), \end{aligned} \tag{4.148}$$

where $Y \subseteq \mathbb{R}^t$,

$$X(\mathbf{y}) = \{\mathbf{x} \in \mathbb{R}^n \ : \ h_i(\mathbf{x},\mathbf{y}) \leq 0,\ i = 1,\dots,m\},$$

and it is assumed that for each fixed $\mathbf{y} \in Y$, $f(\cdot,\mathbf{y})$ and $h_i(\cdot,\mathbf{y})$, $i = 1,\dots,m$, are convex functions (thus, in particular, $X(\mathbf{y})$ is a convex set for each fixed $\mathbf{y} \in Y$). It is also assumed that $X(\mathbf{y})$ satisfies Slater's condition for each $\mathbf{y} \in Y$. Given the Lagrangian function

$$L(\mathbf{x},\mathbf{y},\boldsymbol{\lambda}) = f(\mathbf{x},\mathbf{y}) + \sum_{i=1}^{m} \lambda_i h_i(\mathbf{x},\mathbf{y}),$$

the Lagrangian dual bound is

$$\ell_{LD} = \ell_{LD}(Y) = \sup_{\boldsymbol{\lambda} \geq \mathbf{0}} \inf_{\mathbf{y} \in Y, \mathbf{x} \in \mathbb{R}^n} L(\mathbf{x},\mathbf{y},\boldsymbol{\lambda}).$$

If we denote by f^* the optimal value of (4.148), it turns out that the duality gap $f^* - \ell_{LD}$ might be positive and even very large. In order to reduce the gap, in (Ben-Tal et al., 1994) it is suggested that we introduce a cover of Y, i.e.,

$$\Gamma = \{Y_j \subseteq Y \ : \ j \in J\} \ : \ \bigcup_{j \in J} Y_j = Y$$

(where J might be an infinite index set). Then, we might compute the Lagrangian dual bound over each subset

$$\ell_{LD}(Y_j) = \sup_{\boldsymbol{\lambda} \geq \mathbf{0}} \inf_{\mathbf{y} \in Y_j, \mathbf{x} \in \mathbb{R}^n} L(\mathbf{x},\mathbf{y},\boldsymbol{\lambda})$$

and consider the overall bound associated to the cover

$$\ell_{LD}^{\Gamma} = \inf_{j \in J} \ell_{LD}(Y_j).$$

It turns out that $\ell_{LD} \leq \ell_{LD}^{\Gamma} \leq f^*$. In particular, if $\Gamma = \{Y\}$, then, obviously, $\ell_{LD} = \ell_{LD}^{\Gamma}$, while it is proven that for $\Gamma = \{\{\mathbf{y}\} \ : \ \forall\, \mathbf{y} \in Y\}$, $\ell_{LD}^{\Gamma} = f^*$. Therefore, if

$$\delta(\Gamma) = \max_{j \in J} diam(Y_j)$$

denotes the maximum diameter for the subsets Y_j, then

$$\delta(\Gamma) = 0 \quad \Rightarrow \quad \ell_{LD}^{\Gamma} = f^*. \tag{4.149}$$

Unfortunately, the cover $\Gamma = \{\{\mathbf{y}\} \ : \ \forall\, \mathbf{y} \in Y\}$ is infinite. Thus, in (Ben-Tal et al., 1994) the question is faced whether, under suitable assumptions, for every $\varepsilon > 0$ there exists a fine enough *finite* cover Γ (i.e., a cover with $\delta(\Gamma)$ sufficiently small) such that $f^* - \ell_{LD}^{\Gamma} \leq \varepsilon$.

Ben-Tal et al. observe that, in view of (4.149), this amounts to proving that the gap $f^* - \ell_{LD}^{\Gamma}$, considered as a function of the diameter $\delta(\Gamma)$, is continuous at $\delta(\Gamma) = 0$. They prove continuity for a significative special case. Let

$$f(\mathbf{x},\mathbf{y}) = \mathbf{c}^T\mathbf{x}, \quad X(\mathbf{y}) = \{\mathbf{x} \in \mathbb{R}_+^n \ : \ A(\mathbf{y})\mathbf{x} \leq \mathbf{b}\}, \tag{4.150}$$

so that the problem is a linear one for each fixed $\mathbf{y} \in Y$. Note that in this case

$$\begin{aligned} \ell_{LD} &= \max_{\boldsymbol{\lambda} \geq \mathbf{0}} \min_{\mathbf{y} \in Y, \mathbf{x} \in \mathbb{R}_+^n} \{\mathbf{c}^T\mathbf{x} + \boldsymbol{\lambda}^T(A(\mathbf{y})\mathbf{x} - \mathbf{b})\} \\ &= \max_{\boldsymbol{\lambda} \geq \mathbf{0}} \{-\mathbf{b}^T\boldsymbol{\lambda} \ : \ A(\mathbf{y})^T\boldsymbol{\lambda} + \mathbf{c} \geq \mathbf{0} \quad \forall\, \mathbf{y} \in Y\}, \end{aligned} \tag{4.151}$$

where the last problem is a semi-infinite linear one. Assume that

- Y is a polytope and $A(\mathbf{y})$ is continuous over Y;
- for all $\mathbf{y} \in Y$, there exists $\mathbf{z} \geq \mathbf{0}$ such that

$$A(\mathbf{y})^T\mathbf{z} + \mathbf{c} > \mathbf{0}.$$

Then, Ben-Tal et al. proved that continuity of the gap $f^* - \ell_{LD}^{\Gamma}$ is satisfied at $\delta(\Gamma) = 0$ and, consequently, for all $\varepsilon > 0$ there exists some $\delta > 0$ and a finite cover Γ with $\delta(\Gamma) \leq \delta$ such that

$$f^* - \ell_{LD}^{\Gamma} \leq \varepsilon.$$

For the case where variables $\mathbf{x}$ do not appear in (4.148), an analogous result can be proven under mild conditions such as compactness of Y and continuity of the Lagrangian function (see, e.g., (Barrientos & Correa, 2000)). However, a significative feature of the result proven in (Ben-Tal et al., 1994) is that it is true for covers which involve *only* $\mathbf{y}$ variables. This is particularly significative when the number t of $\mathbf{y}$ variables is much lower than the number n of $\mathbf{x}$ variables, since branching operations (i.e., operations which replace a set with a cover of the same set) only involving $\mathbf{y}$ variables can be employed. These operations are more thoroughly discussed in the framework of BB methods in Section 5.3.

Finally, it is worthwhile to mention that for box-constrained nonconvex quadratic problems, possibly with some additional linear equality constraints, a way to improve the Lagrangian bound, which in this case can be computed through the solution of a semidefinite problem, is discussed in (Xia, Sun, Li, & Zheng, 2011).

Computability of the Lagrangian bounds

Another relevant issue about Lagrangian dual bounds is how simple it is to compute them. For instance, we observed that the dual bound in (4.151) required the solution of a semi-infinite linear program, which is not a simple task. However, in (Ben-Tal et al., 1994) it is proven that if

- Y is a polytope with a known vertex set $V(Y)$;
- the functions

$$a_j(\mathbf{y})^T\mathbf{z} + c_j, \quad j = 1, \ldots, n,$$

where $a_j(\mathbf{y})$ is the jth column of $A(\mathbf{y})$, are quasi-concave (which is true, e.g., if A is linear with respect to $\mathbf{y}$, i.e., the constraints are bilinear ones);

then the problem (4.151) is equivalent to the following linear program, where Y is replaced by $V(Y)$:

$$\max_{\boldsymbol{\lambda} \geq \mathbf{0}} \{-\mathbf{b}^T \boldsymbol{\lambda} \ : \ A(\mathbf{y})^T \boldsymbol{\lambda} + \mathbf{c} \geq \mathbf{0} \quad \forall\, \mathbf{y} \in V(Y)\}.$$

For problems with quadratic objective and constraint functions, the Lagrangian dual bound can be computed through the solution of a semidefinite problem (see, e.g., (Zheng, Sun, Li, & Xu, 2012)). The issue of computability of the Lagrangian dual bounds is also faced in other works, such as (Tuy, 2005a) for partly linear optimization problems generalizing (4.150), (Thoai, 2000b) for problems with a quadratic objective function, linear constraints, and one quadratic constraint, (Thoai, 2002) for problems with linear constraints and a concave objective function, (Nowak, 2005) for mixed-integer problems with block-separable quadratic objective and constraint functions, and (Kuno & Utsunomiya, 2000) for production-transportation problems with concave production costs. Of course, when the Lagrangian dual problem is itself a difficult one, a possibility is solving a relaxation of such a problem by substituting the Lagrangian function with a convex underestimator for it (see, e.g., (Van Voorhis, 2002)).

Further issues about Lagrangian bounds

For problems with a decomposable structure like the following:

$$\begin{array}{ll} \min & f^0(\mathbf{x}) + \sum_{i=1}^N f^i(\mathbf{y}_i) \\ & g^i_j(\mathbf{y}_i) \leq 0, \qquad\qquad i = 1,\ldots,N,\ j = 1,\ldots,m_i, \\ & h^i_r(\mathbf{x},\mathbf{y}_i) \leq 0, \qquad\quad i = 1,\ldots,N,\ r = 1,\ldots,n_i, \end{array}$$

where linking variables $\mathbf{x}$ appear in few linking constraints defined by the functions h^i_r, in (Karuppiah & Grossmann, 2008) the following approach is proposed.[5] First N copies of the linking variables are introduced, giving rise to the following equivalent formulation of the problem:

$$\begin{array}{ll} \min & \sum_{i=1}^N w_i f^0(\mathbf{x}_i) + \sum_{i=1}^N f^i(\mathbf{y}_i) \\ & g^i_j(\mathbf{y}_i) \leq 0, \qquad\qquad i = 1,\ldots,N,\ j = 1,\ldots,m_i, \\ & h^i_r(\mathbf{x}_i,\mathbf{y}_i) \leq 0, \qquad\quad i = 1,\ldots,N,\ r = 1,\ldots,n_i, \\ & \mathbf{x}_i = \mathbf{x}_{i+1}, \qquad\qquad i = 1,\ldots,N-1, \end{array}$$

for some $w_i \geq 0$, $i = 1,\ldots,N$, $\sum_{i=1}^N w_i = 1$. Then, a Lagrangian-based approach is employed where the constraints $\mathbf{x}_i = \mathbf{x}_{i+1}$ are moved into the objective, so that the resulting problem can be separated into N subproblems. The Lagrangian dual problem is then solved

[5] For the sake of precision, in the paper the approach is applied to mixed integer nonlinear (and nonconvex) problems.

by a heuristic approach. The results of the different subproblems are also used to derive valid cuts for the original problem.

Still related to Lagrangian functions, it is worthwhile to cite here methods based on *augmented Lagrangian* and *exact penalty* functions, with the corresponding convergence theory. We recall that a penalty function associated to a set of constraints is a function which is larger than 0 at points which violate the constraints and equal to 0 at all other points. It is called exact if the original problem, and the problem without the penalized constraints and whose objective function is obtained by summing the original objective function with the penalty function multiplied by a parameter, have the same solutions for large enough values of the parameter. In (H. Z. Luo, Sun, & Li, 2007) different classes of augmented Lagrangian functions are discussed. For instance, given the GO problem

$$\begin{array}{ll} \min & f(\mathbf{x}) \\ & h_i(\mathbf{x}) \leq 0, \quad i = 1,\dots,m, \end{array}$$

one can reformulate it as follows for $\alpha > 0$:

$$\begin{array}{ll} \min & f(\mathbf{x}) \\ & \frac{1}{\alpha}\psi(\alpha h_i(\mathbf{x})) \leq 0, \quad i = 1,\dots,m, \end{array}$$

where

- ψ is continuously differentiable and strictly convex over $\mathbb{R}$;
- $\psi(0) = 0$, $\psi'(0) = 1$, $\lim_{t \to -\infty} \psi'(t) = 0$;
- $\lim_{t \to -\infty} \psi(t) > -\infty$, $\lim_{t \to \infty} \frac{\psi(t)}{t} = \infty$.

A suitable choice for ψ is, e.g., $\psi(y) = e^y - 1$. Then, the *exponential-type augmented Lagrangian* functions are the classical Lagrangian functions for such reformulation, i.e.,

$$f(\mathbf{x}) + \frac{1}{\alpha}\sum_{i=1}^{m} \lambda_i \psi(\alpha h_i(\mathbf{x})), \quad \boldsymbol{\lambda} \geq \mathbf{0}.$$

In (H. Z. Luo et al., 2007) convergence results are given for a primal-dual method based on different augmented Lagrangian functions, such as the exponential-type one, in which a sequence of unconstrained GO problems is solved. That is, at each iteration of the method an approximation of a global minimizer for an unconstrained GO problem whose objective function is an augmented Lagrangian one is computed. Different classes of augmented Lagrangian functions and some related convergence results are also presented in (C. Y. Wang & Li, 2009). In (Birgin, Floudas, & Martinez, 2010) an iterative method is proposed where at each iteration an augmented Lagrangian function incorporating all the difficult constraints is minimized over the set of easy (such as linear or even bound) constraints, through the α-BB approach. In (Di Pillo, Lucidi, & Rinaldi, 2012) an iterative method has been proposed with the difficult constraints incorporated into an exact penalty function, and optimization over the easy constraints is performed through the DIRECT algorithm (see Section 3.1.8).

Finally, we remark that there are many works which discuss conditions under which we have a zero duality gap (i.e., $g^* = f^*$) for the Lagrangian dual bound (see, e.g., (Ben-Tal & Teboulle, 1996; Ye & Zhang, 2003; S. Zhang, 2000; Zheng et al., 2012)) and, more generally, for other bounds usually based on augmented Lagrangian and exact penalty functions (see, e.g., (Y. Chen & Chen, 2010; Gao, 2004; X. Huang & Yang, 2003; Nedić & Ozdaglar, 2008; Pang, 1997; Rubinov, Glover, & Yang, 1999; X. Yang & Huang, 2001)).

4.12 Factorable functions

In some cases, once convex underestimators for some functions are available, it is possible to combine them to derive convex underestimators for more complex functions. McCormick relaxations (see (McCormick, 1976, 1983) or (Scott, Stuber, & Barton, 2011) for a generalization) allow us to define convex underestimators and concave overestimators for so-called *factorable functions*.

Definition 4.109. *A function f is called* factorable *if it is obtained by a finite recursive composition of binary sums, binary products, and a library of univariate functions.*

McCormick relaxations are based on the knowledge of convex/concave under/overestimators for the binary operations (sums and products) and for the univariate functions. In order to illustrate them, in what follows, for a given function f and a given convex set X, we will denote by $\underline{f}^X$ and $\overline{f}^X$, respectively, a convex underestimator and a concave overestimator for f over X. The result for the binary sum is stated in the following proposition, whose proof is quite simple.

Proposition 4.110. *Let f_1, f_2 be two functions defined over a convex set X. Then, $\underline{f}_1^X + \underline{f}_2^X$ is a convex underestimator for $f_1 + f_2$, while $\overline{f}_1^X + \overline{f}_2^X$ is a concave overestimator for $f_1 + f_2$.*

For the binary product of two functions, we need to assume that lower and upper bounds for the two functions over X are known. Therefore, let us assume that

$$\ell_i \le f_i(\mathbf{x}) \le u_i \quad \forall\, \mathbf{x} \in X,\ i = 1,2.$$

Let us also define the four functions

$$h_1^X(\mathbf{x}) = \begin{cases} \ell_2 \underline{f}_1^X(\mathbf{x}) + \ell_1 \underline{f}_2^X(\mathbf{x}) - \ell_1\ell_2 & \text{if } \ell_1, \ell_2 \ge 0, \\ \ell_2 \overline{f}_1^X(\mathbf{x}) + \ell_1 \underline{f}_2^X(\mathbf{x}) - \ell_1\ell_2 & \text{if } \ell_1 \ge 0,\ \ell_2 < 0, \\ \ell_2 \underline{f}_1^X(\mathbf{x}) + \ell_1 \overline{f}_2^X(\mathbf{x}) - \ell_1\ell_2 & \text{if } \ell_1 < 0,\ \ell_2 \ge 0, \\ \ell_2 \overline{f}_1^X(\mathbf{x}) + \ell_1 \overline{f}_2^X(\mathbf{x}) - \ell_1\ell_2 & \text{if } \ell_1, \ell_2 < 0; \end{cases}$$

$$h_2^X(\mathbf{x}) = \begin{cases} u_2\underline{f}_1^X(\mathbf{x}) + u_1\underline{f}_2^X(\mathbf{x}) - u_1u_2 & \text{if } u_1, u_2 \geq 0, \\ u_2\overline{f}_1^X(\mathbf{x}) + u_1\underline{f}_2^X(\mathbf{x}) - u_1u_2 & \text{if } u_1 \geq 0,\ u_2 < 0, \\ u_2\underline{f}_1^X(\mathbf{x}) + u_1\overline{f}_2^X(\mathbf{x}) - u_1u_2 & \text{if } u_1 < 0,\ u_2 \geq 0, \\ u_2\overline{f}_1^X(\mathbf{x}) + u_1\overline{f}_2^X(\mathbf{x}) - u_1u_2 & \text{if } u_1, u_2 < 0; \end{cases}$$

$$h_3^X(\mathbf{x}) = \begin{cases} \ell_2\overline{f}_1^X(\mathbf{x}) + u_1\overline{f}_2^X(\mathbf{x}) - u_1\ell_2 & \text{if } u_1, \ell_2 \geq 0, \\ \ell_2\underline{f}_1^X(\mathbf{x}) + u_1\overline{f}_2^X(\mathbf{x}) - u_1\ell_2 & \text{if } u_1 \geq 0,\ \ell_2 < 0, \\ \ell_2\overline{f}_1^X(\mathbf{x}) + u_1\underline{f}_2^X(\mathbf{x}) - u_1\ell_2 & \text{if } u_1 < 0,\ \ell_2 \geq 0, \\ \ell_2\underline{f}_1^X(\mathbf{x}) + u_1\underline{f}_2^X(\mathbf{x}) - u_1\ell_2 & \text{if } u_1, \ell_2 < 0; \end{cases}$$

$$h_4^X(\mathbf{x}) = \begin{cases} u_2\overline{f}_1^X(\mathbf{x}) + \ell_1\overline{f}_2^X(\mathbf{x}) - \ell_1u_2 & \text{if } \ell_1, u_2 \geq 0, \\ u_2\underline{f}_1^X(\mathbf{x}) + \ell_1\overline{f}_2^X(\mathbf{x}) - \ell_1u_2 & \text{if } \ell_1 \geq 0,\ u_2 < 0, \\ u_2\overline{f}_1^X(\mathbf{x}) + \ell_1\underline{f}_2^X(\mathbf{x}) - \ell_1u_2 & \text{if } \ell_1 < 0,\ u_2 \geq 0, \\ u_2\underline{f}_1^X(\mathbf{x}) + \ell_1\underline{f}_2^X(\mathbf{x}) - \ell_1u_2 & \text{if } \ell_1, u_2 < 0. \end{cases}$$

Note that h_1^X, h_2^X are convex functions, while h_3^X, h_4^X are concave functions. Then, we have the following proposition, whose proof generalizes the one for the derivation of the convex and concave envelopes of the bilinear form x_1x_2 over boxes.

Proposition 4.111. *Let f_1, f_2 be defined over a convex set X. Then, the function*

$$\max\{h_1^X(\mathbf{x}), h_2^X(\mathbf{x})\}$$

is a convex underestimator for f_1f_2, while

$$\min\{h_3^X(\mathbf{x}), h_4^X(\mathbf{x})\}$$

is a concave overestimator for the same product function.

A little bit more complicated is the derivation of relaxations for composite functions. Let f be a continuous function defined over a convex set X and let h be a univariate function defined over an interval $[a,b]$ such that

$$\forall\, \mathbf{x} \in X \ :\ f(\mathbf{x}) \in [a,b].$$

Let

$$c_h \in \arg\min_{y\in[a,b]} \underline{h}^{[a,b]}(y), \quad d_h \in \arg\max_{y\in[a,b]} \overline{h}^{[a,b]}(y).$$

Then, it is possible to prove the following proposition.

Proposition 4.112. *Let $g(\mathbf{x}) = h(f(\mathbf{x}))$ be a composite function. Then,*

$$\underline{g}^X(\mathbf{x}) = \begin{cases} \underline{h}^{[a,b]}(\underline{f}^X(\mathbf{x})) & \text{if } c_h < \underline{f}^X(\mathbf{x}), \\ \underline{h}^{[a,b]}(\overline{f}^X(\mathbf{x})) & \text{if } c_h > \overline{f}^X(\mathbf{x}), \\ \underline{h}^{[a,b]}(c_h) & \text{otherwise,} \end{cases}$$

and

$$\overline{g}^X(\mathbf{x}) = \begin{cases} \overline{h}^{[a,b]}(\underline{f}^X(\mathbf{x})) & \text{if } d_h < \underline{f}^X(\mathbf{x}), \\ \overline{h}^{[a,b]}(\overline{f}^X(\mathbf{x})) & \text{if } d_h > \overline{f}^X(\mathbf{x}), \\ \overline{h}^{[a,b]}(d_h) & \text{otherwise.} \end{cases}$$

Proof. We do not give a complete proof of the result. The fact that $\underline{g}^X$ is an underestimator for g easily follows from the observation that $f(\mathbf{x}) \in [\underline{f}^X(\mathbf{x}), \overline{f}^X(\mathbf{x})]$ and the minimum of the convex function $\underline{h}^{[a,b]}$ over this interval lies at the left extreme if $c_h < \underline{f}^X(\mathbf{x})$, at the right extreme if $c_h > \overline{f}^X(\mathbf{x})$, and at c_h otherwise. In a completely similar way one can see that $\overline{g}^X$ is an overestimator for g. For what concerns the convexity of $\underline{g}^X$ (and the concavity of $\overline{g}^X$) we need to prove that for $\mathbf{x}_1, \mathbf{x}_2 \in X$ and $\lambda \in [0,1]$

$$\underline{g}^X(\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) \leq \lambda\underline{g}^X(\mathbf{x}_1) + (1-\lambda)\underline{g}^X(\mathbf{x}_2).$$

We prove this only when

$$c_h < \underline{f}^X(\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2).$$

By convexity of $\underline{f}^X$ it follows that

$$\lambda\underline{f}^X(\mathbf{x}_1) + (1-\lambda)\underline{f}^X(\mathbf{x}_2) \geq \underline{f}^X(\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) > c_h. \tag{4.152}$$

Then, in view of (4.152)

$$\underline{g}^X(\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) = \underline{h}^{[a,b]}(\underline{f}^X(\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2)) \leq \underline{h}^{[a,b]}(\lambda\underline{f}^X(\mathbf{x}_1) + (1-\lambda)\underline{f}^X(\mathbf{x}_2)). \tag{4.153}$$

Moreover, either $\underline{f}^X(\mathbf{x}_1) > c_h$ or $\underline{f}^X(\mathbf{x}_2) > c_h$ is certainly true. Without loss of generality, we assume that $\underline{f}^X(\mathbf{x}_1) > c_h$. Now, we have three possible cases.

- $\underline{f}^X(\mathbf{x}_2) > c_h$: Then, from (4.153) and in view of the convexity of $\underline{h}^{[a,b]}$, we have that

$$\begin{aligned} \underline{g}^X(\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) &\leq \lambda\underline{h}^{[a,b]}(\underline{f}^X(\mathbf{x}_1)) + (1-\lambda)\underline{h}^{[a,b]}(\underline{f}^X(\mathbf{x}_2)) \\ &= \lambda\underline{g}^X(\mathbf{x}_1) + (1-\lambda)\underline{g}^X(\mathbf{x}_2), \end{aligned}$$

  from which the result follows.

- $\underline{f}^X(\mathbf{x}_2) \leq c_h \leq \overline{f}^X(\mathbf{x}_2)$: Then, from (4.152) it also follows that

  $$\lambda \underline{f}^X(\mathbf{x}_1) + (1-\lambda)c_h \geq \lambda \underline{f}^X(\mathbf{x}_1) + (1-\lambda)\underline{f}^X(\mathbf{x}_2) \geq \underline{f}^X(\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2).$$

  Then, from (4.153) and in view of the convexity of $\underline{h}^{[a,b]}$, we have that

  $$\underline{g}^X(\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) \leq \lambda \underline{h}^{[a,b]}(\underline{f}^X(\mathbf{x}_1)) + (1-\lambda)\underline{h}^{[a,b]}(c_h) = \lambda \underline{g}^X(\mathbf{x}_1) + (1-\lambda)\underline{g}^X(\mathbf{x}_2),$$

  from which the result follows.

- $c_h > \overline{f}^X(\mathbf{x}_2)$: Then, from (4.152) it also follows that

  $$\lambda \underline{f}^X(\mathbf{x}_1) + (1-\lambda)\overline{f}^X(\mathbf{x}_2) \geq \lambda \underline{f}^X(\mathbf{x}_1) + (1-\lambda)\underline{f}^X(\mathbf{x}_2) \geq \underline{f}^X(\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2).$$

  Then, from (4.153) and in view of the convexity of $\underline{h}^{[a,b]}$, we have that

  $$\begin{aligned} \underline{g}^X(\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) &\leq \lambda \underline{h}^{[a,b]}(\underline{f}^X(\mathbf{x}_1)) + (1-\lambda)\underline{h}^{[a,b]}(\overline{f}^X(\mathbf{x}_2)) \\ &= \lambda \underline{g}^X(\mathbf{x}_1) + (1-\lambda)\underline{g}^X(\mathbf{x}_2), \end{aligned}$$

  from which the result follows.

The proof for the other cases as well as the proof of the concavity of $\overline{g}^X$ are completely analogous. □

Bompadre and Mitsos (Bompadre & Mitsos, 2012) study the convergence rate for McCormick relaxations, i.e., the relation between (i) the difference of the minimum value of a function over a box and the lower bound returned by the relaxation of the same function over the same box; (ii) the diameter of the box. We remark that in (Mitsos, Chachuat, & Barton, 2009) some rules to derive subgradients of the convex underestimators and concave overestimators are also presented. Knowledge of such subgradients is relevant, as will be clarified in Section 4.13.

Relaxations for factorable functions are proposed in different works. A usual procedure to deal with a problem involving them is by first reformulating the problem by substituting simple univariate or bivariate functions with corresponding new auxiliary variables, and then relaxing the reformulated problem (see, e.g., (Belotti, Lee, Liberti, Margot, & Wächther, 2009; Gatzke, Tolsma, & Barton, 2002; Leyffer, Sartenaer, & Wanufelle, 2008; Marcovecchio, Bergamini, & Aguirre, 2006; Smith & Pantelides, 1999)). We illustrate this through an example.

Example 4.113. Consider the function

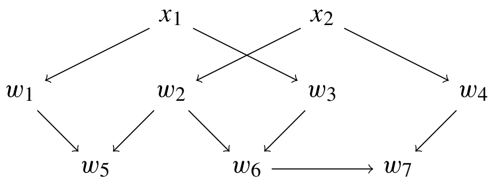
$$x_1^2 \log(x_2) + \frac{\log(x_1)\log(x_2)}{e^{x_2}},$$

with $1 \leq x_1, x_2 \leq 2$. First, we make the substitutions

$$\begin{aligned} w_1 &= x_1^2, \\ w_2 &= \log(x_2), \\ w_3 &= \log(x_1), \\ w_4 &= e^{x_2}, \\ w_5 &= w_1 w_2, \\ w_6 &= w_2 w_3, \\ w_7 &= \tfrac{w_6}{w_4}, \end{aligned} \tag{4.154}$$

so that the original function can be rewritten as a linear one $w_5 + w_7$ but with the additional (nonconvex) constraints (4.154). Then, we proceed by substituting each equality constraint with a convex relaxation for it. Note that such relaxations require the knowledge of bounds also for the new variables w_i, $i = 1,\dots,7$. These can be induced from the bounds on the original variables. ∎

The dependence between variables (the original as well as the newly introduced) can be represented through a directed acyclic graph (see, e.g., (Belotti et al., 2009; Leyffer et al., 2008)), whose nodes correspond to the variables and a directed arc joins some variable w_i (or x_i) with some other variable w_j if the former appears in the equation defining the latter. For the previous example the graph is the following.



Note that, by definition, the original variables (x_1,x_2 in the example) do not have any ingoing arc. While in the example above the univariate functions have simple under- and overestimators, the same may not be true for other univariate functions. In these cases Sherali and Wang (Sherali & Wang, 2001) propose a two-stage approach. First, a (nonconvex) PP relaxation of the problem is defined by substituting univariate functions with bounding polynomials (in particular, obtained by exploiting the mean value theorem and Chebyshev interpolation), and then the RLT technique is used to linearize the PP relaxation.

4.13 Polyhedral convex underestimators

In this section we discuss a topic which has a relevant practical impact. Consider nonconvex problems with the form (4.143), where it is required that X is a polyhedral region. Assume that convex underestimators $\hat{f}$ and $\hat{h}_i$, $i = 1,\dots,m$, over X for the objective and the constraint functions, are available. Then, the most straightforward way to compute a lower bound for (4.143) is to substitute every function with the corresponding convex underestimator, thus ending up with a convex problem. Functions $\hat{f}$ and $\hat{h}_i$, $i = 1,\dots,m$, might be nonlinear and nonpolyhedral (i.e., they are not the maximum of a finite number of affine functions). In this case the convex problem to be solved is a nonlinear one and its solution requires a nonlinear solver. For solving convex mixed integer nonlinear problems, in (Duran & Grossmann, 1986; Geoffrion, 1972) it is suggested that we substitute the convex functions with polyhedral convex underestimators. More formally, if $\hat{f}$ is a nonpolyhedral convex function, we substitute it with a polyhedral convex function

$$\tilde{f}(\mathbf{x}) = \max_{i=1,\dots,K} \boldsymbol{\alpha}_i^T \mathbf{x} + \alpha_{i0} \le \hat{f}(\mathbf{x}) \quad \forall\, \mathbf{x} \in X.$$

Through this substitution we end up with a problem which can be reformulated as a mixed integer linear problem. Tawarmalani and Sahinidis (Tawarmalani & Sahinidis, 2004) extended this observation to nonconvex problems (see also (Linderoth, 2005)): each convex underestimator is substituted by a polyhedral convex underestimator, so that we end up with a problem which can be reformulated as linear. Such substitution has a quite relevant practical impact. Indeed, nonlinear problems are more difficult to solve and pose more numerical difficulties with respect to linear problems. Of course, in doing this we might lose something from the point of view of the quality of the lower bound, but this disadvantage is compensated by the possibility of using linear solvers. Now the question is the following: once the nonpolyhedral convex underestimator $\hat{f}$ is available, how can we compute a polyhedral convex underestimator $\tilde{f}$? Assume that the points $\mathbf{x}_j \in X$, $j = 1,\ldots,K$, are available together with $\mathbf{s}_j \in \partial\hat{f}(\mathbf{x}_j)$ (i.e., $\mathbf{s}_j$ belongs to the subgradient of $\hat{f}$ at $\mathbf{x}_j$). Then, the convexity of $\hat{f}$ implies that

$$\hat{f}(\mathbf{x}) \geq \hat{f}(\mathbf{x}_j) + \mathbf{s}_j(\mathbf{x} - \mathbf{x}_j) \quad \forall\, \mathbf{x} \in X,\ j = 1,\ldots,K,$$

i.e., $\hat{f}(\mathbf{x}_j) + \mathbf{s}_j(\mathbf{x} - \mathbf{x}_j)$ is a supporting hyperplane of $\hat{f}$ at $\mathbf{x}_j$. Consequently, we can set

$$\tilde{f}(\mathbf{x}) = \max_{j=1,\ldots,K} [\hat{f}(\mathbf{x}_j) + \mathbf{s}_j(\mathbf{x} - \mathbf{x}_j)].$$

Of course, different choices of the finite set of points in X lead to different polyhedral functions. The choice of these points should be such that the "distance" between $\hat{f}$ and $\tilde{f}$ is as small as possible. For univariate functions "optimal" strategies to select these points have been developed. These are implemented by the so-called *sandwich algorithm* (Rote, 1992). In this algorithm, given a finite set of points, first an interval, say $[a,b]$, within which some error measure (measuring the distance between $\hat{f}$ and $\tilde{f}$) is largest, is detected. Then, a new point within this interval is selected according to some given rule. Such rules include the following.

Bisection. Bisect the selected interval.

Maximum error rule. Construct the supporting hyperplane at the intersection point of the two supporting lines at the end points of the selected interval.

Slope bisection. Find the supporting line whose slope is the average of the slopes at the end points of the selected interval.

Chord rule. Construct the supporting line with the same slope as the concave envelope of $\hat{f}$ over the selected interval.

The first two rules are based on the selection of a x-value, while the last two select a slope value. The *vertical error* is defined as

$$\sup_{x\in[a,b]} \hat{f}(x) - \tilde{f}(x),$$

i.e., it is equal to the maximum difference between the convex function and its polyhedral underestimator. In (Rote, 1992) it is observed that, with respect to a worst-case analysis, the best possible convergence rate for this error measure is $O(\frac{1}{K^2})$ (the worst case is obtained when $\hat{f}(x) = x^2$), and it is proven that all four rules above are "optimal" with respect to it, in the sense that their largest vertical error is also $O(\frac{1}{K^2})$. Another possible error measure is the *projective error*,

$$\sup_{y\in[a,b]} \inf_{x\in[a,b]} \|(x, \hat{f}(x)) - (y, \tilde{f}(y))\|,$$

where the distance between points on the graph of the two functions is considered. The projective error rule (see (Tawarmalani & Sahinidis, 2004)), where the supporting line is taken at the point on the graph of $\hat{f}$ closest to the intersection of the two supporting lines at the extremes of the selected interval, has quadratic convergence with respect to the projective error measure. In (Guerin, Marcotte, & Savard, 2006) the error is defined as the integral over $[a,b]$ of the difference between the piecewise linear function obtained as the pointwise maximum of the tangent lines at the selected points, and the piecewise linear function obtained as the pointwise maximum of the lines which interpolate $\hat{f}$ at the extremes of the intervals defined by the selected points. A dynamic programming approach is proposed to select the K points in such a way that the error in the worst case is minimized. Polyhedral relaxations for multivariate cases have also been developed (see, e.g., (Linderoth, 2005) for bivariate functions). However, for such cases the development of "optimal" strategies (in the sense specified above) is more complicated and research in this field is still active.

An interesting observation has been made in (Tawarmalani & Sahinidis, 2005). Assume that h_1 is a convex function over a convex set X. Let h_2 be a one-dimensional function defined over an interval $[a,b]$ and such that

- $[a,b] \supseteq \{f(\mathbf{x}) \, : \, \mathbf{x} \in X\}$;
- h_2 is convex over $[a,b]$;
- h_2 is nondecreasing (this assumption might be removed when h_1 is an affine function).

For simplicity assume also that h_1, h_2 are differentiable functions (we refer to (Tawarmalani & Sahinidis, 2005) for the nondifferentiable case). Then, the function $h_3(\mathbf{x}) = h_2(h_1(\mathbf{x}))$ is convex. Given the K points $\mathbf{x}_1, \ldots, \mathbf{x}_K \in X$, the epigraph (see Definition A.25) of a polyhedral convex underestimator of h_3 is the set

$$S_1 = \{(\mathbf{x}\ \gamma) \, : \, \gamma \geq h_3(\mathbf{x}_j) + \nabla h_3(\mathbf{x}_j)(\mathbf{x} - \mathbf{x}_j), \ \ j = 1, \ldots, K\}.$$

However, a different approach is that of finding polyhedral underestimators for h_1 and h_2 separately. If we do this, the resulting polyhedral convex underestimator has the epigraph

$$\begin{aligned} S_2 = \{(\mathbf{x}\ \gamma) \, : \quad & \gamma \geq h_2(h_1(\mathbf{x}_j)) + h_2'(h_1(\mathbf{x}_j))(y - h_1(\mathbf{x}_j)), \ \ j = 1, \ldots, K, \\ & y \geq h_1(\mathbf{x}_j) + \nabla h_1(\mathbf{x}_j)(\mathbf{x} - \mathbf{x}_j), \ \ j = 1, \ldots, K\} \end{aligned}$$

(the last inequalities can be replaced by equalities if h_1 is an affine function). The following theorem is proven in (Tawarmalani & Sahinidis, 2005).

Theorem 4.114. *We have that*

$$S_2 \subseteq S_1.$$

Proof. Recall that h_2 is a nondecreasing function over $[a,b]$, i.e., h_2' is nonnegative over this interval. Then, if $(\mathbf{x}\ \gamma) \in S_2$,

$$\begin{aligned} \gamma &\geq h_2(h_1(\mathbf{x}_j)) + h_2'(h_1(\mathbf{x}_j))(y - h_1(\mathbf{x}_j)) \\ &\geq h_2(h_1(\mathbf{x}_j)) + h_2'(h_1(\mathbf{x}_j))\nabla h_1(\mathbf{x}_j)(\mathbf{x} - \mathbf{x}_j). \end{aligned}$$

Since by the rule for the derivation of composite functions

$$h_2'(h_1(\mathbf{x}_j))\nabla h_1(\mathbf{x}_j) = \nabla h_3(\mathbf{x}_j),$$

we can conclude that

$$\gamma \geq h_3(\mathbf{x}_j) + \nabla h_3(\mathbf{x}_j)(\mathbf{x} - \mathbf{x}_j),$$

so that $(\mathbf{x}\ \gamma) \in S_1$. □

If the function h_1 is affine, then it can be proven that $S_1 = S_2$. The following simple example, taken again from (Tawarmalani & Sahinidis, 2005), shows that strict inclusion is possible when h_1 is not affine.

Example 4.115. Let $h_1(x) = x^2$, $h_2(y) = y^2$, so that $h_3(x) = x^4$. Let $x_1 = 1$ and $x_2 = 4$. Then,

$$S_1 = \{(x\ \gamma)\ :\ \gamma \geq 1 + 4(x-1),\ \gamma \geq 256 + 256(x-4)\},$$

while

$$\begin{aligned} S_2 = \{(x\ \gamma)\ :\quad & \gamma \geq 1 + 2(y-1),\ \gamma \geq 256 + 32(y-16), \\ & y \geq 1 + 2(x-1),\ y \geq 16 + 8(x-4)\}. \end{aligned}$$

The polyhedral function whose epigraph is S_1 is

$$\max\{4x - 3, 256x - 768\},$$

while that related to S_2 is

$$\begin{cases} 4x - 3, & x < \frac{5}{2}, \\ 16x - 33, & \frac{5}{2} \leq x \leq \frac{735}{240}, \\ 256x - 768, & x > \frac{735}{240}. \end{cases}$$

The latter is tighter than the former in the interval $[\frac{5}{2}, \frac{735}{240}]$. ■

We remark that in (Tawarmalani & Sahinidis, 2005) these results are proven for the more general case where h_1 is a vector of functions and h_2 is also a vector of multidimensional functions (all defined over a space whose dimension is equal to the dimension of the vector of functions h_1), so that h_3 is a vector of functions with the same dimension as h_2. The proofs for this more general case do not substantially differ from those presented here. We also remark that the description of S_2 requires more inequalities than that of S_1 for the same number K of points. Therefore, Tawarmalani and Sahinidis also tried to establish whether the quality of the approximation of S_2 can be obtained also for S_1, as soon as in S_1 we allow for the inclusion of a number of linear inequalities equal to that of S_2. They show that for separable functions obtained as the sum of n one-dimensional strictly convex functions, the quality obtained by S_2 with K points can be obtained by S_1 only with exponentially many (K^n) points.

4.14 Convex outer approximations

As already remarked in the introduction, if we want to define a convex relaxation of a GO problem with feasible region

$$\{\mathbf{x} \in D \;:\; g_i(\mathbf{x}) \geq r_i > 0,\; i = 1,\ldots,m\},$$

where D is a convex set, we need to find convex outer approximations for the regions defined by the nonconvex constraints $g_i(\mathbf{x}) \geq r_i$, $i = 1,\ldots,m$. As already seen, one way to do that is to substitute the functions g_i with concave overestimators over D, thus making the constraints convex. But as remarked by Tawarmalani et al. in (Tawarmalani, Richard, & Chung, 2010), in some cases a tighter convex relaxation can be obtained through a convex outer approximation of the upper level set $\{\mathbf{x} \in D \;:\; g_i(\mathbf{x}) \geq r_i\}$. We illustrate this through the simple initial example proposed in (Tawarmalani et al., 2010).

Example 4.116. Consider the nonconvex set

$$\{(x_1\; x_2\; x_3) \in \mathbb{R}^3_+ \;:\; x_1x_2 + x_3 \geq r\}, \tag{4.155}$$

where $r > 0$ is some constant value. One possible way to convexify this set is by first detecting a concave overestimator of the function $x_1x_2 + x_3$ over the nonnegative orthant, and then substituting it on the left-hand side of the inequality. It turns out that the resulting convex relaxation is

$$\{(x_1\; x_2\; x_3) \in \mathbb{R}^3_+ \;:\; x_1, x_2 > 0\} \cup \{(x_1\; x_2\; x_3) \in \mathbb{R}^3_+ \;:\; x_1x_2 = 0,\; x_3 \geq r\}.$$

If we also require closedness of the convex relaxation, we end up with the whole nonnegative orthant $\mathbb{R}^3_+$, i.e., the convexification is obtained by simply dropping the inequality. However, a different and tighter convex relaxation is obtained if we follow another procedure, where we directly search for a convex relaxation of the upper level set (4.155) for the function $x_1x_2 + x_3$, rather than going through the concave overestimator of such function. Indeed, in such a case we might employ the convex outer approximation

$$\left\{(x_1\; x_2\; x_3) \in \mathbb{R}^3_+ \;:\; \sqrt{\frac{x_1x_2}{r}} + \frac{x_3}{r} \geq 1\right\},$$

which, as we will see from the following theory, turns out to be the convex hull of the set (4.155). ∎

Now, following (Tawarmalani et al., 2010), we develop a theory to derive convex hulls of sets like (4.155). We first introduce and comment four assumptions. Consider a set $Y \subseteq \mathbb{R}^{\sum_{i=1}^n d_i}$ and n sets $Y_i \subseteq Y$.

Assumption 4.3. *For each* $i \in \{1,\ldots,n\}$,

$$\mathbf{z} = (\mathbf{z}_1 \ldots \mathbf{z}_i \ldots \mathbf{z}_n) \in Y_i \quad \Rightarrow \quad \mathbf{z}_j = 0 \quad \textit{if } j \neq i.$$

Such an assumption requires that the subsets Y_i belong to linear subspaces which are orthogonal to each other. In fact, in (Tawarmalani et al., 2010) it is observed that in some cases we can also require that the linear subspaces are not necessarily orthogonal to each other but can be made such after the application of a proper linear transformation of the space of variables.

Assumption 4.4. $chull(Y) = chull(\bigcup_{i=1}^{n} Y_i)$.

According to this assumption, each point in $chull(Y)$ can be obtained as a convex combination of points in the sets Y_i, $i = 1,\dots,n$. Since the convex hull of Y is obtained as the convex hull of a union of the orthogonal sets Y_i, a relation can be established with disjunctive programming techniques, whose first reference is a 1974 work by Balas now published in (Balas, 1998). Before introducing the third assumption, the following definition is needed.

Definition 4.117. *A function h is* positively homogeneous *(PH) if, for each* $\lambda > 0$,

$$h(\lambda \mathbf{x}) = \lambda h(\mathbf{x}).$$

Assumption 4.5. *For each* $i \in \{1,\dots,n\}$, *there exist PH functions* t_i^j, $j \in J_i^p$, $p \in \{-1,0,1\}$,

$$t_i^j : \mathbb{R}^{\sum_{i=1}^n d_i} \times \mathbb{R}^{\sum_{i=1}^n d_i'} \rightarrow \mathbb{R},$$

i.e., defined over the original space of $\mathbf{z}_i$ *variables (possibly) augmented with the variables* $\mathbf{u}_i$, *such that*

$$chull(Y_i) \subseteq proj_{\mathbf{z}}(A_i) \subseteq cl(chull(Y_i)),$$

where

$$A_i = \{(\mathbf{0}\dots\mathbf{z}_i\ \mathbf{u}_i\dots\mathbf{0}) \in \mathbb{R}^{\sum_{i=1}^n (d_i+d_i')}\ :\ t_j(\mathbf{z}_i,\mathbf{u}_i) \geq p \quad \forall\ j \in J_i^p,\ p = -1,0,+1\}.$$

This assumption requires that a description of the convex hull of each set Y_i is available through a finite set of inequalities involving PH functions. Note that in some cases $chull(Y_i)$ might be described by sets of inequalities not based on PH functions, but through proper transformations the description can be turned into one based on PH functions. A simple example, discussed in (Tawarmalani et al., 2010) is the set

$$\{(x_1\ x_2) \in \mathbb{R}_+^2\ :\ x_1 x_2 \geq r\},$$

for some $r > 0$. The set is already a convex one, but its current description is based on the function $\frac{x_1 x_2}{r}$, which is not PH. However, it is enough to consider the equivalent formulation

$$\{(x_1\ x_2) \in \mathbb{R}_+^2\ :\ \sqrt{x_1 x_2} \geq \sqrt{r}\}$$

to obtain a description through the PH function $\sqrt{\frac{x_1 x_2}{r}}$.

Assumption 4.6. *Let*

$$C_i = \{(\mathbf{0}\dots\mathbf{z}_i\ \mathbf{u}_i\dots\mathbf{0}) \in \mathbb{R}^{\sum_{i=1}^n (d_i+d_i')}\ :\ t_j(\mathbf{z}_i,\mathbf{u}_i) \geq 0 \quad \forall\ j \in J_i^p,\ p = -1,0,+1\}.$$

Then, for each $i \in \{1,\ldots,n\}$,

$$proj_{\mathbf{z}}(C_i) \subseteq 0^+(cl(chull(\cup_{i=1}^n Y_i))),$$

where, for a given convex set C, $0^+(C)$ denotes the recession cone (see Definition A.13*).*

This assumption requires that the recession directions of all the sets A_i (i.e., the sets C_i) are projected into recession directions for the cone $0^+(cl(chull(\cup_{i=1}^n Y_i)))$. In (Tawarmalani et al., 2010) it is proven that if

$$\forall\, i \in \{1,\ldots,n\} \;:\; Y_i \neq \emptyset, \quad t_j^p \in J_i^p \quad \text{are PH and concave,} \tag{4.156}$$

then Assumption 4.6 is satisfied.

The following theorem gives a description of $chull(Y)$ through inequalities based on the PH functions t_j^p, $j \in J_i^p$, $p \in \{-1,0,1\}$, $i \in \{1,\ldots,n\}$.

Theorem 4.118. *If Assumptions* 4.3–4.6 *are satisfied, then*

$$chull(Y) \subseteq proj_{\mathbf{z}}(X) \subseteq cl(chull(Y)),$$

where X is the set of pairs $(\mathbf{z}\;\mathbf{u})$ such that

$$\begin{array}{ll}
\sum_{i=1}^n t_i^{j_i}(\mathbf{z}_i,\mathbf{u}_i) \geq 1 & \forall\, (j_i)_{i=1}^n \in \prod_{i=1}^n J_i^{+1}, \\
\sum_{i\in I} t_i^{j_i}(\mathbf{z}_i,\mathbf{u}_i) \geq -1 & \forall\, I \subseteq \{1,\ldots,n\},\ (j_i)_{i\in I} \in \prod_{i\in I} J_i^{-1}, \\
t_i^{j_i}(\mathbf{z}_i,\mathbf{u}_i) + t_i^{j_i'}(\mathbf{z}_i,\mathbf{u}_i) \geq 0 & \forall\, i \in \{1,\ldots,n\},\ j_i \in J_i^{+1},\ j_i' \in J_i^{-1}, \\
t_i^{j_i}(\mathbf{z}_i,\mathbf{u}_i) \geq 0 & \forall\, i \in \{1,\ldots,n\},\ j_i \in J_i^{+1}, \\
t_i^{j_i}(\mathbf{z}_i,\mathbf{u}_i) \geq 0 & \forall\, i \in \{1,\ldots,n\},\ j_i \in J_i^{0}.
\end{array} \tag{4.157}$$

The proof of the theorem is based on two lemmas. We first need to introduce the following set for some $T \subseteq \{1,\ldots,n\}$ and some $\lambda_T \geq 0$:

$$R_T(\lambda_T) = \left\{ \begin{array}{lll}
(\mathbf{z}_i\;\mathbf{u}_i)_{i\in T} \;:\; & \sum_{i\in T} t_i^{j_i}(\mathbf{z}_i,\mathbf{u}_i) \geq \lambda_T & \forall\, (j_i)_{i\in T} \in \prod_{i\in T} J_i^{+1} \\
 & \sum_{i\in I} t_i^{j_i}(\mathbf{z}_i,\mathbf{u}_i) \geq -\lambda_T & \forall\, I \subseteq T,\ (j_i)_{i\in I} \in \prod_{i\in I} J_i^{-1} \\
 & t_i^{j_i}(\mathbf{z}_i,\mathbf{u}_i) + t_i^{j_i'}(\mathbf{z}_i,\mathbf{u}_i) \geq 0 & \forall\, i \in T,\ j_i \in J_i^{+1},\ j_i' \in J_i^{-1} \\
 & t_i^{j_i}(\mathbf{z}_i,\mathbf{u}_i) \geq 0 & \forall\, i \in T,\ j_i \in J_i^{+1} \\
 & t_i^{j_i}(\mathbf{z}_i,\mathbf{u}_i) \geq 0 & \forall\, i \in T,\ j_i \in J_i^{0}
\end{array} \right\}.$$

In particular, for a singleton $T = \{i\}$ we use the notation $R_i(\lambda_i)$. Next, we introduce the set

$$Q = \left\{ \begin{array}{lll} (\boldsymbol{\lambda}\ \mathbf{z}\ \mathbf{u}) : & \lambda_i \geq 0 & \forall\, i \in \{1,\dots,n\} \\ & (\mathbf{z}_i\ \mathbf{u}_i) \in R_i(\lambda_i) & \forall\, i \in \{1,\dots,n\} \\ & \sum_{i=1}^n \lambda_i = 1 & \end{array} \right\}.$$

In (Tawarmalani et al., 2010) it is first proven that Q represents a higher-dimensional representation of $chull(Y)$ (Lemma 4.119), and then that the set X is the projection of Q over the space of variables $(\mathbf{z}\ \mathbf{u})$ (Lemma 4.120). The combination of these two results proves Theorem 4.118.

Lemma 4.119. *Under Assumptions* 4.3–4.6*, we have that*

$$chull(Y) \subseteq proj_{\mathbf{z}}(Q) \subseteq cl(chull(Y)).$$

Proof. We first prove that if $\bar{\mathbf{z}} \in chull(\cup_{i=1}^n Y_i)$, then by a suitable choice of $(\boldsymbol{\lambda}\ \mathbf{u})$ such a point can be extended to a point in Q, i.e., we prove that

$$chull(Y) = chull(\cup_{i=1}^n Y_i) \subseteq proj_{\mathbf{z}}(Q),$$

where the equality follows from Assumption 4.4. By Assumption 4.3,

$$\bar{\mathbf{z}} = (\bar{\mathbf{z}}_1 \dots \bar{\mathbf{z}}_n) = \sum_{i=1}^n \lambda_i (\mathbf{0} \dots \mathbf{z}_i' \dots \mathbf{0}),$$

where $\boldsymbol{\lambda} \in \Delta_n$, i.e., $\boldsymbol{\lambda}$ belongs to the unit simplex (see Definition A.3), and

$$(\mathbf{0} \dots \mathbf{z}_i' \dots \mathbf{0}) \in chull(Y_i).$$

It follows from Assumption 4.5 that the point $(\mathbf{0} \dots \mathbf{z}_i' \dots \mathbf{0})$ can be extended to $(\mathbf{0} \dots \mathbf{z}_i'\ \mathbf{u}_i' \dots \mathbf{0}) \in A_i$ and we can define

$$\bar{\mathbf{u}} = (\bar{\mathbf{u}}_1 \dots \bar{\mathbf{u}}_n) = \sum_{i=1}^n \lambda_i (\mathbf{0} \dots \mathbf{u}_i' \dots \mathbf{0}).$$

Now, without loss of generality, assume that the indexes i are ordered in such a way that

$$\lambda_i > 0, \quad i = 1,\dots,t, \quad \lambda_i = 0,\ i = t+1,\dots,n,$$

so that, in particular, $\sum_{i=1}^t \lambda_i = 1$. We have that

$$\bar{\mathbf{z}}_i = \lambda_i \mathbf{z}_i', \quad \bar{\mathbf{u}}_i = \lambda_i \mathbf{u}_i'. \tag{4.158}$$

Notice also that $R_i(1) = proj_{\mathbf{z}_i,\mathbf{u}_i}(A_i)$, so that $(\mathbf{z}_i' \; \mathbf{u}_i') \in R_i(1)$ for all $i \in \{1,\dots,t\}$, i.e.,

$$\begin{array}{ll}
t_i^{j_i}(\mathbf{z}_i',\mathbf{u}_i') \geq 1 & \forall\, j_i \in J_i^{+1}, \\
t_i^{j_i}(\mathbf{z}_i',\mathbf{u}_i') \geq -1 & \forall\, j_i \in J_i^{-1}, \\
t_i^{j_i}(\mathbf{z}_i',\mathbf{u}_i') + t_i^{k_i}(\mathbf{z}_i',\mathbf{u}_i') \geq 0 & \forall\, j_i \in J_i^{+1},\ k_i \in J_i^{-1}, \\
t_i^{j_i}(\mathbf{z}_i',\mathbf{u}_i') \geq 0 & \forall\, j_i \in J_i^{+1}, \\
t_i^{j_i}(\mathbf{z}_i',\mathbf{u}_i') \geq 0 & \forall\, j_i \in J_i^{0}
\end{array}$$

(obviously, the third and fourth sets of constraints are redundant but they are also needed for the proof). Now, let us make the change of variables (4.158) and multiply both sides of all the inequalities by λ_i:

$$\begin{array}{ll}
\lambda_i t_i^{j_i}\left(\frac{\bar{\mathbf{z}}_i}{\lambda_i},\frac{\bar{\mathbf{u}}_i}{\lambda_i}\right) \geq \lambda_i & \forall\, j_i \in J_i^{+1}, \\
\lambda_i t_i^{j_i}\left(\frac{\bar{\mathbf{z}}_i}{\lambda_i},\frac{\bar{\mathbf{u}}_i}{\lambda_i}\right) \geq -\lambda_i & \forall\, j_i \in J_i^{-1}, \\
\lambda_i t_i^{j_i}\left(\frac{\bar{\mathbf{z}}_i}{\lambda_i},\frac{\bar{\mathbf{u}}_i}{\lambda_i}\right) + \lambda_i t_i^{k_i}\left(\frac{\bar{\mathbf{z}}_i}{\lambda_i},\frac{\bar{\mathbf{u}}_i}{\lambda_i}\right) \geq 0 & \forall\, j_i \in J_i^{+1},\ k_i \in J_i^{-1}, \\
\lambda_i t_i^{j_i}\left(\frac{\bar{\mathbf{z}}_i}{\lambda_i},\frac{\bar{\mathbf{u}}_i}{\lambda_i}\right) \geq 0 & \forall\, j_i \in J_i^{+1}, \\
\lambda_i t_i^{j_i}\left(\frac{\bar{\mathbf{z}}_i}{\lambda_i},\frac{\bar{\mathbf{u}}_i}{\lambda_i}\right) \geq 0 & \forall\, j_i \in J_i^{0}.
\end{array}$$

Since all the functions are assumed to be PH, we end up with

$$\begin{array}{ll}
t_i^{j_i}(\bar{\mathbf{z}}_i,\bar{\mathbf{u}}_i) \geq \lambda_i & \forall\, j_i \in J_i^{+1}, \\
t_i^{j_i}(\bar{\mathbf{z}}_i,\bar{\mathbf{u}}_i) \geq -\lambda_i & \forall\, j_i \in J_i^{-1}, \\
t_i^{j_i}(\bar{\mathbf{z}}_i,\bar{\mathbf{u}}_i) + t_i^{k_i}(\bar{\mathbf{z}}_i,\bar{\mathbf{u}}_i) \geq 0 & \forall\, j_i \in J_i^{+1},\ k_i \in J_i^{-1}, \\
t_i^{j_i}(\bar{\mathbf{z}}_i,\bar{\mathbf{u}}_i) \geq 0 & \forall\, j_i \in J_i^{+1}, \\
t_i^{j_i}(\bar{\mathbf{z}}_i,\bar{\mathbf{u}}_i) \geq 0 & \forall\, j_i \in J_i^{0},
\end{array}$$

i.e., $(\bar{\mathbf{z}}_i \; \bar{\mathbf{u}}_i) \in R_i(\lambda_i)$ for $i \in \{1,\dots,t\}$. For $i \in \{t+1,\dots,n\}$ we set $(\bar{\mathbf{z}}_i \; \bar{\mathbf{u}}_i) = (\mathbf{0} \; \mathbf{0})$. Since the functions $t_i^{j_i}$ are PH, then $t_i^{j_i}(\mathbf{0},\mathbf{0}) = 0$ for all of them, i.e., $(\mathbf{0} \; \mathbf{0}) \in R_i(0)$. Therefore, we have proved that for each i, $(\bar{\mathbf{z}}_i \; \bar{\mathbf{u}}_i) \in R_i(\lambda_i)$, i.e., $(\boldsymbol{\lambda} \; \bar{\mathbf{z}} \; \bar{\mathbf{u}}) \in Q$.

Next, we prove that $(\boldsymbol{\lambda} \; \bar{\mathbf{z}} \; \bar{\mathbf{u}}) \in Q$ implies $\bar{\mathbf{z}} \in cl(chull(\cup_{i=1}^n Y_i))$. For $i \in \{1,\dots,t\}$, $\lambda_i > 0$, and the fact that the functions $t_i^{j_i}$ are PH imply that $(\frac{\bar{\mathbf{z}}_i}{\lambda_i} \; \frac{\bar{\mathbf{u}}_i}{\lambda_i}) \in R_i(1)$. Then,

$$\left(\mathbf{0}\dots\frac{\bar{\mathbf{z}}_i}{\lambda_i}\;\frac{\bar{\mathbf{u}}_i}{\lambda_i}\dots\mathbf{0}\right) \in A_i,$$

so that

$$(\bar{\mathbf{z}}_1\ \bar{\mathbf{u}}_1 \dots \bar{\mathbf{z}}_t\ \bar{\mathbf{u}}_t\ \mathbf{0} \dots \mathbf{0}) \in chull(\cup_{i=1}^{t} A_i).$$

Since

$$proj_{\mathbf{z}}(chull(\cup_{i=1}^{n} A_i)) \subseteq chull(\cup_{i=1}^{n} proj_{\mathbf{z}}(A_i)), \quad proj_{\mathbf{z}}(A_i) \subseteq cl(chull(Y_i)),$$

where the last inclusion follows from Assumption 4.5, we have that

$$(\bar{\mathbf{z}}_1 \dots \bar{\mathbf{z}}_t\ \mathbf{0} \dots \mathbf{0}) \in chull(\cup_{i=1}^{n} cl(chull(Y_i))) \subseteq cl(chull(\cup_{i=1}^{n} Y_i)).$$

Now, Assumption 4.6 together with $\lambda_{t+1} = 0$ imply that

$$(\mathbf{0} \dots \bar{\mathbf{z}}_{t+1} \dots \mathbf{0}) \in 0^{+}(cl(chull(\cup_{i=1}^{n} Y_i)))$$

(note that $proj_{\mathbf{z}_i,\mathbf{u}_i}(C_i) = R_i(0)$). Consequently,

$$(\bar{\mathbf{z}}_1 \dots \bar{\mathbf{z}}_{t+1}\ \mathbf{0} \dots\ \mathbf{0}) \in cl(chull(\cup_{i=1}^{n} Y_i)).$$

Then, by induction we can prove that $\bar{\mathbf{z}} \in cl(chull(\cup_{i=1}^{n} Y_i))$. □

Lemma 4.120. *X is the projection of Q, i.e.,*

$$X = proj_{\mathbf{z},\mathbf{u}}(Q).$$

Proof. The proof of this result is by induction. For a given subset $I \subseteq \{1,\dots,n\}$, denote by $\mathbf{z}_I$ the vector $(\mathbf{z}_i)_{i \in I}$, whose dimension is $\sum_{i \in I} d_i$. Now, let A, B be two disjoint subsets of $\{1,\dots,n\}$. Consider the set

$$W = \left\{ \begin{array}{lr} (\lambda_A\ \lambda_B\ \lambda_{AB}\ \mathbf{z}_A\ \mathbf{u}_A\ \mathbf{z}_B\ \mathbf{u}_B) : & \lambda_A \geq 0,\ (\mathbf{z}_A\ \mathbf{u}_A) \in R_A(\lambda_A) \\ & \lambda_B \geq 0,\ (\mathbf{z}_B\ \mathbf{u}_B) \in R_B(\lambda_B) \\ & \lambda_A + \lambda_B = \lambda_{AB} \end{array} \right\},$$

and the set

$$P = \{(\lambda_{AB}\ \mathbf{z}_{AB}\ \mathbf{u}_{AB}) : \lambda_{AB} \geq 0,\ (\mathbf{z}_{AB}\ \mathbf{u}_{AB}) \in R_{A \cup B}(\lambda_{AB})\}.$$

If $\mathbf{z}_{AB} = (\mathbf{z}_A\ \mathbf{z}_B)$ and $\mathbf{u}_{AB} = (\mathbf{u}_A\ \mathbf{u}_B)$, then we prove that the set obtained by projecting λ_A and λ_B out from W is exactly P. First, substitute λ_B with $\lambda_{AB} - \lambda_A$ in W. Therefore, the description of this set becomes

$$\lambda_A \geq 0,\ (\mathbf{z}_A\ \mathbf{u}_A) \in R_A(\lambda_A),$$

$$\lambda_{AB} - \lambda_A \geq 0,\ (\mathbf{z}_B\ \mathbf{u}_B) \in R_B(\lambda_{AB} - \lambda_A).$$

The inequalities

$$t_i^{j_i}(\mathbf{z}_i,\mathbf{u}_i) + t_i^{k_i}(\mathbf{z}_i,\mathbf{u}_i) \geq 0\ \ \forall\, i \in A \cup B,\ j_i \in J_i^{+1},\ k_i \in J_i^{-1}, \tag{4.159}$$

$$t_i^{j_i}(\mathbf{z}_i,\mathbf{u}_i) \geq 0 \qquad \forall\, i \in A \cup B,\ j_i \in J_i^{+1}, \tag{4.160}$$

$$t_i^{j_i}(\mathbf{z}_i,\mathbf{u}_i) \geq 0 \qquad \forall\, i \in A \cup B,\ j_i \in J_i^{0}, \tag{4.161}$$

do not depend on λ_A, so they are not modified by the projection. The inequalities dependent on λ_A can be rewritten as follows for $j_i \in J_i^{+1}$ and $k_i \in J_i^{-1}$:

$$\min\left\{\begin{array}{l}\sum_{i\in A} t_i^{j_i}(\mathbf{z}_i,\mathbf{u}_i)\\ \\ \lambda_{AB}+\min_{B'\subseteq B}\sum_{i\in B'} t_i^{k_i}(\mathbf{z}_i,\mathbf{u}_i)\end{array}\right\} \geq \lambda_A \geq \max\left\{\begin{array}{l}\lambda_{AB}-\sum_{i\in B} t_i^{j_i}(\mathbf{z}_i,\mathbf{u}_i)\\ \\ -\min_{A'\subseteq A}\sum_{i\in A'} t_i^{k_i}(\mathbf{z}_i,\mathbf{u}_i)\end{array}\right\}$$

(the constraint $\lambda_A \geq 0$ corresponds to $A' = \emptyset$, while the constraint $\lambda_{AB} - \lambda_A \geq 0$ corresponds to $B' = \emptyset$). Now, λ_A can be projected out by using the Fourier–Motzkin elimination. Then,

$$\sum_{i\in A\cup B} t_i^{j_i}(\mathbf{z}_i,\mathbf{u}_i) \geq \lambda_{AB} \qquad \forall\, (j_i)_{i\in A\cup B} \in \prod_{i\in A\cup B} J_i^{+1}, \tag{4.162}$$

$$\sum_{i\in A} t_i^{j_i}(\mathbf{z}_i,\mathbf{u}_i) + \sum_{i\in A'} t_i^{k_i}(\mathbf{z}_i,\mathbf{u}_i) \geq 0 \;\forall\, A' \subseteq A,\; (j_i)_{i\in A} \in \prod_{i\in A} J_i^{+1},\; (k_i)_{i\in A'} \in \prod_{i\in A'} J_i^{-1}, \tag{4.163}$$

$$\sum_{i\in B} t_i^{j_i}(\mathbf{z}_i,\mathbf{u}_i) + \sum_{i\in B'} t_i^{k_i}(\mathbf{z}_i,\mathbf{u}_i) \geq 0 \;\forall\, B' \subseteq B,\; (j_i)_{i\in B} \in \prod_{i\in B} J_i^{+1},\; (k_i)_{i\in B'} \in \prod_{i\in B'} J_i^{-1}, \tag{4.164}$$

$$\sum_{i\in A'\cup B'} t_i^{k_i}(\mathbf{z}_i,\mathbf{u}_i) \geq -\lambda_{AB} \;\; \forall\, A' \subseteq A,\; \forall\, B' \subseteq B,\; (k_i)_{i\in A'\cup B'} \in \prod_{i\in A'\cup B'} J_i^{-1}. \tag{4.165}$$

Notice that the inequalities (4.159) for $i \in A'$ (respectively, $i \in B'$) and (4.160) for $i \in A \setminus A'$ (respectively, $i \in B \setminus B'$) imply that the inequalities (4.163) (respectively, (4.164)) are redundant. Note that $A' = B' = \emptyset$ corresponds to $\lambda_{AB} \geq 0$. Therefore, the result of the projection is the set defined by the inequalities (4.159) to (4.162) and (4.165), which is exactly the set P. Now, if this result is applied sequentially with $A = \{1,\ldots,h\}$ and $B = \{h+1\}$, with h ranging from 1 to $n-1$, the result $proj_{\mathbf{z},\mathbf{u}}(Q) = R_{\{1,\ldots,n\}}(1) = X$ is obtained. □

In (Tawarmalani et al., 2010) the following corollary is also proven.

Corollary 4.121. *If Assumptions* 4.3–4.6 *are satisfied, $proj_{\mathbf{z}}(A_i)$ is a closed set, and $proj_{\mathbf{z}}(C_i) = 0^+(cl(chull(Y_i)))$, then $proj_{\mathbf{z}}(X) = cl(chull(Y))$.*

A slightly different result can be proven if Assumption 4.4 is replaced by the more general *convex extension property*.

Definition 4.122. *For a set Y satisfying Assumption* 4.3, *the* convex extension property *is satisfied if*

$$\mathbf{z} \in Y \quad \Rightarrow \quad \mathbf{z} = \sum_{i=1}^{n} \lambda_i \mathbf{v}_i + \sum_{i=1}^{n} \mu_i \mathbf{w}_i,$$

where

$$\mathbf{v}_i \in cl(chull(Y_i)),\; \mathbf{w}_i \in 0^+(cl(chull(Y_i))),\; \sum_{i=1}^{n} \lambda_i = 1,\; \lambda_i,\mu_i \geq 0 \;\forall\, i \in \{1,\ldots,n\}.$$

The property turns out to be equivalent to

$$cl(chull(Y)) = cl(chull(\cup_{i=1}^{n} Y_i)),$$

and if such property replaces Assumption 4.4, it is possible to prove that

$$cl(proj_{\mathbf{z}}(X)) = cl(chull(Y)).$$

The following theorem states quite general conditions under which the convex extension property is satisfied.

Theorem 4.123. *Let g be a function defined over $\mathbb{R}_{+}^{\sum_{i=1}^{n} d_i}$ and*

$$G = \{\mathbf{z} \ : \ g(\mathbf{z}_1,\ldots,\mathbf{z}_n) \geq r\},$$

for $r > 0$. Let

$$G_i = \{(\mathbf{0}\ldots\mathbf{z}_i\ldots\mathbf{0}) \in G\}.$$

Let

$$g_i(\mathbf{z}_i) = g(\mathbf{0},\ldots,\mathbf{z}_i,\ldots,\mathbf{0}).$$

Assume that there exist functions h_i, $i \in \{1,\ldots,n\}$, defined over $\mathbb{R}_{+}^{d_i}$ and f defined over $\mathbb{R}^n$ and convex, such that

(a) $g(\mathbf{z}) \leq f(h_1(\mathbf{z}_1),\ldots,h_n(\mathbf{z}_n))$;

(b) $f(\mathbf{y}_1) > f(\mathbf{y}_2)$ *if* $\mathbf{y}_1 \geq \mathbf{y}_2$ *and* $\mathbf{y}_1 \neq \mathbf{y}_2$;

(c) $g_i(\mathbf{z}_i) = f(0,\ldots,h_i(\mathbf{z}_i),\ldots,0)$;

(d) *for all i : $h_i(\mathbf{0}) = 0$ and for $\lambda \in (0,1]$*

$$\lambda h_i\left(\frac{\mathbf{z}_i}{\lambda}\right) \geq h_i(\mathbf{z}_i);$$

(e) *for all i : $h_i(\mathbf{z}_i) \leq 0 \ \Rightarrow \ (\mathbf{0}\ldots\mathbf{z}_i\ldots\mathbf{0}) \in 0^{+}(cl(chull(G_i)))$.*

Then, the convex extension property is satisfied for G.

Proof. Let $\mathbf{z} \in G$ and let

$$\mathbf{y}(\mathbf{z}) = (h_1(\mathbf{z}_1),\ldots,h_n(\mathbf{z}_n)).$$

Let

$$T = \{i \ : \ h_i(\mathbf{z}_i) \leq 0\}.$$

By Assumption (e), for all $i \in T$

$$(\mathbf{0}\ldots\mathbf{z}_i\ldots\mathbf{0}) \in 0^{+}(cl(chull(G_i))).$$

Therefore,

$$\mathbf{z}' = \mathbf{z} - \sum_{i\in T}(\mathbf{0}\ldots\mathbf{z}_i\ldots\mathbf{0}) \in cl(chull(\cup_{i=1}^{n} G_i)) \quad \Rightarrow \quad \mathbf{z} \in cl(chull(\cup_{i=1}^{n} G_i)).$$

Then, we only need to prove that $\mathbf{z}' \in cl(chull(\cup_{i=1}^n G_i))$. Consider a subgradient $\boldsymbol{\delta}$ of f at $\mathbf{y}(\mathbf{z}')$: we prove that $\boldsymbol{\delta} > \mathbf{0}$. Indeed, by contradiction assume that $\delta_i \leq 0$ for some i. Then, by convexity of f, for $\varepsilon > 0$ we have that

$$f(\mathbf{y}(\mathbf{z}') - \varepsilon \mathbf{e}^i) \geq f(\mathbf{y}(\mathbf{z}')) - \varepsilon\delta_i \geq f(\mathbf{y}(\mathbf{z}')),$$

which contradicts Assumption (b). By definition of $\mathbf{z}'$ we have that

$$h_i(\mathbf{z}_i') = h_i(\mathbf{z}_i), \ \forall\, i \notin T, \ \ h_i(\mathbf{z}_i') = h_i(\mathbf{0}) = 0 \geq h_i(\mathbf{z}_i), \ \ \forall\, i \in T.$$

Therefore,

$$y_i(\mathbf{z}') = \max\{y_i(\mathbf{z}), 0\}.$$

Then, by Assumptions (a) and (b) it follows that

$$f(\mathbf{y}(\mathbf{z}')) \geq f(\mathbf{y}(\mathbf{z})) \geq g(\mathbf{z}) \geq r.$$

Consider first the case where $\boldsymbol{\delta}^T \mathbf{y}(\mathbf{z}') = 0$. In view of $\boldsymbol{\delta} > \mathbf{0}$, necessarily $\mathbf{y}(\mathbf{z}') = \mathbf{0}$ and, consequently, $\mathbf{z}' = \mathbf{0}$. Indeed, if $h_i(\mathbf{z}_i') = 0$, then $h_i(\mathbf{z}_i) \leq 0$, so that $i \in T$ and, by definition, $\mathbf{z}_i' = \mathbf{0}$. From Assumption (c) it follows that $g(\mathbf{z}') = g(\mathbf{0}) = g_i(\mathbf{0}) = f(\mathbf{y}(\mathbf{z}')) \geq r$. Then, $\mathbf{z}' \in G_i$ for all i and $\mathbf{z}' \in cl(chull(\cup_{i=1}^n G_i))$.

Next, assume that $\boldsymbol{\delta}^T \mathbf{y}(\mathbf{z}') > 0$. Let

$$\lambda_i = \frac{\delta_i y_i(\mathbf{z}')}{\boldsymbol{\delta}^T \mathbf{y}(\mathbf{z}')}. \tag{4.166}$$

Then, $\boldsymbol{\lambda} \in \Delta_n$. Let $I = \{i \ : \ \lambda_i > 0\}$. Since $\delta_i > 0$, we have

$$i \notin I \ \Rightarrow \ y_i(\mathbf{z}') = 0 \ \Rightarrow \ i \in T \ \Rightarrow \ \mathbf{z}_i' = \mathbf{0}$$

(the second implication follows from the fact that by definition of T, $i \notin T$ implies $h_i(\mathbf{z}_i') > 0$, while the third implication follows from the definition of $\mathbf{z}'$). Then, setting

$$\mathbf{z}_i'' = (\mathbf{0} \ldots \mathbf{z}_i' \ldots \mathbf{0}),$$

$\mathbf{z}'$ can be rewritten as

$$\mathbf{z}' = \sum_{i \in I} \mathbf{z}_i''.$$

Now, for $i \in I$ let $\mathbf{w}_i = \frac{\mathbf{z}_i''}{\lambda_i}$, so that $\mathbf{z}' = \sum_{i \in I} \lambda_i \mathbf{w}_i$. If it can be proven that $\mathbf{w}_i \in G_i$, then $\mathbf{z}' \in cl(chull(\cup_{i=1}^n G_i))$ follows and the proof is complete.

Indeed,

$$g(\mathbf{w}_i) = g_i\left(\frac{\mathbf{z}_i'}{\lambda_i}\right) = f(\mathbf{y}(\mathbf{w}_i)) \geq f\left(\frac{1}{\lambda_i}\mathbf{y}(\mathbf{z}_i'')\right),$$

where the first equality follows from the definition of g_i, the second equality from Assumption (c), and the inequality from Assumptions (b) and (d). By convexity of f, recalling that $\boldsymbol{\delta}$ is a subgradient of f at $\mathbf{y}(\mathbf{z}')$, and by the definition (4.166) of λ_i, we also have

$$f\left(\frac{1}{\lambda_i}\mathbf{y}(\mathbf{z}_i'')\right) \geq f(\mathbf{y}(\mathbf{z}')) + \delta_i \frac{\boldsymbol{\delta}^T \mathbf{y}(\mathbf{z}')}{\delta_i y_i(\mathbf{z}')} y_i(\mathbf{z}'') - \sum_{j=1}^n \delta_j y_j(\mathbf{z}').$$

Since $y_i(\mathbf{z}'') = h_i(\mathbf{z}_i') = y_i(\mathbf{z}')$, the last expression is also equal to

$$f(\mathbf{y}(\mathbf{z}')) + \delta_i \frac{\boldsymbol{\delta}^T \mathbf{y}(\mathbf{z}')}{\delta_i y_i(\mathbf{z}')} y_i(\mathbf{z}') - \sum_{j=1}^{n} \delta_j y_j(\mathbf{z}') = f(\mathbf{y}(\mathbf{z}')) \geq r,$$

from which $\mathbf{w}_i \in G_i$ follows. □

As a corollary of this result we have the following.

Corollary 4.124. *Let g, g_i, G, G_i be defined as in Theorem* 4.123. *Assume that*

(a1) $g(\mathbf{z}) \leq \sum_{i=1}^{n} g_i(\mathbf{z}_i)$;

(b1) *for all i : $g_i(\mathbf{0}) = 0$ and for $\lambda \in (0,1]$*

$$\lambda g_i\left(\frac{\mathbf{z}_i}{\lambda}\right) \geq g_i(\mathbf{z}_i);$$

(c1)

$$\forall\, i \;:\; g_i(\mathbf{z}_i) \leq 0 \;\Rightarrow\; (\mathbf{0} \ldots \mathbf{z}_i \ldots \mathbf{0}) \in 0^+(cl(chull(G_i))).$$

Then, the convex extension property is satisfied.

Proof. This is a special case of Theorem 4.123 where f is the sum function and $h_i = g_i$ for all i. □

In addition to these results, in (Tawarmalani et al., 2010) conditions under which $chull(G)$ is a closed set are given.

We conclude the section with a description of one of the examples discussed in (Tawarmalani et al., 2010) as applications of all these results, namely the derivation of the convex hull for bilinear covering sets over the nonnegative orthant (we refer to the paper for further examples). Consider the set

$$Z = \left\{ (\mathbf{x}\ \mathbf{y}) \in \mathbb{R}_+^n \times \mathbb{R}_+^n \;:\; \sum_{i=1}^{n} (a_i x_i y_i + b_i x_i + c_i y_i) \geq r \right\}, \tag{4.167}$$

where $r > 0$, and for each i, $a_i, b_i, c_i \geq 0$ and at least one is strictly positive. We first prove that each set

$$Z_i = \{(x_i\ y_i) \in \mathbb{R}_+^2 \;:\; a_i x_i y_i + b_i x_i + c_i y_i \geq r\}$$

is convex and give a representation of this set as an upper level set of a PH function. Let

$$\alpha_i(x_i, y_i) = \frac{1}{2}\left(b_i x_i + c_i y_i + \sqrt{(b_i x_i + c_i y_i)^2 + 4 a_i r x_i y_i} \right).$$

Lemma 4.125. *The set Z_i is convex and*

$$Z_i = \{(x_i\ y_i) \in \mathbb{R}_+^2 \;:\; \alpha_i(x_i, y_i) \geq r\}.$$

Proof. Without loss of generality, we can assume that $c_i \geq b_i$ and that at least one between a_i and c_i is strictly positive, so that for points $(x_i\ y_i) \in Z_i$, $a_i x_i + c_i > 0$. Then, Z_i can be represented as follows:

$$Z_i = \left\{ (x_i\ y_i) \in \mathbb{R}^2_+ \ :\ y_i \geq \frac{r - b_i x_i}{a_i x_i + c_i} \right\}.$$

To see that Z_i is a convex set it is enough to prove that

$$\frac{r - b_i x_i}{a_i x_i + c_i}$$

is a convex function for $x_i \geq 0$. This can be easily seen by computing the second derivative. Now, let us introduce the positive variable h_i and consider the following homogenization of the inequality

$$a_i x_i y_i + b_i h_i x_i + c_i h_i y_i \geq r h_i^2. \tag{4.168}$$

Notice that if $(x_i\ y_i\ h_i)$, with $h_i \geq 1$, satisfies this inequality, then the inequality is satisfied by any point $(x_i\ y_i\ h_i')$ with $h_i' \in [1, h_i]$. Then, Z_i can be viewed as the projection into the space of the variables $(x_i\ y_i)$ of the set

$$\{(x_i\ y_i\ h_i) \ :\ a_i x_i y_i + b_i h_i x_i + c_i h_i y_i \geq r h_i^2,\ h_i \geq 1\}.$$

Now, the inequality (4.168) is satisfied if

$$\frac{1}{2r}\left(b_i x_i + c_i y_i - \sqrt{(b_i x_i + c_i y_i)^2 + 4 a_i r x_i y_i} \right) \leq h_i \leq \frac{1}{2r}\left(b_i x_i + c_i y_i + \sqrt{(b_i x_i + c_i y_i)^2 + 4 a_i r x_i y_i} \right).$$

The lower bound for h_i is a nonpositive quantity, so that it is implied by $h_i \geq 1$. Taking into account the upper bound for h_i, which is equal to $\frac{\alpha_i(x_i, y_i)}{r}$, we have that the set Z_i is defined by the inequality

$$\alpha_i(x_i, y_i) \geq r,$$

as we wanted to prove. □

In (Tawarmalani et al., 2010) it has also been proven that if the upper level set of a PH function is convex, then the function is concave wherever it is positive. Therefore, the function α_i is concave over the nonnegative orthant. Now we are ready to prove the following proposition, returning the convex hull for Z.

Proposition 4.126. *Let Z be defined as in (4.167). We have that*

$$chull(Z) = \left\{ (\mathbf{x}\ \mathbf{y}) \in \mathbb{R}^n_+ \times \mathbb{R}^n_+ \ :\ \sum_{i=1}^n \alpha_i(x_i, y_i) \geq r \right\}.$$

Proof. The proof is based on Theorem 4.118 with $Y = Z$ and

$$Y_i = \{(0\,0 \ldots x_i\ y_i \ldots 0\,0) \ :\ (x_i\ y_i) \in Z_i\}.$$

We do not give all the details of the proof but observe that

- the convex extension property for Z is satisfied in view of Corollary 4.124 with $g_i = \alpha_i$ and $g = \sum_{i=1}^{n} \alpha_i$;
- in Lemma 4.125 we proved that the sets Z_i are already convex ones and can be represented through an inequality involving a PH function, so that Assumption 4.5 is satisfied;
- the functions α_i are concave ones over the nonnegative orthant and for sufficiently large $(x_i\ y_i)$, $\alpha_i(x_i, y_i) \geq r$, i.e., $Z_i \neq \emptyset$. Therefore, in view of (4.156) Assumption 4.6 is satisfied. □

4.15 Further remarks about relaxations

In the previous sections we have discussed relaxations for some large classes of GO problems. We will omit to deal with relaxations for some other smaller, but highly structured, classes of problems. These include (but are certainly not restricted to) problems involving *posynomial functions*, i.e., functions which are linear combinations, with positive coefficients, of posynomial terms $x_1^{\alpha_1} \cdots x_n^{\alpha_n}$, with $\mathbf{x} \in [\boldsymbol{\ell}, \mathbf{u}]$, $\boldsymbol{\ell} > \mathbf{0}$, $\boldsymbol{\alpha} \in \mathbb{R}^n$ (see, e.g., (Lu, Li, Gounaris, & Floudas, 2010)); *signomial* functions, i.e., difference of posynomial functions (see, e.g., (Maranas & Floudas, 1997; Y. Wang, Zhang, & Gao, 2004)); multilinear functions, defined in Section 4.2.7 (see, e.g., (Luedtke et al., 2010; Ryoo & Sahinidis, 2001)); linear complementarity constraints (see, e.g., (Thoai, Yamamoto, & Yoshise, 2005)); products of affine (or convex) functions (see, e.g., (Konno & Kuno, 1995b)); and sum-of-ratios functions (see, e.g., (Schaible, 1995; Schaible & Shi, 2003b)).

We also observe that even for classes of problems which have been previously discussed, there might be subclasses which offer the opportunity of defining further relaxations with respect to those available for the general class. An example of this is represented by network flow problems with concave costs over the arcs: while these can be viewed as general concave optimization problems, where a concave function is minimized over a polyhedral region, their special structure can be exploited. For example, in (Dalila, Fontes, Hadjiconstantinou, & Christofides, 2006), bounds for such problem are computed through a dynamic programming approach.

Although we have pointed out their relation with the material presented in Section 4.14, we have omitted to deal with nonconvex disjunctive programming problems, i.e., problems where also some boolean variables appear and some constraints are imposed only if the value of such variables is true. For the case of nonconvex problems we refer, e.g., to (S. Lee & Grossmann, 2000; Turkay & Grossmann, 1996).

It is also worthwhile to observe that for some mixed integer problems, some relaxations have been proposed which rely on the addition of further discrete variables. For instance, in (Saxena, Bonami, & Lee, 2010) mixed integer quadratic programs

$$\begin{array}{ll} \min & \mathbf{c}^T\mathbf{x} \\ & \mathbf{x}^T\mathbf{A}_i\mathbf{x} + \mathbf{a}_i^T\mathbf{x} + a_{i0} \leq 0, \quad i = 1,\dots,m, \\ & \boldsymbol{\ell} \leq \mathbf{x} \leq \mathbf{u}, \\ & x_j \in \mathbb{Z}, \qquad\qquad\qquad j \in J \subseteq \{1,\dots,n\}, \end{array}$$

are considered. These can be reformulated as follows, adding the matrix variable $\mathbf{X}$:

$$\begin{aligned} \min \quad & \mathbf{c}^T\mathbf{x} \\ & \mathbf{A}_i \bullet \mathbf{X} + \mathbf{a}_i^T\mathbf{x} + a_{i0} \le 0, \quad i = 1,\dots,m, \\ & \ell \le \mathbf{x} \le \mathbf{u}, \\ & x_j \in \mathbb{Z}, \qquad\qquad\qquad j \in J \subseteq \{1,\dots,n\}, \\ & \mathbf{X} = \mathbf{x}\mathbf{x}^T. \end{aligned}$$

For any $\mathbf{z} \in \mathbb{R}^n$, we have that

$$(\mathbf{z}^T\mathbf{x})^2 = \mathbf{z}^T\mathbf{X}\mathbf{z},$$

which can be split into the two inequalities

$$(\mathbf{z}^T\mathbf{x})^2 \le \mathbf{z}^T\mathbf{X}\mathbf{z}, \quad (\mathbf{z}^T\mathbf{x})^2 \ge \mathbf{z}^T\mathbf{X}\mathbf{z}.$$

The former is a convex inequality, while for the latter we have the (2-term) disjunction

$$\begin{aligned} &\text{either } \ell_{\mathbf{z}} \le \mathbf{z}^T\mathbf{x} \le \theta, \quad \mathbf{z}^T\mathbf{x}(\ell_{\mathbf{z}} + \theta) - \theta\ell_{\mathbf{z}} \ge \mathbf{z}^T\mathbf{X}\mathbf{z}, \\ &\text{or } \theta \le \mathbf{z}^T\mathbf{x} \le u_{\mathbf{z}}, \quad \mathbf{z}^T\mathbf{x}(u_{\mathbf{z}} + \theta) - \theta u_{\mathbf{z}} \ge \mathbf{z}^T\mathbf{X}\mathbf{z}, \end{aligned}$$

where $\ell_{\mathbf{z}}, u_{\mathbf{z}}$ are a lower and an upper bound for $\mathbf{z}^T\mathbf{x}$ over the feasible region, and θ is some value in the interval $(\ell_{\mathbf{z}}, u_{\mathbf{z}})$. The disjunction above corresponds to substituting in the nonconvex constraint $(\mathbf{z}^T\mathbf{x})^2 \ge \mathbf{z}^T\mathbf{X}\mathbf{z}$ the function $(\mathbf{z}^T\mathbf{x})^2$ with its piecewise linear overestimator for $\mathbf{z}^T\mathbf{x} \in [\ell_{\mathbf{z}}, u_{\mathbf{z}}]$, interpolating the function at the three points

$$\mathbf{z}^T\mathbf{x} = \ell_{\mathbf{z}}, \theta, u_{\mathbf{z}}.$$

The resulting constraint is still nonconvex, but we can deal with it by disjunctive programming techniques. In fact, the approach above can be generalized by using q-term ($q > 1$) disjunctions, i.e., by using piecewise linear overestimators which interpolate the quadratic function at $q + 1$ points.

A similar idea has been employed in (Bergamini, Grossmann, Scenna, & Aguirre, 2008), where piecewise linear underestimators are employed in place of concave univariate and bilinear terms, with the introduction of additional binary variables which allow us to identify the different pieces of the underestimators.

Piecewise linear underestimators, where the number of additional binary variables scales linearly or logarithmically with respect to the number of linear pieces, are also discussed, e.g., in (Misener, Thompson, & Floudas, 2011; Vielma, Ahmed, & Nemhauser, 2010; Vielma & Nemhauser, 2011; Wicaksono & Karimi, 2008).

As a final remark, we notice that the relaxations we have discussed up to now return rigorous lower bounds under the nontrivial assumption that all computations are performed in exact arithmetic. In practice, one should always take into account numerical errors if rigorous and reliable results are required. For instance, the issue of defining *safe* linear relaxations has been dealt with in different works, such as (Borradaile & Van Hentenryck, 2005; Jansson, 2003; Kearfott, 2011; Neumaier & Shcherbina, 2004).

Chapter 5

Branch and Bound

5.1 Introduction

In this chapter we discuss branch-and-bound (BB) approaches for solving a GO problem. Under suitable assumptions, such approaches allow us to detect in a finite time or, at least, to converge to a globally optimal solution of the problem (if any exists). Throughout the chapter GO problems with the form

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ & g_i(\mathbf{x}) \leq 0, \quad i = 1,\ldots,m, \\ & \mathbf{x} \in X, \end{aligned} \tag{5.1}$$

are considered. It is assumed that

- X is a nonempty compact convex set;
- the functions f and g_i, $i = 1,\ldots,m$, are continuous over X.

Then, the feasible region

$$S = \{\mathbf{x} \in X \ : \ g_i(\mathbf{x}) \leq 0, \ i = 1,\ldots,m\}$$

is a compact set and the Weierstrass theorem guarantees the existence of a global optimal solution if $S \neq \emptyset$. We denote by $f^\star$ the minimum value (we set $f^\star = +\infty$ if $S = \emptyset$). Note that, unless the functions g_i's are of "simple" form (say, convex functions), even establishing whether $S = \emptyset$ might be a hard task. In order to take this fact into account we first need to introduce the definitions of $\boldsymbol{\delta}$-feasible and $(\varepsilon, \boldsymbol{\delta})$-optimal solutions.

Definition 5.1. *Given a vector $\boldsymbol{\delta} \in \mathbb{R}^m_+$, a point $\mathbf{x} \in X$ is a $\boldsymbol{\delta}$-*feasible *point for* (5.1) *if*

$$g_i(\mathbf{x}) \leq \delta_i, \quad i = 1,\ldots,m.$$

The set of $\boldsymbol{\delta}$-feasible solutions is denoted by $S_{\boldsymbol{\delta}}$.

If $\mathbf{x}^\star \in S_{\boldsymbol\delta}$ is such that

$$f(\mathbf{x}^\star) \le f(\mathbf{x}) \quad \forall \mathbf{x} \in S_{\boldsymbol\delta},$$

then $\mathbf{x}^\star$ is called a *$\boldsymbol\delta$-optimal solution*. Note that a different value δ_i for each constraint is used. This makes sense because, e.g., $\delta_i = 0$ might be set for simple constraints like the convex or even linear ones, while a strictly positive δ_i value might be used for nonconvex constraints.[6]

Definition 5.2. *Given a vector $\boldsymbol\delta \in \mathbb{R}^m_+$ and a scalar $\varepsilon \ge 0$, a point $\mathbf{x}^\star$ is an* $(\varepsilon,\boldsymbol\delta)$-optimal solution *for (5.1) if it is $\boldsymbol\delta$-feasible and*

$$f(\mathbf{x}^\star) \le f^\star + \varepsilon.$$

In particular, if $\mathbf{x}^\star \in S$, $f^\star < \infty$ is obviously true, and $\mathbf{x}^\star$ is called an ε-optimal solution.

In the following Algorithm, the general scheme of a BB approach for (5.1) is presented; some details on its individual steps are discussed in what follows.

1. Initialization

```
Data: $\boldsymbol\delta \ge \mathbf{0}$, $\varepsilon \ge 0$
set $\mathcal{C}_0 = \{S\}$, $t = 0$
Compute a lower bound $lb(S)$ for (5.1)
if $\bar{\mathbf{x}} \in S_{\boldsymbol\delta}$ is available then
    set $ub_{\boldsymbol\delta} = f(\bar{\mathbf{x}})$
    set $\mathbf{z} = \bar{\mathbf{x}}$
else
    set $ub_{\boldsymbol\delta} = \infty$
    leave $\mathbf{z}$ undetermined
end
if $ub_{\boldsymbol\delta} \le lb(S) + \varepsilon$ then
    if $ub_{\boldsymbol\delta} < \infty$ then
        return the $(\varepsilon,\boldsymbol\delta)$-optimal solution $\mathbf{z}$
    else
        return $S = \emptyset$
    end
else
    Go to Step 2.
end
```

2. Node selection

Select a subset $S_k \in \mathcal{C}_t$ according to some rule.

3. Branching

Subdivide S_k into $r \ge 2$ subsets $S_{1k},\dots,S_{rk}$ such that $\cup_{i=1}^r S_{ik} = S_k$ and set $\mathcal{C}_{t+1} = \mathcal{C}_t \cup \{S_{1k},\dots,S_{rk}\} \setminus \{S_k\}$.

4. Domain reduction

Possibly reduce the subsets $S_{1k},\dots,S_{rk}$ through the application of domain reduction strategies.

[6] In fact, in order to take into account numerical errors, one can set $\delta_i > 0$ even when the function g_i is convex or even linear.

5. Lower bound computation

Compute a lower bound $lb(S_{ik})$, $i = 1,\ldots,r$, for all the newly generated subsets.

6. Upper bound update

```
Let $T_\delta$ be a (possibly empty) set of $\boldsymbol{\delta}$-feasible points detected during the lower
bound computations
if $\min_{\mathbf{y}\in T_\delta} f(\mathbf{y}) < ub_\delta$ then
    $ub_\delta = \min_{\mathbf{y}\in T_\delta} f(\mathbf{y})$
    $\mathbf{z} \in \arg\min_{\mathbf{y}\in T_\delta} f(\mathbf{y})$
end
```

7. Fathoming

Update $\mathcal{C}_{t+1}$ by discarding all subsets whose lower bound is larger than $ub_\delta - \varepsilon$, i.e.,

$$\mathcal{C}_{t+1} = \mathcal{C}_{t+1} \setminus \{Q \in \mathcal{C}_{t+1} \ : \ lb(Q) \geq ub_\delta - \varepsilon\}. \tag{5.2}$$

If available, apply other fathoming rules in order to (possibly) discard further subsets from $\mathcal{C}_{t+1}$.

8. Stopping rule

```
if $\mathcal{C}_{t+1} = \emptyset$ then
    if $ub_\delta < \infty$ then
        return the $(\varepsilon, \boldsymbol{\delta})$-optimal solution $\mathbf{z}$
    else
        return $S = \emptyset$
    end
else
    Set $t = t + 1$ and go back to Step 2
end
```

The evolution of a BB algorithm can be represented through a so-called BB tree, whose nodes correspond to each subset generated during the execution of the algorithm. In particular, the branching operation at Step 3 generates r *child nodes* $S_{1k},\ldots,S_{rk}$ of a given *father node* S_k. The original feasible region S is the root node of the BB tree. At iteration t, the collection $\mathcal{C}_t$ will contain all the leaves of the BB tree not yet fathomed.

In the following sections the steps of the BB approach are discussed and the conditions under which the approach is able to detect (or at least to converge to) a globally optimal solution are analyzed. The sections do not follow the order in which the steps appear in the algorithm. We first briefly present in Section 5.2 all those steps whose discussion does not need to be too detailed, either because they were already discussed in other parts of the book (in particular, this is true for Step 5 about the lower bound computation) or because they are standard ones, already widely investigated in the literature about BB approaches for mixed integer linear programs, with just small peculiarities for the nonconvex problem (5.1). Next, in Section 5.3 we describe different branching operations. Section 5.4 is dedicated to the proofs of convergence (for $\varepsilon = 0, \boldsymbol{\delta} = \mathbf{0}$) and finiteness (for $\varepsilon > 0, \boldsymbol{\delta} > \mathbf{0}$) under suitable assumptions. Domain reduction techniques, which are not needed to prove convergence and finiteness results, have instead a quite relevant practical impact and are discussed in Section 5.5. Fathoming rules different from the standard one (5.2) are introduced in Section 5.6. They are also not necessary for the convergence and finiteness results, but for some problems they allow us to prove finiteness of the BB approach even when $\varepsilon = 0, \boldsymbol{\delta} = \mathbf{0}$, as discussed in Section 5.7.

5.2 Standard or previously discussed steps

5.2.1 Initialization step

The initialization step is a quite standard one. The collection $\mathcal{C}_0$ of subsets of S is initialized with the set S itself. With respect to the BB approaches for integer programs, some care is needed to take into account the already mentioned difficulty of identifying feasible points even in the case of a nonempty feasible region. Therefore, the upper bound value $ub_{\boldsymbol{\delta}}$ is not the best-observed objective function value at feasible points but rather at $\boldsymbol{\delta}$-feasible points. As long as no $\boldsymbol{\delta}$-feasible point is observed, the value of $ub_{\boldsymbol{\delta}}$ is set equal to ∞. The detection of a feasible or $\boldsymbol{\delta}$-feasible point can be carried on through any available heuristic for the problem (see Chapter 3).

5.2.2 Node selection

The rules for the selection of a node/subset in $\mathcal{C}_t$ do not usually differ from those also employed in the BB approaches for integer programs. One such rule is

$$\text{select } \; S_k \in \arg\min_{Q \in \mathcal{C}_t} lb(Q), \tag{5.3}$$

i.e., a node/subset with the minimum lower bound value among all those not yet fathomed is chosen. This might be called a `best-first` rule. The rationale behind it is that it is worthwhile to explore nodes with a small lower bound, as they are more likely to contain good feasible (or $\boldsymbol{\delta}$-feasible) solutions whose detection allows for a quick decrease of the upper bound value $ub_{\boldsymbol{\delta}}$ and, thus, for a quick fathoming of nodes/subsets. However, a possible drawback of this rule is that it might result in a *breadth-first* investigation of the BB tree, with a quick increase of the memory requirements for storing the information about nodes/subsets in $\mathcal{C}_t$. An alternative is a *depth-first* investigation of the BB tree, where we always select the left-most leaf of the BB tree among those still in $\mathcal{C}_t$. We call this a `depth-first` rule. The memory requirements for it are not too high, but, on the other hand, since it ignores the information given by the lower bounds at the different nodes, it usually decreases the upper bound $ub_{\boldsymbol{\delta}}$ more slowly with respect to the `best-first` rule, thus resulting in a larger BB tree. Moreover, some more care is needed when dealing with nonconvex problems. Differently from BB for bounded integer problems, the BB tree for nonconvex problems might have an infinite depth, and in this case a pure `depth-first` strategy could reduce to following a single infinite descending path through the BB tree, without being able to detect the optimal solution.

Obviously, the `best-` and `depth-first` strategies are not the only possible ones (we refer, e.g., to (Casado, Martinez, & Garcia, 2001; Csendes, 2001) for an alternative selection rule within the framework of interval methods). In fact, we can observe that the `best-` and `depth-first` strategies are somehow complementary, and a hybrid rule is usually a good option. In the first levels of the BB tree a `best-first` strategy is employed, while in the lower levels a `depth-first` strategy is employed most of the time.

This way, during the first iterations of the BB approach a good feasible solution is searched for through the `best-first` strategy while in the last iterations, when, very often, the global minimum point, or at least a good approximation for it, has been already detected but not yet certified as such, we switch to the `depth-first` strategy in order to avoid the explosion of the memory requirements. However, we underline that the `best-first` strategy plays an important role for the convergence proof of BB algorithms. In particular, as we see in Section 5.4, convergence of the BB algorithm can be proven if the `best-first` strategy is employed "sufficiently often." This is stated through the notion of *bound improving selection operation*.

Definition 5.3. *A selection operation is said to be* bound improving *if the number of successive iterations in which the subdivision rule is different from the* `best-first` *one* (5.3) *is always finite.*

5.2.3 Lower bound computation

Given a subset S_k of the feasible region S, we denote by $lb(S_k)$ some value satisfying

$$lb(S_k) \leq \min_{\mathbf{x}\in S_k} f(\mathbf{x}). \tag{5.4}$$

The importance of computing lower bounds within the BB scheme is such that we have dedicated the whole of Chapter 4 to this subject. Here we only briefly recall a few facts. When dealing with the nonconvex problem (5.1), one might directly work on its original formulation. In this case, the set S_k is defined as

$$S_k = \{\mathbf{x} \in X \ : \ g_i(\mathbf{x}) \leq 0,\ i = 1,\ldots,m,\ h_j(\mathbf{x}) \leq 0,\ j = 1,\ldots,t\},$$

where the constraints $h_j(\mathbf{x}) \leq 0,\ j = 1,\ldots,t,$ are (usually simple) constraints added to the original ones defining S along the branches which lead from the root node S to the current node S_k (we are more specific about this in Section 5.3 about branching rules). Very simple examples are $h_j(\mathbf{x}) = x_r - u_r$ or $h_j(\mathbf{x}) = \ell_r - x_r$, which introduce simple bounds (respectively, upper and lower bounds) for a variable x_r. The lower bound $lb(S_k)$ is then computed through the solution of a simpler problem (a convex or even linear one) where the functions $f, g_i, i = 1,\ldots,m$, are substituted by convex underestimators $\hat{f}^{D_k}, \hat{g}_i^{D_k}$, which are valid over some region $D_k \supseteq S_k$. That is,

$$\begin{aligned} lb(S_k) \ = \ \min \quad & \hat{f}^{D_k}(\mathbf{x}) \\ & \hat{g}_i^{D_k}(\mathbf{x}) \leq 0, \quad i = 1,\ldots,m, \\ & h_j(\mathbf{x}) \leq 0, \qquad j = 1,\ldots,t, \\ & \mathbf{x} \in X. \end{aligned} \tag{5.5}$$

Of course, if some of the functions f, g_i are already convex ones the substitution with a convex underestimator is not necessary. The issue of the derivation of convex underestimators was extensively studied in Chapter 4. Recall that (5.4) holds because

$$\hat{S}_k = \{\mathbf{x} \in X \ : \ \hat{g}_i^{D_k}(\mathbf{x}) \leq 0,\ i = 1,\ldots,m,\ h_j(\mathbf{x}) \leq 0,\ j = 1,\ldots,t\} \supseteq S_k,$$

$$\text{and} \quad \hat{f}^{D_k}(\mathbf{x}) \leq f(\mathbf{x}) \quad \forall\, \mathbf{x} \in S_k,$$

i.e., problem (5.5) is a relaxation of the problem $\min_{\mathbf{x}\in S_k} f(\mathbf{x})$. Of course, $\hat{S}_k = \emptyset$ implies $S_k = \emptyset$. In this case we set $lb(S_k) = +\infty$, so that the node/subset is certainly fathomed. As we will see in Section 5.4, convex underestimators should satisfy the property of exactness in the limit, stating that by reducing a region D_k to a singleton, the values of the convex underestimators over D_k approach the values of the corresponding underestimated functions.

Definition 5.4. *The convex underestimators $\hat{f}^{D_k}$, $\hat{g}_i^{D_k}$, $i = 1,\dots,m$, over some region D_k are said to satisfy the property of* exactness in the limit *if*

$$\max\left\{\max_{\mathbf{x}\in D_k}[f(\mathbf{x}) - \hat{f}^{D_k}(\mathbf{x})], \max_{\mathbf{x}\in D_k, i=1,\dots,m}[g_i(\mathbf{x}) - \hat{g}_i^{D_k}(\mathbf{x})]\right\} \leq \eta(diam(D_k)),$$

where

$$diam(D_k) = \max_{\mathbf{x},\mathbf{y}\in D_k} \|\mathbf{x}-\mathbf{y}\|_2$$

is the diameter of D_k, and $\eta(r)$ is a continuous nondecreasing function such that

$$\lim_{r\to 0} \eta(r) = 0.$$

An alternative to working directly with the original formulation is that of first reformulating problem (5.1) into some equivalent problem, and then considering relaxations of the reformulated problem. Again, we can refer to Chapter 4: examples of reformulations have been introduced, e.g., in Sections 4.4.1 and 4.4.2, where QP problems have been reformulated, respectively, as linear problems over the cone of completely positive matrices and as problems with a linear objective function and complementarity constraints, after imposing the KKT conditions and, thus, adding the variables corresponding to the Lagrange multipliers.

5.2.4 Upper bound update

The update of the upper bound value should take into account the difficulties related even to the detection of feasible solutions for problem (5.1). The computation of the lower bounds in Step 5 of the algorithm might deliver some feasible or, at least, $\boldsymbol{\delta}$-feasible points. For instance, if all the functions g_i, $i = 1,\dots,m$, are convex ones, so that they do not need to be substituted by convex underestimators in the relaxed problem (5.5), each optimal solution of the relaxed problem belongs to S and we can evaluate f at it. If at least one of the newly detected $\boldsymbol{\delta}$-feasible points has a function value lower than $ub_{\boldsymbol{\delta}}$, then the value $ub_{\boldsymbol{\delta}}$ is updated with the best function value observed among the newly detected $\boldsymbol{\delta}$-feasible points, and the point $\mathbf{z}$ is updated accordingly. It is important to observe that, unless $\boldsymbol{\delta} = \mathbf{0}$, $ub_{\boldsymbol{\delta}}$ is *not* necessarily an upper bound for the optimal value $f^\star$ of the problem, since $\boldsymbol{\delta}$-feasible points outside S might have objective function value lower than $f^\star$ (this is why we put in evidence the dependency of the upper bound from $\boldsymbol{\delta}$). We also make the following remark.

Remark 5.1. *Since points in $\mathbf{y} \in T_{\boldsymbol{\delta}}$ are usually optimal solutions of the relaxations solved to compute $lb(S_{ik})$, we have that $f(\mathbf{y}) \geq lb(S_{ik})$.*

Finally, we note that heuristic methods can also be applied at nodes of the BB tree in order to possibly update the upper bound.

5.2.5 Standard fathoming rule

The standard fathoming rule (5.2) is completely analogous to the corresponding one in BB approaches for integer programs. It asks for discarding from further consideration a node/subset S_k if $lb(S_k) \geq ub_\delta - \varepsilon$ because in such node we are guaranteed that no δ-feasible solution with function value lower than $ub_\delta - \varepsilon$ can be detected. In some cases other fathoming rules can be applied. These will be discussed in Section 5.6.

5.2.6 Stopping rule

The stopping rule is the standard one for BB approaches: we stop as soon as all the nodes/subsets of the BB tree have been fathomed. Some care is needed when considering the output of the algorithm. If $ub_\delta = \infty$ at stopping, then the algorithm returns a clear conclusion. Indeed, if ub_δ is still equal to ∞, this means that the feasible region of the problem is empty. A less clear conclusion occurs when $ub_\delta < \infty$. In this case the algorithm returns a (ε, δ)-optimal solution $\mathbf{z}$ but, in fact, we have no guarantee even that $S \neq \emptyset$. We can only return a "not-too-unfeasible" (δ-feasible) solution $\mathbf{z}$ for which we can guarantee that, *if* $S \neq \emptyset$, it has function value which can not be larger than $f^\star + \varepsilon$. As already mentioned, this uncertainty in the output is due to the possible difficulty of detecting feasible points for problems with nonconvex constraints, and is basically equivalent to the difficulty of finding the exact global optimum value for a nonconvex problem (so that we usually need to choose a strictly positive value for ε). Although not explicitly included in the above pseudo code of the BB algorithm, one can also keep track of the best *feasible* solution observed during the execution of the algorithm but, again,

Case a) there is no guarantee that a feasible solution will be observed even if $S \neq \emptyset$;

Case b) even if feasible points are observed, the best objective function value at them might be far away from the optimal value.

These situations will be further discussed in Section 5.4 also through some examples.

5.3 Branching

Branching is the operation which allows us to subdivide a given set S_k (a subset of the feasible region) into a finite number $r \geq 2$ of subsets $S_{1k}, \ldots, S_{rk}$. The subdivision is usually obtained by adding constraints. If the original set S_k is defined by some constraints, each of the subsets S_{ik}, $i = 1, \ldots, r$, defined by the branching operation is obtained by adding one or more constraints to those defining S_k, in such a way that each point in S_k belongs to (at least) one subset S_{ik}, $i = 1, \ldots, r$, i.e., in such a way that $\cup_{i=1}^r S_{ik} = S_k$. In the BB approaches for integer programs it is usually guaranteed that each point in S belongs to *exactly* one subset S_{ik}, $i = 1, \ldots, r$, i.e., the collection of subsets S_{ik}, $i = 1, \ldots, r$, is a partition of S. As will be seen, this is not always the case in the field of nonconvex problems. In this field branching operations which we will call *geometric branching* ones are often employed. By geometric branching we mean that the bounded feasible region of the problem is enclosed within a set $D_0 \supset S$ with a given "simple" geometrical shape, and each subset corresponding to a node of the BB tree is enclosed into a set with the same geometrical shape. In other words, a geometric branching operation subdivides a set with a fixed geometrical shape into subsets with the same shape. "Simplicity" of the geometrical shape is related to the simplicity of computing bounds over regions with that shape, e.g., because it is simple to compute convex underestimators over such regions or because minimizing the original objective function over these regions is a simple task. In Sections 5.3.1–5.3.4 we will discuss different possible geometrical shapes used in the literature, namely hyperrectangles, simplices, cones, ellipsoids (the order of presentation more or less reflects the frequency of use of these shapes in the BB approaches proposed up to now, from the most frequent ones down to the least frequent ones). As will be seen in Section 5.4, a feature required by geometric branching in order to prove convergence of the BB algorithm is *exhaustiveness*. If the BB algorithm does not stop after a finite number of iterations, then it generates at least an infinite *nested sequence* of nodes/subsets

$$\{D_{k_j}\}_{j=0}^{\infty} \ : \ D_{k_0} = D_0, \ \ k_j < k_{j+1}, \ \ D_{k_{j+1}} \subseteq D_{k_j} \quad \forall\, j = 0,1,\ldots \tag{5.6}$$

Then, the exhaustiveness property is defined as follows.

Definition 5.5. *A BB algorithm equipped with a geometric branching rule possesses the* exhaustiveness property *if each infinite nested sequence* (5.6) *generated by the algorithm converges to a singleton*

$$\cap_{j=0}^{\infty} D_{k_j} = \{\bar{\mathbf{x}}\}, \quad \bar{\mathbf{x}} \in D_0.$$

or, equivalently,

$$diam(D_{k_j}) \to 0, \quad as \ \ j \to \ \infty.$$

More peculiar geometrical shapes, which can usually be employed for problems with a special structure, are introduced in Section 5.3.5. Finally, we point out that, though extensively used for nonconvex problems, geometric branching operations are not the only option. In Section 5.3.6 we will discuss branching operations based on the KKT conditions for QP problems, which we have already met in Section 4.4.2.

5.3.1 Hyperrectangles

Boxes or hyperrectangles are probably the most widely employed geometrical shapes within BB approaches for nonconvex problems. It is difficult to trace back the first time when they were used (an application for problems with a separable nonconvex objective function can be found in (Falk & Soland, 1969), while an application to problems with a rational objective function can be found in (Skelboe, 1974), where a development of a method proposed in (Moore, 1962) is given). If lower and upper bounds on the variables are not already given in the formulation of the problem, one can compute them by solving the following two convex problems for each variable x_j, $j = 1,\ldots,n$,

$$\ell_j/u_j = \min/\max_{\mathbf{x}\in X} x_j, \tag{5.7}$$

so that the feasible region S is certainly enclosed in the box $B = \prod_{j=1}^{n}[\ell_j,u_j]$. A box

$$B_k = \prod_{j=1}^{n}[\ell_j(S_k),u_j(S_k)] \subseteq \prod_{j=1}^{n}[\ell_j,u_j]$$

is associated to each node/subset S_k of the BB tree and the subset S_k is equal to $S \cap B_k$. Boxes are appealing because they often simplify the computation of a lower bound. For instance, as already seen in Chapter 4, convex underestimators (sometimes even convex envelopes) over boxes can be in many cases obtained in a cheap way. For some bound computation techniques, like those based on interval arithmetic (see Section 4.10) and the α-BB approaches (see Section 4.6), boxes are the most natural choice.

The branching operation on the subset S_k is defined as follows.

- Choose a point $\mathbf{y} \in B_k = \prod_{j=1}^{n}[\ell_j(S_k), u_j(S_k)]$ which is not a vertex of B_k.
- Let
$$J = \{j \in \{1,\dots,n\} \; : \; \ell_j(S_k) < y_j < u_j(S_k)\}$$
(note that $J \neq \emptyset$ since $\mathbf{y}$ is not a vertex of B_k).
- Split each interval $I_j^k = [\ell_j(S_k), u_j(S_k)]$, $j \in J$, into the two subintervals
$$I_j^{1k} = [\ell_j(S_k), y_j], \quad I_j^{2k} = [y_j, u_j(S_k)],$$
while all the intervals $I_j^k = [\ell_j(S_k), u_j(S_k)]$, $j \notin J$, are left unchanged.
- Subdivide the box $\prod_{j=1}^{n}[\ell_j(S_k), u_j(S_k)]$ into the $2^{|J|}$ subboxes $B_{1k}, \dots, B_{2^{|J|}k}$ obtained by taking all the possible combinations of the above subintervals I_j^{qk} for $q = 1,2$, $j \in J$, and I_j^k, for $j \notin J$.

Example 5.6. Let us consider the box $B = [0,2] \times [1,3] \times [0,1]$ and the point $\mathbf{y} = (1\ 2\ 1)$. Then, $J = \{1,2\}$. The interval $[0,2]$ is split into $[0,1]$ and $[1,2]$, the interval $[1,3]$ is split into $[1,2]$ and $[2,3]$, and the last interval $[0,1]$ is left unchanged. The original box is then subdivided into the 2^2 subboxes

$$\begin{array}{ll} B_1 = [0,1] \times [1,2] \times [0,1] & B_2 = [0,1] \times [2,3] \times [0,1] \\ B_3 = [1,2] \times [1,2] \times [0,1] & B_4 = [1,2] \times [2,3] \times [0,1]. \end{array} \quad \blacksquare$$

Remark 5.2. *As already mentioned, opposite to the BB approaches for integer programs, where the branching operations usually generates* disjoint *subsets, when employing the above subdivisions into subboxes we can guarantee that $\cup_{i=1}^{2^{|J|}} S_{ik} = S_k$, but the subsets are not pairwise disjoint. This can be easily seen from Example* 5.6, *where we notice, e.g., that the subboxes B_1 and B_2 share the common facet*

$$[0,1] \times \{2\} \times [0,1].$$

However, it can always be guaranteed that the interiors *$int(B_{ik})$ of the subboxes are disjoint, i.e.,*

$$int(B_{ik}) \cap int(B_{lk}) = \emptyset \quad \forall\, i,l = 1,\dots,2^{|J|},\ i \neq l.$$

What has still to be specified in the above branching operation is how the point $\mathbf{y}$ is chosen. Different rules can be employed. The choice of the rule has a relevant practical impact. Some rules have been evaluated for some BB algorithms in (Tuy, 1991a). The simplest, and probably the most widely employed, is the so-called *bisection rule*. To our knowledge, the first application dates back to (Horst, 1976) in the context of BB approaches based on simplices (see Section 5.3.2). The rule is defined as follows: take $\mathbf{y}$ as the midpoint of one of the longest edges in B_k, i.e., first select

$$h \in \arg \max_{j=1,\dots,n} [u_j(S_k) - \ell_j(S_k)], \tag{5.8}$$

and then set

$$y_j = \begin{cases} \ell_j(S_k) \ \text{(or } u_j(S_k)), & j \neq h, \\ \frac{\ell_j(S_k)+u_j(S_k)}{2}, & j = h. \end{cases} \tag{5.9}$$

Note that by this definition $J = \{h\}$ and the box B_k is subdivided into only two subboxes of equal volume. A mild variant of the bisection rule selects edge h according to rule (5.8) but in the definition (5.9) for the point $\mathbf{y}$, the value y_h can be selected within $(\ell_h(S_k), u_h(S_k))$ and it is not necessarily the midpoint of this interval. For instance, think about the case of a separable objective function $\sum_{j=1}^n f_j(x_j)$ underestimated by the separable convex function $\sum_{j=1}^n \hat{f}_j(x_j)$. In this case, rather than considering the midpoint of the interval $[\ell_h(S_k), u_h(S_k)]$, we might consider the point in this interval where the distance between the function f_h and its convex underestimator $\hat{f}_h$ is largest, or we might even select the interval (not necessarily the largest one) where the largest distance between a function and its underestimator is observed, and the subdivision along this interval is performed at the point where the largest distance is attained (see, e.g., (Schectman & Sahinidis, 1998)). Another variant is to select the edge h rather than through (5.8) by a more general rule,

$$h \in \arg \max_{j=1,\dots,n} \lambda_j p(u_j(S_k) - \ell_j(S_k)),$$

where p is an increasing function such that $p(0) = 0$, and $\lambda_j > 0$ is a positive weight measuring the curvature of the nonlinear part of f_j (the higher the curvature, the higher the weight), thus making the algorithm more sensitive to variables for which the convex underestimation of f_j is least sharp (see, e.g., (Kalantari & Rosen, 1987) for the case of separable concave quadratic functions, where λ_j is related to the coefficient of $-x_j^2$, and $p(t) = t^2$). In (Burer & Chen, 2011), if $\mathbf{x}^\star$ denotes the optimal solution of the current relaxation over S_k, the following quantities are computed for each coordinate $j = 1, \dots, n$:

$$\begin{aligned} \alpha_j &= \frac{(u_j(S_k)-x_j^\star)(x_j^\star-\ell_j(S_k))}{(u_j(S_k)-\ell_j(S_k))^2}, \\ \beta_j &= u_j(S_k) - \ell_j(S_k). \end{aligned}$$

Then,

$$h \in \arg \max_{j=1,\dots,n} \alpha_j \beta_j,$$

and the hth interval is subdivided with respect to $x_h^\star$. This way the selection of the interval to be subdivided takes into account both the longest edge (through the β values) and the largest distance of $\mathbf{x}^\star$ from the extremes of the intervals (through the α values).

The original bisection rule is a purely geometrical rule: given the box, the resulting subdivision is completely independent from the problem to be solved. The above alternatives are an attempt to include some knowledge about the problem into the subdivision rule; the choice of the interval to be subdivided and/or the point at which the interval is subdivided are problem dependent. A rule which is strongly problem dependent is the so-called *ω-subdivision rule*, first introduced in (Tuy, 1964) in the context of BB approaches based on cones (see Section 5.3.3) and later further studied in (Bali, 1973; Zwart, 1974): if a lower bound over S_k is computed by solving the relaxed problem (5.5), then its optimal solution, denoted by $\boldsymbol{\omega}(S_k)$, is a point within the box B_k and we select it as the point $\mathbf{y}$ with respect to which we perform the subdivision. It often happens that convex underestimating functions are exact at the vertices of a box. Therefore, if an ω-subdivision rule is employed, the values of the underestimators of the objective and constraint functions over the subboxes B_{ik}, $i = 1,\dots,2^{|J|}$, will be equal to the values of the functions themselves at $\boldsymbol{\omega}(S_k)$, while in the box B_k the value at $\boldsymbol{\omega}(S_k)$ of at least one of the convex underestimators has to be far enough from the value of the original functions (otherwise, the subset S_k would be fathomed at Step 7 of the BB algorithm). Basically, we are improving the quality of the convex understimators by pushing them up at $\boldsymbol{\omega}(S_k)$, and we are somehow cutting this point away. Such a point will not be the optimal solution of the relaxation in each subbox of B_k, unless the subbox is immediately fathomed. We illustrate all this through an example.

Example 5.7. Consider the concave minimization problem

$$\begin{aligned} \min \quad & -x_1^2 - x_2^2 - 3x_1 \\ & x_1 + 3x_2 \le 7, \\ & 3x_1 + x_2 \le 7, \\ & 0 \le x_1, x_2 \le 2, \end{aligned}$$

whose feasible region is the polytope represented in Figure 5.1. Consider the box $B = [0,2] \times [0,2] \supset S$. A relaxed problem over this box is

$$\begin{aligned} \min \quad & -5x_1 - 2x_2 \\ & x_1 + 3x_2 \le 7, \\ & 3x_1 + x_2 \le 7, \\ & 0 \le x_1, x_2 \le 2, \end{aligned}$$

whose optimal solution is $\boldsymbol{\omega}(B) = (\frac{7}{4}\ \frac{7}{4})$. Since this is a feasible solution of the problem, we can evaluate the objective function at it (the value is $-\frac{91}{8}$) and, consequently, $ub_{\mathbf{0}} \le -\frac{91}{8}$. Notice that the value of the convex underestimator over B of the objective function evaluated at $\boldsymbol{\omega}(B)$ is equal to $-\frac{49}{4}$. Then, the ω-subdivision rule splits the original box into

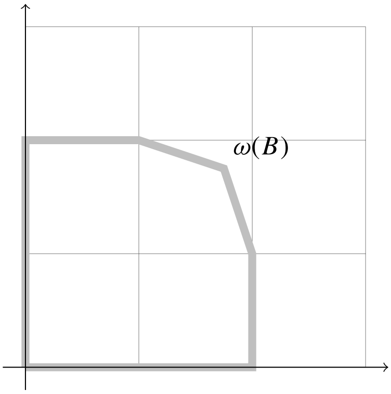


Figure 5.1. *Feasible set for the example (in gray) and optimal solution of the relaxed LP*

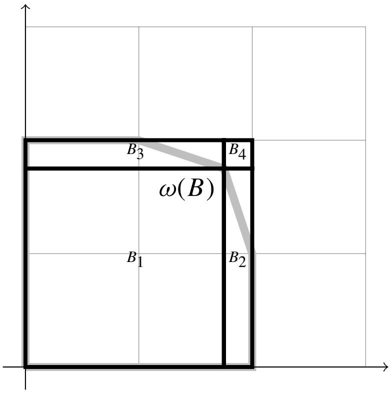


Figure 5.2. *Splitting of the feasible set at $\omega(B)$*

the four subboxes

$$B_1 = \left[0,\tfrac{7}{4}\right] \times \left[0,\tfrac{7}{4}\right], \quad B_2 = \left[\tfrac{7}{4},2\right] \times \left[0,\tfrac{7}{4}\right],$$
$$B_3 = \left[0,\tfrac{7}{4}\right] \times \left[\tfrac{7}{4},2\right], \quad B_4 = \left[\tfrac{7}{4},2\right] \times \left[\tfrac{7}{4},2\right].$$

Figure 5.2 reports these four boxes. In each box it is possible to redefine the convex underestimator of the objective function so that, in particular, it will be equal to the objective function value at $\boldsymbol{\omega}(B)$. Therefore, the value of the convex underestimator over the four subboxes is pushed up from the value $-\frac{49}{4}$ to the higher value $-\frac{91}{8}$ at $\boldsymbol{\omega}(B)$. If we consider, e.g., the subbox B_2, the convex underestimator of the objective function is

$$-\frac{27}{4}x_1 - \frac{7}{4}x_2 + \frac{7}{2},$$

and the relaxed problem is

$$\begin{aligned}\min\quad & -\tfrac{27}{4}x_1-\tfrac{7}{4}x_2+\tfrac{7}{2}\\ & x_1+3x_2\leq 7,\\ & 3x_1+x_2\leq 7,\\ & \tfrac{7}{4}\leq x_1\leq 2,\\ & 0\leq x_2\leq \tfrac{7}{4},\end{aligned}$$

with an optimal solution $\boldsymbol{\omega}(B_2)=(2\ 1)$. If we consider the subbox B_3, the relaxed problem is

$$\begin{aligned}\min\quad & -\tfrac{19}{4}x_1-\tfrac{15}{4}x_2+\tfrac{7}{2}\\ & x_1+3x_2\leq 7,\\ & 3x_1+x_2\leq 7,\\ & \tfrac{7}{4}\leq x_2\leq 2,\\ & 0\leq x_1\leq \tfrac{7}{4},\end{aligned}$$

with an optimal solution $\boldsymbol{\omega}(B_3)=\boldsymbol{\omega}(B)$ and $lb(B_3\cap S)=-\frac{91}{8}\geq ub_{\mathbf{0}}$, so that the subset $S_3=S\cap B_3$ is fathomed. It is easy to see that the same is true for the other two subboxes B_1,B_4. Therefore, the point $\boldsymbol{\omega}(B)$ is somehow cut away in all four subboxes, in the sense that either it can not be the optimal solution of the relaxed problem over $S_i=S\cap B_i$, $i=1,\dots,4$ (which is true for the subbox B_2 in the example), or the subset is immediately fathomed (which, in this example, is indeed the case for the three other subboxes). ■

While defining the point with respect to which the subdivision is performed on the basis of the outcome of the lower bound computation is an appealing property of the ω-subdivision rule, a possible drawback is that the number of children of a given node S_k can be as large as 2^n (this is true if $\boldsymbol{\omega}(B_k)$ is in the interior of B_k and is exactly what happens in the example above), thus making the growth of the BB tree too fast. Moreover, using the ω-subdivision rule alone makes it difficult to prove that the branching operation has the exhaustiveness property (see Definition 5.5), which, as already noted, is needed to prove the convergence of the BB algorithm; this will be made clearer in Section 5.4. On the contrary, the exhaustiveness property for the bisection rule is easily proven.

We note that the subdivision rules suggested above all subdivide each edge of the box into (at most) two parts. Other rules have been proposed where an edge may be subdivided into more than two parts. Just to cite an example, in (Markót et al., 2006), within a BB approach based on interval analysis, the subdivision of a box is based on an index which evaluates both the quality of the function values within the box and the feasibility of the box. According to the value of the index, a simple subdivision of the longest edge in two parts (resulting in two subboxes), a subdivision of the two longest edges in two parts (resulting in four subboxes), or a subdivision of the two longest edges in three parts (resulting in nine subboxes), is performed. We also refer to (Csallner, Csendes, & Markót, 2000; Markót, Csendes, & Csallner, 1999) for the discussion of multiple bisection (simultaneous bisection with respect to more than one edge) and multisplitting (an edge is subdivided into more than two equal subintervals) rules, and for the presentation of different rules to select edges along which the subdivision has to be performed. Finally, even if that algorithm cannot be considered as a BB method, it seems worthwhile to recall also the subdivision rule used by DIRECT (see Section 3.1.8).

5.3.2 Simplices

A n-dimensional simplex F is a polytope defined by $n+1$ affinely independent vertices $\mathbf{v}^1,\dots,\mathbf{v}^{n+1}$, i.e.,

$$F = chull\{\mathbf{v}^1,\dots,\mathbf{v}^{n+1}\}.$$

For example, in $\mathbb{R}^2$ the simplices are the triangles. It is easy to find an initial simplex containing the feasible region S of (5.1). Indeed, assume that the region X is contained in the nonnegative orthant. If this is not the case, we can force it by first taking the lower bounds ℓ_j, $j=1,\dots,n$, computed in (5.7), and then making the change of variables $x'_j = x_j - \ell_j$, $j=1,\dots,n$. Next, we can solve the convex problem

$$\xi = \max_{\mathbf{x}\in X} \sum_{j=1}^{n} x_j. \tag{5.10}$$

Then, X (and, thus, S) is contained in the simplex F whose $n+1$ vertices are $\mathbf{v}^{n+1} = \mathbf{0}$ (the origin) and the n points

$$\mathbf{v}^j = \xi\mathbf{e}^j,$$

where $\mathbf{e}^j$ is the direction of the jth axis, $j=1,\dots,n$. In fact, in the definition (5.10) one might use any linear objective function $\sum_{j=1}^n c_j x_j$, with $c_j > 0$, $j=1,\dots,n$. In this case we need to modify the definition of the vertices $\mathbf{v}^j$, $j=1,\dots,n$, as follows:

$$\mathbf{v}^j = \frac{\xi}{c_j}\mathbf{e}^j.$$

Simplices are appealing because for some functions the computation of a convex underestimator over a simplex is a very simple task. In particular, if a function f is concave, then its convex underestimator (actually, its convex envelope) over the simplex F is equal to the affine function interpolating f at the vertices of F. If f is a DC function (see Section 4.7) with the DC decomposition $f(\mathbf{x}) = g(\mathbf{x}) - h(\mathbf{x})$, with g,h convex, then a convex underestimator of f over F is the sum of g with the affine function interpolating $-h$ at the vertices of F. The first use of simplices within BB approaches dates back to (Horst, 1976) in the context of concave minimization. Further BB approaches based on simplices have been proposed, e.g., in (Ban, 1983; Benson, 1982, 1985; Benson & Sayin, 1994; Horst, 1980; Locatelli & Thoai, 2000; Nast, 1996; Raber, 1998; Tam & Ban, 1985; Wood, 1991; Zhu & Kuno, 2005). Recently, we recall the use of simplicial partitions in (Bundfuss & Duer, 2008, 2009) to approximate the copositive cone, as already discussed in Section 4.4.1. A simplex $F_k = chull\{\mathbf{v}^{1,k},\dots,\mathbf{v}^{n+1,k}\}$ is associated to each node of the BB tree and the corresponding subset is $S_k = F_k \cap S$. Many things are similar to those already seen in the case of boxes. Also in this case the subdivision is performed with respect to a point in F_k. More precisely, the branching operation on the subset S_k is defined as follows.

- Choose a point $\mathbf{y} \in F_k$ which is not a vertex of F_k.

- Let
$$\mathbf{y} = \sum_{j=1}^{n+1} \lambda_j(\mathbf{y})\mathbf{v}^{j,k}, \quad \boldsymbol{\lambda}(\mathbf{y}) \in \Delta_{n+1},$$
where Δ_{n+1} is the $n+1$-dimensional unit simplex (see Definition A.3). In other words, $\boldsymbol{\lambda}(\mathbf{y})$ is the vector made up by the coefficients of the unique convex combination of the vertices of F_k returning $\mathbf{y}$.

- Let
$$J = \{j \in \{1,\ldots,n+1\} \ : \ \lambda_j(\mathbf{y}) > 0\}.$$
Since $\mathbf{y}$ is not a vertex of F_k, $|J| \geq 2$,

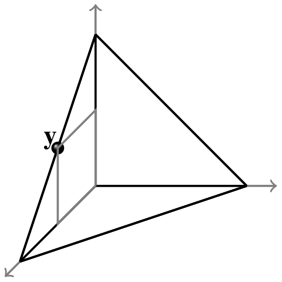
- Split F_k into the $|J|$ simplices
$$F_{ik} = chull\{\mathbf{v}^{1,k},\ldots,\mathbf{v}^{i-1,k},\mathbf{y},\mathbf{v}^{i+1,k},\ldots,\mathbf{v}^{n+1,k}\} \quad \forall\, i \in J.$$

Example 5.8. Let us consider the simplex

$$F = chull\left\{\begin{pmatrix}1\\0\\0\end{pmatrix},\begin{pmatrix}0\\1\\0\end{pmatrix},\begin{pmatrix}0\\0\\1\end{pmatrix},\begin{pmatrix}0\\0\\0\end{pmatrix}\right\},$$

and the point

$$\mathbf{y} = \begin{pmatrix}0\\\frac{1}{2}\\\frac{1}{2}\end{pmatrix}.$$



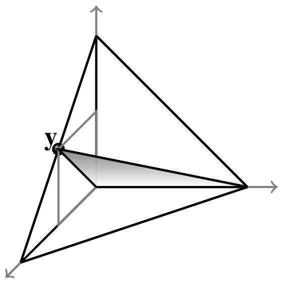
We have that

$$\mathbf{y} = \frac{1}{2}\begin{pmatrix}0\\1\\0\end{pmatrix} + \frac{1}{2}\begin{pmatrix}0\\0\\1\end{pmatrix},$$

i.e., $J = \{2,3\}$. The original simplex F is then subdivided into the 2 subsimplices

$$F_1 = chull\left\{\begin{pmatrix}1\\0\\0\end{pmatrix},\begin{pmatrix}0\\\frac{1}{2}\\\frac{1}{2}\end{pmatrix},\begin{pmatrix}0\\0\\1\end{pmatrix},\begin{pmatrix}0\\0\\0\end{pmatrix}\right\},$$

$$F_2 = chull\left\{\begin{pmatrix}1\\0\\0\end{pmatrix},\begin{pmatrix}0\\1\\0\end{pmatrix},\begin{pmatrix}0\\\frac{1}{2}\\\frac{1}{2}\end{pmatrix},\begin{pmatrix}0\\0\\0\end{pmatrix}\right\}.$$



∎

Notice that, if $\mathbf{y}$ is chosen too close to a facet (without belonging to it) of the simplex, (at least) one of the subsimplices is quite flat and working on it may cause numerical difficulties. Flatness of the simplex F_k can be detected through the computation of the determinant of the n-dimensional square matrix $\mathbf{V}_k$ whose columns are the vectors

$$\mathbf{v}_j^k - \mathbf{v}_{n+1}^k, \quad j = 1, \ldots, n$$

(note that the value of the determinant also allows us to compute the volume of the simplex). If the determinant is close to 0, then the simplex is considered flat. In some cases, a possible way to detect this situation is the following. Let $\mathbf{w}_k^T \mathbf{x} + w_{0k}$ be the affine function interpolating the concave function f at the vertices of the simplex. If we set

$$\mathbf{d}_k = (f(\mathbf{v}_1^k) - f(\mathbf{v}_{n+1}^k), \ldots, f(\mathbf{v}_n^k) - f(\mathbf{v}_{n+1}^k)),$$

then standard computations show that

$$\mathbf{w}_k = \mathbf{d}_k^T \mathbf{V}_k^{-1}, \quad w_{0k} = f(\mathbf{v}_{n+1}^k) - \mathbf{d}_k^T \mathbf{V}_k^{-1} \mathbf{v}_{n+1}^k.$$

Then, requiring that the determinant of $\mathbf{V}_k$ is not too close to 0 also implies that the norm $\|\mathbf{w}_k\|_2$ should not get too large.

Example 5.9. Let us consider the simplex

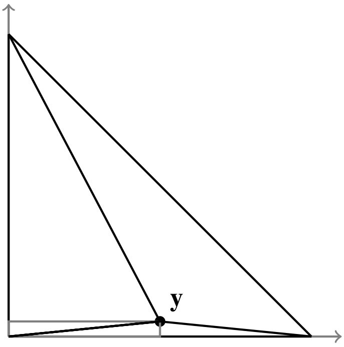
$$F = chull \left\{ \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right\},$$

and the point $\mathbf{y} = (\begin{smallmatrix} 1 \\ \varepsilon \end{smallmatrix})$. We have that

$$\mathbf{y} = \frac{1}{2} \begin{pmatrix} 2 \\ 0 \end{pmatrix} + \frac{\varepsilon}{2} \begin{pmatrix} 0 \\ 2 \end{pmatrix} + \frac{1}{2}(1-\varepsilon) \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

i.e., $J = \{1, 2, 3\}$. The original simplex F is then subdivided into the 3 subsimplices

$$\begin{aligned} F_1 &= chull \left\{ \begin{pmatrix} 1 \\ \varepsilon \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right\}, \\ F_2 &= chull \left\{ \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ \varepsilon \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right\}, \\ F_3 &= chull \left\{ \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 \\ \varepsilon \end{pmatrix} \right\}. \end{aligned}$$



Let us consider the simplex F_2 and the corresponding matrix

$$\mathbf{V}_2 = \begin{pmatrix} 2 & 1 \\ 0 & \varepsilon \end{pmatrix}.$$

The determinant of this matrix is equal to 2ε, which tends to 0 as ε tends to 0 or, equivalently, as the point $\mathbf{y}$ approaches the facet (edge) of F whose vertices are (2 0) and (0 0).

Now, assume that f is concave. The affine function interpolating f at the three vertices of F_2 is

$$\frac{1}{2}(f(2,0)-f(0,0))x_1+\frac{1}{2\varepsilon}(2f(1,\varepsilon)-(f(2,0)+f(0,0)))x_2+f(0,0).$$

If

$$2f(1,\varepsilon)-(f(2,0)+f(0,0)) \to \tau > 0 \quad \text{as } \varepsilon \to 0,$$

(which is certainly the case, e.g., if f is strictly concave), then the norm

$$\sqrt{\frac{1}{4}(f(2,0)-f(0,0))^2+\frac{1}{4\varepsilon^2}(2f(1,\varepsilon)-(f(2,0)+f(0,0)))^2}$$

of the vector of the coefficients for x_1 and x_2 in the affine interpolating function diverges to infinity as ε converges to 0. ■

Another quantity which has to be taken under control is the *eccentricity* $\sigma(F_k;\mathbf{y})$ of the simplex F_k with respect to $\mathbf{y}$. This is defined as follows:

$$\sigma(F_k;\mathbf{y}) = \frac{\max_{i=1,\ldots,n+1} \|\mathbf{y}-\mathbf{v}^{i,k}\|_2}{\max_{i,j=1,\ldots,n+1} \|\mathbf{v}^{i,k}-\mathbf{v}^{j,k}\|_2}. \tag{5.11}$$

In order to guarantee exhaustiveness of the branching process (see Definition 5.5), one should avoid situations where the eccentricity is too close to 1. Indeed, in the latter case the diameter of the simplices in an infinite nested sequence might not decrease to 0, which is a condition required by the exhaustiveness property.

It is easy to see that Remark 5.2 holds true also for simplices. For instance, in Example 5.8 the two subsimplices F_1 and F_2 share the common facet

$$chull\left\{\begin{pmatrix}1\\0\\0\end{pmatrix},\begin{pmatrix}0\\\frac{1}{2}\\\frac{1}{2}\end{pmatrix},\begin{pmatrix}0\\0\\0\end{pmatrix}\right\}.$$

The bisection and ω-subdivision rules for simplices are defined in a completely similar way with respect to the case of boxes. Before illustrating them through an example, we make the following remark.

Remark 5.3. *Given $F_k = chull\{\mathbf{v}^{1,k},\ldots,\mathbf{v}^{n+1,k}\}$, in order to define $S_k = F_k \cap S$ the $n+1$ linear inequalities defining F_k should be added to S. In practice, it is not necessary to compute these linear inequalities but only to perform the change of variables*

$$\mathbf{x} = \bar{V}_k\boldsymbol{\lambda}, \quad \boldsymbol{\lambda} \in \Delta_{n+1},$$

where $\bar{V}_k \in \mathbb{R}^{n\times(n+1)}$ is the matrix whose columns are the vertices of F_k. Note that this usually also simplifies the derivation of the convex underestimator. For instance, if f is a concave function, then after the change of variables its convex envelope over F_k is given by the simple formula

$$\sum_{i=1}^{n+1}\lambda_i f(\mathbf{v}_i^k).$$

Finally, note also that if the optimal solution of the relaxed problem $\boldsymbol{\lambda}^\star(F_k)$ after the change of variable is obtained (so that $\boldsymbol{\omega}(F_k) = \bar{V}_k\boldsymbol{\lambda}^\star(F_k)$), then the set J for the ω-subdivision is immediately available.

In what follows we illustrate the bisection and ω-subdivision rules through an example.

Example 5.10. Consider the problem

$$\begin{aligned}\min \quad & -x_1^2 - x_2^2\\ & 3x_1 - x_2 \le 3,\\ & 0 \le x_1 \le 2,\\ & 0 \le x_2 \le 3.\end{aligned}$$

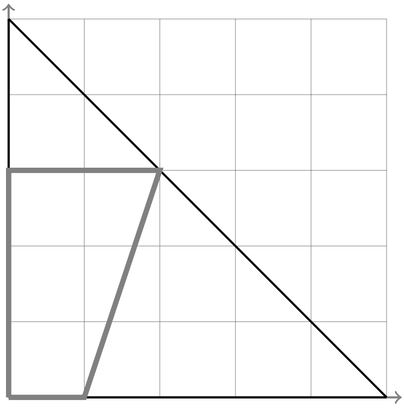
Solving

$$\begin{aligned}\min \quad & x_1 + x_2\\ & 3x_1 - x_2 \le 3,\\ & 0 \le x_1 \le 2,\\ & 0 \le x_2 \le 3,\end{aligned}$$

we get $\xi = 5$, so that the simplex

$$F = chull\{(5\ 0),\ (0\ 5),\ (0\ 0)\} = \{(x_1, x_2) \in \mathbb{R}^2\ :\ x_1 + x_2 \leq 5,\ \ x_1, x_2 \geq 0\} \tag{5.12}$$

contains the feasible region.



The convex envelope of the objective function over F is

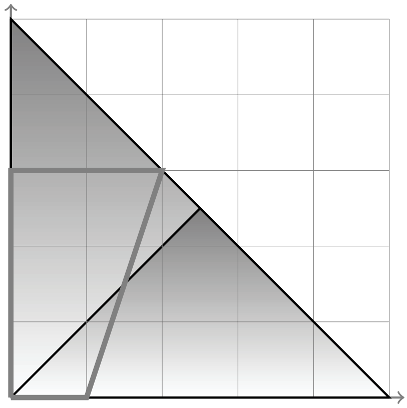
$$-5x_1 - 5x_2,$$

and the relaxed problem over the feasible region S is

$$\begin{aligned} \min \quad & -5x_1 - 5x_2 \\ & 3x_1 - x_2 \leq 3, \\ & 0 \leq x_1 \leq 2, \\ & 0 \leq x_2 \leq 3, \end{aligned}$$

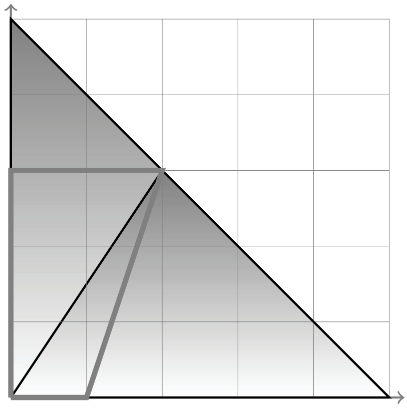
whose optimal solution is $\boldsymbol{\omega}(F) = (2\ 3)$. Since this is a feasible solution of the problem, we can evaluate the objective function at it (the value is -13) and, consequently, $ub_{\mathbf{0}} \leq -13$. Note that we set $\boldsymbol{\delta} = \mathbf{0}$ since the constraints are linear ones. Also note that it is easy to see that -13 is the optimal value of this problem. The longest edge in F is the edge joining the vertices $(5\ 0)$ and $(0\ 5)$. The bisection rule splits F with respect to the midpoint of the edge and results in the two subsimplices

$$F_1 = chull\left\{\begin{pmatrix} 5 \\ 0 \end{pmatrix}, \begin{pmatrix} \frac{5}{2} \\ \frac{5}{2} \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}\right\},$$
$$F_2 = chull\left\{\begin{pmatrix} \frac{5}{2} \\ \frac{5}{2} \end{pmatrix}, \begin{pmatrix} 0 \\ 5 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}\right\}.$$



Instead, the ω-subdivision rule splits F with respect to $\boldsymbol{\omega}(F)$ and results in the two subsimplices

$$F'_1 = chull\left\{\begin{pmatrix} 5 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}\right\},$$
$$F'_2 = chull\left\{\begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 0 \\ 5 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}\right\}.$$



In view of Remark 5.3 the problem over, e.g., F_2 can be written as

$$\begin{aligned}
\min \quad & -\tfrac{25}{2}\lambda_1 - 25\lambda_2 \\
& 5\lambda_1 - 5\lambda_2 \le 3, \\
& 0 \le \tfrac{5}{2}\lambda_1 \le 2, \\
& 0 \le \tfrac{5}{2}\lambda_1 + 5\lambda_2 \le 3, \\
& \lambda_1 + \lambda_2 + \lambda_3 = 1, \\
& \lambda_1, \lambda_2, \lambda_3 \ge 0,
\end{aligned}$$

with an optimal solution $\lambda_1^\star = \frac{4}{5}$, $\lambda_2^\star = \frac{1}{5}$, $\lambda_3^\star = 0$ corresponding to $\boldsymbol{\omega}(F_2) = (2\ 3) \equiv \boldsymbol{\omega}(F)$ (thus, the bisection rule, as expected, does not necessarily "cut away" the optimal solution of the father node). The lower bound over $S_2 = F_2 \cap S$ is equal to -15 and, since the optimal value of the problem is attained at $\boldsymbol{\omega}(F)$ and is equal to -13, the fathoming rule does not allow us to fathom S_2 (at least for $\varepsilon < 2$). It is easy to see that $\boldsymbol{\omega}(F_h') = \boldsymbol{\omega}(F)$ for $h = 1,2$, so that the lower bounds over F_1' and F_2' are both equal to -13, and $S_h' = F_h' \cap S$, $h = 1,2$, are both fathomed. Then, the BB algorithm stops for any $\varepsilon \geq 0$. In this case the ω-subdivision rule appears to be more efficient than the bisection rule, a fact, however, which can not be generalized. ∎

Note that for a given simplex F_k, it might happen that $\boldsymbol{\omega}(F_k)$ is a vertex of the simplex, and thus it cannot be employed for a branching operation. However, at the vertices of a simplex all convex underestimators are usually exact, so that if the optimal solution of the relaxed problem is attained at a vertex, then the lower bound of the node will be at least as large as the current upper bound, which means that the node will be fathomed and no branching is necessary (this is the case for F_1' and F_2' in the example above). Also note that the drawback of the ω-subdivision rule which might occur with boxes (possible exponential number of child nodes), is not true for simplices (the maximum number of child nodes is $n+1$). Instead, it is still true that the ω-subdivision rule alone makes it difficult to prove that the branching operation satisfies the exhaustiveness property (which, again, is trivially proven for the bisection rule).

5.3.3 Cones

Branching based on cones has a more limited applicability. It has been mostly employed for concave optimization problems, i.e., problems where a concave objective function has to be minimized over a polyhedral feasible region, although applications can also be found for other problems (e.g., in (Thoai, 2000a) it is used for the optimization over the efficient set of a multiobjective problem). In the field of concave optimization, the first reference on conical partitions is Tuy's work (Tuy, 1964), later developed in further contributions, such as (Bali, 1973; Hamami & Jacobsen, 1988; Jacobsen, 1981; Tuy, Khachaturov, & Utkin, 1987; Tuy, 1991b; Zwart, 1974). Throughout this section it will be first assumed that f is a concave function and that

$$S = \{\mathbf{x}\ :\ \mathbf{A}\mathbf{x} \leq \mathbf{b}\}$$

is a full-dimensional polytope; other cases will be briefly discussed at the end of the section. A n-dimensional polyhedral cone with n generating rays and with vertex $\mathbf{v}$ is defined as

$$C = cone\{\mathbf{v}\ ;\ \mathbf{r}_1,\ldots,\mathbf{r}_n\} = \left\{\mathbf{x} \in \mathbb{R}^n\ :\ \mathbf{x} = \mathbf{v} + \sum_{j=1}^{n} \mu_j \mathbf{r}_j,\ \ \mu_j \geq 0,\ j = 1,\ldots,n\right\},$$

where the linearly independent vectors $\mathbf{r}_1,\ldots,\mathbf{r_n}$ are the generating rays. For some vertex $\mathbf{v}$ of S it is easy to find a cone with vertex $\mathbf{v}$ and containing S, e.g., through the following procedure proposed in (Balas, 1971):

- identify the set of active constraints at $\mathbf{v}$;

- select n linearly independent active constraints and remove from the definition of S the remaining active constraints at $\mathbf{v}$, thus obtaining a new polyhedron $S' \supseteq S$ such that $\mathbf{v}$ is a nondegenerate vertex of S';
- let $\mathbf{r}_1, \ldots, \mathbf{r}_n$ be the direction of the n one-dimensional faces of S' with one vertex at $\mathbf{v}$.

Example 5.11. Let us consider the concave problem

$$\begin{aligned} \min \quad & -x_1^2 - x_2^2 - x_3^2 \\ & x_1 + x_2 - x_3 \geq 0, \\ & x_1 + x_2 + x_3 \leq 3, \\ & x_1, x_2, x_3 \geq 0. \end{aligned}$$

Let $\mathbf{v} = (0\ 0\ 0)$. The active constraints at $\mathbf{v}$ are $x_1 + x_2 - x_3 \geq 0$ and the three nonnegativity constraints $x_1, x_2, x_3 \geq 0$. Note that $\mathbf{v}$ is a degenerate vertex for the polytope S. If the constraint $x_1 + x_2 - x_3 \geq 0$ is removed, a new polyhedron $S' \supseteq S$ for which $\mathbf{v}$ is a nondegenerate vertex is obtained. It is immediately seen that the directions of the one-dimensional faces of S' with one vertex at the origin are the directions of the three axes, i.e., we can take $\mathbf{r}_i = \mathbf{e}^i$, $i = 1,2,3$. ■

Note that $S' = S$ if $\mathbf{v}$ is a nondegenerate vertex of S. Now, let $adj(\mathbf{v})$ be the set of vertices of S adjacent to $\mathbf{v}$, i.e., the vertices of S connected to $\mathbf{v}$ through an edge of S. We assume that

$$f(\mathbf{v}) \leq f(\mathbf{v}^{j,0}) \quad \forall\ \mathbf{v}^{j,0} \in adj(\mathbf{v}). \tag{5.13}$$

This can be accomplished by performing a local search in the space of the vertices, where the neighborhood of a vertex is made up by its adjacent vertices. By the procedure described above, a cone

$$C_0 = cone\{\mathbf{v}\ ;\ \mathbf{r}^{1,0}, \ldots, \mathbf{r}^{n,0}\} \tag{5.14}$$

can be derived that contains S. We assume that f is evaluated at $\mathbf{v}$, so that in the BB approach

$$f(\mathbf{v}) \geq ub_{\mathbf{0}} \tag{5.15}$$

(note that the linearity of the constraints allows us to set $\boldsymbol{\delta} = \mathbf{0}$). Denoting by $\mathbf{Q}_0$ the square matrix whose columns are the vectors $\mathbf{r}^{j,0}$, after the change of variables

$$\mathbf{x} = \mathbf{v} + \mathbf{Q}_0 \boldsymbol{\mu}, \quad \boldsymbol{\mu} \in \mathbb{R}^n_+,$$

the concave problem is transformed into

$$\begin{aligned} \min \quad & f_0'(\boldsymbol{\mu}) \\ & A\mathbf{Q}_0 \boldsymbol{\mu} \leq \mathbf{b}' \\ & \boldsymbol{\mu} \geq \mathbf{0}, \end{aligned} \tag{5.16}$$

where

$$f_0'(\boldsymbol{\mu}) = f(\mathbf{v} + \mathbf{Q}_0\boldsymbol{\mu}), \quad \mathbf{b}' = \mathbf{b} - \mathbf{A}\mathbf{v}.$$

Now, assume that a full-dimensional subcone with vertex $\mathbf{v}$

$$C_k = cone\{\mathbf{v} \,;\, \mathbf{r}^{1,k}, \dots, \mathbf{r}^{n,k}\} \subseteq C_0$$

is given. The subproblem over this cone is written as

$$\begin{aligned} \min \quad & f_k'(\boldsymbol{\mu}) \\ & A\mathbf{Q}_k\boldsymbol{\mu} \leq \mathbf{b}', \\ & \boldsymbol{\mu} \geq \mathbf{0}, \end{aligned}$$

where $\mathbf{Q}_k$ is the square matrix whose columns are the vectors $\mathbf{r}^{i,k}$, $i = 1, \dots, n$. The subdivision of the subset S_k related to the cone C_k (i.e., $S_k = S \cap C_k$) is quite similar to that previously seen with simplices.

- choose a point $\mathbf{y} \in C_k \setminus \{\mathbf{0}\}$ that does not lie along a generating ray of the cone;
- let
$$\mathbf{y} = \sum_{j=1}^{n} \mu_j(\mathbf{y})\mathbf{r}^{j,k}, \quad \boldsymbol{\mu}(\mathbf{y}) \geq \mathbf{0};$$
- let
$$J = \{j \in \{1, \dots, n\} \;:\; \mu_j(\mathbf{y}) > 0\}$$
(note that $|J| \geq 2$ since $\mathbf{y}$ does not belong to a generating ray of the cone);
- split C_k into the $|J|$ cones
$$C_{ik} = cone\{\mathbf{v} \,;\, \mathbf{r}^{1,k}, \dots, \mathbf{r}^{i-1,k}, \mathbf{y}, \mathbf{r}^{i+1,k}, \dots, \mathbf{r}^{n,k}\} \quad \forall\, i \in J.$$

Example 5.12. Let us consider the cone with vertex the origin

$$C = cone\left\{\begin{pmatrix}0\\0\\0\end{pmatrix} ; \begin{pmatrix}1\\0\\0\end{pmatrix}, \begin{pmatrix}0\\1\\0\end{pmatrix}, \begin{pmatrix}0\\0\\1\end{pmatrix}\right\}$$

and the point $\mathbf{y} = (0\ 1\ 1)$. We have that

$$\mathbf{y} = \begin{pmatrix}0\\1\\0\end{pmatrix} + \begin{pmatrix}0\\0\\1\end{pmatrix},$$

i.e., $J = \{2, 3\}$. The original cone C is then subdivided into the 2 subcones

$$C_1 = cone\left\{\begin{pmatrix}0\\0\\0\end{pmatrix} ; \begin{pmatrix}1\\0\\0\end{pmatrix}, \begin{pmatrix}0\\1\\1\end{pmatrix}, \begin{pmatrix}0\\0\\1\end{pmatrix}\right\},$$

$$C_2 = cone\left\{\begin{pmatrix}0\\0\\0\end{pmatrix} ; \begin{pmatrix}1\\0\\0\end{pmatrix}, \begin{pmatrix}0\\1\\0\end{pmatrix}, \begin{pmatrix}0\\1\\1\end{pmatrix}\right\}.$$

■

As for simplices, if $\mathbf{y}$ is chosen too close to a facet of the cone (without belonging to it), at least one of the subcones is flat. For a given cone C_k this can be established by computing the determinant of the n-dimensional square matrix $\mathbf{Q}_k$. If the determinant is close to 0, then the cone is a flat one. As a further indicator of this situation one might also employ the quantity

$$\eta(C_k) = \|\mathbf{e}\mathbf{Q}_k^{-1}\|_2, \tag{5.17}$$

which should not get too large.

Example 5.13. Let us consider the cone with vertex the origin

$$C = cone\left\{\begin{pmatrix} 0 \\ 0 \end{pmatrix} ; \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}\right\},$$

and the point $\mathbf{y} = (1\ \varepsilon)$. We have that

$$\mathbf{y} = \frac{1}{2}\begin{pmatrix} 2 \\ 0 \end{pmatrix} + \frac{\varepsilon}{2}\begin{pmatrix} 0 \\ 2 \end{pmatrix},$$

i.e., $J = \{1, 2\}$. The original cone C is then subdivided into the 2 subcones

$$C_1 = cone\left\{\begin{pmatrix} 0 \\ 0 \end{pmatrix} ; \begin{pmatrix} 1 \\ \varepsilon \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}\right\},$$

$$C_2 = cone\left\{\begin{pmatrix} 0 \\ 0 \end{pmatrix} ; \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ \varepsilon \end{pmatrix}\right\}.$$

The point $\mathbf{y}$ gets close to the facet corresponding to the generating ray $(2\ 0)$ as ε tends to 0, thus making the cone C_2 flat. Indeed, the corresponding matrix

$$\mathbf{Q}_2 = \begin{pmatrix} 2 & 1 \\ 0 & \varepsilon \end{pmatrix}$$

has a determinant converging to 0 as ε tends to 0. Note that

$$\eta(C_2) = \|\mathbf{e}\mathbf{Q}_2^{-1}\|_2 = \frac{1}{2\varepsilon}\sqrt{1+\varepsilon^2},$$

which diverges to infinity as ε converges to 0. ■

Later we will also introduce, as for simplices, an eccentricity value, which should be kept under control and should not get too large in order to guarantee exhaustiveness. It is easy to verify that Remark 5.2 is true also for cones. For what concerns the bisection and ω-subdivision rules, while defined for cones, they have in this case some peculiarities which we will discuss in what follows. Such peculiarities are strictly related to the lower bound computation in the conical algorithms. In fact, in these algorithms we do *not* compute an explicit lower bound for a subset S_k, but we just try to answer the question whether $lb(S_k) \geq ub_0 - \varepsilon$ or not, which is what we need in Step 7 of the BB approach for the fathoming rule. First, the following definition is needed.

Definition 5.14. *Let f be a continuous concave function and*

$$C_k = cone\{\mathbf{v}\ ;\ \mathbf{r}^{1k},\ldots,\mathbf{r}^{nk}\}$$

be a cone. Let $\gamma \le f(\mathbf{v})$. Then, the γ-extension *along the ray $\mathbf{r}^{jk}$, $j = 1,\ldots,n$, is the point $\bar{\mu}(\mathbf{r}^{jk})\mathbf{r}^{jk}$, where*

$$\bar{\mu}(\mathbf{r}^{jk}) = \sup\{\mu \ge 0\ :\ f(\mathbf{v}+\mu\mathbf{r}^{jk}) \ge \gamma\}.$$

Notice that $\bar{\mu}(\mathbf{r}^{jk}) = +\infty$ may occur for some j. A relevant result is the following.

Proposition 5.15. *If a set T is such that*

$$T \subseteq \Gamma_k = chull\{\mathbf{v},\ \mathbf{v}+\bar{\mu}(\mathbf{r}^{1k})\mathbf{r}^{1k},\ldots,\mathbf{v}+\bar{\mu}(\mathbf{r}^{nk})\mathbf{r}^{nk}\},$$

then

$$\min_{\mathbf{x}\in T} f(\mathbf{x}) \ge \gamma.$$

Proof. Obviously,

$$\min_{\mathbf{x}\in T} f(\mathbf{x}) \ge \min_{\mathbf{x}\in \Gamma_k} f(\mathbf{x}).$$

On the other hand, in view of Theorem 2.11,

$$\min_{\mathbf{x}\in \Gamma_k} f(\mathbf{x}) = \min\{f(\mathbf{v}),\ f(\mathbf{v}+\bar{\mu}(\mathbf{r}^{1k})\mathbf{r}^{1k}),\ldots,f(\mathbf{v}+\bar{\mu}(\mathbf{r}^{nk})\mathbf{r}^{nk})\} \ge \gamma,$$

where the last inequality follows from the assumption $f(\mathbf{v}) \ge \gamma$ and the Definition 5.14 of γ-extension. □

Now, assume that $\gamma = ub_0 - \varepsilon$. If $\varepsilon > 0$, then

$$\bar{\mu}(\mathbf{r}^{jk}) > 0 \quad \forall\, j = 1,\ldots,n.$$

This is also true for $\varepsilon = 0$ if the assumption (5.13) is satisfied and $\mathbf{v}$ is a nondegenerate vertex of S. Then, the representation of Γ_k in the space of the $\boldsymbol{\mu}$ variables is

$$\left\{\boldsymbol{\mu}\ \ge \mathbf{0}\ :\ \sum_{j=1}^{n} \frac{\mu_j}{\bar{\mu}(\mathbf{r}^{jk})} \le 1\right\}$$

(note that $\frac{\mu_j}{\bar{\mu}(\mathbf{r}^{jk})} = 0$ if $\bar{\mu}(\mathbf{r}^{jk}) = \infty$). Therefore, in view of Proposition 5.15, $lb(S_k) \ge \gamma = ub_0 - \varepsilon$ is true if

$$\begin{aligned} \max\quad & \textstyle\sum_{j=1}^{n} \frac{\mu_j}{\bar{\mu}(\mathbf{r}^{jk})} \\ & A\mathbf{Q}_k\boldsymbol{\mu} \le \mathbf{b}', \\ & \boldsymbol{\mu} \ge \mathbf{0}, \end{aligned} \tag{5.18}$$

has an optimal value not larger than 1.

Definition 5.16. *The γ-*concavity cut *is defined as the following constraint:*

$$\sum_{j=1}^{n} \frac{\mu_j}{\bar{\mu}(\mathbf{r}^{jk})} \leq 1. \tag{5.19}$$

γ-concavity cuts (see also Section 5.5.2) were first introduced in (Tuy, 1964) for concave minimization problems (and later introduced also in the context of integer programming by Balas (1971) with the name of *intersection cuts*).

Thanks to the proposition above, a γ-concavity cut does not eliminate any feasible solution in S_k with function value lower than γ.

Remark 5.4. *In Section* 5.2.2 *we commented the node selection step of the BB approach and said that one possible selection rule is to select the node with the lowest lower bound. If at each node we compute the optimal value of problem* (5.18)*, such rule is changed into the selection of the node with the largest optimal value for problem* (5.18)*. Alternatively, one could compute a proper lower bound for the subset S_k as suggested in (Thoai & Tuy,* 1980*). Given the optimal value $\theta(C_k)$ for problem* (5.18)*, let $\mathbf{z}^{jk} = \mathbf{v} + \theta(C_k)\bar{\mu}(\mathbf{r}^{jk})\mathbf{r}^{jk}$, $j = 1,\dots,n$. Then*

$$S_k \subseteq chull\{\mathbf{v}, \mathbf{z}^{1k}, \dots, \mathbf{z}^{nk}\},$$

so that a valid lower bound over S_k is

$$\min\{f(\mathbf{v}), f(\mathbf{z}^{1k}), \dots, f(\mathbf{z}^{nk})\}.$$

Remark 5.5. *It is worthwhile to underline at this point that an alternative way to derive a γ-concavity cut has been suggested in (Carvajal-Moreno,* 1972*) (see also (Horst & Tuy,* 1993*)). Let*

$$\mathbf{r}^j = \mathbf{v}^j - \mathbf{v} \quad \forall\, \mathbf{v}^j \in adj(\mathbf{v}).$$

Let $\bar{\mu}(\mathbf{r}^j)$ be the corresponding γ-extensions. Then, a γ-concavity cut is

$$\mathbf{w}^T(\mathbf{x} - \mathbf{v}) \geq 1,$$

where $\mathbf{w}$ is a basic feasible solution of the system of inequalities

$$\mathbf{w}^T\mathbf{r}^j \geq \frac{1}{\bar{\mu}(\mathbf{r}^j)} \quad \forall\, j \;:\; \mathbf{v}^j \in adj(\mathbf{v}).$$

Indeed, in view of the definition of γ-extension and of the above set of inequalities, for each j we can guarantee that the hyperplane $\mathbf{w}^T(\mathbf{x} - \mathbf{v}) = 1$ crosses the half-line starting at $\mathbf{v}$ and with direction $\mathbf{r}_j$ at some point $\mathbf{z}^j$ such that $f(\mathbf{z}^j) \geq \gamma$, so that

$$f(\mathbf{y}) \geq \gamma \quad \forall\, \mathbf{y} \in S \cap \{\mathbf{x} \;:\; \mathbf{w}^T(\mathbf{x} - \mathbf{v}) \leq 1\}.$$

Remark 5.6. *γ-concavity cuts can also be employed, e.g., to linearize a reverse convex constraint, i.e., a constraint $g(\mathbf{x}) \leq 0$ with g concave. In this case, if a polyhedral cone C_0 with vertex some point $\mathbf{x}_0$ such that $g(\mathbf{x}_0) > 0$ is available, then* 0*-extensions of the function g can be computed and the corresponding γ-concavity cut is such that all the points in C_0*

satisfying the reverse convex constraint also satisfy the cut. If a reverse convex constraint is included in the definition of the feasible region S, and more cones $C_1, \dots, C_t \supset S$ with vertices $\mathbf{x}_i$, $i = 1, \dots, t$, such that $g(\mathbf{x}_i) > 0$ are available, then t valid cuts can be derived which allow us to define an outer approximation of S with the reverse convex constraint replaced by the t linear inequalities.

We also recall here that reverse convex constraints occur, e.g., in DC optimization problems (see Section 4.7). In particular, all canonical DC problems have a single reverse convex constraint besides a convex constraint and a linear objective function.

Now we are ready to describe the bisection and ω-subdivision rules for cones. The ω-subdivision rule uses the optimal solution $\boldsymbol{\mu}^{k^\star}$ of (5.18), i.e., the subdivision takes place with respect to the point

$$\mathbf{y} = \mathbf{v} + \sum_{j=1}^{n} \mu_j^{k^\star} \mathbf{r}^{j,k}.$$

Note that the maximum number of child nodes for this rule is n but, again, it may be difficult to prove that the branching operation has the exhaustiveness property. Similar to simplices, if the optimal solution lies along a generating ray of the cone (so that it could not be employed for the branching operation), the corresponding node is fathomed and no branching operation is required. For the bisection rule the new ray $\mathbf{y}$ is derived from the midpoint of a longest edge of the simplex

$$C_k \cap \left\{ \boldsymbol{\mu} \, : \, \sum_{j=1}^{n} \frac{\mu_j}{\bar{\mu}(\mathbf{r}^{j0})} = 1 \right\},$$

i.e., the intersection of the current cone with the hyperplane defined by the γ-extensions computed for the initial cone C_0.

In all the above development some care is still needed. Indeed, since we have defined $\gamma = ub_{\mathbf{0}} - \varepsilon$, each time $ub_{\mathbf{0}}$ is updated, all the γ-extensions should be recomputed for all the subcones not yet fathomed. In fact, a different approach is followed. If we detect a point $\bar{\mathbf{x}}$ such that $f(\bar{\mathbf{x}}) < ub_{\mathbf{0}}$, then we can compute a vertex $\mathbf{v}'$ such that $f(\mathbf{v}') \leq f(\bar{\mathbf{x}})$, e.g., in the case of a differentiable function f by solving the linear program

$$\min_{\mathbf{x} \in S} \ \nabla f(\bar{\mathbf{x}})^T (\mathbf{x} - \bar{\mathbf{x}}),$$

and, by a local search based on the exploration of the adjacent vertices, we are able to detect a vertex $\mathbf{v}''$ such that $f(\mathbf{v}'') \leq f(\mathbf{v}')$ and such that (5.13) is true with $\mathbf{v}''$ replacing $\mathbf{v}$. Then, in a way completely similar to that previously seen for $\mathbf{v}$, a new initial cone $C_0' \supset S$ is computed and we end up with a reformulation similar to (5.16). The BB approach is restarted with the new cone C_0' while all the previously generated cones are thrown away. Notice that, since we have a restart each time a new improved vertex is detected, and since the number of vertices is finite, then the number of restarts is also finite. A possible inefficiency of this approach is that, after the recomputation of a new initial cone, regions which were previously discarded come into play again. However, this inefficiency stops as soon as a global optimum vertex is detected (obviously, at that point no further restart will take place). Since in BB approaches the time to detect a globally optimal solution is often much lower than the time to certify its optimality (by fathoming of all the nodes), this inefficiency has a relatively mild impact. We illustrate the two subdivision rules and this restarting procedure through an example.

Example 5.17. Let us consider the concave problem

$$\begin{aligned} \min \quad & -x_1^2 - x_2^2 + x_1 + 3x_2 \\ & 3x_1 - x_2 \leq 3, \\ & 0 \leq x_1 \leq 2, \\ & 0 \leq x_2 \leq 3, \end{aligned}$$

reported in Figure 5.3, from which it is easily seen that the global optimum is located at $(2,3)$. The origin is a nondegenerate vertex of the feasible region and satisfies condition (5.13). Considering the vertices adjacent to the origin, we have the following generating rays:

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} 0 \\ 3 \end{pmatrix}.$$

Then, we can take

$$C_0 = cone\left\{\begin{pmatrix} 0 \\ 0 \end{pmatrix}; \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 3 \end{pmatrix}\right\}.$$

We can set $ub_\mathbf{0} = 0$ (the value of the objective function at the origin and, actually, also at its adjacent vertices). By taking $\gamma = ub_\mathbf{0} = 0$ ($\varepsilon = 0$), we can compute the $\bar{\mu}$ values for the

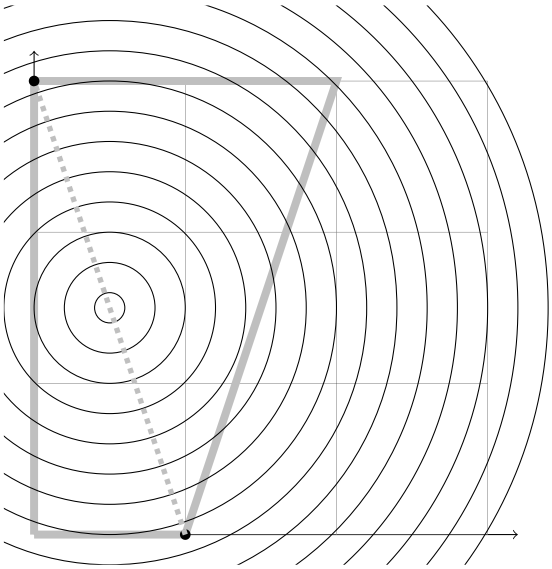


Figure 5.3. *Concave optimization on a polytope; black circles represent the γ-extensions, while the gray dotted line represents the concavity cut*

γ-extensions over the two generating rays (both equal to 1), and then solve the problem

$$\begin{aligned} \max \quad & \mu_1+\mu_2 \\ & 3\mu_1-3\mu_2 \le 3, \\ & 0 \le \mu_1 \le 2, \\ & 0 \le \mu_2 \le 1, \end{aligned}$$

to establish whether we can conclude that $lb(S) \geq ub_{\mathbf{0}}$. Since the optimal value of this problem is attained at $\mu_1^\star = 2$, $\mu_2^\star = 1$ and is equal to $3 > 1$, then we can not fathom S. However, notice that the γ-concavity cut,

$$\mu_1+\mu_2 \geq 1, \tag{5.20}$$

does not exclude any feasible point with function value lower than 0. With respect to the original variables, the optimal solution of the problem above is the nondegenerate vertex of the feasible region S:

$$\mathbf{x}^\star = 2\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 3 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}.$$

Since at this vertex the objective function value is equal to $-2 < ub_{\mathbf{0}} = 0$ and the vertex satisfies (5.13), then $ub_{\mathbf{0}}$ is updated to -2 and the cone C_0 is not split into subcones. Rather, a new initial cone,

$$C_0' = cone\left\{ \begin{pmatrix} 2 \\ 3 \end{pmatrix} ; \begin{pmatrix} -2 \\ 0 \end{pmatrix}, \begin{pmatrix} -1 \\ -3 \end{pmatrix} \right\},$$

is computed (see Figure 5.4).

By taking $\gamma = ub_{\mathbf{0}} = -2$, we can compute the $\bar{\mu}$ values for the γ-extensions over the two generating rays of this cone (equal to $\frac{3}{2}$ and $\frac{6}{5}$, respectively) and solve the problem

$$\begin{aligned} \max \quad & \tfrac{2}{3}\mu_1+\tfrac{5}{6}\mu_2 \\ & \mu_1 \geq 0, \\ & 0 \le 2-2\mu_1-\mu_2 \le 2, \\ & 0 \le 3-3\mu_2 \le 3, \\ & \mu_1,\mu_2 \geq 0, \end{aligned}$$

to establish whether $lb(S) \geq ub_{\mathbf{0}}$. Since the optimal value is $\frac{7}{6} > 1$ attained at $\mu_1^\star = \frac{1}{2}$, $\mu_2^\star = 1$, we can not fathom S. The γ-concavity cut

$$\frac{2}{3}\mu_1+\frac{5}{6}\mu_2 \geq 1$$

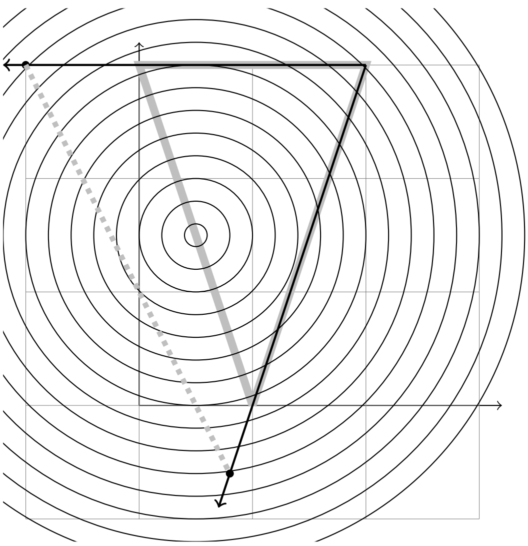


Figure 5.4. *Concave optmization on a polytope; thick arrows represent the cone with vertex* $(2,3)$*, black dots on the rays of the cone are the two* γ*-extensions, and the gray dotted line indicates the* γ*-concavity cut*

does not exclude any feasible solution with function value lower than $ub_\mathbf{0} = -2$. Moreover, since in the original space the optimal solution is

$$\mathbf{x}^\star = \begin{pmatrix} 2 \\ 3 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} -2 \\ 0 \end{pmatrix} + \begin{pmatrix} -1 \\ -3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

and the objective function value at $\mathbf{x}^\star$ is not lower than $ub_\mathbf{0}$, then $ub_\mathbf{0}$ is not updated and now a subdivision of C_0' into subcones is required.[7] Using the bisection rule, we take the midpoint $\bar{\mu}_1 = \frac{3}{4}$, $\bar{\mu}_2 = \frac{3}{5}$ of the segment

$$C_0' \cap \left\{ (\mu_1, \mu_2) \, : \, \frac{2}{3}\mu_1 + \frac{5}{6}\mu_2 = 1 \right\},$$

i.e.,

$$\mathbf{y} = \begin{pmatrix} 2 \\ 3 \end{pmatrix} + \frac{3}{4}\begin{pmatrix} -2 \\ 0 \end{pmatrix} + \frac{3}{5}\begin{pmatrix} -1 \\ -3 \end{pmatrix} = \begin{pmatrix} -\frac{1}{10} \\ \frac{6}{5} \end{pmatrix}.$$

[7] In fact, in a variant of the approach presented here, (at least some of) the previously computed γ-concavity cuts, like (5.20) in our example, can be added to the original problem.

We end up with the two subcones of C'_0

$$C'_1 = cone\left\{\begin{pmatrix} 2 \\ 3 \end{pmatrix};\begin{pmatrix} -2 \\ 0 \end{pmatrix},\begin{pmatrix} -\frac{21}{10} \\ -\frac{9}{5} \end{pmatrix}\right\},$$

$$C'_2 = cone\left\{\begin{pmatrix} 2 \\ 3 \end{pmatrix};\begin{pmatrix} -\frac{21}{10} \\ -\frac{9}{5} \end{pmatrix},\begin{pmatrix} -1 \\ -3 \end{pmatrix}\right\}.$$

If instead an ω-subdivision is chosen, then $\mathbf{y} = \mathbf{x}^\star = (0\ 0)$ and we end up with the two subcones

$$\bar{C}'_1 = cone\left\{\begin{pmatrix} 2 \\ 3 \end{pmatrix};\begin{pmatrix} -2 \\ 0 \end{pmatrix},\begin{pmatrix} -2 \\ -3 \end{pmatrix}\right\},$$

$$\bar{C}'_2 = cone\left\{\begin{pmatrix} 2 \\ 3 \end{pmatrix};\begin{pmatrix} -2 \\ -3 \end{pmatrix},\begin{pmatrix} -1 \\ -3 \end{pmatrix}\right\}.$$ ■

For conical algorithms mixed strategies have also been proposed (in fact, these can also be extended to simplicial algorithms). In particular, in (Tuy, 1991b) *normal conical algorithms* have been proposed, based on a normal subdivision rule, in which ω-subdivisions are employed most of the time but occasionally bisections are also performed in order to guarantee the exhaustiveness property. In particular, bisection of a cone C_k is performed if the value $\eta(C_k)$ defined in (5.17) is too large, or if the eccentricity σ (see (5.11)) of the simplex $F_k = C_k \cap \{\mathbf{x} \in \mathbb{R}^n \ :\ \mathbf{e}\mathbf{Q}_0^{-1}\mathbf{x} = 1\}$ with respect to the subdivision point $\mathbf{y}$ (defined as the intersection of F_k with the ray in C_k with respect to which the subdivision is performed) is too close to 1.

While we have described conical algorithms for concave problems over polyhedral regions, they can also be applied to problems over more general feasible regions S, such as the nonpolyhedral convex ones. In this case, instead of starting with a single cone C_0 containing S, $n+1$ cones, whose interiors are disjoint, and covering S are initially chosen. We briefly sketch how to deal with such cases. Let

$$F_0 = chull\{\mathbf{v}^{1,0},\ldots,\mathbf{v}^{n+1,0}\}$$

be an n-simplex containing S (it can be derived as explained in Section 5.3.2) and $\mathbf{x}^0$ be a point lying in the interior of F_0. Consider the $n+1$ cones with vertex $\mathbf{x}^0$,

$$C_i = cone\{\mathbf{x}^0\ ;\ \mathbf{v}^{1,0}-\mathbf{x}^0,\ldots,\mathbf{v}^{i-1,0}-\mathbf{x}^0,\ \mathbf{v}^{i+1,0}-\mathbf{x}^0,\ldots,\mathbf{v}^{n+1,0}-\mathbf{x}^0\},$$

for $i = 1,\ldots,n+1$. Then, $int(C_i)\cap int(C_j) = \emptyset$, for $i \neq j$, and $\bigcup_{i=1}^{n+1} C_i \supset S$. The algorithm can thus be initialized starting from these cones.

Example 5.18. Consider the problem

$$\begin{aligned} \min\quad & -x_1^2 - x_2^2 \\ & (x_1-1)^2+(x_2-1)^2 \leq 1, \\ & x_1,x_2 \geq 0, \end{aligned}$$

and the point $\mathbf{x}_0 = (1\ 1)$, which lies in the interior of the feasible region S. The feasible region is contained, e.g., in the simplex

$$F_0 = chull\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 4 \end{pmatrix} \right\}.$$

Therefore, the following three cones can be initially chosen, as represented in Figure 5.5.

$$C_1 = cone\left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}; \begin{pmatrix} 3 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 3 \end{pmatrix} \right\},$$
$$C_2 = cone\left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}; \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 3 \end{pmatrix} \right\}, \qquad \blacksquare$$
$$C_3 = cone\left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}; \begin{pmatrix} 3 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix} \right\}.$$

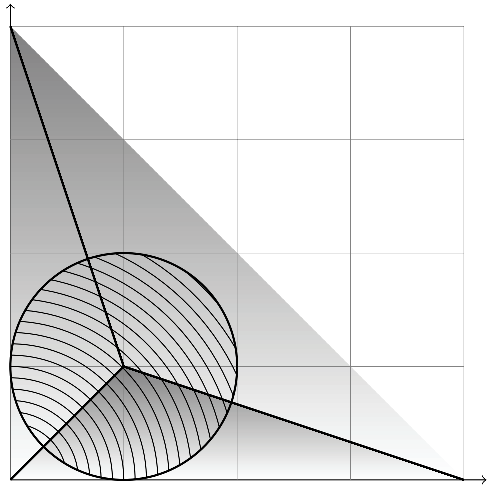


Figure 5.5. *Initial conic subdivision for a convex nonpolyhedral region*

5.3.4 Ellipsoids

Branching based on ellipsoids has been introduced in (Le Thi, 2000) for problems with quadratic objective and strictly convex quadratic constraints (see also (de Angelis, Bomze, & Toraldo, 2004)), and later extended in (Hager & Phan, 2009) to the case of weakly convex functions.

Definition 5.19. *A function f is called* weakly convex *if there exists $\rho > 0$ such that*

$$f(\mathbf{x}) + \rho\|\mathbf{x}\|_2^2$$

is a convex function.

Notice the relation of this definition with the DC decomposition discussed in Theorem 4.81. According to the above definition, given some region X, if a convex underestimator c for the concave function $-\rho\|\mathbf{x}\|_2^2$ is available over X, then

$$f(\mathbf{x})+\rho\|\mathbf{x}\|_2^2+c(\mathbf{x})$$

is a convex underestimator for f over X. Now, consider the case

$$X=E_{\mathbf{B},\mathbf{c}}=\{\mathbf{x}\in\mathbb{R}^n\ :\ (\mathbf{x}-\mathbf{c})^T\mathbf{B}^{-1}(\mathbf{x}-\mathbf{c})\leq 1\},$$

where $\mathbf{B}$ is a positive definite matrix, i.e., $E_{\mathbf{B},\mathbf{c}}$ is an ellipsoid with center $\mathbf{c}$. In (Hager & Phan, 2009) the best affine underestimator of $-\|\mathbf{x}\|_2^2$ over $E_{\mathbf{B},\mathbf{c}}$ is defined as

$$\begin{array}{lll}\min_{\ell \text{ affine}} & \max_{\mathbf{x}\in E_{\mathbf{B},\mathbf{c}}}\ -(\|\mathbf{x}\|_2^2+\ell(\mathbf{x})) & \\ & \ell(\mathbf{x})\leq-\|\mathbf{x}\|_2^2 & \forall\,\mathbf{x}\in E_{\mathbf{B},\mathbf{c}}.\end{array}\tag{5.21}$$

Such affine underestimator is derived in the following theorem.

Theorem 5.20. *The solution of* (5.21) *is the affine function*

$$\ell^\star_{\mathbf{B},\mathbf{c}}(\mathbf{x})=-2\mathbf{c}^T\mathbf{x}+\gamma,\tag{5.22}$$

where

$$\gamma=2\mathbf{c}^T\boldsymbol{\mu}-\|\boldsymbol{\mu}\|_2^2,\quad \boldsymbol{\mu}\in\arg\max_{\mathbf{x}\in E_{\mathbf{B},\mathbf{c}}}\ \|\mathbf{x}-\mathbf{c}\|_2^2.$$

Moreover, the optimal value of (5.21) *is equal to* $\frac{1}{4}diam(E_{\mathbf{B},\mathbf{c}})$.

Notice that in the definition of $\boldsymbol{\mu}$ the solution of a so-called trust region problem is needed; this is a “simple” GO problem (see Section 2.3). Denoting by $\mathbf{u}$ the unit eigenvector associated to the smallest (by definition, positive) eigenvalue of $\mathbf{B}^{-1}$,

$$\boldsymbol{\mu}=\mathbf{c}+r\mathbf{u},$$

where r is the positive scalar such that $\boldsymbol{\mu}$ belongs to the boundary of $E_{\mathbf{B},\mathbf{c}}$. The branching operation for the ellipsoid $E_{\mathbf{B},\mathbf{c}}$ is defined as follows:

- subdivide $E_{\mathbf{B},\mathbf{c}}$ into two parts of equal volume, through a hyperplane $\mathbf{v}^T\mathbf{x}=\mathbf{v}^T\mathbf{c}$ containing the center $\mathbf{c}$, i.e.,

$$H_1=\{\mathbf{x}\in E_{\mathbf{B},\mathbf{c}}\ :\ \mathbf{v}^T\mathbf{x}\leq\mathbf{v}^T\mathbf{c}\},$$
$$H_2=\{\mathbf{x}\in E_{\mathbf{B},\mathbf{c}}\ :\ \mathbf{v}^T\mathbf{x}\geq\mathbf{v}^T\mathbf{c}\};$$

- compute the ellipsoids $E_{\mathbf{B}_1,\mathbf{c}_1}$ and $E_{\mathbf{B}_2,\mathbf{c}_2}$ of minimum volume containing H_1 and H_2, respectively, where

$$\mathbf{c}_1=\mathbf{c}-\tfrac{\mathbf{d}}{n+1},\quad \mathbf{c}_2=\mathbf{c}+\tfrac{\mathbf{d}}{n+1},$$
$$\mathbf{d}=\tfrac{\mathbf{B}\mathbf{v}}{\sqrt{\mathbf{v}^T\mathbf{B}\mathbf{v}}},\qquad \mathbf{B}_1=\mathbf{B}_2=\tfrac{n^2}{n^2-1}\left(\mathbf{B}-\tfrac{2\mathbf{d}\mathbf{d}^T}{n+1}\right).$$

Notice that in this case even the interiors of the two newly generated ellipsoids may overlap. It turns out (see, e.g., (Grotschel, Lovasz, & Schrijver, 1993; N. Z. Shor, 1977)) that

$$\frac{\text{volume}(E_{\mathbf{B}_i,\mathbf{c}_i})}{\text{volume}(E_{\mathbf{B},\mathbf{c}})} = \frac{n}{n+1}\left[\frac{n^2}{n^2-1}\right]^{\frac{n-1}{2}},$$

which is independent from $E_{\mathbf{B},\mathbf{c}}$ and from $\mathbf{v}$. In (Le Thi, 2000) it is suggested that we choose $\mathbf{v}$ in the direction along the major axis of $E_{\mathbf{B},\mathbf{c}}$ (something similar to the selection of a longest edge in the bisection rule for the previously considered geometric branching operations), which guarantees that nested sequences of ellipsoids shrink to a point (i.e., the exhaustiveness property is satisfied). Note that in order to apply the BB algorithm based on ellipsoidal subdivisions we need to compute an initial ellipsoid containing the compact feasible region S. Moreover, we need to be able to efficiently compute a lower bound over $S \cap E_{\mathbf{B},\mathbf{c}}$. In (Hager & Phan, 2009), when S is a convex set an algorithm is proposed to solve the convex problem

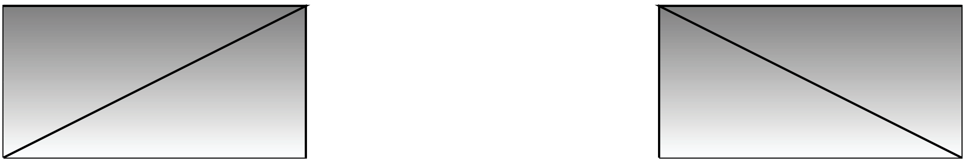
$$\begin{array}{ll} \min & f(\mathbf{x}) + \rho\|\mathbf{x}\|_2^2 + \ell^{\star}_{\mathbf{B},\mathbf{c}}(\mathbf{x}) \\ & \mathbf{x} \in S \cap E_{\mathbf{B},\mathbf{c}}, \end{array}$$

which returns a valid lower bound over the subset $S \cap E_{\mathbf{B},\mathbf{c}}$ of the feasible region. In (Le Thi, 2000) for the case of quadratic objective and strictly convex quadratic constraints, a lower bound is proposed based on Lagrangian duality: the convex quadratic constraints are moved into the objective and the resulting problem is the minimization of a quadratic function over an ellipsoid. The subgradient projection method by Polyak (Polyak, 1987) is then used to solve the dual Lagrangian problem.

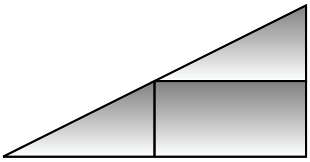
5.3.5 Further geometrical objects

For some problems, the structure of the problem itself suggests possible alternative branching rules.

A variant of the branching operation based on boxes has been proposed in (Linderoth, 2005) for quadratic problems with quadratic constraints. For such problems it is possible to compute lower bounds by using branching based on boxes and deriving convex relaxations obtained through convex lower and concave upper estimating functions for bilinear terms in the quadratic expressions over two-dimensional rectangles. These functions are, in fact, convex and concave envelopes, and can be found from the standard formulae discussed in Section 4.2.2. However, as already mentioned in Section 4.2.7 (see also Section 4.2.3), Linderoth in (Linderoth, 2005) derived the formulae of the convex and concave envelopes of bilinear terms over certain triangles. As a consequence, he also proposes an alternative to branching based on boxes. If only boxes are employed, at each node of the BB tree each pair of variables is associated to a two-dimensional rectangle, which can be subdivided into two or four subrectangles by a branching operation. Linderoth proposes to associate to a pair of variables either a rectangle (as in the standard scheme) or a right-angled triangle. If the current region associated to a pair of variables is a rectangle, then a branching operation might subdivide it into two right-angled triangles (determined by one of the two diagonals).



If the current region associated to a pair of variables is a right-angled triangle, then a branching operation subdivides it in two right-angled triangles and one rectangle, with one vertex corresponding to that forming the right angle of the triangle, while the opposite vertex is the midpoint of the hypotenuse of the triangle.



Another nice example is represented by the sum-of-ratio problems, where the feasible region is a polytope and the objective function is

$$\sum_{i=1}^{p} \frac{\mathbf{c}_i^T \mathbf{x} + c_{0i}}{\mathbf{d}_i^T \mathbf{x} + d_{0i}}$$

(it is assumed that numerator and denominator functions are positive over the feasible region). By introducing the additional variables t_i, s_i, $i = 1, \ldots, p$, and imposing

$$t_i = \mathbf{c}_i^T \mathbf{x} + c_{0i}, \quad s_i = \mathbf{d}_i^T \mathbf{x} + d_{0i},$$

the objective function can be rewritten as

$$\sum_{i=1}^{p} \frac{t_i}{s_i}.$$

To each ratio $\frac{t_i}{s_i}$ we associate a trapezoid

$$T_i = \{(t_i \; s_i) \in \mathbb{R}_+^2 \; : \; \ell_i \le t_i + s_i \le u_i, \, \alpha_i s_i \le t_i \le \gamma_i s_i\},$$

where $0 \le \ell_i \le u_i$ and $0 \le \alpha_i \le \gamma_i$. The convex envelope of the ratio function over T_i is available (see (Kuno, 2002; Benson, 2004)). Therefore, Kuno (Kuno, 2002, 2005) proposed to enclose the feasible region within the Cartesian product

$$T_1^0 \times \cdots \times T_p^0,$$

where the sets T_i^0 are trapezoids which can be efficiently computed by solving suitable linear problems and problems with a single ratio objective function (which can be efficiently solved, as already commented on in Section 2.4.1). Next, he proposed a BB approach, where a Cartesian product of p trapezoids,

$$T_1^{\mathcal{N}} \times \cdots \times T_p^{\mathcal{N}},$$

is associated to each node $\mathcal{N}$ of the BB tree, and the branching operation for the node selects one trapezoid, say

$$T_i^{\mathcal{N}} = \{(t_i\ s_i) \in \mathbb{R}_+^2\ :\ \ell_i \le t_i + s_i \le u_i,\ \alpha_i^{\mathcal{N}} s_i \le t_i \le \gamma_i^{\mathcal{N}} s_i\},$$

and splits it into two trapezoids

$$T_i^1 = \{(t_i\ s_i) \in \mathbb{R}_+^2\ :\ \ell_i \le t_i + s_i \le u_i,\ \alpha_i^{\mathcal{N}} s_i \le t_i \le w_i s_i\}$$

and

$$T_i^2 = \{(t_i\ s_i) \in \mathbb{R}_+^2\ :\ \ell_i \le t_i + s_i \le u_i,\ w_i s_i \le t_i \le \gamma_i^{\mathcal{N}} s_i\},$$

where $w_i \in (\alpha_i^{\mathcal{N}}, \gamma_i^{\mathcal{N}})$. Thus, the two child nodes for the node $\mathcal{N}$ are the Cartesian products of p trapezoids

$$T_1^{\mathcal{N}} \times \cdots \times T_i^1 \times \cdots \times T_p^{\mathcal{N}}$$

and

$$T_1^{\mathcal{N}} \times \cdots \times T_i^2 \times \cdots \times T_p^{\mathcal{N}}.$$

Special branching rules are proposed for the DM problem (4.140) discussed in Section 4.8 and reported here:

$$\min\{f(\mathbf{y})\ :\ \mathbf{y} \in G \cap H\} \tag{5.23}$$

(see also (Tuy, 2000; Tuy & Luc, 2000)). Assume that a reverse polyblock $P(T)$ is given (see Definition 4.95), where $T \subset [\mathbf{a},\mathbf{b}]$ is a finite set and $[\mathbf{a},\mathbf{b}] \supseteq G \cap H$, $\mathbf{a},\mathbf{b} \in \mathbb{R}_+^n$. Then, as has been shown in Section 4.8, a lower bound can be computed solving problem (4.141). The optimal value of such a relaxed problem is attained at some point $\bar{\mathbf{z}} \in T$. Obviously, if $\bar{\mathbf{z}} \in G \cap H$, i.e., if $\bar{\mathbf{z}}$ belongs to the feasible region of the DM problem (5.23), then it is also an optimal solution of (5.23). We can always assume that $\mathbf{z} \in G\ \forall\ \mathbf{z} \in T$. Indeed, $\mathbf{z} \notin G$ implies that no feasible point belongs to $[\mathbf{z},\mathbf{b}]$ and this box can be removed (equivalently, we can remove $\mathbf{z}$ from T). We can also assume that $\mathbf{b} \in H$ (otherwise, $[\mathbf{a},\mathbf{b}] \cap H = \emptyset$ and the problem has an empty feasible region). Obviously

$$\bar{\mathbf{z}} \in G,\ \ \bar{\mathbf{z}} \notin G \cap H \quad \Rightarrow \quad \bar{\mathbf{z}} \notin H.$$

Then, let the point $\tilde{\mathbf{z}}$ be defined as

$$\tilde{\mathbf{z}} = \mathbf{b} + \bar{\alpha}(\bar{\mathbf{z}} - \mathbf{b}), \quad \bar{\alpha} = \max\{\alpha \ge 0\ :\ \mathbf{b} + \alpha(\bar{\mathbf{z}} - \mathbf{b}) \in H\},$$

and, after having defined

$$\mathbf{z}_i = \bar{\mathbf{z}} + (\tilde{z}_i - \bar{z}_i)\mathbf{e}_i, \quad i = 1,\dots,n,$$

consider the reverse polyblock associated to the finite set

$$\bar{T} = T \cup \{\mathbf{z}_1,\dots,\mathbf{z}_n\} \setminus \{\bar{\mathbf{z}}\}.$$

Such an operation is a branching operation, where the original box $[\bar{\mathbf{z}},\mathbf{b}]$ is replaced by the union of boxes $[\mathbf{z}_i,\mathbf{b}]$, $i = 1,\dots,n$, which outer approximate the set $[\bar{\mathbf{z}},\mathbf{b}] \cap G \cap H$.

Notice that the reverse polyblock $P(\bar{T})$ is a cover of the original feasible region. We recall that in the new reverse polyblock $P(\bar{T})$ we can keep *proper* vertices in $\bar{T}$, i.e., all points $\mathbf{z} \in \bar{T}$, such that there does not exist $\mathbf{z}' \in \bar{T} \setminus \{\mathbf{z}\}$ for which $\mathbf{z}' \leq \mathbf{z}$. Moreover, following the standard fathoming rule, if an upper bound $ub_{\mathbf{0}}$ is available, it is possible to remove from $\bar{T}$ all those points whose objective function value is not lower than $ub_{\mathbf{0}} - \varepsilon$. For what concerns the initialization of the reverse polyblock (the initial set T), $T = \{\mathbf{a}\}$ can be chosen, i.e., the initial reverse polyblock is the whole box $[\mathbf{a}, \mathbf{b}]$.

A further branching procedure for problem (5.1), when f, g_i, $i = 1, \ldots, m$, are increasing functions and $X = [\mathbf{a}, \mathbf{b}] \subset \mathbb{R}^n_+$, can be found in (Sun & Li, 2006).

5.3.6 Branching based on KKT conditions

Branching based on KKT conditions was discussed in Section 4.4.2. We briefly recall a few things here. Given the QP problem (4.50), four sets,

$$F_{\mathbf{x}}, F_{\mathbf{z}} \subseteq \{1, \ldots, n\}, F_{\mathbf{y}}, F_{\mathbf{b}-\mathbf{Ax}} \subseteq \{1, \ldots, m\},$$

are associated to each node of the BB tree. The sets satisfy conditions (4.91) and impose the constraints (4.92) in the node (so that a subset of the complementary conditions is satisfied at the node). Branching requires the selection of an index $j \in \{1, \ldots, n\} \setminus F_{\mathbf{x}} \cup F_{\mathbf{z}}$ or an index $i \in \{1, \ldots, m\} \setminus F_{\mathbf{y}} \cup F_{\mathbf{b}-\mathbf{Ax}}$. If an index j is selected, then the four sets associated to the child nodes are

$$\begin{array}{lll} \text{Child I} & : & F_{\mathbf{x}} = F_{\mathbf{x}} \cup \{j\}, \quad \text{the other sets are not changed,} \\[1ex] \text{Child II} & : & F_{\mathbf{z}} = F_{\mathbf{z}} \cup \{j\}, \quad \text{the other sets are not changed.} \end{array}$$

If an index i is selected, then the four sets associated to the child nodes are

$$\begin{array}{lll} \text{Child I} & : & F_{\mathbf{y}} = F_{\mathbf{y}} \cup \{i\}, \quad \text{the other sets are not changed,} \\[1ex] \text{Child II} & : & F_{\mathbf{b}-\mathbf{Ax}} = F_{\mathbf{b}-\mathbf{Ax}} \cup \{i\}, \quad \text{the other sets are not changed.} \end{array}$$

What is still to be clarified is how an index i or j can be chosen. The selection proposed in (Burer & Vandenbussche, 2008) is based on a maximum *normalized* violation of the complementary conditions. Assume that the optimal solution of the relaxed problem at a given node $\mathcal{N}$ is $(\mathbf{x}^\star\ \mathbf{y}^\star\ \mathbf{z}^\star)$. Let $\mathbf{s}^\star = \mathbf{b} - \mathbf{Ax}^\star$ be the value of the slack variables. It is given as understood that, possibly after the computation of upper bounds for $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$ and for the slack variables $\mathbf{s}$ in a preprocessing phase, each of these variables has been rescaled so that it takes values in the interval $[0, 1]$. Then, we compute

$$a_{\mathcal{N}} = \max\left\{ \max_{k=1,\ldots,n} \left\{x_k^\star z_k^\star\right\}, \max_{h=1,\ldots,m} \left\{s_h^\star y_h^\star\right\} \right\}.$$

If $a_{\mathcal{N}} = 0$, then it has been proved in Section 4.4.2 that the node $\mathcal{N}$ is fathomed. Otherwise, the selection is

$$\begin{cases} j \in \arg\max_{k=1,\ldots,n} \left\{x_k^\star z_k^\star\right\} & \text{if } a_{\mathcal{N}} = \max_{k=1,\ldots,n} \left\{x_k^\star z_k^\star\right\}, \\ i \in \arg\max_{h=1,\ldots,m} \left\{s_h^\star y_h^\star\right\} & \text{otherwise,} \end{cases}$$

i.e., the index associated to the maximum normalized violation of the complementary conditions is selected.

5.4 Convergence and finiteness results

In this section BB algorithms are analyzed in order to verify their ability to detect globally optimal solutions. Notice that we postpone to the following sections the description of two steps of the BB approach, namely the domain reduction techniques and the fathoming rules other than the standard one (5.2). While practically relevant, these are not necessary for the theoretical results proven in this section. Throughout the section the following rather standard assumptions are made.

- Geometric branching is considered; thus, to each node of the BB tree some set D_k with known geometrical shape (such as a box, a simplex, a cone, or an ellipsoid, as seen in Section 5.3) is associated. Then, the subset S_k corresponding to the node is equal to $D_k \cap S$. The set D_0 associated to the root node of the tree is such that $D_0 \supseteq S$.
- The lower bound for the node/subset S_k is obtained by solving the convex problem
$$\begin{aligned} lb(S_k) \;=\; \min \quad & \hat{f}^{D_k}(\mathbf{x}) \\ & \hat{g}_i^{D_k}(\mathbf{x}) \le 0, \quad i = 1,\ldots,m, \\ & \mathbf{x} \in X \cap D_k, \end{aligned}$$
where $\hat{f}^{D_k}$ and $\hat{g}_i^{D_k}$, $i = 1,\ldots,m$, are convex underestimators for f and g_i over D_k.
- The underestimators satisfy the *isotonic property*, i.e.,
$$\bar{D} \subseteq \tilde{D} \;\Rightarrow\; \hat{f}^{\bar{D}}(\mathbf{x}) \ge \hat{f}^{\tilde{D}}(\mathbf{x}), \;\; \hat{g}_i^{\bar{D}}(\mathbf{x}) \ge \hat{g}_i^{\tilde{D}}(\mathbf{x}), \;\; i = 1,\ldots,m, \quad \forall\, \mathbf{x} \in \bar{D}. \tag{5.24}$$

5.4.1 Convergence conditions when $\varepsilon = 0$, $\boldsymbol{\delta} = \mathbf{0}$

When $\varepsilon = 0$, $\boldsymbol{\delta} = \mathbf{0}$ finiteness of the BB approach cannot in general be guaranteed. Under suitable assumptions, however, it is at least possible to prove a convergence result if the BB algorithm does not stop in a finite number of iterations. By convergence here we mean that

$$\min_{\bar{S} \in \mathcal{C}_t} lb(\bar{S}) \to f^\star \quad \text{as } t \to \infty, \tag{5.25}$$

i.e., the overall lower bound $\min_{\bar{S} \in \mathcal{C}_t} lb(\bar{S})$ for the original problem converges to the optimal value of the problem. As already remarked, if the BB algorithm does not stop in a finite number of iterations, it generates at least an infinite nested sequence (5.6) of nodes/subsets. Note that under assumption (5.24), the corresponding sequence of lower bounds $\{lb(S_{k_j})\}$ is nondecreasing. Now, we introduce the following assumptions.

***Bound improving selection operation*:** See Definition 5.3.

***Exactness in the limit*:** See Definition 5.4.

***Exhaustiveness of the subdivision process*:** See Definition 5.5.

Remark 5.7. *If the sets D_k's are cones, the exhaustiveness condition is slightly different. It is required that the set of rays of the cones in an infinite nested sequence reduces to a singleton, which corresponds to the direction of the half-line to which the sets D_k's themselves converge. Such difference also implies some differences in the convergence theory for conical algorithms with respect to what we will see below. However, the differences are not particularly significative and we will not discuss them in what follows.*

Some easy examples will show that if at least one of these conditions is not satisfied, then the BB algorithm might be unable to converge, i.e., the value $\min_{\bar{S}\in\mathcal{C}_t} lb(\bar{S})$ might not converge to the optimal value $f^\star$. In the first example the exhaustiveness condition is not satisfied.

Example 5.21. Consider the GO problem

$$\begin{aligned} \min \quad & -x_1^2+3x_1-x_2^2+2x_2 \\ & -x_1+x_2 \le 1, \\ & x_1-x_2 \le 1, \\ & x_1+x_2 \le 3, \\ & -x_1-x_2 \le -1, \\ & 0 \le x_1,x_2 \le 2, \end{aligned}$$

whose global optimal solution is the vertex (0 1) with optimal value $f^\star = 1$. Assume that the subdivision rule for boxes always bisects with respect to the x_1-edge. Then, a nested sequence generated by the BB approach is the following:

$$D_{k_j} = \left[0, \frac{1}{2^{j-1}}\right] \times [0,2], \quad j = 0, \ldots. \tag{5.26}$$

Using the convex envelopes of $-x_1^2+3x_1$ and $-x_2^2+2x_2$ over the given intervals, we have that the underestimating function over these boxes is

$$-\frac{1}{2^{j-1}}x_1+3x_1-2x_2+2x_2 = \left(3-\frac{1}{2^{j-1}}\right)x_1,$$

so that $lb(S_{k_j}) = 0 < f^\star = 1$, i.e., the algorithm does not converge. ∎

In the second example the convex underestimators are not exact in the limit.

Example 5.22. Consider the same instance as in Example 5.21, but

- assume that the subdivision rule now alternatively bisects along the x_1-edge and the x_2-edge;
- the convex underestimator employed is always the one over the initial box $[0,2] \times [0,2]$, i.e., x_1.

Then, an infinite nested sequence generated by the BB approach is $D_{k_0} = [0,2] \times [0,2]$, $D_{k_1} = [0,1] \times [0,2]$, and for $j \geq 2$,

$$D_{k_j} = \left[0, \frac{1}{2^{\left\lfloor \frac{j-1}{2} \right\rfloor}}\right] \times \left[1 - \frac{1}{2^{\left\lfloor \frac{j-2}{2} \right\rfloor}}, 1\right].$$

In spite of the fact that such a sequence converges to the optimal solution (0 1) of the problem, using the suggested convex underestimator x_1 we end up again with the lower bounds $lb(S_{k_j}) = 0 \neq 1$, i.e., we do not have convergence. ∎

Finally, we consider an example where the selection operation is not bound improving.

Example 5.23. Consider the GO problem

$$\begin{aligned} \min \quad & f(x_1,x_2) \;=\; -x_1^2 + (2+\rho)x_1 - x_2^2 + 2x_2 \\ & -x_1 + x_2 \leq 1, \\ & (2-\sqrt{2})x_1 + x_2 \leq 4 - \sqrt{2}, \\ & -\sqrt{2}x_1 + x_2 \geq -\sqrt{2}, \\ & x_1 + 7x_2 \geq 1, \\ & 7x_1 + x_2 \geq 1, \\ & 0 \leq x_1, x_2 \leq 2, \end{aligned}$$

where $\rho > 0$ is a small value. The optimal value of this concave optimization problem is $f^\star = \frac{15+4\rho}{32}$ and is attained at the vertex $(\frac{1}{8}\ \frac{1}{8})$. The objective of the relaxation over $D_0 = [0,2] \times [0,2]$ is ρx_1, so that the optimal value of the relaxation is $lb(D_0 \cap S) = 0$. It is attained at the vertex (0 1), so that we can update the upper bound value $ub_{\mathbf{0}} = f(0,1) = 1$. Assume that the first branching operation subdivides D_0 into

$$D_1^1 = [0,1] \times [0,2] \quad D_1^2 = [1,2] \times [0,2].$$

We notice that the objective of the relaxation over D_1^1 is now $(1+\rho)x_1$. Thus, the optimal value of the relaxation is $lb(D_1^1 \cap S) = 0$ and is still attained at the vertex (0 1). For what concerns D_1^2, we notice that the objective of the relaxation over it is $(\rho - 1)x_1 + 2$. Then, the optimal value of this relaxation is $lb(D_1^2 \cap S) = 2\rho > lb(D_1^1 \cap S)$, attained at the vertex $(2\ \sqrt{2})$. We can update $ub_{\mathbf{0}} = f(2,\sqrt{2}) = 2\rho + 2(\sqrt{2}-1) < 1$. Assume now that the selection rule is not bound improving but instead that it always selects the node with the largest lower bound for x_1. Then, D_1^1 will be selected only if at some iteration all the other nodes have been fathomed. We show that this never happens if we always perform a bisection of the selected nodes. Assume by contradiction that all nodes within D_1^2 are fathomed after a finite number of iterations. In particular, there must exist some node D_{k_j} such that $(2\ \sqrt{2}) \in D_{k_j}$ which is fathomed, i.e.,

$$lb(D_{k_j} \cap S) \geq ub_{\mathbf{0}} = f(2,\sqrt{2}). \tag{5.27}$$

Note that ub_0 can not be lower than $f(2,\sqrt{2})$, because this is the minimum value attained by f in D_1^2 and the unique other point at which f has been evaluated outside D_1^2 is (0 1). However, since $\sqrt{2}$ is irrational, the vertex $(2\ \sqrt{2}) \in D_{k_j}$ will certainly lie in the relative interior of one edge of D_{k_j} (recall that only bisections are performed), so that the value of the convex underestimator of f over D_{k_j} at $(2\ \sqrt{2}) \in D_{k_j}$ will always be strictly lower than $f(2,\sqrt{2})$. Consequently, (5.27) cannot be true. ■

All the above examples show that the lack of exhaustiveness (of the subdivision process) and/or exactness in the limit (of the convex underestimators) and/or bound improvement (of the selection operation) might cause the nonconvergence of the BB algorithm. What if all these properties are satisfied? Can we guarantee convergence of the BB algorithm in this case? The answer is yes and the proof is given in Theorem 5.25. In order to prove it, we first need a lemma.

Lemma 5.24. *For each set D_k such that $D_k \cap S \neq \emptyset$ and for each $\rho_1 > 0$, there exists some value $\rho_2 > 0$ such that*

$$\mathbf{x} \in D_k, \quad \max_{i=1,\dots,m} g_i(\mathbf{x}) \leq \rho_2 \quad \Rightarrow \quad f(\mathbf{x}) \geq \min_{\mathbf{y}\in D_k\cap S} f(\mathbf{y}) - \rho_1.$$

Proof. The proof is an immediate consequence of the continuity of f and g_i, $i = 1,\dots,m$, and of the compactness of S. □

Theorem 5.25. *If the three conditions of exhaustiveness, exactness in the limit, and bound improving selection operation are satisfied, then the BB algorithm with $\varepsilon = 0$, $\boldsymbol{\delta} = \mathbf{0}$ either*

- *stops after a finite number of iterations if $S = \emptyset$ or*
- *stops after a finite number of iterations and the value ub_0 is equal to the optimal value $f^\star$ of the GO problem; or*
- *the algorithm converges, i.e., (5.25) is true, and at least one infinite nested sequence generated by the algorithm converges to a globally optimal solution.*

Proof. Let us first consider the case where $S = \emptyset$. In this case, if we define

$$g^\star = \min_{\mathbf{x}\in X} \max_{i=1,\dots,m} g_i(\mathbf{x}),$$

then $g^\star > 0$. Assume by contradiction that the BB algorithm does not stop after a finite number of iterations. Then, it generates at least an infinite nested sequence (5.6). By exhaustiveness and exactness in the limit we have that for j large enough

$$g_i(\mathbf{x}) - \hat{g}_i^{D_{k_j}}(\mathbf{x}) < g^\star \quad \forall\, \mathbf{x} \in D_{k_j},\ \forall\, i = 1,\dots,m,$$

so that

$$\min_{\mathbf{x}\in D_{k_j}} \max_{i=1,\dots,m} \hat{g}_i^{D_{k_j}}(\mathbf{x}) > 0,$$

and the node $S_{k_j} = D_{k_j} \cap S$ is deleted by infeasibility, thus contradicting the infinity of the nested sequence.

Next, let us assume that $S \neq \emptyset$ and that the algorithm terminates after a finite number of iterations. For a given optimal solution $\mathbf{x}^\star$ of the problem, there certainly exists a finite nested sequence of subsets generated by the algorithm with the property that $\mathbf{x}^\star$ belongs to all the sets of the sequence. Let S_K be the last node/subset of the sequence. Since the algorithm is finite, then $lb(S_K) \geq ub_\mathbf{0} \geq f^\star$ must be true, i.e., the node is fathomed. On the other hand, since $\mathbf{x}^\star \in S_K$, $lb(S_K) \leq f^\star$ must also be true. Therefore, $ub_\mathbf{0} = f^\star$ and the proof is complete.

The third case is also the most complicated one. We first prove that (5.25) is satisfied. Note that under assumption (5.24), the sequence $\{\min_{\bar{S} \in \mathcal{C}_t} lb(\bar{S})\}$ is nondecreasing, and its limit can not be larger than $f^\star$ (since $lb(\bar{S}) \leq f^\star$ for some subset $\bar{S}$ such that $\mathbf{x}^\star \in \bar{S}$). Assume by contradiction that the limit of the nondecreasing sequence is $\tilde{f} < f^\star$. In Lemma 5.24 let $\rho_1 = (f^\star - \tilde{f})/2$ and let ρ_2 be defined as in the statement of the lemma. Because of the conditions of exhaustiveness and exactness in the limit, there exists some finite level of the BB tree such that, for each node/subset $S_k = D_k \cap S$ below this level, either $D_k \cap S = \emptyset$ and infeasibility will be detected after a finite number of further subdivisions (see the first part of the proof), or $D_k \cap S \neq \emptyset$ and

$$lb(S_k) \geq f^\star - \frac{3}{2}\rho_1 > \tilde{f}.$$

Therefore, there exists some level ℓ below which all nodes are either deleted or their lower bound is higher than $\min_{\bar{S} \in \mathcal{C}_t} lb(\bar{S})$ and than the limit $\tilde{f}$. Then, each time the selection operation is a bound improving one (which must occur infinitely often), one node above level ℓ is selected. Since the number of such nodes is finite, after a finite number of iterations all leaves of the BB tree will lie below level ℓ, so that

$$\min_{\bar{S} \in \mathcal{C}_t} lb(\bar{S}) > \tilde{f},$$

which contradicts the initial assumption. In a similar way we prove that at least one infinite nested sequence converges to a globally optimal solution. Again, we prove this by contradiction. Assume that for all t larger than some value $\bar{t}$, all globally optimal solutions lie in some subsets/leaves of the BB tree $\tilde{S}_1, \ldots, \tilde{S}_u$ which are not further subdivided by the algorithm. Note that for all these nodes $lb(\tilde{S}_j) \leq f^\star$, $j = 1, \ldots, u$. Then, if we consider the union of all the sets

$$\mathcal{C}_{\bar{t}} \setminus \{\tilde{S}_1, \ldots, \tilde{S}_u\}$$

this is a compact set not containing any optimal solution of f over S, so that the minimum value of f over this set is $\bar{f} > f^\star$. In Lemma 5.24 let $\rho_1 = (\bar{f} - f^\star)/2$ and ρ_2 be the corresponding value. Then, as before, there exists some level, say ℓ', of the BB tree such that, for each node/subset $S_k = D_k \cap S$ below level ℓ', either the node is deleted by infeasibility or

$$lb(S_k) \geq \bar{f} - \frac{3}{2}\rho_1 > f^\star,$$

so that the lower bound is higher than $lb(\tilde{S}_j)$, $j = 1, \ldots, u$. Therefore, each time the selection operation is a bound improving one, we select one node above level ℓ'. Since the number of such nodes is finite, after a finite number of iterations we need to select a node/subset in $\{\tilde{S}_1, \ldots, \tilde{S}_u\}$ and we are thus led to a contradiction. □

Remark 5.8. *The three conditions of exhaustiveness, exactness in the limit, and bound improvement normally occur in practical BB algorithms and it is often relatively easy to show that they are fulfilled, although some cases also occur where the convergence proof is less obvious (see, e.g., (Duer,* 2001*; Tuy,* 2005a*) for cases where dual Lagrangian bounds are employed).*

In (Horst & Tuy, 1993*) convergence is proven under even more general conditions, such as* strong consistency *of the lower bounding procedure: for each infinite nested sequence* (5.6) *such that*

$$\bigcap_{j=0}^{\infty} D_{k_j} \bigcap S = \bar{S} \neq \emptyset,$$

there exists an infinite subsequence $\{D_{k_{j'}}\}$ *such that*

$$lb(S_{k_{j'}}) \to \min_{\mathbf{x} \in \bar{S}} f(\mathbf{x}), \quad as \ \ j' \to \infty. \tag{5.28}$$

We note that exhaustiveness and exactness in the limit imply (5.28)*, which is thus a more general condition.*

For a discussion about general conditions under which convergence is guaranteed we also refer to (X. Yang & Sun, 2007*).*

Remark 5.9. *As already remarked, the exhaustiveness condition can be easily seen to be satisfied by the bisection subdivision rule. Tuy (Tuy,* 1991b*) proves exhaustiveness of the normal subdivision rule (see Section* 5.3.3*). For concave minimization problems over polytopes, convergence when the ω-subdivision rule alone is employed is proven for conical algorithms in (Jaumard & Meyer,* 2001*; Locatelli,* 1999*), and for simplicial algorithms in (Locatelli & Raber,* 2000*). Within the framework of simplicial algorithms a convergence proof for a variant of the ω-subdivision rule—as well as for a strategy, called ω-bisection, where a bisection is performed with respect to an edge of the face of the current simplex whose interior contains the optimal solution of the current relaxation—is given in (Kuno & Buckland,* 2012*).*

Remark 5.10. *We defined the convergence of the BB algorithm through* (5.25)*. A preferable definition should also involve the convergence of the upper bound ub_0 to $f^\star$. If the constraint functions g_i's are convex, then, if $S \neq \emptyset$ it is usually guaranteed that a finite value will be assigned to ub_0. Indeed, in this case each lower bound computation over nonempty subsets of the feasible region returns a feasible solution which can be employed to possibly update the upper bound value. Then, in such cases convergence of ub_0 to $f^\star$ can also be guaranteed. However, for nonconvex constraints it might happen that, even if $S \neq \emptyset$, no feasible solution is ever observed and ub_0 might remain fixed to the initial infinite value (see, e.g., Example* 5.27*).*

Remark 5.11. *As a final remark, notice that Example* 5.23 *with the additional constraint $x_1 \geq 1$ allows us to see that even under the conditions of Theorem* 5.25*, convergence for $\varepsilon = 0$ can be guaranteed but not finiteness.*

5.4.2 Finiteness result for $\boldsymbol{\varepsilon} > 0$, $\boldsymbol{\delta} > \mathbf{0}$

If $\varepsilon > 0$, $\boldsymbol{\delta} > \mathbf{0}$ (in fact, if some function g_i is convex, we might also set $\delta_i = 0$, although in practice, as already mentioned, a strictly positive feasibility tolerance is usually employed even for linear constraints) under weaker conditions than those of Theorem 5.25, one can prove finiteness of the BB algorithm. This is stated in the following theorem. Notice that in this case bound improvement of the selection operation is not required in order to prove convergence.

Theorem 5.26. *If the conditions of exhaustiveness and exactness in the limit are satisfied, then the BB algorithm with* $\varepsilon > 0$, $\boldsymbol{\delta} > \mathbf{0}$ *terminates after a finite number of iterations and*

- *either establishes that* $S = \emptyset$ *if* $ub_{\boldsymbol{\delta}} = \infty$;

- *or returns a* $(\varepsilon, \boldsymbol{\delta})$*-optimal solution if* $ub_{\boldsymbol{\delta}} < \infty$.

Proof. Assume first that the algorithm terminates after a finite number of iterations. Then, the following possibilities might occur.

- $ub_{\boldsymbol{\delta}} = \infty$: In view of the fathoming rule (5.2), each fathomed leaf node of the BB tree has a lower bound equal to ∞, i.e., the corresponding subset is empty. Since the union of all the subsets corresponding to leaves of the BB tree is equal to the whole feasible region, then the latter must be empty.

- $ub_{\boldsymbol{\delta}} < \infty$: If $S \neq \emptyset$, then the optimal solution $\mathbf{x}^\star$ of the original problem lies in some leaf node/subset of the BB tree, say $S^\star$. In view of the fathoming rule and of the definition of lower bound, we have

  $$ub_{\boldsymbol{\delta}} - \varepsilon \le lb(S^\star) \le f^\star,$$

  so that $ub_{\boldsymbol{\delta}} \le f^\star + \varepsilon$ and the $\boldsymbol{\delta}$-feasible point $\mathbf{z}$ at which $ub_{\boldsymbol{\delta}}$ is attained is an $(\varepsilon, \boldsymbol{\delta})$-optimal solution.

Now, let us assume by contradiction that the algorithm does not terminate after a finite number of iterations. Then, it must generate at least one infinite nested sequence (5.6). Such a sequence must converge to a point $\bar{\mathbf{x}} \in S$ (and $S = \emptyset$ cannot hold; see also the proof of Theorem 5.25). In view of the continuity of the functions g_i's and of exhaustiveness, for j large enough

$$g_i(\mathbf{x}) - g_i(\bar{\mathbf{x}}) \le \delta_i \quad \forall\, \mathbf{x} \in D_{k_j}, \;\; i = 1, \ldots, m,$$

and, in view of the feasibility of $\bar{\mathbf{x}}$, $g_i(\mathbf{x}) \le \delta_i$. Therefore, each point in D_{k_j} is a $\boldsymbol{\delta}$-feasible point. In particular, this means that the optimal solution $\hat{\mathbf{x}}_{k_j}$ of the subproblem which has to be solved to compute the lower bound over $D_{k_j} \cap S$ is a $\boldsymbol{\delta}$-feasible point. The rule to update the upper bound then guarantees that $ub_{\boldsymbol{\delta}} \le f(\hat{\mathbf{x}}_{k_j})$. Moreover, in view of exhaustiveness and exactness in the limit, for j large enough

$$f(\mathbf{x}) - \hat{f}^{D_{k_j}}(\mathbf{x}) < \varepsilon \quad \forall\, \mathbf{x} \in D_{k_j}.$$

In particular,

$$ub_\delta \leq f(\hat{\mathbf{x}}_{k_j}) < \hat{f}^{D_{k_j}}(\hat{\mathbf{x}}_{k_j}) + \varepsilon = lb(D_{k_j} \cap S) + \varepsilon,$$

so that the node/subset $D_{k_j} \cap S$ is fathomed, thus contradicting the infinity of the nested sequence. □

It might appear unsatisfactory that the algorithm is only able to return a $\boldsymbol{\delta}$-feasible solution even when feasible solutions exist. However, two simple one-dimensional examples will clarify that Cases (a) and (b) introduced at the end of Section 5.2.6 might happen even if the exhaustiveness and exactness in the limit conditions are satisfied.

Example 5.27. To see Case (a), one might consider the simple one-dimensional problem

$$\begin{aligned} \min \quad & x \\ & x^2 - 2 = 0, \\ & 0 \leq x \leq 2. \end{aligned}$$

The only feasible (and optimal) point is $x^\star = \sqrt{2}$. After rewriting the problem as

$$\begin{aligned} \min \quad & x \\ & x^2 - 2 \leq 0, \\ & x^2 - 2 \geq 0, \\ & 0 \leq x \leq 2, \end{aligned}$$

a possible convex relaxation over some interval $[\ell, u]$ is

$$\begin{aligned} \min \quad & x \\ & x^2 - 2 \leq 0, \\ & \ell^2 + \tfrac{u^2-\ell^2}{u-\ell}(x-\ell) - 2 \geq 0, \\ & \ell \leq x \leq u. \end{aligned}$$

If a bisection rule is introduced starting with the initial interval $[0,2]$, then exhaustiveness and exactness in the limit are satisfied. A nested sequence $[\ell_j, u_j]$ will be generated such that $\sqrt{2} \in (\ell_j, u_j)$ (indeed, the two extremes ℓ_j and u_j are rational numbers if bisections are performed). The feasible region of the relaxation over $[\ell_j, u_j]$ is some interval $[\ell'_j, \sqrt{2}]$, where $\ell_j < \ell'_j < \sqrt{2}$, and the optimal solution of the relaxation is equal to ℓ'_j. Thus, the unique feasible point $x^\star = \sqrt{2}$ is never detected and the algorithm stops as soon as $2 - \delta \leq (\ell'_j)^2$ (i.e., as soon as ℓ'_j is δ-feasible). ■

Example 5.28. To see Case (b), consider the one-dimensional problem

$$\begin{aligned} \min \quad & -x \\ & (x-1)(x-\sqrt{2})^2 \leq 0, \\ & 0 \leq x \leq 2, \end{aligned}$$

whose feasible region is $[0,1] \cup \{\sqrt{2}\}$ and the optimal value is attained at the isolated feasible point $x^\star = \sqrt{2}$. Again, assume that bisections are performed starting with the interval $[0,2]$. Lower bounds are computed through interval analysis and the function values are computed at midpoints of the intervals generated by the bisections. Exhaustiveness and exactness in the limit are satisfied. The feasible point $\bar{x} = 1$ (which is also the best one in $[0,1]$) is immediately detected. As in the previous example, a nested sequence $[\ell_j, u_j]$ will be generated such that $\sqrt{2} \in (\ell_j, u_j)$ (again, the two extremes ℓ_j and u_j are rational numbers if bisections are performed), and the objective function value will never be evaluated at the optimal solution $x^\star = \sqrt{2}$. Therefore, the best observed value at feasible points (-1) remains bounded away from the optimal value of the problem $(-\sqrt{2})$. ■

In the previous example one can see that, as $\boldsymbol{\delta} \to \mathbf{0}$, $ub_{\boldsymbol{\delta}}$ converges to $f^\star$. In view of the continuity assumption for all the functions, this result is always true when $S \neq \emptyset$ (if $S = \emptyset$, for $\boldsymbol{\delta}$ small enough, similar to Theorem 5.25, we can prove that infeasibility is detected after a finite number of iterations). However, for some fixed $\boldsymbol{\delta}$ there is no guarantee that the value $ub_{\boldsymbol{\delta}}$ is "close" to the optimal value $f^\star$ (and $\mathbf{z}$ close to a global optimal solution). This can be seen from the following example, similar to the previous one.

Example 5.29. Consider the one-dimensional problem

$$\begin{aligned} \min \quad & -x \\ & (x-1)(x-\sqrt{2})^2 + \rho \leq 0 \\ & 0 \leq x \leq 2, \end{aligned}$$

where $\rho > 0$ is a small value. The feasible region is $[0,\eta]$ for some $\eta < 1$. The optimal value is attained at $x^\star = \eta$. If we set $\delta > \rho$, then there exists some interval $[\xi_1, \xi_2]$ such that $\eta < \xi_1$, $\sqrt{2} \in (\xi_1, \xi_2)$, and all its points are δ-feasible. Therefore, if the BB algorithm evaluates the objective function value within this interval, the value $ub_{\boldsymbol{\delta}}$ will be lower than and bounded away from the optimal value of the problem. ■

The examples above show that the notion of $(\varepsilon, \boldsymbol{\delta})$-optimal solution might be a critical one. Observing that the main difficulty associated with this notion is the presence of isolated feasible solutions, in a series of papers (Tuy, 2005b; Tuy & Hoai-Phuong, 2007; Tuy, 2010) Tuy proposes an alternative definition of approximate solution. He introduces the notion of *essential feasible set* $S^\star$, made up by all feasible solutions which are not isolated. Then, for some $\varepsilon > 0$ he defines an *essential ε-optimal solution* $\bar{\mathbf{x}}$ as a point in $S^\star$ such that

$$f(\bar{\mathbf{x}}) - \varepsilon \leq \inf\{f(\mathbf{x}) \ : \ g_i(\mathbf{x}) \leq -\varepsilon,\ i = 1,\ldots,m,\ \mathbf{x} \in X\}.$$

Tuy also proposes a method which, under suitable assumptions, fulfilled, e.g., by difference-of-convex and difference-of-monotonic optimization problems, for any $\varepsilon > 0$ guarantees that after a finite time either an essential ε-optimal solution is returned or $S^\star = \emptyset$ is established.

5.5 Domain reduction

Domain reduction techniques are not necessary to guarantee convergence or finiteness results for BB algorithms. However, they might have a strong impact on their practical performance. To support this claim we may, for instance, note that the "R" in the name of the software BARON (see, e.g., (Sahinidis, 1996)), which, according to a computational study carried on in (Neumaier, Shcherbina, Huyer, & Vinkó, 2005) is the best-performing software for the solution of nonconvex problems, stands for Reduce and refers to domain reduction strategies. Such techniques add "simple" inequalities to the original problem (e.g., linear ones, also called *cutting planes*), or strengthen existing ones. They can be classified into two broad categories.

Feasibility based: The added inequalities or the strengthened ones are satisfied by all feasible points. They are usually employed for problems with nonconvex feasible regions in order to get improved convex relaxations of such regions.

Optimality based: The added inequalities or the strengthened ones are satisfied by at least one optimal solution if no optimal solution has been previously observed. In this case some feasible solutions (and, in some cases, even some optimal ones) might not satisfy such inequalities.

In the field of domain reductions an important role is played by *range reduction* techniques. These strengthen existing bounds over the variables of the problem and, as we will further comment later on, they allow us to improve the quality of convex underestimators. In view of their importance we discuss them separately in Section 5.5.1, while more general domain reduction strategies will be discussed in Section 5.5.2.

5.5.1 Range reduction strategies

A range reduction (RR) strategy can be defined as follows. Let $B = \prod_{i=1}^{n}[\ell_i, u_i]$ be a box containing the feasible region S of the nonconvex problem. Let $\mathcal{B}_n$ be the set of all n-dimensional boxes. A RR strategy can be seen as a function

$$R_{f,S} : \mathcal{B}_n \rightarrow \mathcal{B}_n$$

satisfying the property

$$R_{f,S}(B) \subseteq B \quad \forall\, B \in \mathcal{B}_n.$$

Many RR strategies have been proposed in the literature (see, e.g., (Hamed & McCormick, 1993; Hansen, Jaumard, & Lu, 1991; Maranas & Floudas, 1997; Ryoo & Sahinidis, 1996; Zamora & Grossmann, 1999)). As for general domain reduction strategies, they are classified into two categories: *feasibility based* and *optimality based* RRs. A valid feasibility based RR is such that

$$\forall\, B \supseteq S \;:\; R_{f,S}(B) \supseteq S,$$

i.e., the result of a reduction is a box which also contains the feasible region S. Therefore, feasibility based RRs do not remove feasible points. For optimality based strategies, assume that a finite upper bound ub_{δ} is available. Then, a valid optimality based RR satisfies

$$\forall\, B \supseteq \{\mathbf{x} \in S \;:\; f(\mathbf{x}) \leq ub_{\delta}\}, \quad R_{f,S}(B) \supseteq \{\mathbf{x} \in S \;:\; f(\mathbf{x}) \leq ub_{\delta}\},$$

i.e., optimality based RR strategies might remove feasible points but no feasible point with function value not larger than ub_{δ} (and, therefore, no feasible solution which is a δ-optimal solution) is removed. In particular, if $\delta = \mathbf{0}$ (e.g., if the constraints are convex ones) no optimal solution is removed. Of course, we would like that a reduction is as tight as possible. For what concerns feasibility based RRs, the tightest possible reduction for a single variable x_k is obtained by solving the pair of problems

$$\min / \max_{\mathbf{x}\in S} x_k, \tag{5.29}$$

while for optimality based RRs the tightest reduction is obtained by solving the pair of problems

$$\min / \max_{\mathbf{x}\in S\,:\, f(\mathbf{x})\leq ub_{\delta}} x_k. \tag{5.30}$$

If the feasible region S is a convex set, then problems (5.29) are convex (linear if the feasible region is a polyhedron). With respect to problems (5.30), these are GO problems themselves if f is a nonconvex function, even when S is a convex set. Therefore, identification of the tightest optimality based RR may be a very hard task (the same is true for feasibility based RR strategies when S is not a convex set). However, the formulation of the two problems (5.29) and (5.30) returning the tightest, respectively, feasibility and optimality based RRs immediately suggests how to define valid RR strategies: any lower bound for minimization problem and any upper bound of the maximization problem in (5.29) and (5.30) define a valid (feasibility or optimality based) RR. Therefore, we are not only interested in detecting lower bounds for the original problem, but also lower and upper bounds for the problems (5.29) and (5.30). These further bound computations require an extra computational effort. Which is the advantage of such effort? For simplicity throughout the section we will only discuss the case of a convex set S and optimality based RRs, but the discussion can be extended to the case of a nonconvex set and of feasibility based RRs. If S is a convex set, then a lower bound for the original GO problem can be computed by replacing the original objective function with a convex underestimator $\hat{f}$ of f over S. As seen in Chapter 4, many convex underestimators depend on the range of the variables, and the tighter the range, the sharper the underestimator. Therefore, reducing the range of the variables has the effect of improving the convex underestimator, which in turn has the effect of improving the lower bound of the problem. But a closer look reveals that the effect of the reduction is not merely restricted to the improvement of the lower bound of the original problem, through the improvement of the convex underestimators, but there is also a feedback effect

$$\text{improved range} \quad \leftrightarrow \quad \text{improved convex understimators.}$$

Indeed, the convex underestimator $\hat{f}$ can also be employed in (5.30) in place of f, thus making the problem a convex one (recall that we are assuming that S is a convex set). This way, once we have reduced the range of the variables and obtained a better convex underestimator, we can use such an underestimator for a further RR, and so on until this iterative procedure converges or, in practice, until it produces significative reductions. An example will clarify what we stated above.

Example 5.30. Consider the following problem:

$$\begin{aligned} \min \quad & x_1x_2 + x_1 + x_2 \\ & x_1 + 3x_2 \geq 1, \\ & 3x_1 + x_2 \geq 1, \\ & a \leq x_1, x_2 \leq b. \end{aligned}$$

Initially we set $a = 0$, $b = 1$ and consider the upper bound $ub_{\mathbf{0}} = \frac{9}{16}$ (which is, in fact, the optimal value of the problem). A lower bound for this problem, using the convex envelope of the bilinear term x_1x_2 over the box $[a,b]^2$ (see Section 4.2.7), is obtained by solving the following linear problem:

$$\begin{aligned} \min \quad & z + x_1 + x_2 \\ & z \geq ax_1 + ax_2 - a^2, \\ & z \geq bx_1 + bx_2 - b^2, \\ & x_1 + 3x_2 \geq 1, \\ & 3x_1 + x_2 \geq 1, \\ & a \leq x_1, x_2 \leq b. \end{aligned}$$

Easy computations lead to the lower bound

$$\begin{cases} \frac{a+1}{2} - a^2 & \text{if } b + a \geq \frac{1}{2}, \\ \frac{b+1}{2} - b^2 & \text{otherwise.} \end{cases}$$

We can reduce the range of the variable x_1 (and, by symmetry, also that of x_2) by solving the two problems

$$\begin{aligned} \min/\max \quad & x_1 \\ & z + x_1 + x_2 \leq \frac{9}{16}, \\ & z \geq ax_1 + ax_2 - a^2, \\ & z \geq bx_1 + bx_2 - b^2, \\ & x_1 + 3x_2 \geq 1, \\ & 3x_1 + x_2 \geq 1, \\ & a \leq x_1, x_2 \leq b. \end{aligned}$$

Again, after easy computations the RR $[a', b']$ with

$$\begin{aligned} a' &= \frac{1}{2} - \frac{1}{32}\frac{9+16a^2}{a+1}, \\ b' &= \frac{3}{32}\frac{9+16a^2}{a+1} - \frac{1}{2} \end{aligned}$$

is obtained. Therefore, if in the initial box we have $a = 0$ and $b = 1$, the initial lower bound is $\frac{1}{2}$, while the range of the two variables is reduced to $[\frac{7}{32}, \frac{11}{32}]$. If we recompute the lower bound after the reduction we get the new bound $\frac{575}{1024}$, while a further reduction leads to the new range

$$\left[\frac{623}{2496}, \frac{209}{832}\right] \approx [0.249599, 0.2512019].$$

Iterating this procedure, the lower bound converges to the upper bound $ub_{\mathbf{0}} = \frac{9}{16}$, while the range of the variables reduces to the single point $\frac{1}{4}$, i.e., the box is reduced to the singleton corresponding to the optimal solution. ∎

Range reductions and Lagrange multipliers

In some cases when solving a subproblem to reduce the range of a variable, it is possible to exploit the information collected during the solution of the subproblem to reduce the range of some other variable and/or to improve the lower bound of the problem. Elaborating on (Tawarmalani & Sahinidis, 2004) and (Zamora & Grossmann, 1999), consider a function h, for which an upper bound h_{ub} over the current box is known. As before, assume that the functions g_i's are convex. Let us solve the following problem dependent on the perturbation vector $\mathbf{q} = (q_0, q_1, \dots, q_m) \in \mathbb{R}^{m+1}$:

$$\begin{aligned} \phi(\mathbf{q}) = \min \quad & h(\mathbf{x}) \\ & g_0(\mathbf{x}) = \hat{f}(\mathbf{x}) - ub_{\mathbf{0}} \le q_0, \\ & g_i(\mathbf{x}) \le q_i, \qquad\qquad i = 1, \dots, m, \\ & \mathbf{x} \in B, \end{aligned} \tag{5.31}$$

where $\hat{f}$ is a convex underestimator for f over S. Let $\bar{\lambda} \in \mathbb{R}^{m+1}_{+}$ be the optimal dual Lagrange multiplier for $\mathbf{q} = \mathbf{0}$. Then, standard results (see, e.g., (Boyd & Vandenberghe, 2004)) show that

$$\phi(\mathbf{q}) \ge \phi(\mathbf{0}) - \bar{\lambda}^T \mathbf{q} \quad \forall\, \mathbf{q} \in \mathbb{R}^{m+1}. \tag{5.32}$$

Assume now that the constraint $g_i(\mathbf{x}) \le q_i$, $i \in \{0, 1, \dots, m\}$, is active at the optimal solution of problem (5.31) for $\mathbf{q} = \mathbf{0}$, and that $\bar{\lambda}_i > 0$. If we consider a vector

$$\mathbf{q} = (0 \dots 0\ q_i\ 0 \dots 0),$$

where $q_i \le 0$, then it follows from (5.32) that

$$\phi(\mathbf{q}) \ge \phi(\mathbf{0}) - \bar{\lambda}_i g_i(\mathbf{x}),$$

for each $\mathbf{x}$ such that $g_i(\mathbf{x}) = q_i$. Equivalently,

$$g_i(\mathbf{x}) \ge \frac{\phi(\mathbf{0}) - \phi(\mathbf{q})}{\bar{\lambda}_i}.$$

In view of $\phi(\mathbf{q}) \le h_{ub}$, the dependency on $\mathbf{q}$ can be removed and we end up with

$$g_i(\mathbf{x}) \ge \frac{\phi(\mathbf{0}) - h_{ub}}{\bar{\lambda}_i}. \tag{5.33}$$

In order to see possible applications of this result, assume that $h(\mathbf{x}) = x_j$, i.e., the problem (5.31) is the problem that we solve to improve the lower bound for the variable x_j. Then, an upper bound for h is the current upper bound u_j for the variable x_j, while $\phi(\mathbf{0})$ is the current lower bound ℓ_j for x_j, after solving the reduction subproblem. For $i = 0$, (5.33) reduces to

$$\hat{f}(\mathbf{x}) \geq ub_{\mathbf{0}} - \frac{u_j - \ell_j}{\bar{\lambda}_0},$$

i.e., the right-hand side is a valid lower bound for the GO problem. For $i = 1, \ldots, m$, and $g_i(\mathbf{x}) = \ell_i - x_i$, (5.33) reduces to

$$x_i \leq \ell_i + \frac{u_j - \ell_j}{\bar{\lambda}_i},$$

i.e., the right-hand side is a valid upper bound for x_i. We note that the above results have been employed to (possibly) improve the lower bound of the problem or to tighten the bounds for a variable. In fact, they can be used in a more general way. For instance, if g_i is a linear function, then (5.33) is a valid linear inequality satisfied by the optimal solutions of the problem.

Relations between some range reductions

Standard range reduction (SRR) is defined as the RR obtained from (5.30) by substituting the function f with a convex underestimator $\hat{f}$, so that the resulting problem is a convex one (once again we recall that we are assuming that S is a convex set). Denote the corresponding new lower and upper bound for x_k as ℓ_k^S and u_k^S, respectively. An obvious way to improve the results of SRR is by iterating it (this has already been done in Example 5.30 for each variable).

Iterated Standard Range Reduction (ISRR)

Step 0 Set $\ell_k^0 = \ell_k$, $u_k^0 = u_k$, and $i = 0$.

Step 1 Solve the two problems

$$\min/\max\left\{x_k : \hat{f}^i(\mathbf{x}) \leq ub_{\mathbf{0}},\ \mathbf{x} \in S \cap B_i\right\}, \tag{5.34}$$

where

$$B_i = \{\mathbf{x} \in \mathbb{R}^n \ :\ \ell_k^i \leq x_k \leq u_k^i,\ \ell_j \leq x_j \leq u_j,\ j \neq k\} \tag{5.35}$$

is the current box and $\hat{f}^i$ is the underestimator of f with respect to B_i. Denote by ℓ_k^{i+1} and u_k^{i+1}, respectively, the newly obtained lower and upper bounds on x_k.

Step 2 If $\ell_k^{i+1} = \ell_k^i$ and $u_k^{i+1} = u_k^i$, then stop. Otherwise, set $i = i + 1$ and repeat Step 1.

The above procedure either terminates after a finite number of iterations or generates an infinite nondecreasing sequence $\{\ell_k^i\}$ of lower bounds and an infinite nonincreasing sequence $\{u_k^i\}$ of upper bounds. In both cases the values

$$\ell_k^{IS} = \sup_i \ell_k^i = \lim_{i\to\infty} \ell_k^i \quad \text{and} \quad u_k^{IS} = \inf_i u_k^i = \lim_{i\to\infty} u_k^i$$

define a valid RR for variable x_k. We introduce now a different RR strategy (see (Caprara & Locatelli, 2010)). Start by fixing $x_k = t$ in the original problem, thus reducing by one the dimension of the problem and eliminating all the nonlinearities due to variable x_k. This fact justifies the name *nonlinearities removal range reduction* (NRRR). For any vector $\mathbf{x} \in \mathbb{R}^n$ define

$$\mathbf{x}^{-k} = (x_1 \dots x_{k-1}\ x_{k+1} \dots x_n) \in \mathbb{R}^{n-1}$$

and

$$\mathbf{x}^{(k)}(t) = (x_1 \dots\ x_{k-1}\ t\ x_{k+1} \dots x_n).$$

By fixing $x_k = t$ we are led to the problem

$$\min\{f_k^t(\mathbf{x}^{-k}) : \mathbf{x}^{-k} \in S_k^t \cap B^{-k}\}, \tag{5.36}$$

where

$$B^{-k} = \{\mathbf{x}^{-k} \in \mathbb{R}^{n-1}\ :\ \ell_j \leq x_j \leq u_j,\ j \neq k\} \tag{5.37}$$

is the box for the variables $x_1, \dots, x_{k-1}, x_{k+1}, \dots, x_n$;

$$S_k^t = \{\mathbf{x}^{-k} \in \mathbb{R}^{n-1}\ :\ \mathbf{x}^{(k)}(t) \in S\}$$

is the (closed and nonempty) convex set given by the projection of the intersection of S with the hyperplane $\{\mathbf{x} \in \mathbb{R}^n : x_k = t\}$ on the subspace of the variables $x_1, \dots, x_{k-1}, x_{k+1}, \dots, x_n$; and

$$f_k^t(\mathbf{x}^{-k}) = f(\mathbf{x}^{(k)}(t))$$

is the associated restriction of f. Then, we consider a convex underestimator $\hat{f}_k^t$ for f_k^t over B^{-k}. Finally, define the function h_k as the lower bound for (5.36) derived as

$$h_k(t) = \min\{\hat{f}_k^t(\mathbf{x}^{-k}) : \mathbf{x}^{-k} \in S_k^t \cap B^{-k}\}. \tag{5.38}$$

Once function h_k is available along with the upper bound $ub_{\mathbf{0}}$ on the optimal value, compute the values

$$\ell_k^{NR} = \inf\{t\ :\ h_k(t) \leq ub_{\mathbf{0}},\quad t \in [\ell_k, u_k]\} \tag{5.39}$$

and

$$u_k^{NR} = \sup\{t\ :\ h_k(t) \leq ub_{\mathbf{0}},\quad t \in [\ell_k, u_k]\}. \tag{5.40}$$

Then, obviously, the domain of x_k can be reduced to $[\ell_k^{NR}, u_k^{NR}]$. For the sake of precision, we point out that the two extremes in (5.39) and (5.40) are attained, i.e., "inf" and "sup" can be replaced by "min" and "max," if the function h_k is a continuous one, which is the case under mild assumptions (see (Caprara & Locatelli, 2010)).

Having defined the new domain reduction, we would like to establish its relation with SRR and ISRR. In order to establish the relation with SRR we introduce the following natural assumption.

Assumption 5.1. *For each $t \in [\ell_k, u_k]$ and for each $\mathbf{x} \in B \cap S$ with $x_k = t$, we have that*

$$\hat{f}_k^t(\mathbf{x}^{-k}) \geq \hat{f}(\mathbf{x}).$$

The assumption simply states that, after fixing the value of the variable x_k to t, we are able to find an underestimator $\hat{f}_k^t$ of f which is better (i.e., larger or, at least, not smaller) than $\hat{f}$.

Under this assumption we can prove that NRRR dominates SRR.

Observation 5.1. *Under Assumption* 5.1*, we have that*

$$[\ell_k^{NR}, u_k^{NR}] \subseteq [\ell_k^S, u_k^S].$$

Proof. We just prove that $\ell_k^S \leq \ell_k^{NR}$. The other inequality can be proven in a completely analogous way. Notice that in order to prove $\ell_k^S \leq \ell_k^{NR}$ it is sufficient to prove that

$$h_k(t) > ub_{\mathbf{0}}, \quad t \in [\ell_k, \ell_k^S). \tag{5.41}$$

For all feasible solutions of the problem with $x_k < \ell_k^S$ we have, by definition, that $\hat{f}(\mathbf{x}) > ub_{\mathbf{0}}$. Moreover, also in view of Assumption 5.1, we have that

$$\hat{f}_k^t(\mathbf{x}^{-k}) \geq \hat{f}(\mathbf{x}) \text{ for } \mathbf{x} \text{ such that } x_k = t \quad \Rightarrow \quad h_k(t) \geq \min_{\mathbf{x} \in B \cap S \,:\, x_k = t} \hat{f}(\mathbf{x}),$$

which, combined with $\hat{f}(\mathbf{x}) > ub_{\mathbf{0}}$ for $\mathbf{x}$ such that $x_k < \ell_k^S$, gives (5.41) and then the desired result. □

Simple examples, such as Example 5.31 below, show that strict dominance holds. The dominance of NRRR with respect to SRR can also be extended to ISRR, if the following extension of Assumption 5.1 is satisfied.

Assumption 5.2. *For each i, for each $t \in [\ell_k^i, u_k^i]$, and for each $\mathbf{x} \in B \cap S$ with $x_k = t$, we have that*

$$\hat{f}_k^t(\mathbf{x}^{-k}) \geq \hat{f}^i(\mathbf{x}).$$

Indeed, if Assumption 5.2 is satisfied, the proof of Observation 5.1 can be immediately extended to show that each intermediate bound ℓ_k^i computed by ISRR is dominated by ℓ_k^{NR}. Then, the following dominance result holds.

Observation 5.2. *Under Assumption* 5.2*, we have that*

$$[\ell_k^{NR}, u_k^{NR}] \subseteq [\ell_k^{IS}, u_k^{IS}].$$

In fact, under mild assumptions we will prove that equality holds. This equivalence result is of theoretical interest because it allows a new interpretation of ISRR. Indeed, it states the nonobvious fact that iterating SRR has the same effect as removing all the nonlinearities involving the variable x_k in the original problem and studying the resulting relaxation. Moreover, the result also shows that we can substitute the iterative solutions of problems performed by ISRR with the direct analysis of the function h_k performed by

NRRR. Of course, the latter requires us to know how the function h_k is defined, which depends on the particular problem at hand. In some special cases (such as concave separable functions, bilinear problems) parametric programming can be exploited to study the function h_k (see (Caprara & Locatelli, 2010)). Here we just report a simple example which is, however, significative because in this case ISRR only converges to the final bounds, while NRRR reaches them in a finite time.

Example 5.31. Consider the following example:

$$\begin{array}{ll}\min & -x_1^2 - x_2^2 \\ & x_1 + 3x_2 \leq 3, \\ & 44x_1 + 16x_2 \leq 45, \\ & 0 \leq x_1, x_2 \leq 1.\end{array}$$

Assume that the objective function has been evaluated at the point $(\frac{3}{4}\ \frac{3}{4})$, which is the global optimal solution and gives an upper bound $ub_{\mathbf{0}} = -\frac{9}{8}$. Let us try to reduce the upper bound for x_1. The function $h_{x_1}(t)$, needed in NRRR, is defined as

$$h_{x_1}(t) = -t^2 + \begin{array}{l}\min\ -x_2 \\ 3x_2 \leq 3 - t, \\ 16x_2 \leq 45 - 44t, \\ 0 \leq x_2 \leq 1,\end{array}$$

so that for $t \in [0,1]$

$$h_{x_1}(t) = -t^2 - \min\left\{\frac{3-t}{3}, \frac{45-44t}{16}\right\}$$

or, equivalently,

$$h_{x_1}(t) = -t^2 - \begin{cases} \frac{45-44t}{16}, & t \geq \frac{3}{4}, \\ \frac{3-t}{3}, & t \leq \frac{3}{4}. \end{cases}$$

In NRRR we need to search for

$$\max\{t\ :\ h_{x_1}(t) \leq -9/8\}$$

to get the upper bound for x_1,

$$u_1^{NR} = \frac{11 - \sqrt{13}}{8} = 0.923406.$$

Let's now move to ISRR. At iteration k, let u_k be the current upper bound on x_1 (obviously, $u_0 = 1$). At such iteration the upper bound is updated through the solution of the following problem:

$$\begin{array}{ll}\max & x_1 \\ & -u_k x_1 - x_2 \leq -\frac{9}{8}, \\ & x_1 + 3x_2 \leq 3, \\ & 44x_1 + 16x_2 \leq 45, \\ & 0 \leq x_2 \leq 1, \\ & 0 \leq x_1 \leq u_k.\end{array}$$

The optimal value of this problem and, thus, the new upper bound for x_1, is equal to

$$u_{k+1} = \frac{27}{44 - 16u_k} \quad (u_0 = 1)$$

($u_1 = 0.9643, u_2 = 0.945$, and so on). Therefore, we notice that in this case ISRR generates an infinite sequence of upper bounds. It is also easy to verify that the limit of this sequence is exactly equal to u_1^{NR}, as expected from the theory. ∎

Now, let us introduce the following two, quite reasonable, assumptions.

Assumption 5.3. *The functions $\hat{f}_k^t$'s are continuous both with respect to the variables $x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n$ and with respect to the parameter $t \in [\ell_k, u_k]$.*

Assumption 5.3 immediately implies continuity of h_k. The proof of this fact is reported in (Caprara & Locatelli, 2010) but can also be derived from other results already existing in the literature (Choquet, 1969; Rockafellar, 1971; Wets, 1974).

Assumption 5.4. *There exists some $\eta^\star > 0$ such that for each i, for each $t \in [\ell_k^i, u_k^i]$, and for each $\mathbf{x}^{-k} \in B^{-k} \cap S_k^t$*

$$\hat{f}^i(\mathbf{x}^{(k)}(t)) \geq \hat{f}_k^t(\mathbf{x}^{-k}) - \eta^\star \min\{t - \ell_k^i, u_k^i - t\}.$$

Assumption 5.4 simply states that the gap between $\hat{f}^i$ and $\hat{f}_k^t$ can not grow too fast as we move away from the borders of the current domain $[\ell_k^i, u_k^i]$ for x_k. Note that Assumptions 5.2 and 5.4 together give rise to the following bracketing property:

$$\hat{f}_k^t(\mathbf{x}^{-k}) \geq \hat{f}^i(\mathbf{x}^{(k)}(t)) \geq \hat{f}_k^t(\mathbf{x}^{-k}) - \eta^\star \min\{t - \ell_k^i, u_k^i - t\} \quad \forall\, \mathbf{x}^{(k)}(t) \in B.$$

In particular, we have that

$$\hat{f}_k^{\ell_k^i}(\mathbf{x}^{-k}) = \hat{f}^i(\mathbf{x}^{(k)}(\ell_k^i)) \text{ for } \mathbf{x}^{-k} \in B^{-k} \cap S_k^{\ell_k^i},$$

and

$$\hat{f}_k^{u_k^i}(\mathbf{x}^{-k}) = \hat{f}^i(\mathbf{x}^{(k)}(u_k^i)) \text{ for } \mathbf{x}^{-k} \in B^{-k} \cap S_k^{u_k^i}.$$

Under the above assumptions we can state the announced equivalence result.

Theorem 5.32. *Under Assumptions 5.2, 5.3, and 5.4, we have that*

$$[\ell_k^{NR}, u_k^{NR}] \equiv [\ell_k^{IS}, u_k^{IS}].$$

Proof. We just prove that $\ell_k^{NR} = \ell_k^{IS}$. The proof that $u_k^{NR} = u_k^{IS}$ is completely analogous. Let us assume, by contradiction, that $\ell_k^{IS} < \ell_k^{NR}$. The function h_k is continuous over the interval $[\ell_k, u_k]$. Then, under the assumption by contradiction we must have

$$h_k(\ell_k^i) \geq ub_{\mathbf{0}} + \rho,$$

for some $\rho > 0$ and for each i. In view of Assumption 5.4 we have that

$$\hat{f}^i(\mathbf{x}) \geq h_k(t) - \eta^\star \min\{t - \ell_k^i, u_k^i - t\},$$

for each $\mathbf{x} \in B \cap S$ with $x_k = t$. By definition we also have that there exists a feasible (in fact, optimal) solution $\mathbf{x}$ of problem (5.34) with $x_k = \ell_k^{i+1}$. Then

$$ub_\mathbf{0} \geq \hat{f}^i(\mathbf{x}) \geq h_k(\ell_k^{i+1}) - \eta^\star(\ell_k^{i+1} - \ell_k^i) \geq ub_\mathbf{0} + \rho - \eta^\star(\ell_k^{i+1} - \ell_k^i),$$

from which we have that for each i

$$\ell_k^{i+1} - \ell_k^i \geq \frac{\rho}{\eta^\star} > 0,$$

which contradicts the boundedness of the sequence $\{\ell_k^i\}$. □

We might wonder how strict the assumptions of the theorem are. In (Caprara & Locatelli, 2010) it is proven that Assumptions 5.2–5.4 are satisfied, e.g., if the original objective function f is Lipschitzian and the underestimators $\hat{f}^i$ are the convex envelopes of f over the corresponding boxes B_i.

Sequences of variables and range reductions

A final question about RR strategies is the following. Any RR strategy can be applied to all the variables and the whole procedure can be iterated until the range of at least one variable is reduced. Therefore, we might wonder whether the final result depends on the order in which the variables are considered or not. It turns out that the order has no impact on the final result (although it may have an impact on the computation times required to reach the final result). Any RR applied to a single variable x_k can be viewed as a procedure RR that takes in input the variable x_k and the current ranges $[\ell_j, u_j]$ for all the variables x_j, and returns a new, possibly reduced, range for x_k, i.e.,

$$[\ell_k, u_k] = \mathrm{RR}(x_k\ ;\ [\ell_j, u_j],\ j = 1, \ldots, n).$$

A common practice is to fix some order T of the variables, to reduce the domain of each variable in the given order, and to iteratively repeat this procedure until there are reductions (in practical implementations, large enough reductions) from one iteration to the next. The procedure is the following.

Multiple range reduction

Step 0 Let $T = \{x_{j_1}, \ldots, x_{j_n}\}$ be a given order of the variables. Set $\ell_j^0 = \ell_j$, $u_j^0 = u_j$ for $j = 1, \ldots, n$ and $i = 0$.

Step 1 For $k = 1, \ldots, n$ set

$$[\ell_{i_k}^{i+1}, u_{i_k}^{i+1}] = \mathrm{RR}(x_{i_k};\ [\ell_{i_j}^{i+1}, u_{i_j}^{i+1}],\ j = 1, \ldots, k-1,\ [\ell_{i_j}^i, u_{i_j}^i],\ j = k, \ldots, n).$$

Step 2 If $\ell_j^{i+1} = \ell_j^i$ and $u_j^{i+1} = u_j^i$ for $j = 1, \ldots, n$, then stop. Otherwise, set $i = i + 1$ and repeat Step 1.

We introduce the following, rather natural, monotonicity assumption.

Assumption 5.5. *The* RR *procedure is* monotone*, i.e., given domains* $[\tilde{\ell}_j, \tilde{u}_j]$ *and* $[\ell_j, u_j]$ *that satisfy*

$$[\tilde{\ell}_j, \tilde{u}_j] \subseteq [\ell_j, u_j], \; j = 1, \dots, n,$$

we have that, for any variable x_k*,*

$$\mathrm{RR}(x_k;\; [\tilde{\ell}_j, \tilde{u}_j],\; j = 1, \dots, n) \subseteq \mathrm{RR}(x_k;\; [\ell_j, u_j],\; j = 1, \dots, n).$$

Proposition 5.33. *Under Assumption* 5.5*, the domains computed by* multiple range reduction *satisfy*

$$\ell_j^i \to \bar{\ell}_j \quad \textit{and} \quad u_j^i \to \bar{u}_j \quad \textit{as} \quad i \to \infty, \; j = 1, \dots, n,$$

and the limits $\bar{\ell}_j$ *and* $\bar{u}_j$ *do not depend on the order* T *of the variables.*

Proof. Let $T = \{x_{j_1}, \dots, x_{j_n}\}$ and $T' = \{x_{k_1}, \dots, x_{k_n}\}$ be two distinct orderings of the variables. We denote by $\ell_j^q(T)$ and $u_j^q(T)$ ($\ell_j^q(T')$ and $u_j^q(T')$) the lower and upper bound for the variable x_j after q iterations of the multiple range reduction procedure when the ordering T (T') is employed. Note that lower and upper bounds are initialized independently of the ordering of the variables. It follows from the monotonicity condition that

$$[\ell_{j_1}^1(T'), u_{j_1}^1(T')] \subseteq [\ell_{j_1}^1(T), u_{j_1}^1(T)].$$

Next, a further application of the monotonicity condition leads to

$$[\ell_{j_2}^2(T'), u_{j_2}^2(T')] \subseteq [\ell_{j_2}^1(T), u_{j_2}^1(T)].$$

Iterating the above results we can prove that for any $q = 1, \dots, n$

$$[\ell_{j_q}^q(T'), u_{j_q}^q(T')] \subseteq [\ell_{j_q}^1(T), u_{j_q}^1(T)].$$

Therefore, after n iterations of the iterative procedure with the variable ordering T', the domain of each variable is at least as tight as its domain after one iteration with the variable ordering T, i.e.,

$$[\ell_j^n(T'), u_j^n(T')] \subseteq [\ell_j^1(T), u_j^1(T)], \; j = 1, \dots, n.$$

Next, we can iterate the proof to show that for each $\nu = 1, 2, \dots$, after at most νn iterations with the variable ordering T', the domain of each variable is at least as good as its domain after ν iterations with the variable ordering T. Symmetrically, we have that after at most νn iterations with the variable ordering T, the domain of each variable is at least as good as its domain after ν iterations with the variable ordering T'. This is only possible if

$$\ell_j^q(T), \ell_j^q(T') \to \bar{\ell}_j \quad \text{and} \quad u_j^q(T), u_j^q(T') \to \bar{u}_j \quad \text{as} \quad q \to \infty, \; j = 1, \dots, n,$$

and the above limits do not depend on the variable orderings. □

Some further remarks

In this section we present a few further remarks about RR strategies.

Remark 5.12. *It is possible to define a RR strategy where the upper limit u_j (respectively, the lower limit ℓ_j) of some variable x_j is updated by testing different possible increasing (respectively, decreasing) values for the lower (respectively, upper) limit of x_j. We only discuss the update of the upper limit (the update of the lower limit is completely analogous). Let $lb^t(S)$ be a lower bound for the problem* when the lower limit for x_j is fixed to

$$\ell_j^t = \ell_j + t\Delta, \quad t = 0, 1, \dots,$$

where $\Delta > 0$ is a prefixed step length. Let

$$\bar{t} = \arg\min\{t \ : \ lb^t(S) \geq ub_{\mathbf{0}}\},$$

i.e., $\ell_j^{\bar{t}}$ is the smallest value for ℓ_j^t such that the lower bound for the problem, when the lower limit for x_j is fixed to ℓ_j^t, is not lower than the current upper bound for the problem. Then, we can set the upper limit for x_j to $\ell_j^{\bar{t}}$. This technique, also known as probing, *is described, e.g., in (Belotti et al.,* 2009*; Faria & Bagajewicz,* 2011*; Tawarmalani & Sahinidis,* 2002b*).*

Remark 5.13. *For problems with a special structure, problem specific RRs can be defined. This has been done in different papers, such as (Caprara & Monaci,* 2009*) for bilinear problems employed to compute lower bounds for two-dimensional packing problems; (Locatelli & Raber,* 2002*) for a quadratic formulation of circle packing problems over the unit square; (Benson,* 2010*) for convex problems with a multiplicative constraint; and (Shen, Ma, & Chen,* 2011*) for the DM reformulation of problems with polynomial constraints and an objective function which is a sum of ratios of polynomial functions. In (Casado, Martinez, Garcia, & Sergeyev,* 2003*) Lipschitz optimization techniques are employed within the framework of interval methods and, in particular, results are discussed to reduce boxes by eliminating points which are guaranteed not to be global minimizers. Still within the same framework, in (Tóth & Casado,* 2007*) given a box X, gradient information is exploited to define a polytope P for which it can be guaranteed that it does not contain optimal solutions, and the box X is substituted by (a limited number of) subboxes X_i which cover $X \setminus P$.*

Remark 5.14. *In Section* 4.12 *we discussed problems with factorable functions and presented the directed acyclic graphs associated to such problems. In (Belotti et al.,* 2009*) a RR strategy based on forward and backward propagation along this graph has been presented. We illustrate this through an example also discussed in that paper. Consider the region*

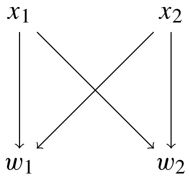
$$\begin{aligned} x_1 &= ax_2, \\ x_2 &= ax_1, \\ 0 &\leq x_1, x_2 \leq 1, \end{aligned}$$

where $a \in (0,1)$. The example is rather simple (it is even linear) but allows for a simple illustration of the forward and backward propagation along the directed acyclic graph.

By introducing the auxiliary variables w_1, w_2 *we can rewrite the system as*

$$\begin{aligned}&w_1 = ax_1,\\&w_2 = ax_2,\\&w_2 = x_1,\\&w_1 = x_2,\\&0 \le x_1, x_2 \le 1.\end{aligned}$$

The directed acyclic graph associated to this reformulation is the following.



Starting by nodes/variables with no ingoing arcs (the original variables x_1, x_2*) and moving forward along the graph, it is possible to deduce the bounds* $[0,a]$ *for* w_1, w_2*. Now, moving backward from the nodes/variables with no outgoing arcs (*w_1, w_2 *in the example), we can deduce the bounds* $[0,a]$ *for the variables* x_1, x_2*. The procedure can be iterated. By moving forward again, we obtain the bounds* $[0,a^2]$ *for* w_1, w_2*, which, by backward propagation allows us to deduce the bounds* $[0,a^2]$ *for* x_1, x_2*, and so on.*

Remark 5.15. *The RR strategies discussed above can be generalized into domain reduction strategies. For instance, given some linear function* $\boldsymbol{\alpha}^T\mathbf{x}$*, the best possible reduction for this function can be computed as in* (5.29)*, if we want to consider all feasible points, or as in* (5.30)*, if we only want to consider points with objective function value not larger than* ub_δ*. In both cases we only need to replace* x_k *in the objective function with the linear function* $\boldsymbol{\alpha}^T\mathbf{x}$*. We have previously seen that an important justification of RR strategies is that the reduction of the ranges allows us to define better convex underestimators and, thus, better lower bounds. A domain reduction involving a linear function* $\boldsymbol{\alpha}^T\mathbf{x}$ *might allow a tighter outer convex approximation of the feasible set S (if this is nonconvex), but in some cases it may also allow us to define better convex underestimators, similar to those seen for RR strategies. We illustrate this through the example below.*

Example 5.34. Assume that $x_1, x_2 \in [0,1]^2$. Then, a convex underestimator for the function $-x_1^2 - x_2^2$ is $-x_1 - x_2$. Let us assume that a domain reduction with the linear function $x_1 + x_2$ allows us to identify the lower and upper bound $\frac{1}{2}$ and $\frac{3}{2}$, while a domain reduction with the linear function $x_1 - x_2$ returns the lower and upper bound $-\frac{1}{2}$ and $\frac{1}{2}$. Therefore, we can conclude that for all the feasible points (or, at least, all the optimal solutions) the coordinates x_1 and x_2 lie in the two-dimensional polytope

$$\{2x_1 + 2x_2 \ge 1,\ 2x_1 + 2x_2 \le 3,\ 2x_1 - 2x_2 \le 1,\ -2x_1 + 2x_2 \le 1\} \subset [0,1]^2,$$

whose vertices are $(0\ \frac{1}{2})$, $(\frac{1}{2}\ 1)$, $(1\ \frac{1}{2})$, $(\frac{1}{2}\ 0)$. We can compute the convex envelope of the concave function over this polytope, which is equal to the affine function interpolating $-x_1^2 - x_2^2$ at the four vertices of the polytope, namely $-x_1 - x_2 + \frac{1}{4}$. Obviously, this convex underestimator improves the one over the unit square. ■

5.5.2 Domain reduction strategies

In this section we go through some feasibility or optimality based domain reduction techniques that have been proposed in the literature.

A possible way to reduce the domain is by *constraint propagation* (see, e.g., (Domes & Neumaier, 2010)). In particular, the following theorem is proven (Dallwig, Neumaier, & Schichl, 1997).

Theorem 5.35. *Let a constraint*

$$\sum_{k=1}^{t} p_k(\mathbf{x}) \geq c$$

and a box B be given. Let

$$\bar{p}_k \geq p_k(\mathbf{x}) \quad \forall\, \mathbf{x} \in B, \quad \bar{c} \geq \sum_{k=1}^{t} \bar{p}_k.$$

Then,

$$p_k(\mathbf{x}) \geq c - \bar{c} + \bar{p}_k.$$

Similarly, let a constraint

$$\sum_{k=1}^{t} p_k(\mathbf{x}) \leq c$$

be given, and let

$$\hat{p}_k \leq p_k(\mathbf{x}) \quad \forall\, \mathbf{x} \in B, \quad \hat{c} \leq \sum_{k=1}^{t} \hat{p}_k.$$

Then,

$$p_k(\mathbf{x}) \leq c - \hat{c} + \hat{p}_k.$$

Proof. The proof is rather simple. We give it just for the first part. For each $\mathbf{x} \in B$, we have that

$$p_k(\mathbf{x}) \geq c - \sum_{h=1;\, h \neq k}^{t} p_h(\mathbf{x}) \geq c - \sum_{h=1;\, h \neq k}^{t} \bar{p}_h \geq c - \bar{c} + \bar{p}_k. \qquad \square$$

Note that the use of this theorem is possible if a constraint can be decomposed into a sum of functions for which a lower (or upper) bound over the current box is easily available. We also note that among the constraints we can also include the constraint $f(\mathbf{x}) \leq ub_\delta$, which is obviously satisfied by all the δ-optimal solutions of the problem (5.1). The following example, taken from (Neumaier, 2004) illustrates some applications of the above theorem.

Example 5.36. Consider the problem

$$\begin{aligned} \min \quad & -x_1 - 2x_2 \\ & (x_1-1)^2 + (x_2-1)^2 = 1, \\ & -1 \leq x_1, x_2 \leq 1. \end{aligned}$$

Then, $B = [-1,1]^2$. Assume that the local minimizer $(0 \ \ 1)$, with objective function value equal to -2, has been already detected, so that the constraint

$$x_1 + 2x_2 \geq 2$$

can be added without cutting any optimal solution. Consider this additional constraint and set

$$p_1(x_1) = x_1, \quad p_2(x_2) = 2x_2,$$

so that we can take

$$\bar{p}_1 = 1, \quad \bar{p}_2 = 2, \quad \bar{c} = 3.$$

Then, from Theorem 5.35 it follows that

$$x_1 \geq 2 - 3 + 1 = 0, \quad 2x_2 \geq 2 - 3 + 2 = 1,$$

so that we can update the lower bounds for x_1 and x_2, respectively, into 0 and $\frac{1}{2}$. Now, in the feasible region of the problem the following constraint is obviously satisfied:

$$(x_1-1)^2 + (x_2-1)^2 \geq 1.$$

Let us set

$$p_1(x_1) = (x_1-1)^2, \quad p_2(x_2) = (x_2-1)^2,$$

so that we can take

$$\bar{p}_1 = 1, \quad \bar{p}_2 = \frac{1}{4}, \quad \bar{c} = \frac{5}{4}$$

over the two new intervals $x_1 \in [0,1]$ and $x_2 \in [\frac{1}{2},1]$. Then, from Theorem 5.35 it follows that

$$(1-x_1)^2 \geq 1 - \frac{5}{4} + 1 = \frac{3}{4},$$

from which we get to $x_1 \leq 1 - \frac{\sqrt{3}}{2}$, thus improving the upper bound for x_1. ∎

Interval analysis is another way to reduce the domain of a nonconvex problem. We use again the example taken from (Neumaier, 2004) to illustrate a possible way to use interval analysis.

Example 5.37. Recall the mean value theorem which states that for two given points $\mathbf{x}, \mathbf{y} \in X$, where X is a convex set, and a differentiable function f,

$$f(\mathbf{x}) = f(\mathbf{y}) + \nabla f(\lambda \mathbf{y} + (1-\lambda)\mathbf{x})(\mathbf{x} - \mathbf{y})$$

for some $\lambda \in [0,1]$. Therefore,

$$f(\mathbf{x}) \in f(\mathbf{y}) + \nabla f(X)(\mathbf{x} - \mathbf{y}), \tag{5.42}$$

where $\nabla f(X)$ is the box enclosing the values of the gradient of f over X. Consider the function

$$f(x_1, x_2) = (x_1 - 1)^2 + (x_2 - 1)^2.$$

Since

$$\nabla f(x_1, x_2) = (2x_1 - 2, 2x_2 - 2),$$

over the box $X = \left[0, 1 - \frac{\sqrt{3}}{2}\right] \times \left[\frac{1}{2}, 1\right]$ we can take

$$\nabla f(X) = [-2, -\sqrt{3}] \times [-1, 0].$$

Then, by taking $\mathbf{y} = (0 \;\; 1)$ (the already observed local minimizer) in (5.42) and taking into account the constraint $f(x_1, x_2) = 1$, we can impose

$$1 \in [1 - 2x_1, 2 - \sqrt{3}x_1 - x_2].$$

In particular, the upper limit of the above interval leads to the linear inequality

$$\sqrt{3}x_1 + x_2 \leq 1,$$

which is satisfied by all the feasible points of the original problem with function value not larger than -2. ■

In the field of concave minimization problems over polytopes, we have introduced in Section 5.3.3 the notion of γ-concavity cut. Such a cut is a linear inequality which certainly cuts some vertex $\mathbf{v}$ of the feasible polytope S away. It might even cut some global optimal solutions. This is the case if $\mathbf{v}$ is already a globally optimal solution (recall that concave minimization problems over polytopes attain at least a globally optimal solution at a vertex of the feasible polytope S; see Theorem 2.11). Now, noticing that a vertex is a face of dimension 0 for S, we would like to see whether it is possible to define other linear inequalities which remove higher dimensional faces of S. In particular, we are interested in inequalities which cut *extreme faces* of S with respect to a given polyhedron C away.

Definition 5.38. *Let H be a face of S. Then, H is called an* extreme face *of S with respect to the polyhedron C if $H \cap C \neq \emptyset$ and*

$$H \cap C \subset ri(H),$$

where ri denotes the relative interior.

Note that in the field of conical algorithms, C is chosen to be a polyhedral cone. Also note that, except for the case where H is a 0-dimensional face, i.e., a vertex, of S, the above definition states that no vertex of S lies in $H \cap C$, so that we can cut the extreme face without removing vertices of S also lying in C. This is important for all those problems for which we are sure that at least an optimal solution is attained at a vertex of S. Therefore, we are now interested in two problems: (i) establishing whether a face of S is also an extreme face; (ii) defining a linear inequality which cuts the extreme face away without cutting vertices of S lying in C. Assume that the feasible region S is a polytope represented in standard form, i.e.,

$$S = \{\mathbf{x} \in \mathbb{R}^n \ : \ \mathbf{A}\mathbf{x} = \mathbf{b}, \ \mathbf{x} \geq \mathbf{0}\},$$

with $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^m$. Then, a face H is defined as

$$\{\mathbf{x} \in \mathbb{R}^n \ : \ \mathbf{x} \in S, \ x_j = 0, \ j \in I_H\},$$

for some set $I_H \subset \{1,\dots,n\}$. The following result is proven (see, e.g., (Horst & Tuy, 1993)).

Theorem 5.39. *Let C be a polyhedron and H be a face of S such that $H \cap C \neq \emptyset$. H is an extreme face of S with respect to C if and only if the following linear programs for all $k \in \{1,\dots,n\} \setminus I_H$*

$$\begin{aligned} \min \quad & x_k \\ & \mathbf{x} \in S \cap C, \\ & x_i = 0 \qquad \forall\, i \in I_H, \end{aligned} \tag{5.43}$$

have a strictly positive optimal value.

Proof. Note that the feasible region of the linear programs is $H \cap C$. If H is an extreme face, then by definition $H \cap C$ lies in the relative interior of H. Therefore, any point $\mathbf{x} \in H \cap C$ has a strictly positive component x_k for any $k \in \{1,\dots,n\} \setminus I_H$. Indeed, if $x_k = 0$ in the relative interior of H, then $x_k = 0$ must be true over H, i.e., $k \in I_H$. Then, by compactness of the feasible region, the linear problem (5.43) has a strictly positive optimal value. The converse is also true. Indeed, $x_k > 0$ for all $\mathbf{x} \in H \cap C$ and $k \in \{1,\dots,n\} \setminus I_H$ implies that $H \cap C$ lies in the relative interior of H. □

Now, let us introduce the notion of facial cut.

Definition 5.40. *Let H be an extreme face of S with respect to a given polyhedron C, different from a vertex and from the whole region S. Then, a linear inequality $\mathbf{w}^T\mathbf{x} \geq w_0$ is a* facial cut *if*

$$\mathbf{w}^T\mathbf{y} < w_0 \quad \forall\, \mathbf{y} \in H \cap C,$$

while for any vertex $\mathbf{v}$ of S lying in C

$$\mathbf{w}^T\mathbf{v} \geq w_0.$$

In (Majthai & Whinston, 1974) a procedure has been suggested to derive a facial cut. Let $\alpha_i > 0$, $i \in I_H$, be some fixed coefficients. For each $k \notin I_H$ solve the following parametric linear problem with respect to the parameter q:

$$\begin{aligned} \xi_k(q) \ = \ \min \quad & x_k \\ & \mathbf{x} \in S \cap C, \\ & \textstyle\sum_{i \in I_H} \alpha_i x_i \leq q. \end{aligned}$$

By Theorem 5.39, $\xi_k(0) > 0$ for all $k \notin I_H$ if H is an extreme face. Let

$$q_k = \sup\{q \ : \ \xi_k(q) > 0\}. \tag{5.44}$$

Then, the following theorem can be proven.

Theorem 5.41. *Let H be an extreme face of S with respect to a given polyhedron C such that $0 < |I_H| < n - m$. Let $\beta = \min_{k \notin I_H} q_k > 0$. Then,*

$$\sum_{i \in I_H} \alpha_i x_i \geq \beta$$

is a facial cut.

Proof. Since $\beta > 0$ and $x_i = 0$ for all $i \in I_H$, the inequality is certainly violated by the points in H. Now, consider a point $\mathbf{x} \in C \cap S$ violating the inequality. By the definition (5.44) of q_k and the definition of β as the minimum of the q_k's, $x_k > 0$ for any $k \notin I_H$. Since, by assumption, $|I_H| < n - m$, then $\mathbf{x}$ cannot be a basic feasible solution (i.e., a vertex of S). □

Note that we may have $\beta = +\infty$. In this case we can conclude that $S \cap C$ does not contain any vertex of S.

An interesting way to produce cuts deeper than the γ-concavity ones has been proposed in (Konno, 1976) for concave quadratic functions and later generalized in (Horst & Tuy, 1993). As already mentioned, if a locally optimal vertex for a concave problem is available, then we can reformulate the problem in such a way that the vertex coincides with the origin and the feasible polytope

$$S = \{\mathbf{x} \in \mathbb{R}^n \ : \ \mathbf{A}\mathbf{x} \leq \mathbf{b}\}$$

is contained in the nonnegative orthant, i.e., we can impose $\mathbf{x} \geq \mathbf{0}$. Assume that $\gamma \leq f(\mathbf{0})$. If

$$f(s_i \mathbf{e}^i) \geq \gamma, \quad s_i > 0, \quad i = 1, \ldots, n,$$

then we can conclude that

$$\mathbf{x} \in S \quad \text{and} \quad \sum_{i=1}^{n} \frac{x_i}{s_i} \leq 1 \quad \Rightarrow \quad f(\mathbf{x}) \geq \gamma. \tag{5.45}$$

We call the inequality

$$\sum_{i=1}^{n} \frac{x_i}{s_i} \geq 1$$

a *γ-valid cut*. The deepest possible γ-valid cuts are γ-concavity cuts, where the values s_i are chosen so that the points $s_i \mathbf{e}^i$ are the γ-extensions (see Definition 5.14) along the rays $\mathbf{e}^i$. Assume that a function h is available with the following properties

$$h(\mathbf{x}, \mathbf{x}) = f(\mathbf{x}), \tag{5.46}$$

$$h(\mathbf{x}, \mathbf{y}) \geq \min\{f(\mathbf{x}), f(\mathbf{y})\} \quad \forall\, \mathbf{x}, \mathbf{y}, \tag{5.47}$$

$$h \ \text{ concave in } \mathbf{x} \text{ for fixed } \mathbf{y}, \text{ affine in } \mathbf{y} \text{ for fixed } \mathbf{x}. \tag{5.48}$$

Consider the two regions

$$D_1(\mathbf{s}) = \left\{ \mathbf{x} \in S \; : \; \sum_{i=1}^{n} \frac{x_i}{s_i} \leq 1 \right\}$$

and

$$D_2(\mathbf{s}) = \left\{ \mathbf{x} \in S \; : \; \sum_{i=1}^{n} \frac{x_i}{s_i} \geq 1 \right\}.$$

Define

$$g_{\mathbf{s}}(\mathbf{x}) = \inf_{\mathbf{y} \in D_2(\mathbf{s})} h(\mathbf{x},\mathbf{y}).$$

Being the infimum of concave functions, $g_{\mathbf{s}}$ is a concave function. Moreover, the following proposition is proven.

Proposition 5.42.

$$\min_{\mathbf{x} \in D_2(\mathbf{s})} g_{\mathbf{s}}(\mathbf{x}) = \min_{\mathbf{x} \in D_2(\mathbf{s})} f(\mathbf{x}).$$

Proof. For any $\mathbf{x},\mathbf{y} \in D_2(\mathbf{s})$,

$$h(\mathbf{x},\mathbf{y}) \geq \min\{f(\mathbf{x}), f(\mathbf{y})\} \geq \min_{\mathbf{x} \in D_2(\mathbf{s})} f(\mathbf{x})$$

follows from (5.47). Then,

$$g_{\mathbf{s}}(\mathbf{x}) \geq \min_{\mathbf{x} \in D_2(\mathbf{s})} f(\mathbf{x}).$$

Moreover,

$$g_{\mathbf{s}}(\mathbf{x}) = \inf_{\mathbf{y} \in D_2(\mathbf{s})} h(\mathbf{x},\mathbf{y}) \leq h(\mathbf{x},\mathbf{x}) = f(\mathbf{x}),$$

from which the result follows. □

Now we can prove the following theorem.

Theorem 5.43. *Assume that* $\mathbf{s}$ *defines a* γ*-valid cut*

$$\sum_{i=1}^{n} \frac{x_i}{s_i} \geq 1,$$

i.e., (5.45) *is true. Let* $t_i\mathbf{e}^i$*,* $i = 1,\ldots,n$*, be the* γ*-extension (see Definition* 5.14*) along the* i*th axis for the function* $g_{\mathbf{s}}$*. If*

$$g_{\mathbf{s}}(s_i\mathbf{e}^i) \geq \gamma, \quad i = 1,\ldots,n,$$

then $\mathbf{t} \geq \mathbf{s}$*, and*

$$\sum_{i=1}^{n} \frac{x_i}{t_i} \geq 1$$

is a γ*-valid cut at least as good as the one defined by* $\mathbf{s}$.

Proof. The fact that $\mathbf{t} \geq \mathbf{s}$ immediately follows from the definition of γ-extension. Then, $D_1(\mathbf{t}) \cap D_2(\mathbf{s})$ is contained in a polytope whose vertices are

$$\mathbf{v}^i = s_i \mathbf{e}^i, \quad \mathbf{v}^{n+i} = t_i \mathbf{e}^i, \quad i = 1, \ldots, n.$$

Since by assumption and by the definition of γ-extension,

$$g_{\mathbf{s}}(\mathbf{v}^j) \geq \gamma, \quad j = 1, \ldots, 2n,$$

then the concavity of $g_{\mathbf{s}}$ and (5.46) imply that

$$\gamma \leq g_{\mathbf{s}}(\mathbf{x}) \leq h(\mathbf{x}, \mathbf{x}) = f(\mathbf{x}) \quad \forall\, \mathbf{x} \in D_1(\mathbf{t}) \cap D_2(\mathbf{s}).$$

Combining the result above with (5.45), we can also state that

$$f(\mathbf{x}) \geq \gamma \quad \forall\, \mathbf{x} \in D_1(\mathbf{t}),$$

which implies the result. □

An immediate corollary is the following.

Corollary 5.44. *Let* $\mathbf{s}$ *define a* γ*-concavity cut, i.e., the points* $s_i \mathbf{e}^i$*,* $i = 1, \ldots, n$*, are the* γ*-extensions of* f *along the axes. Let* t_i *be defined as in Theorem* 5.43*, and let*

$$\mathbf{y}^i \in \arg\min\{h(s_i \mathbf{e}^i, \mathbf{y}) \;:\; \mathbf{y} \in D_2(\mathbf{s})\}. \tag{5.49}$$

If $f(\mathbf{y}^i) \geq \gamma$*, then*

$$\sum_{i=1}^{n} \frac{x_i}{t_i} \geq 1$$

is a γ*-valid cut.*

Proof. It is enough to observe that

$$g_{\mathbf{s}}(s_i \mathbf{e}^i) = h(s_i \mathbf{e}^i, \mathbf{y}^i) \geq \min\{f(s_i \mathbf{e}^i), f(\mathbf{y}^i)\} \geq \gamma. \quad \square$$

Remark 5.16. *If for some* i*,* $f(\mathbf{y}^i) < \gamma$*, then* $\mathbf{y}^i$ *is a point that improves with respect to* γ*, which is usually an upper bound for the optimal value of the problem. Then, we can detect, by a proper local search starting at* $\mathbf{y}^i$*, a new vertex that improves the current upper bound for the problem.*

Getting back to the original work (Konno, 1976), Konno observed that for concave quadratic functions $\frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{c}^T\mathbf{x}$, with $\mathbf{Q}$ negative semidefinite, the following bilinear function satisfies the conditions (5.46)–(5.48):

$$h(\mathbf{x}, \mathbf{y}) = \frac{1}{2}(\mathbf{c}^T\mathbf{x} + \mathbf{c}^T\mathbf{y} + \mathbf{x}^T\mathbf{Q}\mathbf{y}).$$

Conditions (5.46) and (5.48) are trivially satisfied. Condition (5.47) follows by observing that

$$(h(\mathbf{x},\mathbf{y})-h(\mathbf{x},\mathbf{x}))+(h(\mathbf{x},\mathbf{y})-h(\mathbf{y},\mathbf{y}))=-\frac{1}{2}(\mathbf{x}-\mathbf{y})^T\mathbf{Q}(\mathbf{x}-\mathbf{y})\geq 0,$$

where the last inequality follows from the fact that $\mathbf{Q}$ is negative semidefinite. Then,

$$(h(\mathbf{x},\mathbf{y})-f(\mathbf{x}))+(h(\mathbf{x},\mathbf{y})-f(\mathbf{y}))\geq 0,$$

so that

$$h(\mathbf{x},\mathbf{y})\geq \min\{f(\mathbf{x}),f(\mathbf{y})\}.$$

Note that, in order to detect the points $\mathbf{y}^i$ defined in (5.49), one need only solve the linear program

$$\begin{array}{ll} \min & h(s_i\mathbf{e}^i,\mathbf{y}) \\ & \sum_{i=1}^n \frac{y_i}{s_i}\geq 1, \\ & \mathbf{y}\in S. \end{array} \tag{5.50}$$

With respect to the computation of the γ-extensions for the function $g_\mathbf{s}$ we have the following result.

Theorem 5.45. *The value t_i is the optimal value of the linear problem*

$$\begin{array}{ll} \min & \frac{1}{2}\mathbf{c}^T\mathbf{z}-\gamma z_0 \\ & -\mathbf{A}\mathbf{z}+z_0\mathbf{b}\geq \mathbf{0}, \\ & \sum_{j=1}^n \frac{z_j}{s_j}-z_0\geq 0, \\ & -\frac{1}{2}\sum_{j=1}^n Q_{ij}z_j-\frac{1}{2}c_iz_0=1, \\ & \mathbf{z},z_0\geq \mathbf{0}. \end{array}$$

Proof. Similarly to (5.50), we see that for each t

$$\begin{array}{ll} g_\mathbf{s}(t\mathbf{e}^i)=\frac{1}{2}tc_i+\min & \frac{1}{2}\mathbf{c}^T\mathbf{y}+\frac{1}{2}t\sum_{j=1}^n Q_{ij}y_j \\ & -\mathbf{A}\mathbf{y}\geq -\mathbf{b}, \\ & \sum_{j=1}^n \frac{y_j}{s_j}\geq 1, \\ & \mathbf{y}\geq \mathbf{0}. \end{array}$$

By duality, we also have that

$$\begin{array}{ll} g_\mathbf{s}(t\mathbf{e}^i)=\frac{1}{2}tc_i+\max & -\mathbf{b}^T\mathbf{w}+w_0 \\ & -\mathbf{A}^T\mathbf{w}+w_0Diag(1/s_1,\ldots,1/s_n)\mathbf{e}\leq \frac{1}{2}\mathbf{c}+\frac{1}{2}t[\mathbf{Q}]_i^T, \\ & \mathbf{w}\geq \mathbf{0},w_0\geq 0, \end{array}$$

where $[\mathbf{Q}]_i$ is the ith row of $\mathbf{Q}$. Then, t_i is defined as follows:

$$\begin{aligned} t_i = \max \quad & t \\ & \tfrac{1}{2}tc_i - \mathbf{b}^T\mathbf{w} + w_0 \geq \gamma, \\ & -\mathbf{A}^T\mathbf{w} + w_0 Diag(1/s_1,\ldots,1/s_n)\mathbf{e} - \tfrac{1}{2}t[\mathbf{Q}]_i^T \leq \tfrac{1}{2}\mathbf{c}, \\ & \mathbf{w} \geq \mathbf{0}, w_0 \geq 0. \end{aligned}$$

Then, the result follows by taking the dual of the above linear problem. □

Now, let us take a closer look at γ-concavity cuts. As we have seen, the requirements to generate a γ-concavity cut for some target value γ are

(a) a vertex $\mathbf{v}$ of the feasible polytope S such that $f(\mathbf{v}) \geq \gamma$;

(b) a polyhedral cone $C(\mathbf{v})$ with vertex $\mathbf{v}$ and containing S.

In fact, both these requirements can be relaxed. We first relax requirement (a): we do not need to consider only vertices of S, and we do not even need to consider points in S. As an instance, we might consider the cut proposed by Porembski in (Porembski, 2001). Assume that the concavity cut $\boldsymbol{\pi}^T(\mathbf{x}-\mathbf{v})$ has been computed, where $\mathbf{v}$ is some vertex of the feasible region S. Denote by $C(\mathbf{v})$ the smallest cone containing S and with vertex $\mathbf{v}$. Its extreme rays are the vectors $\mathbf{v}_i - \mathbf{v}$, for each $\mathbf{v}_i \in adj(\mathbf{v})$. Consider the point $\mathbf{w} = \mathbf{v} - \bar{\beta}\boldsymbol{\pi}$, where

$$\bar{\beta} \in \arg\max\{\beta \ : \ f(\mathbf{v} - \beta\boldsymbol{\pi}) \geq \gamma\}.$$

Then, consider $C(\mathbf{w})$, the smallest cone containing S and with vertex $\mathbf{w}$. Its extreme rays are $\mathbf{w}_i - \mathbf{w}$, $i = 1,\ldots,s$, $(s \geq n)$, where $\mathbf{w}_i$ are vertices of S, and a procedure to compute them is described in (Porembski, 2001). Assume that $f(\mathbf{w}_i) \geq \gamma$, $i = 1,\ldots,s$ (otherwise one of these vertices would improve the current target value). Then, a cut is derived as follows: (i) take the hyperplane $\boldsymbol{\pi}(\mathbf{x}-\mathbf{v}) = \xi$; (ii) let $\mathbf{z}_i(\xi)$, $i = 1,\ldots,s$, be the intersection of the hyperplane with the half-line whose origin is the point $\mathbf{w}$ and whose direction is the ith extreme ray of $C(\mathbf{w})$; (iii) let

$$\bar{\xi} = \arg\max\{\xi \ : \ f(\mathbf{z}_i(\xi)) \geq \gamma, \ \ i = 1,\ldots,s\}.$$

Then, Porembski (Porembski, 2001) proves that $\boldsymbol{\pi}(\mathbf{x}-\mathbf{v}) \geq \bar{\xi}$ is a valid cut and that $\bar{\xi} \geq 1$, i.e., this cut is at least as deep as the concavity cut. We illustrate this through an example.

Example 5.46. Consider the following problem:

$$\begin{aligned} \min \quad & -x_1^2 - x_2^2 \\ & 2x_1 - x_2 \leq 2, \\ & -x_1 + 2x_2 \leq 2, \\ & x_1, x_2 \geq 0, \end{aligned}$$

where the feasible region is the polytope whose vertices are (0 0), (1 0), (0 1), (2 2). Let $\gamma = -8$ be the target value (equal to the optimal value of the problem). If we take $\mathbf{v} = (0\ 0)$, then the smallest cone with vertex $\mathbf{v}$ and containing S is the nonnegative orthant, with extreme rays the directions of the two axes. The γ-extensions are $(2\sqrt{2}\ 0)$ and $(0\ 2\sqrt{2})$. Therefore, the γ-concavity cut is

$$x_1 + x_2 \geq 2\sqrt{2},$$

and $\boldsymbol{\pi} = (1\ 1)$. Then, let us consider the point $\mathbf{w} = \mathbf{v} - \bar{\beta}\boldsymbol{\pi}$, where

$$\bar{\beta} \in \arg\max\{\beta\ :\ -2\beta^2 \geq -8\},$$

i.e., $\bar{\beta} = 2$ and $\mathbf{w} = (-2\ -2)$. The smallest cone with vertex $\mathbf{w}$ and containing S has the extreme rays

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} - \begin{pmatrix} -2 \\ -2 \end{pmatrix} = \begin{pmatrix} 3 \\ 2 \end{pmatrix}, \quad \begin{pmatrix} 0 \\ 1 \end{pmatrix} - \begin{pmatrix} -2 \\ -2 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}.$$

The intersections of $x_1 + x_2 = \xi$ with the half-lines whose origin is the point $\mathbf{w}$ and whose directions are the extreme rays are

$$\mathbf{z}_1(\xi) = \left(\frac{2}{5}\xi - \frac{2}{5}\ \ \frac{3}{5}\xi + \frac{2}{5}\right), \quad \mathbf{z}_2(\xi) = \left(\frac{3}{5}\xi + \frac{2}{5}\ \ \frac{2}{5}\xi - \frac{2}{5}\right),$$

so that $\bar{\xi} = \frac{48}{13} > 2\sqrt{2}$, i.e., the inequality

$$x_1 + x_2 \geq \frac{48}{13}$$

is a valid cut, deeper than the concavity cut. ∎

Porembski also proves the finiteness of a pure cutting plane algorithm (i.e., with no branching operations) based on these deeper cuts.

Next, we move to a relaxation of requirement (b). What we need is not necessarily a cone C containing S, but rather any (usually unbounded) polyhedron S' such that (i) S' contains S; (ii) its vertices are known and such that the value of f at them is not lower than the target value γ; (iii) its extreme rays are known; (iv) we can easily compute the γ-extensions along the infinite edges of S'. We present an example below, while we refer to (Porembski, 2001) for a technique to derive valid cuts, deeper than the concavity ones, based on this observation.

Example 5.47. Let us consider the instance of Example 5.46. Let S' be the polyhedron whose vertices are $(0\ 0)$ and $(0\ 1)$, at which the objective function value is not lower than $\gamma = -8$, while its extreme rays are the direction $(1\ 0)$ of the x_1-axis and the vector $(2\ 1)$ (the direction of the half-line along which the edge joining $(0\ 1)$ and $(2\ 2)$ lies). The γ-extensions computed along the infinite edges of S' are the points $(2\sqrt{2}\ 0)$ and $(2\ 2)$. Then, if we consider the half-space generated by the line through these two γ-extensions, we end up with the valid cut

$$\frac{\sqrt{2}}{4}x_1 + \frac{2-\sqrt{2}}{4}x_2 \geq 1$$

(which, in this case, removes the whole feasible region S, except the optimal vertex $(2\,2)$). ∎

With respect to quadratic problems, we recall here the cuts already discussed in Section 4.4.2 for BoxQP problems, which are based on the one row relaxation, where the KKT conditions related to a single index i are required to be satisfied. For problems with quadratic objective and constraint functions, cuts, which remove local minimizers satisfying some regularity assumption and require the computation of the Lagrange multipliers corresponding to such local minimizers, have been proposed in (Nowak, 2000). In the context of QP problems, in (Hu et al., 2012) optimality based cuts, derived from second-order necessary optimality conditions, are introduced within the mixed integer reformulation of the linear problem with complementarity constraints (4.79)–(4.84) equivalent to the QP problem (see also Remark 4.5).

5.6 Fathoming rules

Aside from the standard fathoming rule (5.2), nodes can also be fathomed through other rules. In particular, an obvious but relevant observation is that any globally optimal solution is also a locally optimal solution and, consequently, has to satisfy local optimality conditions. In Section 4.4.2 we have seen that this fact is already taken into account in the relaxation of QP problems, where the KKT conditions are added to the constraints of the problem. This fact can also be taken into account in the fathoming step.

Methods based on interval analysis may include a *monotonicity test* (see, e.g., (Kearfott, 1996; Neumaier, 2004; Ratschek & Rokne, 1995)). Assume that the objective function f of our problem is differentiable and that the feasible region is a box, i.e., $S = \prod_{i=1}^n [\ell_i, u_i]$. Assume also that we are considering a node/subbox $S_k = \prod_{i=1}^n [\ell_i^k, u_i^k]$. For each $i = 1, \ldots, n$, let F_i be an inclusion function for the partial derivative $\frac{\partial f}{\partial x_i}$, i.e.,

$$F^i([a_i, b_i]) \supseteq \left\{ \frac{\partial f}{\partial x_i}(\mathbf{x}) \ : \ \mathbf{x} \in S_k \right\}.$$

Then,

- If for some i, $\ell_i < \ell_i^k$, $u_i^k < u_i$ (i.e., the interval in S_k related to the variable x_i is in the interior of the initial interval for this variable), and $0 \not\in F^i([\ell_i^k, u_i^k])$, then node S_k can be fathomed. Indeed, in S_k all local optimizers of f should have null partial derivative with respect to x_i.

- If for some i, $\ell_i = \ell_i^k$, $u_i^k < u_i$, and $F_u^i([\ell_i^k, u_i^k]) < 0$, then node S_k can be fathomed. Indeed, in S_k all local optimizers of f should have null partial derivative with respect to x_i if they belong to the interior of the interval $[\ell_i, u_i]$, or nonnegative partial derivative with respect to x_i if they lie at the left extreme ℓ_i of the same interval.

- If for some i, $\ell_i < \ell_i^k$, $u_i^k = u_i$, and $F_\ell^i([\ell_i^k, u_i^k]) > 0$, then node S_k can be fathomed. Indeed, in S_k all local optimizers of f should have null partial derivative with respect to x_i if they belong to the interior of the interval $[\ell_i, u_i]$, or nonpositive partial derivative with respect to x_i if they lie at the right extreme u_i of the same interval.

Note that the above monotonicity test can also be employed for some variable fixing. For instance, if for some i, $\ell_i = \ell_i^k$, $u_i^k < u_i$, and $F_\ell^i([\ell_i^k, u_i^k]) > 0$, then we can fix the value of

x_i in node S_k to ℓ_i. Indeed, under this condition all local optimizers of f in S_k can only lie at the left extreme ℓ_i of $[\ell_i, u_i^k]$. If f is twice-continuously differentiable, then we can also consider interval analysis of the Hessian. For example, if such analysis establishes that the Hessian is nowhere positive semidefinite over a given subbox that lies strictly in the interior of the initial box, then we can fathom the node/subbox (the second-order necessary conditions can not be satisfied within the box). In the opposite case, if the analysis establishes that the Hessian is everywhere positive semidefinite over the subbox, then f is convex over the subbox, and after having found a local (in fact, global) optimizer of f over the subbox, we can fathom it.

Now assume that the feasible region S is a polytope, i.e., given $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$,

$$S = \{\mathbf{x} \ : \ \mathbf{A}\mathbf{x} = \mathbf{b}, \ \ \ell_i \le x_i \le u_i, \ i = 1, \ldots, n\}.$$

Following (Al-Khayyal & Sherali, 2000) assume that all data $\mathbf{A}, \mathbf{b}, \ell_i, u_i$, $i = 1, \ldots, n$, are integral. For some problems it is possible to guarantee that at least one globally optimal solution lies at a vertex of the feasible region. As we already know (see Theorem 2.11), this is true, e.g., when the objective function is quasi-concave. This is also true when the problem is a disjoint bilinear one (bilinear objective function $\mathbf{x}\mathbf{Q}\mathbf{y} + \mathbf{c}^T\mathbf{x} + \mathbf{d}^T\mathbf{y}$ with disjoint linear constraints for $\mathbf{x}$ and $\mathbf{y}$). In these cases we might introduce a fathoming rule which discards some subset S_k if we can guarantee that it contains no vertex of S, or if it contains a single vertex of S at which f has already been evaluated. Assume that S_k lies within a subbox $\prod_{i=1}^n [\ell_i^k, u_i^k]$ and as in (Al-Khayyal & Sherali, 2000) define

$$\mathcal{L}_\ell(S_k) = \{i \ : \ \ell_i^k = \ell_i\}, \quad \mathcal{L}_u(S_k) = \{i \ : \ u_i^k = u_i\}, \quad \mathcal{L}_0(S_k) = \{i \ : \ u_i^k = \ell_i^k\}.$$

Then the following fathoming rules can be introduced:

- If $|(\mathcal{L}_\ell(S_k) \cup \mathcal{L}_u(S_k))| < n - m$, then the number of nonbasic variables (at their lower or upper bound) is lower than $n - m$. That is, the subset can not contain any basic feasible solution (i.e., any vertex) of S and it can be fathomed.

- If $|\mathcal{L}_\ell(S_k) \cup \mathcal{L}_u(S_k)| = n - m$ and $\mathcal{L}_\ell(S_k) \cap \mathcal{L}_u(S_k) = \emptyset$, then the only possible basic solution is obtained by fixing to their lower bound the variables whose index is in $\mathcal{L}_\ell(S_k)$ and to their upper bound the variables whose index is in $\mathcal{L}_u(S_k)$. If this basic solution is also feasible, i.e., it is a vertex of S, then we compute f at such vertex, possibly update $ub_\mathbf{0}$ and, finally, fathom subset S_k.

- If $|\mathcal{L}_0(S_k)| \ge n - m$, then we check whether by fixing the value of the variables in $\mathcal{L}_0(S_k)$ we get a basic feasible solution. If yes, we compute f at the corresponding vertex, possibly update the upper bound $ub_\mathbf{0}$, and fathom the node. If $\mathcal{L}_0(S_k) = \mathcal{L}_\ell(S_k) \cup \mathcal{L}_u(S_k)$ we can fathom the node in any case (the node either contains a vertex, detected by fixing the value of the variables in $\mathcal{L}_0(S_k)$ as previously discussed, or contains no vertex).

Again, some variable fixing is possible. Integrality of the data allows us to establish that if $\mathbf{v}^r, \mathbf{v}^s$ are two distinct vertices of S, then it is well known (see, e.g., (Nemhauser & Wolsey, 1988)) that for each i, $i = 1, \ldots, n$,

$$\text{either} \quad v_i^r = v_i^s \quad \text{or} \quad v_i^r - v_i^s \ge \frac{1}{|det_{\max}|}, \tag{5.51}$$

where $det_{\max}$ is the largest (in absolute value) determinant of a basis of $\mathbf{A}$. Then, let L be a large enough integer such that $2^L > |det_{\max}|$. We make the (reasonable) assumption that for each lower bound ℓ_i, $i = 1,\ldots,n$, there exists a vertex $\mathbf{v}$ of S such that $v_i = \ell_i$. We require the same for the upper bound values. Then,

- if $i \in \mathcal{L}_\ell(S_k)$ and $u_i^k - \ell_i \leq 2^{-L}$, then we can fix x_i and u_i^k to ℓ_i in S_k (indeed, according to (5.51) there can not be a basic feasible solution in S_k with $x_i > \ell_i$);
- similarly, if $i \in \mathcal{L}_u(S_k)$ and $u_i - \ell_i^k \leq 2^{-L}$, then we can fix x_i and ℓ_i^k to u_i in S_k.

5.7 Some finiteness results for the case $\varepsilon = 0$

In the previous sections it has been shown that, under suitable conditions, the BB algorithm terminates after a finite number of iterations if $\varepsilon > 0$, $\boldsymbol{\delta} > \mathbf{0}$, and converges (or establishes that the feasible region is empty) when $\varepsilon = 0$, $\boldsymbol{\delta} = \mathbf{0}$. We have also pointed out that if all the constraint functions g_i, $i = 1,\ldots,m$, are convex ones, then the same results can also be attained for $\boldsymbol{\delta} = \mathbf{0}$. Now, we might wonder whether finiteness can be guaranteed also when $\varepsilon = 0$. In general, the answer is no. Hamami and Jacobsen (Hamami & Jacobsen, 1988), as well as Horst and Tuy (Horst & Tuy, 1993), show that a BB algorithm for concave minimization over polytopes based on bisections may not terminate in a finite number of iterations if $\varepsilon = 0$ (see also Remark 5.11). However, finiteness can be attained if the structure of the problem reveals that a finite subset $S^\star$ certainly contains at least an optimal solution. Then, we can introduce in the BB algorithm some tools which take into account this fact and, in the worst case, enumerate all the elements in $S^\star$ before stopping, after a finite number of iterations, the algorithm. These tools can be of different nature (such as fathoming rules, branching operations, and domain reductions).

Obviously any globally optimal solution is also a local optimal one. This simple observation can be exploited to make a BB algorithm finite. As seen in Section 5.6, for box constrained problems and differentiable objective function, interval analysis applied to the gradient allows us to fathom nodes/subboxes which certainly do not contain any local minimizer, and if the function is twice-continuously differentiable, interval analysis of the Hessian allows us to identify whether the function is convex over a subbox or not. If it is, we can identify a local (and global) minimizer over the subbox and then fathom the node/subbox. Therefore, under suitable conditions (for instance, we may require that the local minimizers be in a finite number, they lie in the interior of the initial box, they all satisfy second-order sufficient conditions of local optimality) the above interval analysis tools allow us to make the BB algorithm finite. In this case finiteness is guaranteed by the fathoming rules.

A similar observation has been exploited in the approaches discussed in Section 4.4.2 ((Audet et al., 1999; Burer & Vandenbussche, 2008, 2009; Hansen et al., 1993; Vandenbussche & Nemhauser, 2005b, 2005a)): global optimal solutions for QP problems with box or linear constraints must also satisfy the KKT conditions. Branching operations enforce some KKT conditions, and this finally leads to a finite algorithm, as already discussed in Section 4.4.2 (note that in all these approaches the properties of exhaustiveness, exactness in the limit, and bound improvement are not required).

For some problems, whose feasible region is a polytope, it is guaranteed that at least one global optimal solution lies at a vertex of the polytope. As seen in Section 5.6, in (Al-Khayyal & Sherali, 2000) fathoming rules are proposed which are able to identify whether some subbox contains no vertex or a single vertex of the feasible region. In both cases the node/subbox can be fathomed (in the latter case after identification of the unique vertex). Such fathoming rules lead to a finite algorithm.

For concave minimization problems over polytopes different strategies have been developed. In (Benson & Sayin, 1994) a neighbor generation mechanism has been added to a simplicial algorithm to ensure finiteness. Different subdivision rules have been suggested (see (Nast, 1996; Tam & Ban, 1985)). While the approach suggested in (Nast, 1996) has the drawback that a simplex may be subdivided into a large number of subsimplices, in (Tam & Ban, 1985) (see also (Horst & Tuy, 1993)) each simplex is subdivided into two subsimplices through a subdivision similar to a bisection. The subdivision is performed with respect to a point in the interior of an edge of the simplex, with the property that a linear constraint (strictly satisfied and strictly violated, respectively, at the extremes of the edge) is active at that point. Appropriate subdivision rules are also the key for proving the finiteness of the conical approach proposed in (Hamami & Jacobsen, 1988), of the rectangular BB algorithm for minimizing separable concave functions over polytopes proposed in (Schectman & Sahinidis, 1998), and of the rectangular BB algorithm for linear multiplicative programming (minimization of a product of affine functions over a polytope) proposed in (Kuno, 2001).

Introduction of cutting planes is another tool to ensure finiteness of BB algorithms for concave minimization. The basic idea is to introduce cuts, such as the γ-concavity ones, in order to cut all the globally optimal vertices of the problem. However, some care is needed in doing that. When adding a cut, we might also create new vertices with respect to those of the original polytope, namely the intersection of the cut with edges of the polytope. These new vertices might also be globally optimal solutions, so that we cut a vertex but we add new ones, without being able to guarantee that at some point all globally optimal solutions will be discarded. Such a situation does not occur, e.g., if f is a strictly concave function. However, if f is not strictly concave, what we can do is modify the function f so that its globally optimal solutions are only its globally optimal vertices. In (Locatelli & Thoai, 2000) the following modification has been proposed. Assume that the feasible region is the polytope

$$S = \{\mathbf{x} \in \mathbb{R}^n \ : \ \mathbf{a}_j^T \mathbf{x} \leq b_j, \ j = 1, \ldots, m\}.$$

Let

$$g(\mathbf{x}) = \min_{J \subset \{1,\ldots,m\} \ : \ |J|=n, \ (\mathbf{a}_j, \ j \in J, \text{ independent})} \sum_{j \in J} (b_j - \mathbf{a}_j^T \mathbf{x}).$$

The function g is concave and can be computed, for a fixed $\mathbf{x}$, in polynomial time through a greedy algorithm (see, e.g., (Papadimitriou & Steiglitz, 1998)). Moreover, g is equal to 0 at all vertices of S, and strictly positive at all other points in S. Therefore, the globally optimal vertices of $f + g$ are the same as those of f, but $f + g$ has no further globally optimal solutions. This way, when we add cuts, we will not create new globally optimal vertices for the function $f + g$.

Chapter 6

Problems

This chapter is dedicated to a discussion about GO test problems and applications. Neither subject will be discussed in an exhaustive way. With respect to test problems, we have restricted our attention to a class of GO test problems, and discussed through it the issue of carefully selecting a subset of functions which is meaningful with respect to the kind of approach which has to be tested. Other GO test problems are not discussed in detail, but references are given to the existing literature and for some class of test problems the challenging aspects of the class are briefly discussed. With respect to the applications, listing all of them is quite hard due to the large number of publications in this field. Driven by our personal experience, we restrict our attention to four applications (namely molecular conformation, distance geometry, packing, and space trajectory planning problems) and give some pointers to a few others.

6.1 Test problems for GO heuristics

Validating a global optimization algorithm is not an easy task, as many are the performance indices which might be interesting to control. Depending on the algorithm used and on the available information, it might be important to compare methods on the basis of the number of function evaluations, of gradient evaluations, or of calls to a local optimization method. Of course CPU time is another relevant quantity to be taken into account, although the differences in performance of different hardware and of the same hardware with different software (such as different compilers) make a sound comparison very difficult to perform.

In this section we chose not to give any comparison of performance of the algorithms presented. Instead we chose to collect a few test problems of moderately small dimension, for which we performed an extensive set of tests with the most basic method, Multistart. The statistics we report here should be considered as lower bounds on the performance required by any implementation of any heuristic technique which uses gradient-based local searches. All problems are box-constrained and most of them are smooth. We chose to solve them using the AMPL modeling language (Fourer, Gay, & Kernighan, 2002) and the LBFGS-B 3.0 solver (Morales & Nocedal, 2011) with default parameters. We point out that the use of different local solvers might also lead to (usually slightly) different results, but we do not discuss this issue here. Unless otherwise specified, we ran 100 000 local optimizations starting from uniform randomly generated points in the feasible box of every problem.

Tables 6.1 and 6.2 report a set of results collected on a selection of test problems taken from Ismael Vaz.[8] In particular, Table 6.1 reports: in the first column the problem name; in the second column its dimension; in the third column the average number of function evaluations (per local search) performed which, for this algorithm, coincides with the number of gradient evaluations as well; and in the fourth column the best optimum found in 100 000 local searches. When a putative optimum whose value is lower than that found in these experiments is known, it is reported in parenthesis.

Table 6.1. *General statistics on a set of test functions*

Problem name	Dim	Fun/Grad evaluations	Record found
ack	10	17.798	0.947697*(0.)*
ap	2	15.253	−0.352386
bf1	2	23.233	0
bf2	2	22.352	0
bhs	2	16.742	−3.42849
bl	2	4	0
bp	2	12.033	0.397887
cb3	2	13.37	0
cb6	2	19.243	−1.03163
cm2	2	14.707	−0.2
cm4	4	18.839	−0.4
da	2	22.427	−24776.5
em-10	10	100.61	−9.17124*(-9.66015)*
em-5	5	46.515	−4.49589*(-4.68766)*
ep	2	4.6642	−1
exp	10	9.3677	−1
fls	2	11.591	−2.02181
fr	2	20.578	0
fx-10	10	59.532	−10.2088
fx-5	5	39.099	−10.4056
gp	2	28.189	3
grp	3	7.274	0
gw	10	57.588	0
h3	3	18.717	−3.86278
h6	6	27.923	−3.32237
hm	2	22.721	0
hm1	1	10.392	0
hm2	1	9.2013	−4.81447

(cont.)

[8] These are available at `http://www.norg.uminho.pt/aivaz/pswarm/software/prob_linear.zip` and at the web site devoted to this volume: `http://www.siam.org/books/mo15`.

Problem name	Dim	Fun/Grad evaluations	Record found
hm3	1	11.42	−2.26754
hm4	2	12.599	0
hm5	3	4	0
hsk	2	17.172	−2.34581
hv	3	21.629	0
ir2	2	36.112	0
ir5	2	1.7288	0.00199601
kl	4	32.506	0.000307486
ks	1	2.2519	0
lm1	3	21.975	0
lm2-10	10	144.86	0
lm2-5	5	40.367	0
lv8	3	27.322	0
mc	2	14.6	−1.91322
mcp	4	45.589	0
mgp	2	19.252	−1.29695
mgw-10	10	43.941	0
mgw-2	2	11.251	0
mgw-20	20	64.928	0
ml-10	10	1.0002	−0.517*(-0.965)*
ml-5	5	1.1927	−0.965
mr	3	53	0.00190015
mrp	2	22.936	0
ms1	20	44.966	5.23265
ms2	20	55.893	11.7464
nf2	4	714.8	0
nf3-10	10	32.479	−210
nf3-15	15	51.811	−665
nf3-20	20	74.036	−1520
nf3-25	25	98.13	−2900
nf3-30	30	120.98	−4930
pp	10	35.018	−45.7785
prd	2	9.8104	0.9
pwq	4	57.919	0
rb	10	114.08	0
rg-10	10	21.497	4.9748*(0.)*
rg-2	2	16.252	0
s10	4	35.824	−10.5364
s5	4	34.34	−10.1532
s7	4	36.391	−10.4029
sal-10	10	11.238	0
sal-5	5	11.253	0
sbt	2	20.369	−186.731

(cont.)

Problem name	Dim	Fun/Grad evaluations	Record found
sf1	2	9.7071	0
sf2	2	12.034	0.00536448*(0.)*
shv1	1	8.1683	−1
shv2	2	13.78	0
sin-10	10	35.283	−3.5
sin-20	20	52.596	−3.5
stg	1	21.812	0
swf	10	17.171	−3952.95*(-4189.83)*
sz	1	11.79	−12.0312
szzs	1	11.227	−1.60131
wf	4	63.529	0
xor	9	32.596	0.867827
zkv-10	10	34.932	0
zkv-2	2	13.795	0
zkv-20	20	45.394	0
zkv-5	5	25.263	0
zlk1	1	10.803	−1.90596
zlk2a	1	8.6878	−1.125
zlk2b	1	13.342	−1.125
zlk3a	1	7.0469	−1
zlk3b	1	7.1656	−1
zlk3c	1	6.7973	−1
zlk4	2	12.039	0.397888
zlk5	3	18.741	−3.86278
zzs	1	10.938	−0.824239

Table 6.2 reports the computational effort required, on average, to detect the global minimum, if Multistart was able to find it. The table reports the problem name (first column), the percentage of successes in detecting the putative global optimum (second column), the average number of local searches per success (third column), and the average number of function evaluations required to detect the putative global optimum (fourth column). When the number of successes is 0, we report an F (failure) in the third and fourth columns. These quantities depend on the precision used to state that the optimum has been observed. We used the following method: let $\varepsilon > 0$ be a chosen precision level, and denote by $f^\star$ the value of the global minimum known from the literature and by $\hat{f}$ the final value found in a local search. Then a local optimization is labeled as a success if

$$\hat{f} \leq f^\star + \varepsilon \max\{1, |f^\star|\}$$

(we set $\varepsilon = 10^{-2}$ in our experiments). This way, we combine a relative error measure with an absolute one, which comes into play for problems with a global minimum value close or equal to zero. The column "Local searches per success" reports the ratio between the total number of local searches performed and the total number of times the global optimum was found; the meaning of the following column is analogous.

Table 6.2. *Success statistics on the test functions, $\varepsilon = 10^{-2}$*

Problem name	Successes %	Local Src. per success	Fun/Grad evals per success
ack	0	*F*	*F*
ap	51.591	1.9383	29.571
bf1	10.601	9.4331	246.51
bf2	16.555	6.0405	149.18
bhs	61.175	1.6347	28.05
bl	100.0	1.0	4.0
bp	99.974	1.0003	12.036
cb3	82.183	1.2168	16.413
cb6	83.954	1.1911	23.354
cm2	33.137	3.0178	44.514
cm4	7.436	13.448	254.09
da	93.941	1.0645	24.011
em-10	0	*F*	*F*
em-5	0	*F*	*F*
ep	6.66	15.004	70.04
exp	100.0	1.0	9.3677
fls	75.847	1.3184	15.38
fr	49.967	2.0013	42.117
fx-10	0.074	1351.4	81285
fx-5	1.051	95.147	3762.5
gp	55.567	1.7996	54.332
grp	60.513	1.6525	12.027
gw	15.994	6.2523	360.28
h3	61.325	1.6307	30.638
h6	68.660	1.4565	40.725
hm	83.603	1.1961	28.416
hm1	100.0	1.0	10.392
hm2	63.405	1.5772	14.516
hm3	35.641	2.8058	32.419
hm4	67.441	1.4828	18.811
hm5	100.0	1.0	4.0
hsk	64.914	1.5405	27.116
hv	100.0	1.0	21.75
ir2	99.953	1.0005	36.129
ir5	100.0	1.0	1.7289
kl	100.0	1.0	32.538
ks	100.0	1.0	2.2519
lm1	31.816	3.1431	69.446
lm2-10	4.117	24.29	3519.0
lm2-5	4.298	23.267	944.17
lv8	65.898	1.5175	41.588

(cont.)

Problem name	Successes %	Local Src. per success	Fun/Grad evals per success
mc	67.157	1.489	22.139
mcp	89.035	1.1232	51.209
mgp	3.512	28.474	551.76
mgw-10	31.866	3.1381	140.39
mgw-2	30.909	3.2353	37.964
mgw-20	21.778	4.5918	304.57
ml-10	0	*F*	*F*
ml-5	0.29	344.83	411.28
mr	100.0	1.0	53.29
mrp	99.977	1.0002	23.087
ms1	100.0	1.0	44.976
ms2	100.0	1.0	55.904
nf2	99.864	1.0014	715.77
nf3-10	100.0	1.0	33.322
nf3-15	100.0	1.0	54.794
nf3-20	100.0	1.0	82.768
nf3-25	100.0	1.0	118.21
nf3-30	100.0	1.0	155.08
pp	100.0	1.0	35.902
prd	2.575	38.835	380.99
pwq	100.0	1.0	57.919
rb	82.996	1.2049	137.47
rg-10	0	*F*	*F*
rg-2	0.188	531.91	8806.7
s10	37.602	2.6594	96.284
s5	39.419	2.5368	88.039
s7	42.304	2.3638	86.946
sal-10	0.084	1190.5	135324
sal-5	0.079	1265.8	14414
sbt	5.800	17.241	368.36
sf1	0.261	383.14	3731.6
sf2	0.004	25000	312790
shv1	62.199	1.6077	13.132
shv2	100.0	1.0	14.098
sin-10	16.701	5.9877	211.98
sin-20	4.838	20.67	1095.2
stg	29.307	3.4122	119.09
swf	0	*F*	*F*
sz	29.221	3.4222	41.919
szzs	40.166	2.4897	28.297
wf	99.988	1.0001	63.537
xor	100.0	1.0	32.725
zkv-10	100.0	1.0	34.932

(cont.)

Problem name	Successes %	Local Src. per success	Fun/Grad evals per success
zkv-2	100.0	1.0	13.795
zkv-20	100.0	1.0	45.394
zkv-5	100.0	1.0	25.263
zlk1	40.026	2.4984	27.55
zlk2a	6.943	14.403	127.83
zlk2b	14.484	6.9042	94.473
zlk3a	96.579	1.0354	7.2966
zlk3b	98.226	1.0181	7.2951
zlk3c	91.900	1.0881	7.3964
zlk4	99.978	1.0002	12.042
zlk5	61.233	1.6331	30.731
zzs	24.968	4.0051	44.058

The results reported in Table 6.2 suggest that the use of those test problems for which the percentage of successes of Multistart is above, say, 5–10%, is questionable, at least for methods which rely on gradient-based local searches. Also, test functions for which the number of function/gradient evaluations per success is low are probably not challenging enough. A threshold on the number of function/gradient evaluations depending on the dimension of the problem might be chosen in order to select the most challenging test problems, e.g., for GO methods which do not make use of the gradient. The rationale might be that of avoiding the use of test functions for which a gradient-based method in which gradients are numerically estimated can easily detect the putative global optimum.

The test functions reported above have been chosen both as they are quite often cited in the GO literature, but also thanks to the fact that Multistart experiments were very easy to perform starting from the AMPL implementation made available by Ismael Vaz. Other interesting collections of test functions exist. Without claiming any exhaustiveness, we report some of them here.

- Many interesting and challenging GO problems are proposed each year in the form of a competition associated with the IEEE World Congress on Evolutionary Computation (CEC) . As an example, in (Tang, Li, Suganthan, Yang, & Weise, 2010) many test functions are proposed, some of which consist of generalizations which make some of the test problems described above more challenging. Some test functions are generalized to any dimension and some, like Rastrigin's (`rg`) and Ackley's (`ack`), have their feasible regions rigidly rotated and translated. Rotation destroys separability, which is a characteristics of many classical test functions which might be implicitly or explicitly exploited by a GO method in order to reduce an N–dimensional problem into the solution of N one-dimensional ones. Translation is a further modification which avoids locating the global optimum at the origin (or at the center of the feasible box); sometimes, GO methods use the center of the feasible box as a default starting point, and it is desirable to avoid that this initial point is already the global minimum.

- An infinite set of test functions with known global minima and maxima can be defined using interpolation techniques. One such method, based on Shepard's interpolation technique (Gordon & Wixom, 1978) was suggested in (Schoen, 1993). These test functions are defined as

$$f(\mathbf{x}) = \frac{\sum_{i=1}^{k} f_i \prod_{j \neq i} \|\mathbf{x} - \mathbf{z}_j\|^{\alpha_j}}{\sum_{i=1}^{k} \prod_{j \neq i} \|\mathbf{x} - \mathbf{z}_j\|^{\alpha_j}},$$

where $\mathbf{x} \in [0,1]^N$, $k \in \mathbb{N}$ and, for all $j \in 1,\ldots,k$, $\mathbf{z}_j \in [0,1]^N$, $f_j \in \mathbb{R}$, $\alpha_j \geq 0$. These functions interpolate the values $(\mathbf{z}_j, f_j)$ for all j; each parameter α_j can be used to tune the "flatness" of these functions around the point $\mathbf{z}_j$, with the guarantee that, if $\alpha_j > 1$,

$$\lim_{\mathbf{x} \to \mathbf{z}_j} \nabla f(\mathbf{x}) = 0.$$

When using these functions for GO it is important to notice that the global minimum and maximum of $f(\mathbf{x})$ are, respectively, $\min_{j=1,k} f_j$ and $\max_{j=1,k} f_j$. Thus, thanks to all these properties, it is possible to generate test functions in any dimension, with a known set of stationary points whose location can be controlled, and with known global minimum and maximum. The functions might possess other stationary points, at locations different from the $\mathbf{z}_j$'s, but it is guaranteed that one of the $\mathbf{z}_j$'s is the global minimum.

- In (Addis & Locatelli, 2007) another generator of test functions is defined, which depends on a set of parameters to be chosen by the user. The main characteristic of these functions is that, by suitably choosing the parameters, it is possible to build test functions which have a global minimum which is known and which admits a number of local optima which can be controlled by the user. Moreover, and this is the most interesting characteristic of the test suite, the user can also decide on the number of different local optima at different levels (see the discussion on levels and funnels in Section 3.1). In particular, it is possible to build test functions with any prescribed number of funnels. Thus these functions are very well suited for testing local search-based GO methods.

- The Advanced Concept Team of the European Space Agency (ESA-ACT)[9] made available its GTOP Database,[10] a collection of spacecraft trajectory GO problems. These problems are highly multimodal and challenging; a brief description of space trajectory planning problems and of the resulting optimization models can be found in this book in Section 6.2.4.

- Many interesting test functions, including a set of problems whose constraints are not limited to boxes, are contained in (Floudas et al., 1999). In particular, some of the examples, related to "pooling problems" can be used as challenging problems for exact GO methods. These problems originate from industrial chemistry and are very well described in the cited book by Floudas.

- Another vast collection of test problems, coded in the GAMS modeling language, is available under the name of GLOBALlib,[11] and consists of several nonlinearly constrained tests.

[9] `http://www.esa.int/act`

[10] `http://www.esa.int/gsp/ACT/inf/op/globopt.htm`

[11] `http://www.gamsworld.org/global/globallib.htm`

6.2 GO problems arising in applications

GO problems arise in many applicative fields. Driven by our personal experience, in this section we restrict our attention to four of them, namely molecular conformation, distance geometry, packing, and space trajectory problems. However, we point out that many other GO problems arising from applications can be found in the literature. These include, but are certainly not limited to, pooling problems (e.g., the survey (Misener & Floudas, 2009)), inventory theory and scheduling problems (Lootsma & Pearson, 1970; Vandenbussche & Nemhauser, 2005a), problems about the optimal location of objects in the plane (e.g., (Blanquero, Carrizosa, & Hansen, 2009)), molecular replacement problems in X-ray crystallography (Jamrog, Phillips Jr, Tapia, & Zhang, 2005), parameter estimation problems (e.g., (Esposito & Floudas, 1998)), design of a magnetic resonance device (Liuzzi, Lucidi, Piccialli, & Sotgiu, 2004), and multiview geometry problems, where the 3-D structure of a scene is recovered from multiple images (Kahl, Agarwal, Chandraker, Kriegman, & Belongie, 2008). It is also worthwhile to recall that, as seen in Chapter 2, combinatorial optimization problems can be reformulated as GO problems. Examples of these reformulations include the one of the maximum clique problem presented in Section 2.5 and the one of the max cut problem presented in Section 2.6.

6.2.1 Finding the lowest energy conformation of atomic clusters

Chemistry and biology are not only fascinating fields in their own: they are also an extremely rich source of challenging, relevant, difficult, large-scale GO problems. It is widely recognized, and confirmed by experiments, that complex molecules tend to dispose themselves in a configuration in the three-dimensional space which corresponds to the most stable position; this conformation is associated to the minimum possible energy for the system. Of course molecules and atoms are not static, so that it is incorrect to state that they tend to assume a specific conformation. However, the minimal energy conformation happens in general to be the most stable one and the one to which typically each molecule tends.

If the above consideration is taken for granted, then it should be clear that in order to predict the most stable position of a molecule, or, even more challenging, to be able to design a new molecule so that it will display a specific conformation, a fundamental tool is a GO method applied to a function which describes the energy of the complex; in this case GO is used to find the conformation associated to the lowest possible energy. After this phase, the geometry of the resulting molecule can be inspected in order to check if it has the desired shape. We will not go into details in the field of finding accurate models for the potential energy, but will limit ourselves to use the most elementary ones in order to introduce the wide subject of *molecular conformation* problems. Before discussing them, it is worth recalling that very difficult and complex problems arise in the fields of protein folding and docking. Protein folding, from the point of view of GO, is related to the problem of predicting the three-dimensional shape of a bio-molecule when its energy is minimal. The problem has received large attention in the optimization literature (see, just to cite a few important contributions, (Liwo, Lee, Ripoll, Pillardy, & Scheraga, 1999; Ripoll, Liwo, & Scheraga, 2009; J. Lee et al., 2000; Prentiss, Wales, & Wolynes, 2008; McAllister & Floudas, 2010)). One of the reasons we do not describe in detail the underlying models is that on one hand they are quite complex and their description would necessitate a long introduction, and on the other hand, there is no single model which is universally accepted as the best one in describing the energy of a bio-molecule. Many models, called *force fields*, exist (see, e.g., AMBER (Case et al., 2005), GROMACS (Van Der Spoel et al., 2005), ECEPP (Arnautova, Jagielska, & Scheraga, 2006), but other, more recent, force fields are also available) with different characteristics, assumptions, and capabilities of obtaining a reasonable compromise between model accuracy and model simplicity. The most successful algorithms for protein folding are quite complex, but some of them can be related to a few important techniques described elsewhere in this book, like the α-BB approach (see Section 4.6) and population-based basin hopping (BH) (or “conformational space annealing”) methods (see Section 3.1.3). The second problem mentioned above, protein docking, is even more challenging: in this case two or more large bio-molecules are assumed to interact in such a way that they eventually form a single complex, and during this docking phase local modifications to the shapes of the molecules occur. Descriptions of these problems as GO ones can be found, e.g., in (Klepeis, Ierapetritou, & Floudas, 1998; K. Lee, 2008). The problem, given a suitable force field, can be modeled following an approach similar to those which can be used in protein folding. These problems, typically, give rise to huge optimization problems with a very large number of variables. A version of docking exists in which molecules are considered as rigid bodies; in this case the dimension of the optimization problem is only 6 (three variables for the rigid translation and three for the rotation of one body with respect to the other, considered fixed). Even in this case, which is a strongly simplified model, the presence of a huge number of local minima makes the resulting optimization problem very challenging.

Here we describe the simplest possible model, namely that of *atomic clusters*. These are groups of atoms interacting via pairwise potential, with no chemical bond. A conformation problem for atomic clusters can be defined as the following unconstrained GO problem:

$$\min \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} v_{i,j}(\|\mathbf{x}_i - \mathbf{x}_j\|_2),$$

where N is the number of atoms in the cluster, $\mathbf{x}_i \in \mathbb{R}^3$, $i = 1,\dots,N$, represent the coordinates of the center of the ith atom, and $v(\cdot,\cdot)$ is a *pair potential* whose analytic expression might depend on each atom's type. If the cluster is composed of identical atoms, the subscripts of v can be omitted. Frequently used potential functions are

$$v(r) = LJ(r) = \varepsilon \left(\frac{r_0}{r}\right)^6 \left(\left(\frac{r_0}{r}\right)^6 - 2\right),$$

known as the *Lennard-Jones* (LJ) potential (usually $\varepsilon = r_0 = 1$), and the *Morse potential*

$$v(r) = M(r;\rho) = e^{\rho(1-r)} \left(e^{\rho(1-r)} - 2\right),$$

where $\rho > 0$ is a parameter.

These models are approximate potential energy models frequently used to describe clusters made of specific atom types, like some rare gases. The shape of both pairwise potential functions, as can be seen in Figure 6.1, displays a steep increase when the distance

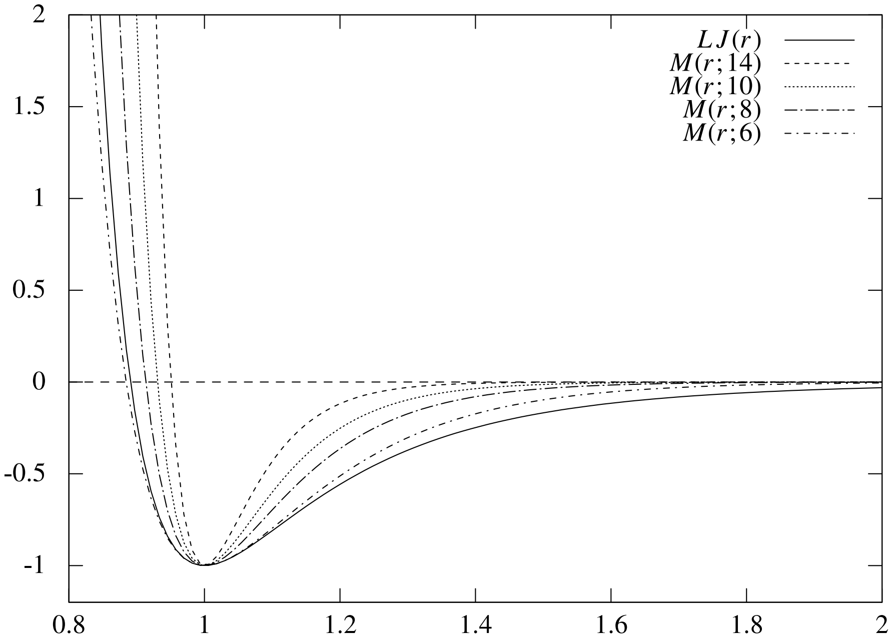


Figure 6.1. *Graph of Lennard-Jones and Morse pair potential, as a function of pairwise distance; here $\varepsilon = r_0 = 1$*

between two atoms approaches zero, a unique minimum point, which corresponds to the most stable distance in a single, isolated pair of atoms, and an asymptotic convergence to 0 when the pairwise distance diverges to infinity. The parameter ρ in the definition of the Morse potential influences the shape of the pairwise contribution. Larger values of ρ correspond to short-range potential, i.e., the effect of an atom can be “felt” only by atoms which are quite close. If ρ is increased to ∞, the Morse potential converges to a discrete potential which is ∞ for distances less than 1, -1 at unit distance, 0 elsewhere.

Both pair potentials are nonconvex functions of the relative distance between two atoms; when these pairwise contributions are summed up over all pairs of atoms in a cluster, the resulting objective function is a highly multimodal, nonconvex function of the coordinates of the N atoms.

In recent years a few attempts have been made to prove some theoretical properties concerning the global minima of LJ and Morse potential functions, although most of the literature has been devoted to computational approaches. Even if the definition of the GO problem is quite elementary, the theoretical analysis of properties enjoyed by global minima is exceptionally hard; only a few rigorous results exist in the literature and they invariably prove properties whose validity (typically in a much stronger form) is taken for granted by chemists. As an example of this kind of results, some papers concentrated on finding a positive lower bound on the minimum interatomic distance of globally optimal clusters. This information might prove useful in designing BB exact methods and also in guiding the exploration of new solutions in heuristic methods. However, the results, among which we cite (Xue, 1997; Locatelli & Schoen, 2002; Schachinger, Addis, Bomze, & Schoen, 2007), even after quite refined and elaborated analyses, offer lower bounds that are significantly lower than the minimal interatomic distances observed in putative optimal configurations.

From the computational point of view, first attempts in finding lowest energy configurations for clusters were based on a priori knowledge about the structures most often observed. For LJ clusters of up to 200–300 atoms, icosahedral structures are, by far, the most common ones. Based on this fact, the first methods (see, e.g., (Xue, 1994)) were based on deciding the best position for N atoms on an icosahedral mesh. This initial solution was then refined by running a local descent algorithm.

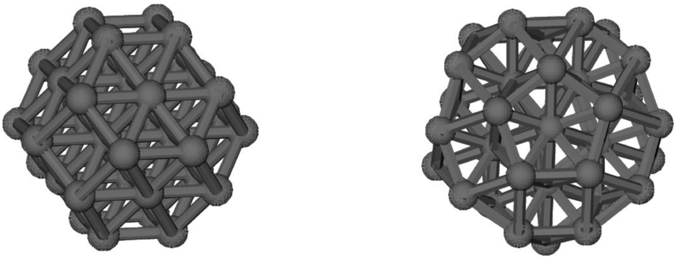


Figure 6.2. *The putative global optimum configuration (left) and the putative best icosahedral-based configuration (right) under the LJ potential with* 38 *atoms*

A clear defect of this approach is that starting from the assumption of an icosahedral lattice, those methods are limited to that geometry and it is virtually impossible to obtain configurations of radically different shapes and, possibly, significantly lower energy. Often there is not a huge difference in energy between a good icosahedral cluster and a good cluster with a different shape. As an example, the LJ 38-atom cluster putative global optimum, well known to be the smallest exception to icosahedral putative optima, is based on a truncated octahedron shape. Its LJ potential, in standard units with $r_0 = 1, \varepsilon = 1$, is equal to -173.928427, while the energy of the best-known icosahedral structure with 38 atoms is equal to -173.252378. Figure 6.2 graphically shows the shape of these clusters. However, due to the rugged energetic landscape of the potential energy, any path which, in the 38×3–dimensional space of variables as well as in the three-dimensional space of atomic centers, continuously connects the two configurations must inevitably traverse intermediate configurations with very high potential. In other words, there is a high energetic barrier which separates the two shapes. This is also the reason by which we can explain the importance of searching for the global optimum. Given its depth and the high barriers around it, when that configuration, thanks to thermodynamical atom movements, is reached, it becomes a quite

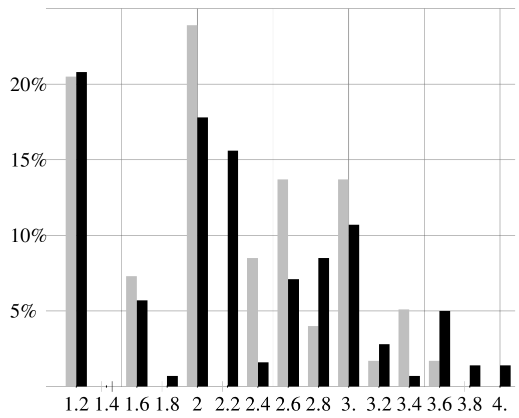


Figure 6.3. *Frequency distribution of the pairwise distances. On the left, in gray, the putative optimum for LJ*$_{38}$*; on the right, in black, the best-known icosahedral geometry for the same cluster*

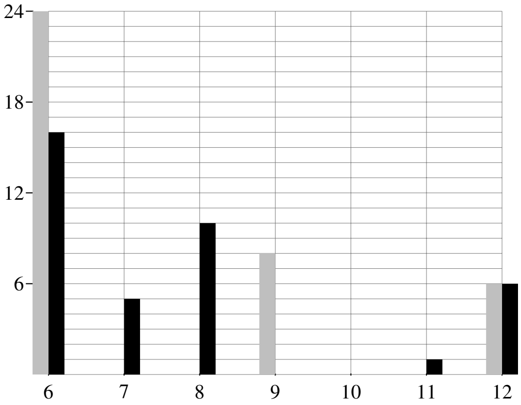


Figure 6.4. *Frequency distribution of the number of atoms having a given number of neighbors. On the left (gray) the statistics for the putative optimum of LJ*$_{38}$*; on the right (black) that of the best-known icosahedral geometry*

stable one, difficult to escape from. For the interested reader, we cite (Wales, 2003), a very deep and complete analysis of potentials and of cluster conformation problems. Just to get a quick idea of the difference between clusters of similar energy but different geometry, in Figures 6.3–6.4 we report nearest neighbor statistics of two LJ$_{38}$ clusters. In particular, in the first figure we report the frequency distribution of pairwise distances between all pairs of atoms. The figure reports the percentage of pairs having distance up to 1.2, the one for pairs whose distance is between 1.2 and 1.4, and so on. Figure 6.4 gives the total number of atoms having $6, 7, \ldots, 12$ "nearest neighbors," i.e., atoms within a given distance from the current one. In the figure the threshold used was 1.2. It is quite evident from both figures that the geometrical structures of the two clusters are radically different. Many papers (see, e.g., (Doye & Wales, 1997)) analyze the geometry of putative optimal clusters and the "energetic barriers" between pairs of low energy clusters of the same dimension.

A significant breakthrough in cluster optimization started in the late 90s when (Wales & Doye, 1997) introduced the basin hopping method (see Section 3.1.3). This paper started a large research effort both in the direction of understanding the properties of the method (why and when the energetic landscape transformed by means of local optimization is easier to explore than the original one) and in the direction of extending it to different problems in chemical physics (other potential functions, binary compounds, water clusters, fullerene, crystals, proteins, etc.).

Nowadays it is fully recognized that exploration of the rough energetic landscape of a molecule or a cluster is more efficiently performed if this landscape is transformed by means of local optimization in order to obtain two important results: to lower the number of different configurations to be explored and to eliminate some barriers between adjacent configurations.

Recently several approaches have been proposed in order to go some steps further in the direction of applying local optimization to ease the exploration of the conformational space.

Some research efforts have been devoted to analyzing the effect of different *GloballyGenerate* procedures in a BH (see Section 3.1.3) algorithm. Leary (1997) introduced the "big bang" method, where the initial configuration was randomly generated in a very small box, so that each pairwise distance was significantly lower than the equilibrium one. A detailed analysis was also carried out in (Marques, Pais, & Abreu, 2010), where the correlation between the shape of the box used in the starting point configuration and the local optimum obtained was analyzed. It should be remarked, however, that as in most BH implementations, the strongest impact on the efficiency and on the discovery capabilities of BH methods derives on one hand from the *LocallyGenerate* procedure and, on the other hand, from the use of a population-based approach.

The first implementations of BH for atomic clusters were based on the classical BH scheme. Many variants of these methods appeared in the last few years, some of which (Locatelli & Schoen, 2003; Doye et al., 2004) were based on the introduction of a *two-phase* local search to be employed in place of a standard one in the BH scheme. This kind of local search is implemented by first running a standard local descent method on a modified objective function, which tends to favor some specific geometrical shapes through penalties, followed by a regular local search. The effect of this double local search in cluster optimization was analyzed Doye (2000), who showed experimentally that the introduction of such a modified local search changes the shapes of the funnels of LJ clusters in such a way that the deepest one is, in some cases, much more easily accessible through BH methods. In (J. Lee et al., 2000, 2003) the *conformational space annealing* method was introduced, where the idea of using a population and a dissimilarity criterion among elements in the population was exploited. This method, extended also to the optimization of complex bio-molecules, was extremely successful in obtaining low-energy conformations of moderately large clusters, avoiding the greedy behavior of sequential methods.

Recently some very promising approaches have been reported in the scientific literature. One of the basic ideas is that local optimization, although very powerful, is not suitable for obtaining stable cluster configurations when the dimension of the cluster increases, so that, even after using local optimization, there might be funnels with far too many minima to explore and too large barriers between them. Recent approaches tend to refine the local generation phase by designing algorithms that go very deep in energy by means of a clever use of local optimization and special purpose modifications to the cluster geometry. One precursor of this kind of modification was (Hartke, 1999) where the concept of *direct mutation* was introduced. In that paper, mutation was defined as the relocation of single atoms belonging to the surface; the idea was that the poorest contribution to the total energy of a cluster comes from the surface, where atoms have a lower number of nearest neighbors with respect to the interior of the cluster. So changing the position of a surface atom might have a good effect on energy minimization. In (Cheng et al., 2009) this idea is refined and forms the basis of a method that the authors call *funnel hopping*, a BH method with a tailored local generation phase. The basic scheme can be described as follows. First a build-up phase is executed to find good candidate points for the relocation of atoms; these points are called *dynamic lattice* (as opposed to the static lattice used in biased methods, such as those based on the icosahedral lattice). What follows is a summary of this procedure, which should be considered as a part of the *LocallyGenerate* procedure of BH.

Identification of "surface atoms." There is no explicit surface identification here, but a simple evaluation of the contribution of each atom to the total energy:

$$v_i = \sum_{j \neq i} v(\|X_i - X_j\|_2)$$

(notice that the total energy is simply half the sum of the above contributions). The "surface" is defined as a set of atoms with highest energy.

Neighborhood graph definition. A graph is built whose nodes are atoms and whose edges connect pairs of atoms whose Euclidean distance is approximately equal to the equilibrium pair distance

$$r^\star \in \arg\min_{r \geq 0} v(r)$$

(in LJ clusters, an upper bound around 1.2 might be safely assumed to be a threshold).

Triangle construction. Triangles, i.e., cliques of cardinality 3, in this graph are identified.

Lattice definition. An almost regular tetrahedron is built over each triangle by adding a fourth atom position in one of the two possible ways (above or below the triangle). These new points, if they are not too close to other atoms or other points in the lattice, are inserted in a structure called the "dynamic lattice."

Lattice quenching. For each point in the lattice a three-variable minimization problem is solved which consists in minimizing the total energy of a cluster with N fixed atoms plus a single variable one, whose position is initialized at the lattice point.

This procedure generates a bunch of candidate empty locations where other atoms might be profitably placed. After the lattice has been built, a combinatorial search strategy is applied, whose aim is to move some atoms from their high-energy location to a lattice point in such a way that the energy is minimized.

This procedure, possibly included in a population-based scheme, is a special purpose *LocallyGenerate* procedure whose definition is strongly problem-dependent; the introduction of such a kind of specific generation tools might produce dramatic effects in the behavior of GO methods.

It should also be remarked that conformational problems being so difficult, it is not expected that a single modification is enough to guarantee quick minima discovery. Indeed this is the case: even the excellent method of (Cheng et al., 2009) fails when implemented in a nonpopulation framework. Local optimization improvements are of highest importance in the efficient exploration of funnels; however, using just these techniques there is a strong danger of evolving toward a too-greedy approach, one in which a few funnels are very effectively explored, while others, initially less promising, are left out. Population-based approaches, through the direct or indirect enforcement of diversity among population members, are the key for successful exploration of multiple funnels. In (Cheng et al., 2009) the funnel hopping procedure is embedded in a quite standard genetic scheme, in which new generations are obtained by cut-and-paste operations on two parent clusters (typically two clusters are cut in halves by means of a random plane). The resulting cluster is optimized and the resulting molecule enters the new generation provided no similar cluster is already present or the energy is sufficiently low. This way, clusters in the population have a certain degree of dissimilarity, which is extremely beneficial in terms of discovery capabilities. In (Grosso et al., 2007b) a generic scheme for population-based algorithms has been introduced in which diversity is first defined by introducing dissimilarity measures; then it is enforced as one of the rules in population evolution (see Section 3.1.3). The need for dissimilarity measures is evident from the fact that in cluster optimization it has been recognized that the energy alone is not sufficient to discriminate among clusters. Pairs of clusters of largely different geometry might possess a similar energy value and, on the contrary, clusters with quite similar geometry might have quite different energy, just as a consequence of the location of one or two atoms.

Similar approaches have been used with success in different situations, such as in the optimization of *binary clusters*, i.e., clusters made of two different kinds of atoms. For a recent survey on the application of GO methods to cluster conformation we refer to (Locatelli & Schoen, 2012a).

6.2.2 Distance geometry

Closely related, although significantly different with respect to cluster energy minimization, is the so-called *distance geometry* problem. Again the final aim is to obtain a good three-dimensional representation of a large bio-molecule. However, while in the approaches described previously in this section the conformation was obtained starting from "first principles," in distance geometry a set of measurements obtained from the molecule under study is available. Depending on the technology used, which might be based on spectrographic analysis or on NMR observations, these measurements typically take the form of a set of distances between pairs of atoms. These distances are only seldom exact ones: often they are reported in the form of an uncertainty interval, so that the distance measurement between a single atom pair is given through a lower and an upper bound. The aim of distance geometry is that of reconstructing the three-dimensional shape of the molecule given partial information on a subset of pairwise distances. Here no potential function is assumed and the problem is purely geometrical. Let N be the number of atoms in the molecule, $E \subseteq \{1,\dots,N\} \times \{1,\dots,N\}$ be the set of pairs of atoms for which we have a distance measurement, and $\{(\ell_{ij}, u_{ij})\}$ be the lower and upper bounds on the distance of the pair of atoms $(i,j) \in E$. The distance geometry problem is defined as the following feasibility problem: find N three-dimensional vectors $\mathbf{x}_1,\dots,\mathbf{x}_N$ such that

$$\|\mathbf{x}_i - \mathbf{x}_j\|_2 \in [\ell_{ij}, u_{ij}] \qquad \forall (i,j) \in E. \tag{6.1}$$

The distance geometry feasibility problem can be formulated as a GO problem in several different ways. One of the most frequently used GO formulations is the following (see, e.g., (Morè & Wu, 1999)):

$$\min_{\mathbf{x}_1,\dots,\mathbf{x}_N} \sum_{(i,j)\in E} \left(\frac{\min\{0, \|\mathbf{x}_i - \mathbf{x}_j\|_2 - \ell_{ij}\}^2}{\ell_{ij}^2} + \frac{\min\{0, u_{i,j} - \|\mathbf{x}_i - \mathbf{x}_j\|_2\}^2}{u_{ij}^2} \right),$$

where a penalty is associated to violations of the constraints (6.1). It has been experimentally observed in (Grosso et al., 2009) that if the absolute constraint violation

$$\min_{\mathbf{x}_1,\dots,\mathbf{x}_N} \sum_{(i,j)\in E} \left(\min\{0, \|\mathbf{x}_i - \mathbf{x}_j\|_2 - \ell_{ij}\}^2 + \min\{0, u_{i,j} - \|\mathbf{x}_i - \mathbf{x}_j\|_2\}^2 \right)$$

is optimized in place of the relative one, significant advantages may be obtained. This problem has many variants depending on the available prior information and on the quality and quantity of pair distances known. As an extreme case, if all pairwise distances were known with no uncertainty, then the problem could be solved in polynomial time (for this classical result see, e.g., (Crippen & Havel, 1988)). Frequently some information is available on the atom "identities," which makes the three-dimensional reconstruction easier. In particular, when measurements come from the observation of a protein, it is often possible to label the atoms according to their positions in the amino acid chain of the protein. This way, if enough distances are known, an elementary build-up procedure based on the solution of second-order equations can lead to the solution of a distance geometry problem (see (Q. Dong & Wu, 2002)). When this build-up phase fails as a consequence of a too sparse set of known distances, similar ideas can be exploited in a BB-like scheme (see, e.g., (Lavor, Liberti, Maculan, & Mucherino, 2012)). A recent survey on the distance geometry problem is (Mucherino, Lavor, Liberti, & Maculan, 2013), where these and other related approaches are discussed in detail.

It is worth observing in closing this section on distance geometry that this problem is very closely related to a problem in telecommunication networks known as the *sensor localization* problem. Here, similar to the case of molecules, a set of pairwise distances between points, typically in $\mathbb{R}^2$ and only seldom in $\mathbb{R}^3$, are given. Points are considered as sensors distributed on a region, which communicate with neighbors located at a sufficiently small distance through radio signals. The important difference between this problem and the one outlined above is that distance information is available, possibly affected by noise, for *all* pairs which are located within a threshold distance r one from the other. Thus, an important piece of information available in this case is that all pairs of points for which distance information is not available are known to be located at least r units apart from each other. Another, somewhat less relevant, difference with respect to generic distance geometry problems is that in sensor localization sometimes a few sensors are precisely localized. This corresponds to the fact that some specific sensor might have been placed at a known position or that some sensors, more powerful than others, are able to estimate and communicate their coordinates. These problems have been studied by many authors, among which we cite (Biswas, Ye, Wang, & Liang, 2006; Carter, Jin, Saunders, & Ye, 2006; Tseng, 2007; Man-Cho So & Ye, 2007; Pong & Tseng, 2011).

6.2.3 Optimal packing of disks, spheres, and other geometrical objects

Geometrical packing problems consist in finding the "best" conformation of a set of geometrical objects in a container, without overlap. The most widely studied problem is that of packing circles in squares, circles, or rectangles, but different object and/or container shapes are also sometimes analyzed; three-dimensional problems involving spheres have also received quite a lot of attention. The applications of these packing problems are quite evident, as they concern placing objects in containers in such a way as to minimize wasted surface or volume or maximize the quantity of placed objects. In a typical packing problem there are a few characteristic quantities which might be optimized:

1. The number N of objects. In a typical packing problem, given a fixed container and the shape of a set of objects that are usually, but not necessarily, identical, the problem is to maximize the number N of objects which can be placed without overlapping in the container. A part of the problem is also finding the correct localization of the chosen N objects.

2. The "size" of N objects. Given the number N of objects, their shape and a fixed container, if the object shape depends on one or a few parameters, it is required to optimize these parameters so that the "largest" N objects can be placed in the container.

3. The "size" of the container. Given a fixed number of known objects, all of them have to be placed in the "smallest" possible container.

In what follows we restrict our attention to a rather classical packing problem, namely that of placing identical disks in a square, but we point out that other packing problems involving different objects and/or containers have been discussed in the literature. Results for many such problems are reported at `www.packomania.com`.

Let us denote by

$$\text{dist}(i,j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

the Euclidean distance of two points $(x_i\ y_i)$ and $(x_j\ y_j)$. For the case of packing disks in a square, the three problems discussed above become the following:

1. Given the unit square $[0,1]^2$ and disks of a prefixed radius $r < 1/2$, find the largest possible value for N so that the following problem is feasible:

$$\begin{aligned} \text{dist}(i,j) &\geq 2r & \forall i,j : 1 \leq i < j \leq N, \\ x_i &\in [r, 1-r] & \forall i \in 1,\ldots,N, \\ y_i &\in [r, 1-r] & \forall i \in 1,\ldots,N. \end{aligned}$$

2. Given the number N of disks and a unit square container, maximize the common radius r of all disks, maximize the total covered surface $N\pi r^2$, or minimize the waste $1 - N\pi r^2$:

$$\begin{aligned}
&\max_{\mathbf{x},\mathbf{y},r} r && && (6.2)\\
&\text{dist}(i,j) \ge 2r && \forall i,j : 1 \le i < j \le N, &&\\
&x_i \in [r, 1-r] && \forall i \in 1,\dots,N, &&\\
&y_i \in [r, 1-r] && \forall i \in 1,\dots,N. &&
\end{aligned}$$

3. Given a fixed number N of disks with unit radius $r = 1$, find the smallest square containing them:

$$\begin{aligned}
&\min_{\mathbf{x},\mathbf{y},L} L && && (6.3)\\
&\text{dist}(i,j) \ge 2 && \forall i,j : 1 \le i < j \le N, &&\\
&x_i \in [1, L-1] && \forall i \in 1,\dots,N, &&\\
&y_i \in [1, L-1] && \forall i \in 1,\dots,N. &&
\end{aligned}$$

Problem (6.2) is by far the one which has received the most attention in the scientific literature. It can also be observed that problem (6.3) is equivalent to problem (6.2). To see this, just transform (6.2) as

$$\begin{aligned}
&\max_{\mathbf{x},\mathbf{y},r} r && \\
&\sqrt{((x_i - x_j)^2 + (y_i - y_j)^2)/r^2} \ge 2 && \forall i,j : 1 \le i < j \le N,\\
&x_i/r \in [1, 1/r - 1] && \forall i \in 1,\dots,N,\\
&y_i/r \in [1, 1/r - 1] && \forall i \in 1,\dots,N,
\end{aligned}$$

and the equivalence becomes clear choosing $L = 1/r$.

Problem (6.2) can be equivalently reformulated as the problem of scattering N points in the unit square so that their minimal pairwise distance is maximized:

$$\max\min\{\text{dist}(i,j) : 1 \le i < j \le N, (x_i\ y_i) \in [0,1]^2\},$$

which can be formulated as the following GO problem:

$$\begin{aligned}
&\max d && && (6.4)\\
&\text{dist}(i,j) \ge d, && 1 \le i < j \le N, &&\\
&x_i \in [0,1], && &&\\
&y_i \in [0,1]. && &&
\end{aligned}$$

It is easy to show that a strong relation links this problem with that of packing circles. In fact the values of the optimal objectives (6.2) and (6.4) are linked by the following relation:

$$r^\star = \frac{d^\star}{2(d^\star + 1)}.$$

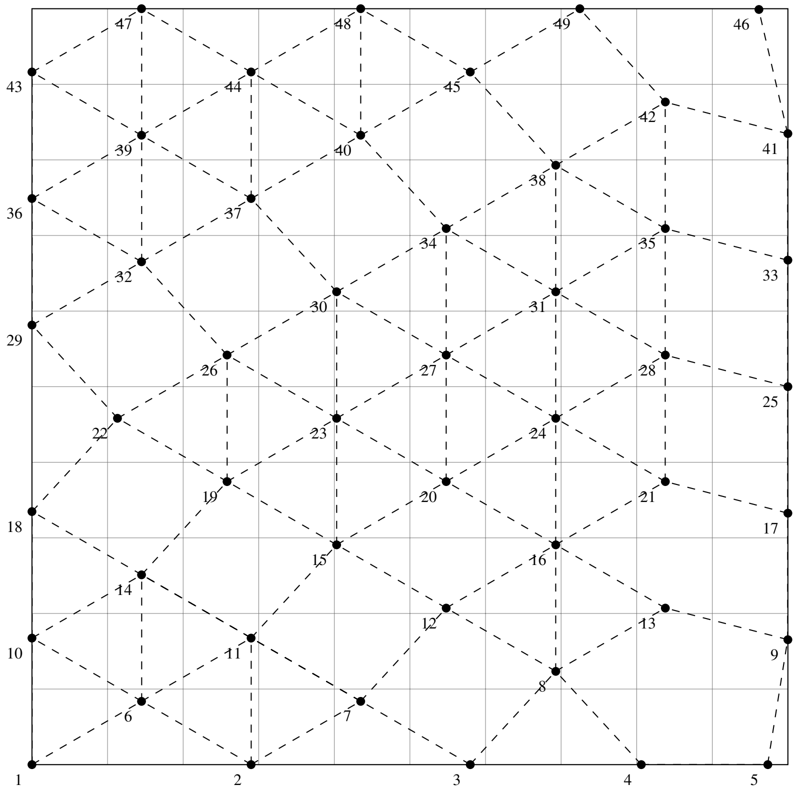


Figure 6.5. *Putative optimal scattering of* 49 *points in the unit square*

In Figure 6.5 an image is reported of the putative optimal scattering of 49 points in the unit square (while the equivalent optimal packing was displayed in Figure 1.1 on page 4). It might be somewhat surprising to notice that these conformations, with $r = 0.07169268170362$ ($d = 0.16738607686832$), are significantly better than the one obtained with a regular 7×7 grid, which corresponds to $r = 0.07142857142857$ ($d = 1/6$).

As can be seen, we used the term "putative optimum" to denote this configuration. Indeed, theoretical proofs of global optimality for the problem of optimally placing equal disks in the unit square are confined to a very limited set of values for N.

Optimal packings have been theoretically proven for $N \leq 9$ in (Schaer, 1965; Schaer & Meir, 1965; Schwartz, 1970); proofs of optimality have also been derived for $N = 14, 16, 25, 36$ (Wengerodt, 1987a, 1983, 1987b; Wengerodt & Kirchner, 1987). Thanks to the improvements in exact GO methods, some of which have been presented in this book, some computational proofs of optimality have been obtained. By this we mean that a special-purpose BB algorithm has been coded and run for specific problem instances, returning an optimal or an ε-optimal conformation. For example, computational proofs of optimality have been found for all values of $N \leq 30$ in (de Groot, Peikert, Würtz, & Monagan, 1991; Nurmela & Oestergard, 1999; Markót & Csendes, 2005). In (Locatelli & Raber, 2002) optimality within precision 10^{-5} has been proven for $N \leq 35$, $N = 38, 39$.

These and many other theoretical and computational results on disk packing, are included in the volume (Szabó et al., 2007).

Most of the recent research in circle packing has been devoted to heuristic techniques aimed at confirming and possibly improving previously known configurations. The already mentioned `www.packomania.com` site maintains the best-known configurations for this problem and several of its variants.

Problems (6.2) and (6.4) are nonconvex, both with a linear objective function and quadratic reverse convex constraints, in these formulations. Of course many other equivalent formulations are possible, some of which might reveal themselves as preferable from the computational point of view. As an example, in (Lòpez & Beasley, 2011) some interesting numerical results have been obtained by using a polar coordinate representation, which, for our problem, turns out to be

$$\begin{aligned} &\max d \\ r_i^2 + r_j^2 - 2r_i r_j \cos(\theta_i - \theta_j) &\geq d^2, \qquad\qquad 1 \leq i < j \leq N, \\ r_i \sin(\theta_i) &\in [0,1], \\ r_i \cos(\theta_i) &\in [0,1], \\ \theta_i &\in [0, \pi/2]. \end{aligned}$$

Actually, in that paper the authors propose to mix coordinate representations: some circles are represented through their centers' Cartesian coordinates while some others are represented in polar form.

We would like to avoid mentioning the vast literature devoted to computational approaches for packing problems, which is quite large even when restricted to circle packing. We just cite some references, such as (Addis et al., 2008a; Addis, Locatelli, & Schoen, 2008b; Boll, Donovan, Graham, & Lubachevsky, 2000; Casado, Garcia, Szabó, & Csendes, 1998; Castillo, Kampas, & Pinter, 2008; Donev, Torquato, Stillinger, & Connelly, 2004; Grosso et al., 2010; X. Huang & Yang, 2003; Huang and Li and Akeb and Li, 2005; Kallrath, 2009; Liu, Xue, Liu, & Xu, 2009; Locatelli & Raber, 2002; Lòpez & Beasley, 2011; Maranas, Floudas, & Pardalos, 1995; Markót & Csendes, 2005; Nurmela & Oestergard, 1997, 1999; Szabó, Markót, & Csendes, 2005; Szabó et al., 2007; H. Wang, Huang, Zhang, & Xu, 2002). Most of the best-performing methods rely on local optimization as a powerful tool for improving starting configurations which are either randomly generated or obtained by perturbation of a solution. Thus, most of the proposed approaches perfectly fit into the general framework for heuristics introduced in this book. If we refer to the scheme presented in Section 3.1 and, in particular, to the scheme presented in Algorithm 3, some specific components can be identified:

GloballyGenerate. Most methods just generate initial solutions by choosing the relevant parameters uniformly in the search space or in a region in some way connected to the feasible space. For disk packing in a square, the coordinates of the centers are uniformly generated in the unit box; in most formulations, these are not the unique variables of the problem. For instance, in our formulations a distance, or radius, variable is also necessary. This variable might be itself randomly generated, or it might be computed after the generation of centers, e.g., as

$$d \leq \bar{d} = \min_{i \neq j} \|(x_i \; y_i) - (x_j \; y_j)\|_2.$$

In the experiments performed in (Addis et al., 2008b) it was observed that choosing $d = 0$ produced much better results than $d = \bar{d}$. This can be explained by observing that the first choice gives much more freedom to the current configuration, enabling the subsequently called local optimization refinement to perform longer steps.

LocallyGenerate. Various forms of local perturbation have been used in the literature in order to generate a neighboring solution which, hopefully, after local optimization will lead to an improved configuration with respect to the starting one. In packing points in the unit square a possible perturbation for each coordinate x is the uniform one in

$$[\max\{0, x - 1/(2\sqrt{N})\}, \min\{1, x + 1/(2\sqrt{N})\}].$$

Many of the earlier methods did not use local optimization at all and tried to mimic a physical rearrangement of disks in order to improve a starting configuration. One of the best known such methods is the *billiard simulation* introduced in (Lubachevsky, 1991), where disks were considered as rigid bodies, each characterized by a position (its center) as well as a velocity, which is maintained constant until a collision happens. In that case, the component of the velocity vector normal to the line passing through the centers of the colliding disks is kept, while the component parallel to that line is reversed.

Another, quite interesting approach (Donev et al., 2004) is based on the detection, by means of the solution of a linear programming problem, of a direction by which all disks can be moved, unless they form a rigid configuration. Here, although the authors use optimization, its role is not that of finding a locally optimal configuration of the current configuration, but only that of building a feasible direction which allows the movement of disks.

Some authors introduced the idea of “direct moves” in circle packing; these perturbations are problem specific, quite similar to what has been done in the field of molecular optimization (see Section 6.2.1). As an example, in (W. Huang & Ye, 2010) a set of “vacant” places is defined relative to the current disk configuration and, systematically, each disk is tentatively placed in one of the top vacant positions prior to local optimization. In order to measure the vacancy and to rank empty parts in a disk packing configuration, the authors suggest to use a probe disk of the same radius r as the currently placed disks. Assuming that this disk is placed at position (x, y) in the square $[\ell, u]^2$, its overlap with disk i located at $(x_i\ y_i)$ is defined as

$$Ov_i = \max\{0; 2r - \sqrt{(x - x_i)^2 + (y - y_i)^2}\}^2,$$

the overlap measures with respect to the lower bounds are

$$Ov_{\ell,x} = \max\{0; \ell + r - x\}^2,$$
$$Ov_{\ell,y} = \max\{0; \ell + r - y\}^2,$$

and, for the upper bounds,

$$Ov_{u,x} = \max\{0; r + x - u\}^2,$$
$$Ov_{u,y} = \max\{0; r + y - u\}^2.$$

This way two overlapping disks are weighted according to the closeness of their centers, while a probe disk outside the box is weighted on the basis of the closest point to the border.

The total measure of vacancy is then defined as

$$\sum_{i=1}^{N} Ov_i + \sum_{i \in \{\ell,u\}, j \in \{x,y\}} Ov_{i,j}. \tag{6.5}$$

In order to find the most vacant empty places, a two-variable local optimization is started several times, in a Multistart framework, from randomly chosen starting points with the total measure of vacancy as an objective function to be minimized when $(x\ y)$ is free to move inside the box. Once this optimization phase is finished, different solutions are ranked according to the value of (6.5) and a few solutions are kept.

The direct move consists in trying to place each of the N disks in one of the top vacant places and to run a local optimization, based on the original objective function and with all N disks free to move, starting from this configuration. An acceptance criterion based on strict improvement of the objective function is used.

As an example, in Figure 6.6 we report the 10 most vacant places we found for the putative optimal configuration of 49 disks in the unit square.

Local Search. Most recent methods employ a standard local optimization routine in order to refine a randomly chosen starting point. In (Addis et al., 2008b, 2008a; Grosso et al., 2010) we usually employed SNOPT (Gill, Murray, & Saunders, 2005), as other authors also did (see, e.g., (Liu et al., 2009; Lòpez & Beasley, 2011)).

Some authors used not only different solvers but, in the local phase, optimized a different objective function in order to promote dense enough patterns. We mention here (Nurmela & Oestergard, 1997), where the local search is based on the minimization of an "energy" function

$$\sum_{i<j} \left(\frac{\lambda}{\|(x_i\ y_i) - (x_j\ y_j)\|_2^2} \right)^m,$$

where m is a positive integer and $\lambda > 0$ is a constant term. The authors progressively increase m from one run to the next and perform a local search with a standard continuous descent method.

IsAcceptable. Acceptance of a new configuration is usually based on strict improvement with respect to the current one. However some authors used "simulated annealing" types of criteria (see Section 3.1.10).

Many other approaches might be cited which are based on the application of GO heuristics to this and other related geometrical problems. This introduction to the topic was mainly aimed at defining the disk packing models and showing how some of the ideas introduced in this volume may be successfully applied to this challenging problem.

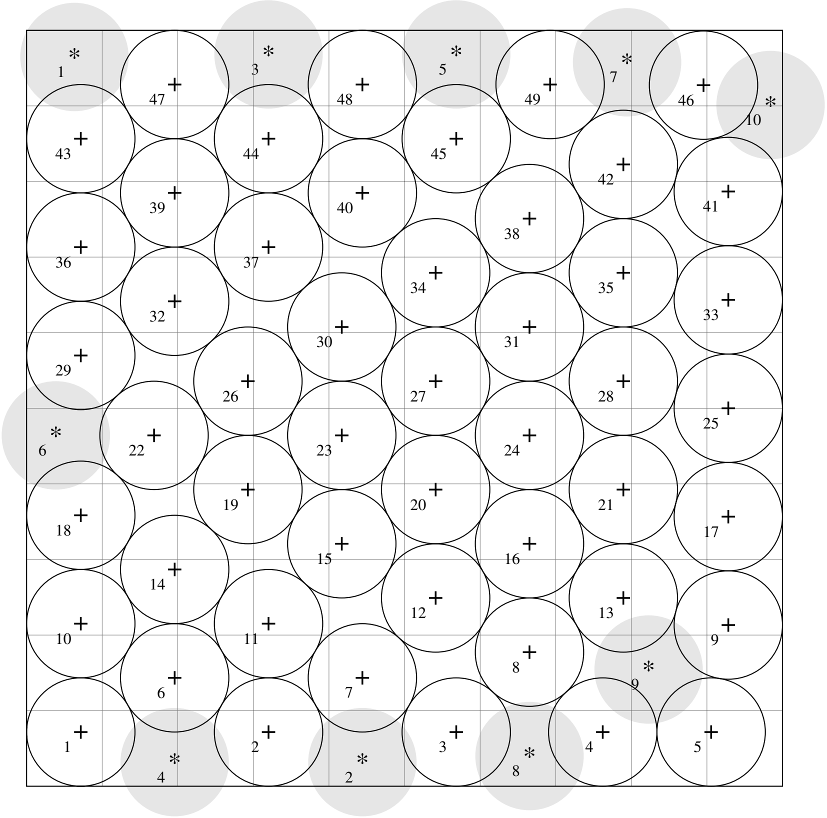


Figure 6.6. *The* 10 *most vacant sites (in gray) in the putative optimal packing of* 49 *disks in the unit square*

6.2.4 Optimal space trajectory planning

The problem of optimally designing a trajectory for a spacecraft leaving Earth in a mission planned to terminate at some planet is a challenging problem which can be formulated as a GO model and approximately solved thanks to strategies based on the methods presented in this book. Without going into much detail on the subject, we introduce here a few basic concepts and models. The literature on space trajectory planning and on its solution through GO techniques is quite vast. We cite the recent book (Fasano & Pinter, 2013) and the references cited in that volume, and some papers on the subject ((Izzo et al., 2007; Olympio & Marmorat, 2008; Addis et al., 2011; Locatelli & Vasile, 2009; Vasile et al., 2011)). The main decision variables to be defined when planning a space mission are departure time and velocity from Earth and the analogous arrival date and velocity to the target planet. These variables might suffice in completely determining a spacecraft trajectory if no *deep space maneuver* or any *planet swing-by* is allowed. A deep space maneuver corresponds to the decision of using the on-board engine (frequently a chemically propelled one); variables can be associated to the decision on when to switch on the engine, for how long, and on the change in the velocity vector obtained thanks to the engine. A planetary swing-by is obtained when the orbit of the spacecraft gets so close to a planet (possibly Earth itself) that the gravitational attraction of the planet becomes much higher than that caused by the sun. In these situations, the spacecraft can "use" some of the kinetic energy of the planet to obtain a change in its velocity vector. Planning a trajectory when these kinds of maneuvers are allowed entails choosing the sequence of planets to be "visited," which is a combinatorial decision, and choosing dates and other relevant quantities (like velocities or the distance from the planet's center when the swing-by is performed). Usually, as swing-bys are quite limited in their total number, the combinatorial part of the problem is solved by enumeration of all admissible swing-by sequences. The resulting problem, thus, is a continuous GO problem of moderate size. The objective of the GO problem is usually related to the total energy consumption: it might be a measure of fuel consumption, or a sum of the modules of all difference in velocities, or other measures of the energetic cost of the mission.

The problem, even in its most basic formulation without deep space maneuvers and swing-bys, is highly multimodal, as can be seen even from a very simple example in two variables, like the one depicted in Figure 6.7.

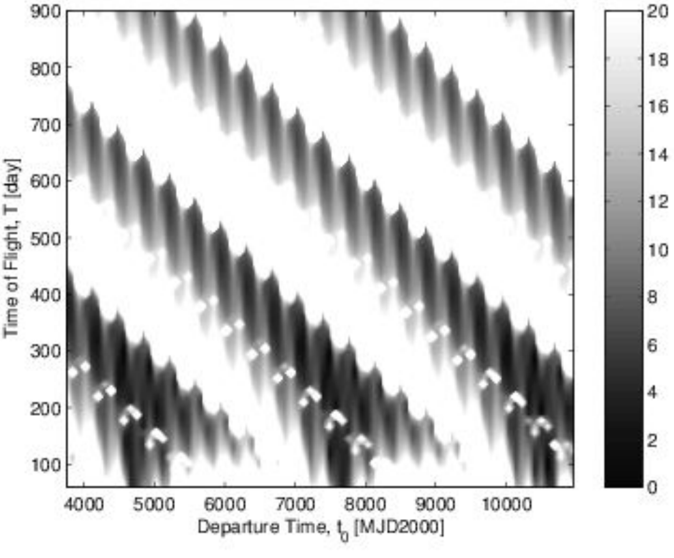


Figure 6.7. *Level sets of a two-variable space planning problem for an Earth–Apophis transfer. The plot represents level sets of the function which to each possible departure date and travel time associates the absolute variation between initial and final velocity vectors.*

The origin of multimodality is mainly due to the many periodicities which characterize the problem, generating many solutions with similar energy consumption and widely different variable values (in particular, those associated with departure and landing dates). In (Myatt, Becerra, Nasuto, & Bishop, 2004), as an example, several plots are displayed of the level sets of a simple Earth–Mars transfer, a problem with just two variables associated, respectively, to the start and the arrival dates. In that report, as well as in many other independent studies on the problem, the presence of a huge number of local optima is always observed.

Formulations of these planning problems are basically of two types: analytic formulations, derived from the application of gravitational laws and of the fundamental laws of motion (see, for a survey of these models, (Cassioli, Di Lorenzo, Locatelli, & Schoen, 2012)), or "black boxes," in which the user specifies values of the variables and a set of numerical methods is then used to compute the trajectory parameters and to return the fuel consumption, or any other chosen objective function. As already mentioned in Section 6.1 devoted to test problems in GO, the ESA-ACT team has made available several of these black boxes. Here we simply comment on the fact that for both kind of formulations, either analytical or implicit (hidden in a black box), the most successful computational approaches belong to the family of heuristics introduced in this volume. In particular, differential evolution (see Section 3.1.6) and BH (see Section 3.1.3) are the most widely used approaches. When the problem is in the form of a black box, some ingenuity has to be used in order to profit from the capabilities of local solvers within a GO scheme. In particular, analytical derivatives are not available from the black box, so that either derivative-free methods have to be employed or the necessity arises of resorting to numerical differentiation. In our experience, this last approach, for black-box space trajectory design, seems to be particularly attractive. As has been reported in (Addis et al., 2011), the use of difference estimates for the objective function in the black boxes provided by ESA displayed very good performance when the numerical estimates were obtained using a large step δ in the derivative approximation formula:

$$f'(x) \approx \frac{f(x+\delta)-f(x)}{\delta}.$$

This large step approximation of the derivative, inspired by Kelley's *filter method* (Kelley, 2011), has a smoothing effect on the highly oscillating landscape of the objective function of the problem. Different from what happens in local optimization, where an accurate estimate of the gradient is necessary in order to be able to build sufficiently accurate local models of the objective function, in GO there is no necessity to approximate, at high precision each local optimum location. Instead, by using a large step in numerical differentiation, in some sense the overall picture of the objective function is captured and even local optimization algorithms display a tendency of jumping over local optima with small region of attraction.

We end this section observing that these problems are quite different from the ones presented in this chapter: most of the previous examples were quite easy to formulate, yet very hard to solve. Here, similar to what happens, e.g., in protein folding, models are quite refined and several competing models exist, each capturing some relevant part of the problem. This, in addition to the quite intricate formulae required, is why we do not explicitly give a formulation of the problem in this section. Moreover, in space mission design, modern approaches tend to consider an even more challenging problem, i.e., that of designing an efficient, yet *robust*, trajectory. By this we mean a trajectory which does not deteriorate too much if some variables are changed by a small amount. As an example, variables associated to the dates of departure, arrival, and swing-bys are subject to last-minute modifications, caused by atmospheric problems or unpredictable mechanical or communication difficulties. Robust optimization is a fascinating subject in local optimization which received only limited attention in the GO community. Space trajectory design might turn out to be a challenging testbed for robust GO methods. Somehow related to this issue is also the fact that for these problems it usually makes sense not only to search for the globally optimal solution but also for a whole set of good locally optimal solutions, in order to have a selection of different reasonable options among which the mission designer can choose.

Appendix A
Basic Definitions and Results on Convexity

A.1 Convex sets

A subset X of the n-dimensional Euclidean space is convex if it contains all the segments whose extreme points are distinct points in X. More formally, we have the following definition.

Definition A.1. *A set X is said to be* convex *if*

$$\mathbf{x}_1,\ \mathbf{x}_2 \in X,\ \ \lambda \in (0,1) \quad \Rightarrow \quad \lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2 \in X.$$

Special convex sets are polyhedra.

Definition A.2. *A set P defined by a finite number of linear equalities and/or inequalities, i.e.,*

$$P = \{\mathbf{x} \in \mathbb{R}^n \ :\ \mathbf{A}_1\mathbf{x} \leq \mathbf{b}_1,\ \mathbf{A}_2\mathbf{x} = \mathbf{b}_2\}$$

is called a polyhedron. *If the polyhedron is a bounded set, it is called a* polytope.

A special polytope is the unit simplex.

Definition A.3. *The polytope*

$$\Delta_n = \{\mathbf{x} \in \mathbb{R}^n \ :\ \mathbf{e}^T\mathbf{x} = 1,\ \mathbf{x} \geq \mathbf{0}\}$$

is called n-dimensional unit simplex.

Given some set $Z \subseteq \mathbb{R}^n$, the tightest convex set containing Z is called the convex hull of Z.

Definition A.4. *The* convex hull *of a set $Z \subseteq \mathbb{R}^n$, denoted by $chull(Z)$, is the smallest convex set containing Z, i.e., $chull(Z) \supseteq Z$ is a convex set, and for any convex set C such that $C \supseteq Z$, $chull(Z) \subseteq C$.*

It is worthwhile to recall here Carathéodory's theorem (see, e.g., (Rockafellar, 1970)).

Theorem A.5. *Let $Z \subseteq \mathbb{R}^n$. If $\mathbf{x} \in chull(Z)$, then $\mathbf{x}$ is equal to a convex combination of (at most) $n+1$ points in Z, i.e., there exist $\mathbf{x}_1,\ldots,\mathbf{x}_{n+1} \in Z$ such that*

$$\mathbf{x} = \sum_{i=1}^{n+1} \lambda_i \mathbf{x}_i,$$

where

$$\sum_{i=1}^{n+1} \lambda_i = 1, \quad \lambda_i \geq 0, \;\; i = 1,\ldots,n+1.$$

The tightest affine set containing a set Z is called affine hull of Z.

Definition A.6. *For a given set $Z \subseteq \mathbb{R}^n$, the* affine hull *of Z is denoted by $ahull(Z)$ and is defined as follows*

$$\begin{aligned} ahull(Z) \;\; = \;\; & \{\mathbf{x} \in \mathbb{R}^n \; : \\ & \mathbf{x} = \textstyle\sum_{i=1}^{r} \lambda_i \mathbf{z}_i, \; \mathbf{z}_i \in Z, \;\; i = 1,\ldots,r, \; \sum_{i=1}^{r} \lambda_i = 1 \; r \in \mathbb{N}\}. \end{aligned}$$

Next, we define some special points within a set X.

Definition A.7. *Given a set $X \subseteq \mathbb{R}^n$, an* extreme point *of X is a point $\mathbf{x} \in X$ such that there do not exist two points $\mathbf{x}_1, \mathbf{x}_2 \in X$, $\mathbf{x}_1 \neq \mathbf{x}_2$, such that*

$$\mathbf{x} = \frac{1}{2}\mathbf{x}_1 + \frac{1}{2}\mathbf{x}_2.$$

The set of extreme points of X is denoted by $Ext[X]$.

Special hyperplanes for a convex set are so called supporting hyperplanes at extreme points of the set.

Definition A.8. *Given a convex set $X \subseteq \mathbb{R}^n$ and a point $\mathbf{x} \in Ext[X]$, a* supporting hyperplane *of X at $\mathbf{x}$ is a hyperplane $\mathbf{a}^T\mathbf{y} = a_0$ such that*

- $\mathbf{a}^T\mathbf{x} = a_0$ *(i.e., $\mathbf{x}$ belongs to the hyperplane);*
- $\mathbf{a}^T\mathbf{z} \leq a_0 \; \forall \; \mathbf{z} \in X$.

Next we introduce a subset of the extreme points.

Definition A.9. *Given a convex set $X \subseteq \mathbb{R}^n$, a* vertex *of X is an extreme point $\mathbf{v}$ such that there exist n linearly independent supporting hyperplanes of X at $\mathbf{v}$. The set of vertices of X is denoted by $V(X)$.*

In case X is a polyhedron, $Ext[X] = V(X)$ and the cardinality of $V(X)$ is finite.

Definition A.10. *Given the m vectors $\mathbf{v}_1, \dots, \mathbf{v}_m \in \mathbf{R}^n$, we say that they are* affinely independent *if the vectors $\mathbf{v}_2 - \mathbf{v}_1, \dots, \mathbf{v}_m - \mathbf{v}_1$ are linearly independent.*

The dimension of a convex set X is defined as follows.

Definition A.11. *A convex set X has dimension d if the maximum number of affinely independent points in X is equal to $d+1$.*

Relevant convex sets are the convex cones.

Definition A.12. *A subset C of some vector space is a cone if*

$$\mathbf{x}_1,\ \mathbf{x}_2 \in C, \quad \lambda_1, \lambda_2 \geq 0 \quad \Rightarrow \quad \lambda_1 \mathbf{x}_1 + \lambda_2 \mathbf{x}_2 \in C.$$

Given a convex set, we can associate to it a convex cone, called a recession cone, whose definition is given below.

Definition A.13. *For a given convex set $X \subseteq \mathbb{R}^n$,*

$$0^+(X) = \{\mathbf{d} \in \mathbb{R}^n \ :\ \mathbf{x} + \lambda \mathbf{d} \in X,\ \ \forall\, \mathbf{x} \in X,\ \forall\, \lambda \geq 0\}$$

is the recession cone *for X. Note that X is a bounded set if and only if $0^+(X) = \{\mathbf{0}\}$.*

A well-known convex cone is the nonnegative orthant $\mathbb{R}^n_+$. Another important convex cone is the cone of semidefinite matrices, whose definition is given in what follows.

Definition A.14. *Let $\mathcal{S}_n$ denote the space of the symmetric matrices of order n. The space $\mathcal{P}_n \subset \mathcal{S}_n$ of* positive semidefinite *matrices of order n is composed by all the symmetric matrices $\mathbf{M}$ such that*

$$\mathbf{x}^T \mathbf{M} \mathbf{x} \geq 0 \quad \forall\, \mathbf{x} \in \mathbb{R}^n.$$

We also write $\mathbf{M} \succeq \mathbf{O}$ to denote that $\mathbf{M}$ is positive semidefinite. If

$$\mathbf{x}^T \mathbf{M} \mathbf{x} > 0 \quad \forall\, \mathbf{x} \in \mathbb{R}^n \setminus \{0\},$$

then $\mathbf{M}$ is called positive definite *and is also denoted as $\mathbf{M} \succ \mathbf{O}$. If $-\mathbf{M}$ is positive semidefinite (respectively, positive definite), then $\mathbf{M}$ is called* negative semidefinite *(respectively,* negative definite*).*

Recall that a matrix is positive semidefinite (definite) if and only if all of its eigenvalues are nonnegative (strictly positive). Similarly, a matrix is negative semidefinite (definite) if and only if its eigenvalues are all nonpositive (strictly negative). For two matrices $\mathbf{A}, \mathbf{B}$ the notation $\mathbf{A} \succeq\ (\succ)\ \mathbf{B}$ stands for $\mathbf{A} - \mathbf{B} \succeq\ (\succ)\ \mathbf{O}$.

Definition A.15. *Let $\mathbf{X}$ be a symmetric matrix with the structure*

$$\mathbf{X} = \begin{pmatrix} \mathbf{X}_1 & \mathbf{X}_2 \\ \mathbf{X}_2^T & \mathbf{X}_3 \end{pmatrix}, \tag{A.1}$$

where $\mathbf{X}_1, \mathbf{X}_3$ *are square matrices and* $\mathbf{X}_1$ *is invertible. Then, the* Schur complement *of* $\mathbf{X}_1$ *in* $\mathbf{X}$ *is the matrix*

$$\mathbf{X}_3 - \mathbf{X}_2^T \mathbf{X}_1^{-1} \mathbf{X}_2.$$

We have the following result.

Observation A.1. *Let* $\mathbf{X}$ *be defined as in (A.1). If* $\mathbf{X}_1$ *is positive definite, then* $\mathbf{X}$ *is positive semidefinite (definite) if and only if its Schur complement is positive semidefinite (definite).*

Now let us define an inner product between matrices. First we introduce the notion of trace for a square matrix.

Definition A.16. *Given a square matrix* $\mathbf{A} \in \mathbb{R}^{n \times n}$*, its* trace *is the sum of its diagonal elements, i.e.,*

$$tr(\mathbf{A}) = \sum_{i=1}^{n} A_{ii}.$$

Definition A.17. *Given two matrices* $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$*, their* Frobenius inner product *is defined as*

$$\mathbf{A} \bullet \mathbf{B} = tr(\mathbf{AB}) = \sum_{i,j=1}^{n} A_{ij} B_{ij}.$$

Now we are ready to introduce the notion of dual cone.

Definition A.18. *Given a cone* $\mathcal{K}$ *and an inner product* $\cdot$ *between members of* $\mathcal{K}$*, the cone*

$$\mathcal{K}^* = \{\mathbf{x} \ : \ \mathbf{x} \cdot \mathbf{y} \geq 0 \quad \forall\, \mathbf{y} \in \mathcal{K}\},$$

is called the dual cone *of* $\mathcal{K}$.

We have that

$$\mathbb{R}_+^{n\,*} = \mathbb{R}_+^n, \quad \mathcal{P}_n^* = \mathcal{P}_n,$$

i.e., both the nonnegative orthant and the cone of semidefinite matrices are self-dual. For the nonnegative orthant the inner product is the standard one between vectors, while for the semidefinite cone the inner product is the Frobenius one. A further self-dual cone is that of the nonnegative matrices.

Definition A.19. *The convex cone of n-dimensional* nonnegative matrices *is made up by all the symmetric matrices whose entries are nonnegative, i.e.,*

$$\mathcal{N}_n = \{\mathbf{A} \in \mathcal{S}_n \ : \ \mathbf{A} \geq \mathbf{O}\}.$$

A further cone is the intersection of the cones of semidefinite and nonnegative matrices.

Definition A.20. *The cone*

$$\mathcal{DNN}_n = \mathcal{P}_n \cap \mathcal{N}_n$$

is called the cone of doubly nonnegative matrices.

The dual cone $\mathcal{DNN}_n^*$ of $\mathcal{DNN}_n$ is the cone $\mathcal{P}_n + \mathcal{N}_n$. Another convex cone which is not self-dual is that of the copositive matrices.

Definition A.21. *The convex cone of n-dimensional* copositive matrices *is made up of all the symmetric matrices defining quadratic forms which are nonnegative over the nonnegative orthant, i.e.,*

$$\mathcal{C}_n = \{\mathbf{A} \in \mathcal{S}_n \ : \ \mathbf{x}^T\mathbf{A}\mathbf{x} \geq 0 \quad \forall\, \mathbf{x} \in \mathbb{R}^n_+\}.$$

Its dual cone is the cone of completely positive matrices.

Definition A.22. *The cone*

$$\mathcal{C}_n^* = \left\{\mathbf{A} \in \mathcal{S}_n \ : \ \mathbf{A} = \sum_{k=1}^{K} \mathbf{x}_k\mathbf{x}_k^T, \ \ \mathbf{x}_k \in \mathbb{R}^n_+, \ K \in \mathbb{N}\right\}$$

is called the cone of completely positive matrices.

Finally, we introduce the second-order cone.

Definition A.23. *The* second-order cone *in* $\mathbb{R}^n$ *is*

$$\mathcal{SOC}_n = \left\{\mathbf{x} \in \mathbb{R}^n \ : \ \sqrt{x_2^2 + \cdots + x_n^2} \leq x_1\right\}.$$

A.2 Convex and concave functions

We first introduce the definitions of convex and concave functions.

Definition A.24. *A function f is said to be a* convex function *over a convex set X if*

$$\forall\, \lambda \in [0,1], \quad \forall\, \mathbf{x},\mathbf{y} \in X \ : \ f(\lambda\mathbf{x} + (1-\lambda)\mathbf{y}) \leq \lambda f(\mathbf{x}) + (1-\lambda) f(\mathbf{y}).$$

If we have strict inequality for any $\lambda \in (0,1)$ and $\mathbf{x} \neq \mathbf{y}$, then we call the function a strictly convex function. *The function f is a* concave function *if*

$$\forall\, \lambda \in [0,1], \quad \forall\, \mathbf{x},\mathbf{y} \in X \ : \ f(\lambda\mathbf{x} + (1-\lambda)\mathbf{y}) \geq \lambda f(\mathbf{x}) + (1-\lambda) f(\mathbf{y}).$$

If we have strict inequality for any $\lambda \in (0,1)$ and $\mathbf{x} \neq \mathbf{y}$, then we call the function a strictly concave function.

For differentiable functions we have the following equivalent definition.

Observation A.2. *A continuously differentiable function f is convex over X if and only if*

$$\forall\, \mathbf{x}, \mathbf{y} \in X, \quad f(\mathbf{x}) \geq f(\mathbf{y}) + \nabla f(\mathbf{y})(\mathbf{x} - \mathbf{y}).$$

If we have strict inequality for $\mathbf{x} \neq \mathbf{y}$, then the function is strictly convex. For concave and strictly concave functions we have analogous results by simply reversing the inequality.

Finally, for twice-differentiable functions, we have a further equivalent definition.

Observation A.3. *A twice-continuously differentiable function f is convex over X if and only if its Hessian is positive semidefinite over X, i.e.,*

$$\forall\, \mathbf{x} \in X, \ : \ \nabla^2 f(\mathbf{x}) \in \mathcal{P}_n.$$

If the Hessian is positive definite over X, then the function is strictly convex (note that the reverse is not true, as the function $f(x) = x^4$ shows). For concave and strictly concave functions we have analogous results by simply substituting the requirements of semidefinite and definite positiveness with those of semidefinite and definite negativeness, respectively.

In terms of sets we can also say that a function f is convex over X if and only if its epigraph is a convex set.

Definition A.25. *The* epigraph *of a function f over $X \subseteq \mathbb{R}^n$ is denoted by $epi_X[f]$ and defined as*

$$epi_X[f] = \{(\mathbf{x}\ \mu) \in \mathbb{R}^{n+1} \ : \ \mu \geq f(\mathbf{x}),\ \mathbf{x} \in X\}.$$

Now we introduce the definition of subgradient of a convex function at some point.

Definition A.26. *Let f be a convex function over a convex set X. The* subgradient *of f at $\mathbf{x} \in X$ is a set, denoted by $\partial f(\mathbf{x})$, such that*

$$\forall\, \mathbf{s} \in \partial f(\mathbf{x}),\ \forall\, \mathbf{y} \in X \ : \ f(\mathbf{y}) \geq f(\mathbf{x}) + \mathbf{s}(\mathbf{y} - \mathbf{x}).$$

In other words, the vectors in $\partial f(\mathbf{x})$ identify all the possible supporting hyperplanes of $epi_X[f]$ at the point $(\mathbf{x}\ f(\mathbf{x}))$.

Notice that for differentiable functions $\partial f(\mathbf{x})$ is a singleton containing only $\nabla f(\mathbf{x})$.

Many results about convex and concave functions exist (we refer, e.g., to (Rockafellar, 1970)). Here we only mention a couple of well-known results.

Observation A.4. *Let f_1, f_2 be two convex (concave) functions over a convex set X and let $\alpha_1, \alpha_2 \geq 0$. Then, the function $\alpha_1 f_1 + \alpha_2 f_2$ is also convex (concave) over X.*

Observation A.5. *Let I be a set of indices (possibly of infinite cardinality) and f_i, $i \in I$, be convex functions over X. Then, the pointwise supremum of all these functions, i.e., the function*

$$f(\mathbf{x}) = \sup_{i \in I} f_i(\mathbf{x}),$$

is convex over X. Similarly, if f_i, $i \in I$, are concave functions over X, then the pointwise infimum of all these functions, i.e., the function

$$f(\mathbf{x}) = \inf_{i \in I} f_i(\mathbf{x}),$$

is a concave function.

A more general notion is that of quasiconvex and quasiconcave functions.

Definition A.27. *A function f is said to be a* quasiconvex function *over a convex set X if*

$$\forall\, \lambda \in [0,1], \quad \forall\, \mathbf{x}, \mathbf{y} \in X \;:\; f(\lambda\mathbf{x} + (1-\lambda)\mathbf{y}) \le \max\{f(\mathbf{x}), f(\mathbf{y})\}.$$

A function f is said to be a quasiconcave function *over a convex set X if*

$$\forall\, \lambda \in [0,1], \quad \forall\, \mathbf{x}, \mathbf{y} \in X \;:\; f(\lambda\mathbf{x} + (1-\lambda)\mathbf{y}) \ge \min\{f(\mathbf{x}), f(\mathbf{y})\}.$$

Obviously, any convex (concave) function is also quasiconvex (quasiconcave).

Appendix B
Notation

For ease of reference, we report here a list of some frequently used mathematical symbols. We also point out that, throughout the book, vectors are denoted in bold letters, while matrices are denoted in bold capital letters.

$\mathbb{R}$: the set of real numbers;

$\mathbb{N}_0$: the set of nonnegative integers;

$\mathbb{N}$: the set of positive integers;

$\mathbb{Z}$: the set of integers;

$\mathbb{Q}$: the set of rational numbers;

$\mathbb{R}^n$: the n-dimensional Euclidean space;

$\mathbb{R}^n_+$: the n-dimensional nonnegative orthant;

$\mathbf{e}$: the vector whose components are all equal to 1 (its dimension is usually clear from the context);

$\mathbf{E}$: the matrix whose entries are all equal to 1 (dimension clear from the context);

$\mathbf{0}$: the null vector (dimension clear from the context);

$\mathbf{O}$: the null matrix (dimension clear from the context);

$diag(\mathbf{A})$: for a given square matrix $\mathbf{A}$, this is the vector whose components are equal to the diagonal elements of $\mathbf{A}$;

$Diag(\mathbf{v})$: for a given vector $\mathbf{v}$, this is the diagonal matrix whose diagonal elements are equal to the components of $\mathbf{v}$;

$tr(\mathbf{A})$: for a square matrix $\mathbf{A}$, tr is the trace of the matrix $\mathbf{A}$;

$rank(\mathbf{A})$: the rank of a matrix $\mathbf{A}$, i.e., the maximum number of linear independent rows (or columns) of the matrix;

$det(\mathbf{A})$: the determinant of a square matrix $\mathbf{A}$;

$\mathcal{S}_n$: the set of the symmetric matrices of order n;

$\mathcal{P}_n$: the set of the semidefinite matrices of order n;

$\mathcal{N}_n$: the set of the nonnegative matrices of order n;

$\mathcal{C}_n$: the set of the copositive matrices of order n;

$\mathcal{C}_n^*$: the set of the completely positive matrices of order n;

Δ_n: the n-dimensional unit simplex;

$\mathbb{R}_m[\mathbf{x}]$, $\mathbf{x} = (x_1, \ldots, x_n)$: the set of polynomials in n variables of degree at most m;

g', g'': first and second derivative for a function g defined over $\mathbb{R}$;

$\frac{\partial f}{\partial x_i}$, $i = 1, \ldots, n$: the partial derivative of a function f defined over $\mathbb{R}^n$ with respect to x_i;

$\nabla f \in \mathbb{R}^n$: the gradient of a function f defined over $\mathbb{R}^n$;

$\nabla^2 f \in \mathbb{R}^{n \times n}$: the Hessian of a function f defined over $\mathbb{R}^n$;

$\mathcal{C}^k(X)$: the set of k-times continuously differentiable functions over a set X;

$card(X)$: the cardinality of a finite set X;

$int(X)$: the interior of a set X;

$ri(X)$: the relative interior of a set X;

$cl(X)$: the closure of a set X;

$bd(X)$: the boundary of a set X;

$proj_{\mathbf{x}}(X)$: the projection of set X over the space of the $\mathbf{x}$ variables;

$epi_X[f]$: the epigraph of a function f over a set X;

$0^+(X)$: the recession cone of a convex set X;

$Ext[X]$: the set of extreme points of a convex set X;

$V(X)$: the set of vertices of a convex set X;

$chull(X)$: the convex hull of a set X;

$ahull(X)$: the affine hull of a set X;

$diam(X)$: the diameter of a set X;

$\|\mathbf{x}\|$: a generic norm for some vector $\mathbf{x}$;

$\|\mathbf{x}\|_2$: the Euclidean norm for some vector $\mathbf{x}$.

References

Addis, B., Cassioli, A., Locatelli, M., & Schoen, F. (2011). A global optimization method for the design of space trajectories. *Computational Optimization and Applications*, *48*, 635–652. (Cited on pp. 63, 386, 388)

Addis, B., & Leyffer, S. (2006). A trust-region algorithm for global optimization. *Computational Optimization and Applications*, *35*(3), 287–304. (Cited on p. 81)

Addis, B., & Locatelli, M. (2007). A new class of test functions for global optimization. *Journal of Global Optimization*, *38*(3), 479–501. (Cited on p. 370)

Addis, B., Locatelli, M., & Schoen, F. (2005). Local optima smoothing for global optimization. *Optimization Methods and Software*, *20*(4–5), 417–437. (Cited on p. 81)

Addis, B., Locatelli, M., & Schoen, F. (2008a). Disk packing in a square: A new global optimization approach. *INFORMS Journal on Computing*, *20*(4), 516–524. (Cited on pp. 62, 63, 383, 385)

Addis, B., Locatelli, M., & Schoen, F. (2008b). Efficiently packing unequal disks in a circle. *Operations Research Letters*, *36*(1), 37–42. doi: 10.1016/j.orl.2007.03.001 (Cited on pp. 383, 384, 385)

Adjiman, C. S., Androulakis, I. P., & Floudas, C. A. (1998). A global optimization method, αBB, for general twice-differentiable constrained NLPs - II. Implementation and computational results. *Computers and Chemical Engineering*, *22*, 1159–1179. (Cited on p. 230)

Adjiman, C. S., Androulakis, I. P., Maranas, C. D., & Floudas, C. A. (1996). A global optimization method, αBB, for process design. *Computers and Chemical Engineering Supplement*, *20*, S419–S424. (Cited on p. 230)

Adjiman, C. S., Dallwig, S., Floudas, C. A., & Neumaier, A. (1998). A global optimization method, αBB, for general twice differentiable NLPs- I. Theoretical advances. *Computers and Chemical Engineering*, *22*, 1137–1158. (Cited on p. 230)

Akrotirianakis, I. G., & Floudas, C. A. (2004). A new class of improved convex underestimators for twice continuously differentiable constrained NLPs. *Journal of Global Optimization*, *30*, 367–390. (Cited on pp. 230, 239, 240)

Al-Khayyal, F. A., & Sherali, H. D. (2000). On finitely terminating branch-and-bound algorithms for some global optimization problems. *SIAM Journal on Optimization*, *10*, 1049–1057. (Cited on pp. 359, 360)

Allgower, E. L., & Georg, K. (2003). *Introduction to numerical continuation methods*. SIAM, Philadelphia. (Cited on p. 83)

Anstreicher, K. M. (2009). Semidefinite programming versus the reformulation-linearization technique for nonconvex quadratically constrained quadratic programming. *Journal of Global Optimization*, *43*, 471–484. (Cited on pp. 170, 171, 212)

Anstreicher, K. M. (2012). On convex relaxations for quadratically constrained quadratic programming. *Mathematical Programming*, *136*, 233–251. (Cited on pp. 212, 213)

Anstreicher, K. M., & Burer, S. (2005). D.C. versus copositive bounds for standard QP. *Journal of Global Optimization*, *33*, 299–312. (Cited on pp. 247, 248)

Anstreicher, K. M., & Burer, S. (2010). Computable representations for convex hulls of low-dimensional quadratic forms. *Mathematical Programming B*, *124*, 33–43. (Cited on pp. 161, 210)

Arnautova, Y. A., Jagielska, A., & Scheraga, H. A. (2006). A new force field (ecepp-05) for peptides, proteins, and organic molecules. *The Journal of Physical Chemistry B*, *110*(10), 5025–5044. (Cited on p. 372)

Arora, S., & Barak, B. (2009). *Computational complexity: A modern approach*. Cambridge University Press. (Cited on p. 8)

Audet, C., Hansen, P., Jaumard, B., & Savard, G. (1999). A symmetrical linear maxmin approach to disjoint bilinear programming. *Mathematical Programming*, *85*, 573–592. (Cited on pp. 198, 199, 360)

Audet, C., Hansen, P., Jaumard, B., & Savard, G. (2000). A branch and cut algorithm for nonconvex quadratically constrained quadratic programming. *Mathematical Programming*, *87*, 131–152. (Cited on pp. 163, 167, 212)

Balas, E. (1971). Intersection cuts - a new type of cutting planes for integer programming. *Operations Research*, *19*, 19–39. (Cited on pp. 309, 314)

Balas, E. (1998). Disjunctive programming: properties of the convex hull of feasible points. *Discrete Applied Mathematics*, *89*, 3–44. (Cited on p. 276)

Bali, S. (1973). *Minimization of a concave function on a bounded convex polyhedron*. Unpublished doctoral dissertation, University of California, Los Angeles, CA. (Cited on pp. 299, 309)

Ban, V. T. (1983). A finite algorithm for minimizing a concave function under linear constraints and its application. In (Vols. Proceedings IFIP Working Conference on Recent Adavances on System Modeling and Optimization, Hanoi). (Cited on p. 302)

Banks, A., Vincent, J., & Anyakoha, C. (2007). A review of particle swarm optimization. Part I: background and development. *Natural Computing*, *6*(4), 467–484. doi: 10.1007/s11047-007-9049-5 (Cited on p. 63)

Banks, A., Vincent, J., & Anyakoha, C. (2008). A review of particle swarm optimization. Part II: hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications. *Natural Computing*, *7*(1), 109–124. doi: 10.1007/s11047-007-9050-z (Cited on p. 63)

Bao, X., Sahinidis, N. V., & Tawarmalani, M. (2009). Multiterm polyhedral relaxations for nonconvex, quadratically constrained quadratic programs. *Optimization Methods and Software*, *24*(4), 485–504. (Cited on p. 211)

Bao, X., Sahinidis, N. V., & Tawarmalani, M. (2011). Semidefinite relaxations for quadratically constrained quadratic programming: A review and comparisons. *Mathematical Programming*, *129*, 129–157. (Cited on p. 217)

Baritompa, W., & Stephens, C. (1998). Global optimization requires global information. *Journal of Optimization Theory and Applications*, *9*, 575–588. (Cited on pp. 119, 122, 123)

Barrientos, O., & Correa, R. (2000). An algorithm for global minimization of linearly constrained quadratic functions. *Journal of Global Optimization*, *16*, 77–93. (Cited on p. 264)

Barvinok, A. I. (2007). Integration and optimization of multivariate polynomials by restriction onto a random subspace. *Foundations of Computational Mathematics*, *7*, 229–244. (Cited on p. 26)

Baumann, E. (1988). Optimal centered forms. *BIT*, *28*, 80–87. (Cited on pp. 257, 258)

Beck, A., Ben-Tal, A., & Teboulle, M. (2006). Finding a global optimal solution for a quadratically constrained fractional quadratic problem with applications to the regularized total least squares. *SIAM Journal on Matrix Analysis and Applications*, *28*, 425–445. (Cited on p. 14)

Beck, A., & Teboulle, M. (2009). A convex optimization approach for minimizing the ratio of indefinite quadratic functions over an ellipsoid. *Mathematical Programming*, *118*, 13–35. (Cited on pp. 12, 13, 14, 16)

Bellare, M., Goldwasser, S., Lund, C., & Russell, A. (1993). Efficient probabilistically checkable proofs and applications to approximation. In *Proceedings of the twenty fifth annual symposium on the theory of computing, ACM.* (Cited on p. 29)

Bellare, M., & Rogaway, P. (1995). The complexity of approximating a nonlinear program. *Mathematical Programming*, *69*, 429–441. (Cited on p. 29)

Belotti, P., Lee, J., Liberti, L., Margot, F., & Wächther, A. (2009). Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods and Software*, *24*, 597–634. (Cited on pp. 270, 271, 346)

Benson, H. P. (1982). On the convergence of two branch-and-bound algorithms for non-convex programming problems. *Journal of Optimization Theory and Applications*, *36*, 129–134. (Cited on p. 302)

Benson, H. P. (1985). A finite algorithm for concave minimization over a polyhedron. *Naval Research Logistics Quarterly*, *32*, 165–177. (Cited on p. 302)

Benson, H. P. (2004). On the construction of convex and concave envelope formulas for bilinear and fractional functions on quadrilaterals. *Computational Optimization and Applications*, *27*, 5–22. (Cited on pp. 161, 323)

Benson, H. P. (2010). Simplicial branch-and-reduce algorithm for convex programs with a multiplicative constraint. *Journal of Optimization Theory and Applications*, *145*, 213–233. (Cited on p. 346)

Benson, H. P., & Sayin, S. (1994). A finite concave minimization algorithm using branch and bound and neighbor generation. *Journal of Global Optimization*, *5*, 1–14. (Cited on pp. 302, 361)

Ben-Tal, A., Eiger, G., & Gershovitz, V. (1994). Global minimization by reducing the duality gap. *Mathematical Programming*, *63*, 193–212. (Cited on pp. 263, 264)

Ben-Tal, A., & Nemirovski, A. (2001). *Lectures on modern convex optimization: analysis, algorithms, and engineering applications.* MOS-SIAM Series on Optimization 2. (Cited on p. 9)

Ben-Tal, A., & Teboulle, M. (1996). Hidden convexity in some nonconvex quadratically constrained quadratic programming. *Mathematical Programming*, *72*, 51–63. (Cited on p. 267)

Bergamini, M. L., Grossmann, I. E., Scenna, N., & Aguirre, P. A. (2008). An improved piecewise outer-approximation algorithm for the global optimization of MINLP models involving concave and bilinear terms. *Computers and Chemical Engineering*, *32*, 477–493. (Cited on p. 287)

Berman, A., & Xu, C. (2004). 5 x 5 completely positive matrices. *Linear Algebra Appl.*, *393*, 55–71. (Cited on p. 185)

Bigi, G., Frangioni, A., & Zhang, Q. (2010). Outer approximation algorithms for canonical DC problems. *Journal of Global Optimization*, *46*, 163–189. (Cited on p. 243)

Birgin, E. G., Floudas, C. A., & Martinez, J. M. (2010). Global minimization using an augmented Lagrangian method with variable lower-level constraints. *Mathematical Programming*, *125*, 139–162. (Cited on p. 266)

Biswas, P., Ye, Y., Wang, T.-C., & Liang, T.-C. (2006). Semidefinite programming approaches for sensor network localization with noisy distance measurements. *IEEE Transactions on Automation Science and Engineering*, *3*(4), 360–371. (Cited on p. 380)

Blanquero, R., Carrizosa, E., & Hansen, P. (2009). Locating objects in the plane using global optimization techniques. *Mathematics of Operations Research*, *34*, 837–858. (Cited on p. 371)

Blekherman, G. (2006). There are significantly more nonnegative polynomials than sums of squares. *Israel Journal of Mathematics*, *153*, 355–380. (Cited on p. 221)

Bodlaender, H., Gritzmann, P., Klee, V., & van Leeuwen, J. (1990). Computational complexity of norm-maximization. *Combinatorica*, *10*, 203–225. (Cited on p. 38)

Boender, C. G. E., Caron, R. J., Mcdonald, J. F., Rinnooy Kan, A. H. G., Romeijn, H. E., Smith, R. L., … Vorst, A. C. F. (1991). Shake-and-bake algorithms for generating uniform points on the boundary of bounded polyhedra. *Operations Research*, *39*(6), 945–954. (Cited on p. 50)

Boll, D. V., Donovan, J., Graham, R. L., & Lubachevsky, B. D. (2000). Improving dense packings of equal disks in a square. *The Electronic Journal of Combinatorics*, *7*(R46), 1–9. (Cited on p. 383)

Bompadre, A., & Mitsos, A. (2012). Convergence rate of McCormick relaxations. *Journal of Global Optimization*, *52*, 1–28. (Cited on p. 270)

Bomze, I. M. (2002). Branch-and-bound approaches to standard quadratic optimization problems. *Journal of Global Optimization*, *22*, 17–37. (Cited on p. 246)

Bomze, I. M. (2011). Copositive optimization - recent developments and applications. *European Journal of Operations Research*, *216*, 509–520. (Cited on p. 173)

Bomze, I. M., & de Klerk, E. (2002). Solving standard quadratic optimization problems via linear, semidefinite and copositive programming. *Journal of Global Optimization*, *24*, 163–185. (Cited on pp. 26, 27, 28)

Bomze, I. M., Duer, M., de Klerk, E., Roos, C., Quist, A., & Terlaky, T. (2000). On copositive programming and standard quadratic optimization problems. *Journal of Global Optimization*, *18*, 301–320. (Cited on pp. 26, 173, 174)

Bomze, I. M., Frommlet, F., & Locatelli, M. (2010). Copositive bounds for improving SDP bounds on the clique number. *Mathematical Programming B*, *124*, 13–32. (Cited on p. 184)

Bomze, I. M., & Jarre, F. (2010). A note on Burer's copositive representation of mixed-binary QPs. *Optimization Letters*, *4*, 465–472. (Cited on p. 181)

Bomze, I. M., & Locatelli, M. (2004). Undominated DC Decompositions of Quadratic Functions and Applications to Branch-and-Bound Approaches. *Computational Optimization and Applications*, *28*(2), 227–245. (Cited on pp. 243, 244, 246)

Bomze, I. M., & Locatelli, M. (2012). Separable standard quadratic optimization problems. *Optimization Letters*, *6*, 857–866. (Cited on pp. 16, 23)

Bomze, I. M., Locatelli, M., & Tardella, F. (2008). New and old bounds for standard quadratic optimization: dominance, equivalence and incomparability. *Mathematical Programming*, *115*(1), 31–64. (Cited on p. 249)

Bomze, I. M., Schachinger, W., & Uchida, G. (2012). Think co(mpletely)positive! Matrix properties, examples and a clustered bibliography on copositive optimization. *Journal of Global Optimization*, *52*, 423–445. (Cited on p. 173)

Boros, E., & Hammer, P. L. (1993). Cut-polytopes, boolean quadric polytopes and nonnegative quadratic pseudo-boolean functions. *Mathematics of Operations Research*, *18*, 245–253. (Cited on p. 207)

Borradaile, G., & Van Hentenryck, P. (2005). Safe and tight linear estimators for global optimization. *Mathematical Programming*, *102*, 495–517. (Cited on p. 287)

Boyd, S. P., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press. Retrieved from `http://www.stanford.edu/~boyd/cvxbook/`. (Cited on p. 338)

Bricker, D. L. (1980). Bounding a class of nonconvex linearly-constrained resource allocation problems via the surrogate dual. *Mathematical Programming*, *18*, 68–83. (Cited on p. 259)

Brieden, A. (2002). Geometric optimization problems likely not contained in APX. *Discrete and Computational Geometry*, *28*, 201–209. (Cited on p. 29)

Brieden, A., Gritzmann, P., & Klee, V. (2000). Inapproximability of some geometric and quadratic optimization problems. In P. M. Pardalos (Ed.), (Vol. Approximation and Complexity in Numerical Optimization: Continuous and Discrete Problems, p. 96–115). Kluwer Academic Publishers. (Cited on p. 38)

Buhmann, M. D. (2003). *Radial basis functions*. Cambridge, U.K.: Cambridge University Press. (Cited on pp. 91, 93)

Bundfuss, S., & Duer, M. (2008). Algorithmic copositivity detection by simplicial partition. *Linear Algebra and its Applications*, *428*, 1511–1523. (Cited on pp. 183, 302)

Bundfuss, S., & Duer, M. (2009). An adaptive linear approximation algorithm for copositive programs. *SIAM Journal on Optimization*, *20*, 30–53. (Cited on pp. 183, 184, 302)

Burer, S. (2009). On the copositive representation of binary and continuous nonconvex quadratic programs. *Mathematical Programming*, *120*, 479–495. (Cited on pp. 174, 178, 180, 198, 213)

Burer, S., & Chen, J. (2011). Relaxing the optimality conditions of box QP. *Computational Optimization and Applications*, *48*, 653–673. (Cited on pp. 201, 298)

Burer, S., & Dong, H. (2012). Representing quadratically constrained quadratic programs as generalized copositive programs. *Operations Research Letters*, *40*, 203–206. (Cited on pp. 180, 181)

Burer, S., & Dong, H. (2013). Separation and relaxation for cones of quadratic forms. *Mathematical Programming*, *137*, 343–370. (Cited on p. 186)

Burer, S., & Letchford, A. (2012). Non-convex mixed-integer nonlinear programming: a survey. *Surveys in Operatirons Research and Management Science*, *17*, 97–106. (Cited on p. 4)

Burer, S., & Letchford, A. N. (2009). On nonconvex quadratic programming with box constraints. *SIAM Journal on Optimization*, *20*, 1073–1089. (Cited on pp. 202, 204, 209, 210)

Burer, S., & Saxena, A. (2009). *Old wine in new bottle: The MILP road to MIQC* (Tech. Rep.). Department of Management Sciences, University of Iowa. Retrieved from `http://www.optimization-online.org/DB_FILE/2009/07/2338.pdf` (Cited on p. 217)

Burer, S., & Vandenbussche, D. (2008). A finite branch-and-bound algorithm for nonconvex quadratic programming via semidefinite relaxations. *Mathematical Programming*, *113*, 259–282. (Cited on pp. 187, 193, 194, 195, 197, 198, 200, 325, 360)

Burer, S., & Vandenbussche, D. (2009). Globally solving box-constrained nonconvex quadratic programs with semidefinite-based finite branch-and-bound. *Computational Optimization and Applications*, *43*, 181–195. (Cited on pp. 187, 200, 201, 360)

Caprara, A., & Locatelli, M. (2010). Global optimization problems and domain reduction strategies. *Mathematical Programming*, *125*, 123–137. (Cited on pp. 340, 342, 343, 344)

Caprara, A., & Monaci, M. (2009). Bidimensional packing by bilinear programming. *Mathematical Programming*, *118*, 75–108. (Cited on p. 346)

Carrizosa, E., Hansen, P., & Messine, F. (2004). Improving interval analysis bounds by translations. *Journal of Global Optimization*, *29*, 157–172. (Cited on p. 258)

Carter, M. W., Jin, H. H., Saunders, M. A., & Ye, Y. (2006). Spaseloc: An adaptive subproblem algorithm for scalable wireless sensor network localization. *SIAM Journal on Optimization*, *17*(4), 1102–1128. doi: 10.1137/040621600. (Cited on p. 380)

Carvajal-Moreno, R. (1972). *Minimization of concave functions subject to linear constraints* (Tech. Rep. No. ORC 72–3). University of Berkeley, California. (Cited on p. 314)

Casado, L. G., Garcia, I., Szabó, P., & Csendes, T. (1998). Equal circles packing in square II: New results for up to 100 circles using the TAMSASS-PECS algorithm. In F. Giannessi, P. M. Pardalos, & T. Rapcsak (Eds.), *Optimization Theory: Recent developements from Mátraháza* (pp. 207–224). Kluwer Academic Publishers. (Cited on p. 383)

Casado, L. G., Martinez, J. A., & Garcia, I. (2001). Experiments with a new selection criterion in a fast interval optimization algorithm. *Journal of Global Optimization*, *19*, 247–264. (Cited on p. 292)

Casado, L. G., Martinez, J. A., Garcia, I., & Sergeyev, Y. D. (2003). New interval analysis support functions using gradient information in a global minimization algorithm. *Journal of Global Optimization*, *25*, 345–362. (Cited on p. 346)

Case, D. A., Cheatam, T. E. I., Darden, T., Gohlke, H., Luo, R., Merz, K. M. J., ... Woods, R. (2005). The Amber biomolecular simulation programs. *Journal of Computational Chemistry*, *26*, 1668–1688. (Cited on p. 372)

Cassier, G. (1984). Problème des moments sur un compact de $\mathbb{R}^n$ et décomposition de polynômes à plusieurs variables. *J. Funct. Anal.*, *58*, 254–266. (Cited on p. 222)

Cassioli, A., Di Lorenzo, D., Locatelli, M., & Schoen, F. (2012). Global optimization approaches for optimal trajectory planning. In G. Fasano & J. D. Pinter (Eds.), *Modeling and optimization in space engineering* (Vol. Springer Optimization and Its Applications). Springer. (Cited on p. 388)

Cassioli, A., Di Lorenzo, D., Locatelli, M., Schoen, F., & Sciandrone, M. (2012). Machine learning for global optimization. *Computational Optimization and Applications*, *51*, 279–303. (Cited on p. 85)

Cassioli, A., Locatelli, M., & Schoen, F. (2009). Global optimization of binary Lennard–Jones clusters. *Optimization Methods and Software*, *24*(4-5), 819–835. (Cited on p. 62)

Cassioli, A., Locatelli, M., & Schoen, F. (2010). Dissimilarity measures for population-based global optimization algorithms. *Computational Optimization and Applications*, *45*(2), 257–281. (Cited on p. 62)

Castillo, I., Kampas, F. J., & Pinter, J. D. (2008). Solving circle packing problems by global optimization: Numerical results and industrial applications. *European Journal of Operational Research*, *191*, 786–802. (Cited on p. 383)

Chen, J., & Burer, S. (2011). *Globally solving nonconvex QPs via completely positive programming* (Tech. Rep. No. ANL/MCS-P1837-0211). Argonne National Laboratory, Argonne, IL 60439, USA. (Cited on p. 198)

Chen, Y., & Chen, M. (2010). Extended duality for nonlinear programming. *Computational Optimization and Applications*, *47*, 33–59. (Cited on p. 267)

Cheng, L., Feng, Y., Yang, J., & Yang, J. (2009). Funnel hopping: Searching the cluster potential energy surface over the funnels. *The Journal of Chemical Physics*, *130*(21), 214112. (Cited on pp. 59, 377, 378)

Choquet, G. (1969). *Outlis topologiques et métriques de l'analyse mathématique,* (Tech. Rep.). Centre de documentation universitaire et SEDES, Paris. (Cited on p. 343)

Clausen, J., & Žilinskas, A. (2002). Subdivision, sampling, and initialization strategies for simplical branch and bound in global optimization. *Computers & Mathematics with Applications*, *44*, 943–955. (Cited on p. 254)

Clerc, M., & Kennedy, J. (2002). The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, *6*(1), 58–73. doi: 10.1109/4235.985692 (Cited on pp. 63, 65)

Conn, A. R., Gould, N. I. M., & Toint, P. L. (2000). *Trust-region methods.* Philadelphia (USA): SIAM. (Cited on p. 12)

Cooren, Y., Clerc, M., & Siarry, P. (2009). Performance evaluation of TRIBES, an adaptive particle swarm optimization algorithm. *Swarm Intelligence*, *3*(2), 149–178. doi: 10.1007/s11721-009-0026-8 (Cited on p. 63)

Crama, Y. (1989). Recognition problems for polynomials in 0-1 variables. *Mathematical Programming*, *44*, 139–155. (Cited on p. 128)

Crama, Y. (1993). Concave extensions for nonlinear 0-1 maximization problems. *Mathematical Programming*, *61*, 53–60. (Cited on p. 162)

Crippen, G. M., & Havel, T. F. (1988). *Distance geometry and molecular conformation.* John Wiley & Sons. (Cited on p. 379)

Csallner, A. E., Csendes, T., & Markót, M. C. (2000). Multisection in interval branch-and-bound methods for global optimization I. Theoretical results. *Journal of Global Optimization*, *16*, 371–392. (Cited on p. 301)

Csendes, T. (2001). New subinterval selection criteria for interval global optimization. *Journal of Global Optimization*, *19*, 307–327. (Cited on p. 292)

Cvijović, D., & Klinowski, J. (2002). Taboo search: An approach to the multiple-minima problem for continuous functions. In P. M. Pardalos & E. H. Romeijn (Eds.), *Handbook of Global Optimization - Vol. 2* (pp. 387–406). Kluwer Academic Publishers. (Cited on pp. 48, 84)

Dalila, B. M., Fontes, M., Hadjiconstantinou, E., & Christofides, N. (2006). A branch-and-bound algorithm for concave network flow problems. *Journal of Global Optimization*, *34*, 127–155. (Cited on p. 286)

Dallwig, S., Neumaier, A., & Schichl, H. (1997). GLOPT: a program for constrained global optimization. In I. M. Bomze, T. Csendes, R. Horst, & P. M. Pardalos (Eds.), (Vol. Developments in Global Optimization, p. 19–36). Kluwer Academic Publishers, Dordrecht. (Cited on p. 348)

de Angelis, P. L., Bomze, I. M., & Toraldo, G. (2004). Ellipsoidal approach to box-constrained quadratic problems. *Journal of Global Optimization*, *28*, 1–15. (Cited on p. 320)

de Klerk, E. (2008). The complexity of optimizing over a simplex, hypercube or sphere: A short survey. *CEJOR*, *16*, 111–125. (Cited on p. 11)

de Klerk, E., Laurent, M., & Parrillo, P. A. (2006). A PTAS for the minimization of polynomials of fixed degree over the simplex. *Theoretical Computer Science*, *361*, 210–225. (Cited on p. 28)

de Klerk, E., & Pasechnik, D. V. (2002). Approximation of the stability number of a graph via copositive programming. *SIAM Journal of Optimization*, *12*, 875–892. (Cited on pp. 26, 181, 182)

de Groot, C., Peikert, R., Würtz, D., & Monagan, M. (1991). *Packing circles in a square: A review and new results.* Laxenburg, Austria. (Cited on p. 382)

Depetrini, D., & Locatelli, M. (2009). A FPTAS for a class of linear multiplicative problems. *Computational Optimization and Applications*, *44*, 275–288. (Cited on p. 17)

Depetrini, D., & Locatelli, M. (2011). Approximation of linear fractional/multiplicative problems. *Mathematical Programming*, *128*, 437–443. (Cited on pp. 17, 18)

Diananda, P. H. (1967). On non-negative forms in real variables some or all of which are non-negative. *Proceedings of the Cambridge Philosophical Society*, *58*, 17–25. (Cited on p. 184)

Dickinson, P., Eichfelder, G., & Povh, J. (2012). Erratum to "on the set-semidefinite representation of nonconvex quadratic programs over arbitrary feasible sets". *Optimization Letters*, to appear. (Cited on p. 174)

Dinkelbach, W. (1967). On nonlinear fractional programming. *Management Science*, *13*, 492–498. (Cited on p. 18)

Di Pillo, G., Lucidi, S., & Rinaldi, F. (2012). An approach to constrained global optimization based on exact penalty functions. *Journal of Global Optimization*, *54*, 251–260. (Cited on p. 266)

Dixon, L. C. W., & Szegö, G. P. (1975). *Towards global optimization.* North-Holland. (Cited on pp. vii, 52)

Dixon, L. C. W., & Szegö, G. P. (1978). *Towards global optimization 2.* North-Holland. (Cited on pp. vii, 52)

Domes, F., & Neumaier, A. (2010). Constraint propagation on quadratic constraints. *Constraints*, *15*, 404–429. (Cited on p. 348)

Donev, A., Torquato, S., Stillinger, F. H., & Connelly, R. (2004). A linear programming algorithm to test for jamming in hard–sphere packings. *Journal of Computational Physic*, *197*, 139–166. (Cited on pp. 383, 384)

Dong, H., & Anstreicher, K. M. (2010). *Separating doubly nonnegative and completely positive matrices.* Retrieved from `http://www.optimization-online.org/DB_HTML/2010/03/2562.html` (Cited on pp. 184, 186)

Dong, Q., & Wu, Z. (2002). A linear-time algorithm for solving the molecular distance geometry problem with exact inter-atomic distances. *Journal of Global Optimization*, *22*, 365–375. (Cited on p. 379)

Doye, J. P. K. (2000). The effect of compression on the global optimization of atomic clusters. *Physical Review E*, *62*, 8753–8761. (Cited on p. 376)

Doye, J. P. K., Leary, R. H., Locatelli, M., & Schoen, F. (2004). Global optimization of Morse clusters by potential energy transformations. *INFORMS Journal On Computing*, *16*, 371–379. (Cited on pp. 62, 376)

Doye, J. P. K., & Wales, D. J. (1997). Structural consequences of the range of the interatomic potential: A menagerie of clusters. *Journal of the Chemical Society, Faraday Transactions*, *93*, 4233–4244. (Cited on p. 376)

Duer, M. (2001). Dual bounding procedures lead to convergent branch-and-bound algorithms. *Mathematical Programming*, *91*, 117–125. (Cited on p. 331)

Duer, M. (2002). A class of problems where dual bounds beat underestimation bounds. *Journal of Global Optimization*, *22*, 49–57. (Cited on p. 261)

Duer, M., & Horst, R. (1997). Lagrange duality and partitioning techniques in nonconvex global optimization. *Journal of Optimization Theory and Applications*, *95*, 347–369. (Cited on p. 260)

Duran, M. A., & Grossmann, I. E. (1986). An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, *36*, 307–339. (Cited on p. 271)

Egea, J. A., Martí, R., & Banga, J. R. (2010). An evolutionary method for complex-process optimization. *Computers & Operations Research*, *37*(2), 315–324. (Cited on p. 84)

Eichfelder, G., & Povh, J. (2012). On the set-semidefinite representation of nonconvex quadratic programs over arbitrary feasible sets. *Optimization Letters*, to appear. (Cited on p. 174)

Esposito, W., & Floudas, C. (1998). Global optimization in parameter estimation of nonlinear algebraic models via the error-in-variables approach. *Industrial & Engineering Chemical Research*, *35*, 1841–1998. (Cited on p. 371)

Evtushenko, Y. G. (1971). Numerical methods for finding global extrema (case of a nonuniform mesh). *USSR Computational Mathematics and Mathematical Physics*, *11*, 38–54. (Cited on p. 253)

Falk, J. E. (1969). Lagrange multipliers and nonconvex programs. *SIAM Journal on Control*, *7*, 534–545. (Cited on pp. 128, 260)

Falk, J. E., & Soland, R. M. (1969). An algorithm for separable nonconvex programming problems. *Management Science*, *15*, 550–569. (Cited on p. 296)

Faria, D. C., & Bagajewicz, M. J. (2011). Novel bound contraction procedure for global optimization of bilinear MINLP problems with applications to water management problems. *Computers and Chemical Engineering*, *35*, 446–455. (Cited on p. 346)

Fasano, G., & Pinter, J. D. (Eds.). (2013). *Modeling and optimization in space engineering*. Springer. (Cited on p. 386)

Feige, U., & Kilian, J. (1994). Two prover protocols: low error at affordable rates. In *Proceedings of the twenty sixth annual symposium on the theory of computing, ACM.* (Cited on p. 29)

Feo, T. A., & Resende, M. G. C. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization*, *6*, 109–133. (Cited on p. 68)

Ferrer, A., & Martinez-Legaz, J. E. (2009). Improving the efficiency of DC global optimization methods by improving the DC representation of the objective function. *Journal of Global Optimization*, *43*, 513–531. (Cited on p. 243)

Fletcher, R. L. (2000). *Practical methods of optimization* (2nd ed.). John Wiley & Sons. (Cited on p. 12)

Floudas, C. A., Pardalos, P. M., Adjiman, C. S., Esposito, W. R., Gumus, Z. H., Harding, S. T., ... Schweiger, C. (1999). *Handbook of test problems in local and global optimization* (Vol. 33). Dordrecht: Kluwer Academic Publishers. (Cited on p. 370)

Forrester, A. I. J., & Keane, A. J. (2009). Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, *45*(1-3), 50–79. (Cited on pp. 86, 102, 110)

Fortin, C., & Wolkowicz, H. (2004). The trust region subproblem and semidefinite programming. *Optimization Methods and Software*, *19*, 41–67. (Cited on p. 12)

Fourer, R., Gay, D. M., & Kernighan, B. W. (2002). *AMPL: A modeling language for mathematical programming* (Second edition ed.). Duxbury Press. (Cited on p. 363)

Galperin, E. A. (1985). The cubic algorithm. *Journal of Mathematical Analysis and Applications*, *112*, 635–640. (Cited on p. 254)

Galperin, E. A. (1988). Precision, complexity, and computational schemes of the cubic algorithm. *Journal of Optimization Theory and Applications*, *57*, 223–238. (Cited on p. 254)

Gao, D. Y. (2004). Canonical duality theory and solutions to constrained nonconvex quadratic programming. *Journal of Global Optimization*, *29*, 377–399. (Cited on p. 267)

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of NP-completeness*. New York, NY, USA: W. H. Freeman & Co. (Cited on p. 8)

Gatzke, E. P., Tolsma, J. E., & Barton, P. I. (2002). Construction of convex relaxations using automated code generation techniques. *Optimization and Engineering*, *3*, 305–326. (Cited on p. 270)

Geoffrion, A. M. (1971). Duality in nonlinear programming : a simplified applications-oriented development. *SIAM Review*, *13*, 1–37. (Cited on p. 260)

Geoffrion, A. M. (1972). Generalized Benders decomposition. *Journal of Optimization Theory and Applications*, *10*, 237–260. (Cited on p. 271)

Giannessi, F., & Tomasin, E. (1973). Nonconvex quadratic programs, linear complementarity problems, and integer linear programs. In *Proceedings of the 5th Conference on Optimization Techniques Part I*, Lecture Notes in Computer Science 3, 437–449. Springer, Berlin Heidelberg New York. (Cited on p. 188)

Gill, P. E., Murray, W., & Saunders, M. A. (2005). SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Review*, *47*(1), 99–131. (Cited on p. 385)

Goemans, M. X., & Williamson, D. P. (1995). Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the Association for Computing Machinery*, *42*, 1115–1145. (Cited on pp. 30, 34)

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA. (Cited on p. 73)

Gordon, W. J., & Wixom, J. A. (1978). Shepard's method of "metric interpolation" to bivariate and multivariate interpolation. *Mathematics of Computation*, *32*, 253–264. (Cited on p. 369)

Gounaris, C., & Floudas, C. A. (2008a). Tight convex underestimators for C^2-continuous problems : II. Multivariate functions. *Journal of Global Optimization*, *42*, 69–89. (Cited on pp. 237, 238, 239)

Gounaris, C., & Floudas, C. A. (2008b). Tight convex underestimators for C^2-continuous problems : I. Univariate functions. *Journal of Global Optimization*, *42*, 51–67. (Cited on p. 237)

Gourdin, E., Hansen, P., & Jaumard, B. (1994). Finding maximum likelihood estimators for the three-parameter Weibull distribution. *Journal of Global Optimization*, *5*, 373–397. (Cited on p. 254)

Goyal, V., Genc-Kaya, L., & Ravi, R. (2011). An FPTAS for minimizing the product of two non-negative linear cost functions. *Mathematical Programming*, *126*, 401–405. (Cited on p. 17)

Grosso, A., Jamali, A., Locatelli, M., & Schoen, F. (2010). Solving the problem of packing equal and unequal circles in a circular container. *Journal of Global Optimization*, *47*(1), 63–81. (Cited on pp. 63, 383, 385)

Grosso, A., Locatelli, M., & Schoen, F. (2007a). An experimental analysis of a population based approach for global optimization. *Computational Optimization and Applications*, *38*(3), 351–370. doi: 10.1007/s10589-007-9026-z (Cited on pp. 49, 62)

Grosso, A., Locatelli, M., & Schoen, F. (2007b). A population based approach for hard global optimization problems based on dissimilarity measures. *Mathematical Programming*, *110*(2), 373–404. (Cited on pp. 62, 378)

Grosso, A., Locatelli, M., & Schoen, F. (2009). Solving molecular distance geometry problems by global optimization algorithms. *Computational Optimization and Applications*, *43*, 23–37. (Cited on pp. 62, 379)

Grotschel, M., Lovasz, L., & Schrijver, A. (1993). *Geometric algorithms and combinatorial optimization*. Springer-Verlag - Berlin. (Cited on p. 322)

Guerin, J., Marcotte, P., & Savard, G. (2006). An optimal adaptive algorithm for the approximation of concave functions. *Mathematical Programming*, *107*, 357–366. (Cited on p. 273)

Gutmann, H. M. (2001a). A radial basis function method for global optimization. *Journal of Global Optimization*, *19*, 201–227. (Cited on pp. 87, 91, 110)

Gutmann, H. M. (2001b). *Radial basis function methods for global optimization*. Unpublished doctoral dissertation, University of Cambridge, United Kingdom. (Cited on pp. 98, 100, 110, 114)

Gvozdenovic, N., & Laurent, M. (2007). Semidefinite bounds for the stability number of a graph via sums of squares of polynomials. *Mathematical Programming*, *110*, 145–173. (Cited on p. 182)

Hager, W., & Phan, D. (2009). An ellipsoidal branch and bound algorithm for global optimization. *SIAM Journal on Optimization*, *20*, 740–758. (Cited on pp. 320, 321, 322)

Hamami, M., & Jacobsen, S. E. (1988). Exhaustive nondegenerate conical processes for concave minimization on convex polytopes. *Mathematics of Operations Research*, *13*, 479–487. (Cited on pp. 309, 360, 361)

Hamed, A. S. E., & McCormick, G. P. (1993). Calculation of bounds on variables satisfying nonlinear inequality constraints. *Journal of Global Optimization*, *3*, 25–47. (Cited on p. 335)

Hansen, P. (1992). *Global optimization using interval analysis*. New York: Marcel Dekker. (Cited on p. 259)

Hansen, P., Jaumard, B., & Lu, S. H. (1991). An analytical approach to global optimization. *Mathematical Programming*, *52*, 227–254. (Cited on p. 335)

Hansen, P., Jaumard, B., & Lu, S.-H. (1992). Global optimization of univariate Lipschitz functions: II. New algorithms and computational comparison. *Mathematical Programming*, *55*, 273–292. (Cited on p. 253)

Hansen, P., Jaumard, B., Ruiz, M., & Xiong, J. (1993). Global minimization of indefinite quadratic functions subject to box constraints. *Naval Research Logistics*, *40*, 373–392. (Cited on pp. 188, 360)

Hansen, P., Mladenović, N., & Moreno Pérez, J. (2008). Variable neighbourhood search: Methods and applications. *4OR: A Quarterly Journal of Operations Research*, *6*(4), 319–360. doi: 10.1007/s10288-008-0089-1 (Cited on p. 63)

Hanzon, B., & Jibetean, D. (2003). Global minimization of a multivariate polynomial using matrix methods. *Journal of Global Optimization*, *27*, 1–23. (Cited on p. 222)

Hartke, B. (1999). Global cluster geometry optimization by a phenotype algorithm with niches: Location of elusive minima, and low-order scaling with cluster size. *Journal of Computayional Chemistry*, *20*, 1752–1759. (Cited on p. 377)

Hartke, B. (2006). Efficient global geometry optimization of atomic and molecular clusters. In J. D. Pinter (Ed.), *Global optimization* (Vol. 85, p. 141–168). Springer US. doi: 10.1007/0-387-30927-6_6 (Cited on p. 73)

Hastad, J. (1999). Clique is hard to approximate within $| V |^{1-\varepsilon}$. *Acta Mathematica*, *182*, 105–142. (Cited on pp. 25, 26)

Hastad, J. (2001). Some optimal inapproximability results. *Journal ACM*, *48*, 798–859. (Cited on p. 30)

He, S., Li, Z., & Zhang, S. (2010). Approximation algorithms for homogeneous polynomial optimization with quadratic constraints. *Mathematical Programming*, *125*, 353–383. (Cited on p. 37)

Hedar, A.-R., & Fukushima, M. (2006). Tabu search directed by direct search methods for nonlinear global optimization. *European Journal of Operational Research*, *170*(2), 329–349. (Cited on pp. 48, 84)

Herrera, F., Lozano, M., & Molina, D. (2006). Continuous scatter search: An analysis of the integration of some combination methods and improvement strategies. *European Journal of Operational Research*, *169*(2), 450–476. (Cited on p. 84)

Hilbert, D. (1888). Ueber die Darstellung definiter Formen als Summe von Formenquadraten. *Mathematische Annalen*, *32*, 342–350. (Cited on p. 218)

Hiriart-Urruty, J., & Lemaréchal, C. (1993). *Convex analysis and minimization algorithms I.* Springer Verlag. (Cited on p. 235)

Hiriart-Urruty, J.-B. (1995). Conditions for global optimality. In R. Horst & P. M. Pardalos (Eds.), (Vol. Handbook of Global Optimization, pp. 1–26). Kluwer Academic Publishers. (Cited on p. 4)

Hiriart-Urruty, J.-B. (1998). Conditions for global optimality 2. *Journal of Global Optimization*, *13*, 349–367. (Cited on p. 4)

Hirsch, M. J., Meneses, C. N., Pardalos, P. M., & Resende, M. G. C. (2007). Global optimization by continuous GRASP. *Optimization Letters*, *1*, 201–212. (Cited on p. 68)

Holmström, K. (2008). An adaptive radial basis algorithm (ARBF) for expensive black-box global optimization. *Journal of Global Optimization*, *41*(3), 447–464. doi: 10.1007/s10898-007-9256-8 (Cited on pp. 115, 116)

Horst, R. (1976). An algorithm for nonconvex programming problems. *Mathematical Programming*, *10*, 312–321. (Cited on pp. 298, 302)

Horst, R. (1980). A note on the convergence of an algorithm for nonconvex programming problems. *Mathematical Programming*, *19*, 237–238. (Cited on p. 302)

Horst, R., & Tuy, H. (1993). *Global optimization: Deterministic approaches* (2nd ed.). Springer Verlag. (Cited on pp. 242, 314, 331, 351, 352, 360, 361)

Hu, J., Mitchell, J. E., & Pang, J. S. (2012). An LPCC approach to nonconvex quadratic programs. *Mathematical Programming*, *133*, 243–277. (Cited on pp. 199, 200, 358)

Hu, J., Mitchell, J. E., Pang, J. S., Bennett, K., & Kunapuli, G. (2008). On the global solution of linear programs with linear complementarity constraints. *SIAM Journal on Optimization*, *19*, 445–471. (Cited on pp. 199, 200)

Huang, W., Li, Y., Akeb, H., & Li, C. (2005). Greedy algorithms for packing unequal circles into a rectangular container. *Journal of the Operational Research Society*, *56*(5), 539–548. (Cited on p. 383)

Huang, W., & Ye, T. (2010). Greedy vacancy search algorithm for packing equal circles in a square. *Operations Research Letters*, *38*(5), 378–382. (Cited on p. 384)

Huang, X., & Yang, X. (2003). A unified augmented Lagrangian approach to duality and exact penalization. *Mathematics of Operations Research*, *28*, 533–552. (Cited on pp. 267, 383)

Izzo, D., Becerra, V., Myatt, D., Nasuto, S., & Bishop, J. (2007). Search space pruning and global optimisation of multiple gravity assist spacecraft trajectories. *Journal of Global Optimization*, *38*, 283–296. (Cited on p. 386)

Jach, M., Michaels, D., & Weismantel, R. (2008). The convex envelope of (n–1)-convex functions. *SIAM Journal on Optimization*, *19*(3), 1451–1466. (Cited on pp. 141, 161)

Jacobsen, S. E. (1981). Convergence of a Tuy-type algorithm for concave minimization subject to linear inequality constraints. *Applied Mathematics and Optimization*, *7*, 1–9. (Cited on p. 309)

Jakobsson, S., Patriksson, M., Rudholm, J., & Wojciechowski, A. (2010). A method for simulation based optimization using radial basis functions. *Optimization and Engineering*, *11*, 501–532. doi: 10.1007/s11081-009-9087-1 (Cited on pp. 108, 109)

Jamrog, D., Phillips Jr, G., Tapia, R., & Zhang, Y. (2005). A global optimization method for the molecular replacement problem in X-ray crystallography. *Mathematical Programming B*, *103*, 399–426. (Cited on p. 371)

Jansson, C. (2003). Rigorous error bounds for the optimal value of linear programming problems. In (Vols. First International Workshop on Global Constrained Optimization and Constraint Satisfaction, COCOS 2002, p. 59–70). (Cited on p. 287)

Jarre, F. (2012). Burer's key assumption for semidefinite and doubly nonnegative relaxations. *Optimization Letters*, *6*, 593–599. (Cited on p. 181)

Jaumard, B., & Meyer, C. (2001). On the convergence of cone splitting algorithms with ω-subdivisions. *Journal of Optimization Theory and Applications*, *110*, 119–144. (Cited on p. 331)

Jeyakumar, V., & Li, G. (2011). Necessary global optimality conditions for nonlinear programming problems with polynomial constraints. *Mathematical Programming*, *126*, 393–399. (Cited on p. 4)

Jeyakumar, V., Rubinov, A., & Wu, Z. (2007). Non-convex quadratic minimization problems with quadratic constraints: Global optimality conditions. *Mathematical Programming*, *110*, 521–541. (Cited on p. 4)

Jibetean, D., & de Klerk, E. (2006). Global optimization of rational functions: A semidefinite programming approach. *Mathematical Programming*, *106*, 93–109. (Cited on p. 230)

Jibetean, D., & Laurent, M. (2005). Semidefinite approximations for global unconstrained polynomial optimization. *SIAM Journal on Optimization*, *16*(2), 490–514. (Cited on pp. 222, 230)

Jones, D. R. (2001a). The DIRECT global optimization algorithm. In C. A. Floudas & P. M. Pardalos (Eds.), (Vol. Encyclopedia of Optimization, p. 431–440). Kluwer Academic Publishers, Dordrecht. (Cited on p. 71)

Jones, D. R. (2001b). A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, *21*(4), 345–383. (Cited on pp. 86, 88, 110, 116)

Jones, D. R., Perttunen, C. D., & Stuckman, B. E. (1993). Lipschitzian optimization without the Lipschitz constant. *Journal of optimization Theory and Applications*, *79*, 157–181. (Cited on pp. 69, 70)

Jones, D. R., Schonlau, M., & Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, *13*(4), 455–492. (Cited on p. 117)

Kahl, F., Agarwal, S., Chandraker, M. K., Kriegman, D., & Belongie, S. (2008). Practical global optimization for multiview geometry. *International Journal of Computer Vision*, *79*, 271–284. (Cited on p. 371)

Kalantari, B., & Rosen, J. B. (1987). An algorithm for global minimization of linearly constrained concave quadratic functions. *Mathematics of Operations Research*, *12*, 544–561. (Cited on p. 298)

Kallrath, J. (2009). Cutting circles and polygons from area-minimizing rectangles. *Journal of Global Optimization*, *43*(2-3), 1–30. (Cited on p. 383)

Karuppiah, R., & Grossmann, I. E. (2008). A Lagrangean based branch-and-cut algorithm for global optimization of nonconvex mixed-integer nonlinear programs with decomposable structures. *Journal of Global Optimization*, *41*, 163–186. (Cited on p. 265)

Kearfott, R. B. (1996). *Rigorous global search: continuous problems*. Dordrecht: Kluwer Academic Publishesrs. (Cited on pp. 259, 358)

Kearfott, R. B. (2011). Interval computations, rigor and non-rigor in deterministic continuous global optimization. *Optimization Methods and Software*, *26*, 259–279. (Cited on p. 287)

Kelley, C. T. (2011). *Implicit filtering*. C.T. Kelley. (Cited on p. 388)

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings IEEE International Conference on Neural Networks* (Vol. 4, pp. 1942–1948). doi: 10.1109/icnn.1995.488968 (Cited on p. 63)

Kern, W., & Woeginger, G. (2007). Quadratic programming and combinatorial minimum weight product problems. *Mathematical Programming*, *110*(3), 641–649. doi: 10.1007/s10107-006-0047-7 (Cited on p. 17)

Khajavirad, A., & Sahinidis, N. (2012b). Convex envelopes of products of convex and component-wise concave functions. *Journal of Global Optimization*, *52*, 391–409. (Cited on pp. 147, 148)

Khajavirad, A., & Sahinidis, N. V. (2013). Convex envelopes generated from finitely many compact convex sets. *Mathematical Programming*, *137*, 371–408. (Cited on p. 147)

Kiefer, J., & Wolfowitz, J. (1952). Stochastic estimation of the maximum of a regression function. *Annals of Mathematical Statistics*, *23*(3), 462–466. (Cited on p. 102)

Klepeis, J. L., Ierapetritou, M. G., & Floudas, C. A. (1998). Protein folding and peptide docking: A molecular modeling and global optimization approach. *Computers and Chemical Engineering*, *22 Suppl. 1*, S3–S10. (Cited on p. 372)

Kolda, T. G., Lewis, R. M., & Torczon, V. (2003). Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Review*, *45*, 385–482. (Cited on p. 66)

Konno, H. (1976). Maximization of a convex quadratic function under linear constraints. *Mathematical Programming*, *11*, 117–127. (Cited on pp. 352, 354)

Konno, H., & Kuno, T. (1995a). Multiplicative programming problems. In R. Horst & P. M. Pardalos (Eds.), (Vol. Handbook of Global Optimization, p. 369–405). Kluwer Academic Publishers. (Cited on p. 18)

Konno, H., & Kuno, T. (1995b). Multiplicative programming problems. In R. Horst & P. M. Pardalos (Eds.), (Vol. Handbook of Global Optimization, p. 369–405). Kluwer Academic Publishers. (Cited on p. 286)

Kuno, T. (2001). A finite branch-and-bound algorithm for linear multiplicative programming. *Computational Optimization and Applications*, *20*, 119–135. (Cited on p. 361)

Kuno, T. (2002). A branch-and-bound algorithm for maximizing the sum of several linear ratios. *Journal of Global Optimization*, *22*, 155–174. (Cited on p. 323)

Kuno, T. (2005). A revision of the trapezoidal branch-and-bound algorithm for linear sum-of-ratios problems. *Journal of Global Optimization*, *33*, 215–234. (Cited on p. 323)

Kuno, T., & Buckland, P. E. K. (2012). A convergent simplicial algorithm with ω-subdivision and ω-bisection strategies. *Journal of Global Optimization*, *52*, 371–390. (Cited on p. 331)

Kuno, T., & Utsunomiya, T. (2000). A Lagrangian based branch-and-bound algorithm for production-transportation problems. *Journal of Global Optimization*, *18*, 59–73. (Cited on p. 265)

Kushner, H. J. (1964). A new method of locating the maximum of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*(86), 97–106. (Cited on pp. 102, 118)

Lagarias, J. C., Reeds, J. A., Wright, M. H., & Wright, P. E. (1998). Convergence properties of the Nelder–Mead simplex method in low dimensions. *SIAM Journal on Optimization*, *9*, 112–147. (Cited on p. 67)

Lagouanelle, J. L., Csendes, T., & Vinkó, T. (2004). A new inclusion function for optimization: Kite - The one dimensional case. *Journal of Global Optimization*, *30*, 435–456. (Cited on p. 258)

Laguna, M., Molina, J., Peréz, F., Caballero, R., & Hernàndez-Dìaz, A. G. (2010). The challenge of optimizing expensive black boxes: A scatter search/rough set theory approach. *Journal of the Operational Research Society*, *61*(1), 53–67. (Cited on p. 84)

Laraki, R., & Lasserre, J. B. (2008). Computing uniform convex approximations for convex envelopes and convex hulls. *Journal of Convex Analysis*, *15*, 635–654. (Cited on pp. 146, 147)

Lasserre, J. B. (2001). Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, *11*(3), 796–817. (Cited on pp. 222, 227, 229)

Lasserre, J. B. (2002). Semidefinite programming vs LP relaxations for polynomial programming. *Mathematics of Operations Research*, *27*, 347–360. (Cited on p. 229)

Lasserre, J. B. (2005). Polynomial programming: Lp-relaxations also converge. *SIAM Journal on Optimization*, *15*(2), 383–393. (Cited on p. 229)

Lasserre, J. B. (2006). A sum of squares approximation of nonnegative polynomials. *SIAM Journal on Optimization*, *16*(3), 751–765. (Cited on pp. 223, 230)

Lasserre, J. B. (2009). Moments and sums of squares for polynomial optimization and related problems. *Journal of Global Optimization*, *45*, 39–61. (Cited on p. 146)

Lasserre, J. B., & Netzer, T. (2006). SOS approximations of nonnegative polynomials via simple high degree perturbations. *Mathematische Zeitschrift*, *256*, 99–112. (Cited on p. 223)

Laurent, M. (2007). Semidefinite representations for finite varieties. *Mathematical Programming*, *109*, 1–26. (Cited on p. 222)

Laurent, M. (2009). Sums of squares, moment matrices and optimization over polynomials. In M. Putinar & S. Sullivant (Eds.), *Emerging Applications of Algebraic Geometry*, IMA Volumes in Mathematics and Its Applications 149, 157–270. Springer. (Cited on p. 217)

Lavor, C., Liberti, L., Maculan, N., & Mucherino, A. (2012). The discretizable molecular distance geometry problem. *Computational Optimization and Applications*, *52*, 115–146. (Cited on p. 379)

Leary, R. H. (1997). Global optima of Lennard-Jones clusters. *Journal of Global Optimization*, *11*(1), 35–53. (Cited on p. 376)

Leary, R. H. (2000). Global optimization on funneling landscapes. *Journal of Global Optimization*, *18*, 367–383. (Cited on p. 57)

Lee, J., Lee, I.-H., & Lee, J. (2003). Unbiased global optimization of Lennard-Jones clusters for $N \leq 201$ by conformational space annealing method. *Physical Review Letters*, *91*(8), 1–4. (Cited on pp. 62, 376)

Lee, J., & Leyffer, S. (Eds.). (2011). *Mixed integer nonlinear programming*. Berlin: Springer. (Cited on p. 4)

Lee, J., Pillardy, J., Czaplewski, C., Arnautova, Y. A., Ripoll, D. R., Liwo, A., … Scheraga, H. A. (2000). Efficient parallel algorithms in global optimization of potential energy functions for peptides, proteins, and crystals. *Computer Physics Communications*, *128*, 399–411. (Cited on pp. 62, 371, 376)

Lee, J., Ripoll, D. R., Czaplewski, C., Pillardy, J., Wedemeyer, W. J., & Scheraga, H. A. (2001). Optimization of parameters in macromolecular potential energy functions by conformational space annealing. *The Journal of Physical Chemistry B*, *105*, 7291–7298. (Cited on p. 62)

Lee, K. (2008). Computational study for protein-protein docking using global optimization and empirical potentials. *International Journal of Molecular Sciences*, *9*, 65–77. (Cited on p. 372)

Lee, S., & Grossmann, I. E. (2000). New algorithms for nonlinear generalized disjunctive programming. *Computers and Chemical Engineering*, *24*, 2125–2141. (Cited on p. 286)

Le Thi, H. A. (2000). An efficient algorithm for globally minimizing a quadratic function under convex quadratic constraints. *Mathematical Programming*, *87*, 401–426. (Cited on pp. 320, 322)

Levy, A. V., & Montalvo, A. (1985). The tunneling method for global optimization. *SIAM Journal on of Scientific and Statistical Computing*, *1*, 15–29. (Cited on p. 76)

Leyffer, S., Sartenaer, A., & Wanufelle, E. (2008). *Branch-and-refine for mixed-integer nonconvex global optimization* (Tech. Rep. No. Preprint ANL/MCS-P1547-0908,).

Argonne National Laboratory, Mathematics and Computer Science Division. (Cited on pp. 270, 271)

Liberti, L. (2005). Linearity embedded in nonconvex programs. *Journal of Global Optimization*, *33*, 157–196. (Cited on p. 171)

Liberti, L., & Pantelides, C. C. (2003). Convex envelopes of monomials of odd degree. *Journal of Global Optimization*, *25*, 157–168. (Cited on p. 161)

Lin, Q., & Rokne, J. G. (1996). Interval approximation of higher-order to the ranges of functions. *Computers Mathematical Applications*, *31*, 101–109. (Cited on p. 258)

Lin, T. C., & Vandenbussche, D. (2008). Box-constrained quadratic programs with fixed charge variables. *Journal of Global Optimization*, *41*, 75–102. (Cited on p. 188)

Linderoth, J. (2005). A simplicial branch-and-bound algorithm for solving quadratically constrained quadratic programs. *Mathematical Programming*, *103*, 251–282. (Cited on pp. 161, 212, 271, 273, 322)

Ling, C., Nie, J., Qi, L., & Ye, Y. (2009). Bi-quadratic optimization over unit spheres and semidefinite programming relaxations. *SIAM Journal on Optimization*, *20*, 1286–1310. (Cited on pp. 26, 37)

Liu, J., Xue, S., Liu, Z., & Xu, D. (2009). An improved energy landscape paving algorithm for the problem of packing circles into a larger containing circle. *Computers and Industrial Engineering*, *57*, 1144–1149. (Cited on pp. 383, 385)

Liuzzi, G., Lucidi, S., & Piccialli, V. (2010). A DIRECT-based approach exploiting local minimizations for the solution of large-scale global optimization problems. *Computational Optimization and Applications*, *45*(2), 353–375. doi: 10.1007/s10589-008-9217-2 (Cited on p. 72)

Liuzzi, G., Lucidi, S., Piccialli, V., & Sotgiu, A. (2004). A magnetic resonance device designed via global optimization techniques. *Mathematical Programming B*, *101*, 339–364. (Cited on p. 371)

Liwo, A., Lee, J., Ripoll, D. R., Pillardy, J., & Scheraga, H. A. (1999). Protein structure prediction by global optimization of a potential energy function. *Proceedings of the National Academy of Sciences USA*, *96 Biophysics*, 5482–5485. (Cited on p. 371)

Locatelli, M. (1999). Finiteness of conical algorithms with ω-subdivisions. *Mathematical Programming*, *85*, 593–616. (Cited on p. 331)

Locatelli, M. (2002). Simulated annealing algorithms for continuous global optimization. In P. M. Pardalos & H. E. Romeijn (Eds.), (Handbook of Global Optimization, Vol. 2, p. 179–229). Kluwer Academic Publishers. (Cited on p. 75)

Locatelli, M. (2005). On the multilevel structure of global optimization problems. *Computational Optimization and Applications*, *30*(1), 5–22. (Cited on pp. 46, 47, 57)

Locatelli, M. (2009). Complexity results for some global optimization problems. *Journal of Optimization Theory and Applictions*, *140*(1), 93–102. (Cited on p. 37)

Locatelli, M. (2013). Approximation algorithm for a class of global optimization problems. *Journal of Global Optimization*, *55*, 13–25. (Cited on p. 17)

Locatelli, M. (2012b). *Polyhedral subdivisions and functional forms for the convex envelopes of bilinear, fractional and other bivariate functions over general polytopes.* (submitted) (Cited on p. 157)

Locatelli, M., & Raber, U. (2000). On convergence of the simplicial branch-and-bound algorithm based on ω-subdivisions. *Journal of Optimization Theory and Applications*, *107*, 69–79. (Cited on p. 331)

Locatelli, M., & Raber, U. (2002). Packing equal circles in a square: A deterministic global optimization approach. *Discrete Applied Mathematics*, *122*, 139–166. (Cited on pp. 346, 382, 383)

Locatelli, M., & Schoen, F. (1996). Simple linkage: Analysis of a threshold-accepting global optimization method. *Journal of Global Optimization*, *9*, 95–111. (Cited on p. 55)

Locatelli, M., & Schoen, F. (1999). Random linkage: A family of acceptance/rejection algorithms for global optimisation. *Mathematical Programming*, *85*(2), 379–396. (Cited on pp. 55, 56)

Locatelli, M., & Schoen, F. (2002). Minimal interatomic distance in Morse clusters. *Journal of Global Optimization*, *22*, 175–190. (Cited on p. 374)

Locatelli, M., & Schoen, F. (2003). Efficient algorithms for large scale global optimization: Lennard-Jones clusters. *Computational Optimization and Applications*, *26*, 173–190. (Cited on pp. 62, 376)

Locatelli, M., & Schoen, F. (2010). *On convex envelopes and underestimators for bivariate functions. Mathematical Programming*, to appear. (Cited on pp. 142, 150, 154, 155, 156, 161)

Locatelli, M., & Schoen, F. (2012a). Local search based heuristics for global optimization: atomic clusters and beyond. *European Journal of Operational Research*, *222*, 1–9. (Cited on p. 378)

Locatelli, M., & Schoen, F. (2012b). On the relation between concavity cuts and the surrogate dual for convex maximization problems. *Journal of Global Optimization*, *52*, 411–421. (Cited on p. 259)

Locatelli, M., & Thoai, N. V. (2000). Finite exact branch-and-bound algorithms for concave minimization over polytopes. *Journal of Global Optimization*, *18*, 107–128. (Cited on pp. 302, 361)

Locatelli, M., & Vasile, M. (2009). A hybrid multiagent approach for global trajectory optimization. *Journal of Global Optimization*, *44*(4), 461–479. (Cited on p. 386)

Lootsma, F., & Pearson, J. (1970). An indefinite-quadratic-programming model for a continuous production problem. *Philips Research Reports*, *25*, 244–254. (Cited on p. 371)

Lòpez, C. O., & Beasley, J. E. (2011). A heuristic for the circle packing problem with a variety of containers. *European Journal of Operational Research*, *214*(3), 512–525. (Cited on pp. 383, 385)

Lourenço, H. R., Martin, O. C., & Stützle, T. (2003). Iterated local search. In F. W. Glover & G. A. Kochenberger (Eds.), *Handbook of Metaheuristics* (pp. 321–353). Boston, Dordrecht, London: Kluwer Academic Publishers. (Cited on pp. 45, 57)

Lovász, L. (1982). Submodular functions and convexity. In M. Grötschel & B. Korte (Eds.), (Mathematical programming : The state of the art, p. 235–257). Springer. (Cited on p. 138)

Lu, H. C., Li, H. L., Gounaris, C., & Floudas, C. A. (2010). Convex relaxation for solving posynomial programs. *Journal of Global Optimization*, *46*, 147–154. (Cited on p. 286)

Lubachevsky, B. D. (1991). How to simulate billiards and similar systems. *Journal of Computational Physics*, *94*, 255–283. (Cited on p. 384)

Lucidi, S., & Piccioni, M. (1989). Random tunneling by means of acceptance-rejection sampling for global optimization. *Journal of Optimization Theory and Applications*, *62*, 255–275. (Cited on p. 77)

Luedtke, J., Namazifar, M., & Linderoth, J. (2010). *Some results on the strength of relaxations of multilinear functions* (Tech. Rep.). UW-Madison. (Cited on pp. 210, 286)

Luo, H. Z., Sun, X. L., & Li, D. (2007). On the convergence of augmented Lagrangian methods for constrained global optimization. *SIAM Journal on Optimization*, *18*(4), 1209–1230. (Cited on p. 266)

Luo, Z. Q., Sidiropoulos, N., Tseng, P., & Zhang, S. Z. (2007). Approximation bounds for quadratic optimization with homogeneous quadratic constraints. *SIAM Journal on Optimization*, *18*, 1–28. (Cited on p. 37)

Luo, Z. Q., & Zhang, S. Z. (2011). A semidefinite relaxation scheme for multivariate quartic polynomial optimization with quadratic constraints. *SIAM Journal on Optimization*, *20*, 1716–1736. (Cited on p. 37)

Majthai, A., & Whinston, A. (1974). Quasi-concave minimization subject to linear constraints. *Discrete Mathematics*, *9*, 35–59. (Cited on p. 351)

Man-Cho So, A., & Ye, Y. (2007). Theory of semidefinite programming for sensor network localization. *Mathematical Programming*, *108*(2), 367–384. (Cited on p. 380)

Maranas, C. D., & Floudas, C. A. (1994). Global minimum potential energy conformations of small molecules. *Journal of Global Optimization*, *4*, 135–170. (Cited on p. 230)

Maranas, C. D., & Floudas, C. A. (1997). Global optimization in generalized geometric programming. *Computers and Chemical Engineering*, *21*, 351–370. (Cited on pp. 286, 335)

Maranas, C. D., Floudas, C. A., & Pardalos, P. M. (1995). New results in the packing of equal circles in a square. *Discrete Mathematics*, *142*, 287–293. doi: 10.1016/0012-365x(93)e0230-2 (Cited on p. 383)

Marcovecchio, M. G., Bergamini, M. L., & Aguirre, P. A. (2006). Improve-and-branch algorithm for the global optimization of nonconvex NLP problems. *Journal of Global Optimization*, *34*, 339–368. (Cited on p. 270)

Markót, M. C., & Csendes, T. (2005). A new verified optimization technique for the "packing circles in a unit square" problem. *SIAM Journal on Optimization*, *16*, 193–219. (Cited on pp. 382, 383)

Markót, M. C., Csendes, T., & Csallner, A. E. (1999). Multisection in interval branch-and-bound methods for global optimization II. Numerical tests. *Journal of Global Optimization*, *16*, 219–228. (Cited on p. 301)

Markót, M. C., Fernandez, J., Casado, L. G., & Csendes, T. (2006). New interval methods for constrained global optimization. *Mathematical Programming*, *106*, 287–318. (Cited on pp. 258, 301)

Marques, J. M. C., Pais, A. A. C. C., & Abreu, P. E. (2010). Generation and characterization of low-energy structures in atomic clusters. *Journal of Computational Chemistry*, *31*(7), 1495–1503. doi: 10.1002/jcc.21436 (Cited on p. 376)

Matheron, G. (1963). Principles of geostatistics. *Economic Geology*, *58*, 1246–1266. (Cited on p. 101)

Matsui, T. (1996). NP-hardness of linear multiplicative programming and related problems. *Journal of Global Optimization*, *9*, 113–119. (Cited on p. 17)

Mayne, D. Q., & Polak, E. (1984). Outer approximation algorithm for nondifferentiable optimization problems. *Journal of Optimization Theory and Applications*, *42*, 19–30. (Cited on p. 253)

McAllister, S., & Floudas, C. A. (2010). An improved hybrid global optimization method for protein tertiary structure prediction. *Computational Optimization and Applications*, *45*, 377–413. doi: 10.1007/s10589-009-9277-y (Cited on p. 371)

McCormick, G. P. (1976). Computability of global solutions to factorable nonconvex programs. I. Convex underestimating problems. *Mathematical Programming*, *10*, 147–175. (Cited on pp. 132, 160, 267)

McCormick, G. P. (1983). *Nonlinear programming: Theory, algorithms and applications*. New York: John Wiley and Sons. (Cited on p. 267)

Meewella, C. C., & Mayne, D. Q. (1988). An algorithm for global optimization of Lipschitz functions. *Journal of Optimization Theory and Applications*, *57*, 307–323. (Cited on p. 254)

Meyer, C. A., & Floudas, C. A. (2004). Convex envelopes of trilinear monomials with positive or negative domains. *Journal of Global Optimization*, *29*, 125–155. (Cited on p. 161)

Meyer, C. A., & Floudas, C. A. (2005a). Convex envelopes for edge-concave functions. *Mathematical Programming B*, *103*, 207–224. (Cited on pp. 133, 134, 160, 162)

Meyer, C. A., & Floudas, C. A. (2005b). Convex underestimation of twice continuously differentiable functions by piecewise quadratic perturbation: Spline αBB underestimators. *Journal of Global Optimization*, *32*, 221–258. (Cited on pp. 230, 232, 233, 236)

Misener, R., & Floudas, C. (2009). Advances for the pooling problem: Modeling, global optimization, and computational studies. *Applied & Computational Mathematics*, *8*, 3–22. (Cited on p. 371)

Misener, R., Thompson, J. P., & Floudas, C. A. (2011). APOGEE: Global optimization of standard, generalized, and extended pooling problems via linear and logarithmic partitioning schemes. *Computers and Chemical Engineering*, *35*, 876–892. (Cited on p. 287)

Mitsos, A., Chachuat, B., & Barton, P. (2009). McCormick-based relaxations of algorithms. *SIAM Journal on Optimization*, *20*, 573–601. (Cited on p. 270)

Mittel, S., & Schulz, A. (2012). An FPTAS for optimizing a class of low-rank functions over a polytope. *Mathematical Programming*, to appear. (Cited on p. 17)

Mladenovic, N., & Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research*, *24*(11), 1097–1100. (Cited on p. 45)

Mladineo, R. H. (1986). An algorithm for finding the global maximum of a multimodal multivariate function. *Mathematical Programming*, *34*, 188–200. (Cited on p. 253)

Mockus, J., Eddy, W., & Reklaitis, G. (1996). *Bayesian heuristic approach to discrete and global optimization: Algorithms, visualization, software, and applications*. Kluwer Academic Publishers. (Cited on p. 117)

Molina, D., Lozano, M., Sànchez, A., & Herrera, F. (2011). Memetic algorithms based on local search chains for large scale continuous optimisation problems: MA-SSW-Chains. *Soft Computing*, *15*, 2201–2220. (Cited on p. 48)

Molinaro, A., Pizzuti, C., & Sergeyev, Y. D. (2001). Acceleration tools for diagonal information global optimization algorithms. *Computational Optimization and Applications*, *18*, 5–26. (Cited on p. 255)

Moore, R. E. (1962). *Interval arithmetic and automatic error analysis in digital computing*. Unpublished doctoral dissertation, Stanford Univeristy, Stanford, CA. (Cited on p. 296)

Moore, R. E. (1966). *Interval analysis*. Englewood Cliffs, NJ: Prentice-Hall. (Cited on p. 257)

Morales, J. L., & Nocedal, J. (2011). Remark on "Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound constrained optimization." *ACM Transactions on Mathematical Software*, *38*(1), 7:1–7:4. (Cited on p. 363)

Moré, J. J., & Wu, Z. (1996). Smoothing techniques for macromolecular global optimization. In G. Di Pillo & F. Giannessi (Eds.), *Nonlinear optimization and applications* (pp. 297–312). Plenum Press. (Cited on p. 79)

Moré, J. J., & Wu, Z. (1997a). Global continuation for distance geometry problems. *SIAM Journal on Optimization*, *7*, 814–836. (Cited on pp. 79, 81)

Moré, J. J., & Wu, Z. (1997b). Issues in large scale global molecular optimization. In L. T. Biegler, T. F. Coleman, A. R. Conn, & F. N. Santosa (Eds.), *Large scale optimization with applications: Part III: Molecular structure and optimization* (pp. 99–121). New York: Springer. (Cited on p. 79)

Morè, J. J., & Wu, Z. (1999). Distance geometry optimization for protein structures. *Journal of Global Optimization*, *15*, 219–234. (Cited on p. 379)

Moscato, P., & Cotta, S. (2010). A modern introduction to memetic algorithms. In M. Gendreau & J.-Y. Potvin (Eds.), *Handbook of Metaheuristics* (second ed., pp. 141–184). Springer. (Cited on p. 48)

Motzkin, T. S., & Strauss, E. G. (1965). Maxima for graphs and a new proof of a theorem of Turán. *Canadian Journal of Mathematics*, *17*, 533–540. (Cited on pp. 23, 28)

Mucherino, A., Lavor, C., Liberti, L., & Maculan, N. (Eds.). (2013). *Distance geometry – Theory, methods, and applications*. Springer. (Cited on p. 379)

Myatt, D., Becerra, V., Nasuto, S., & Bishop, J. (2004). *Advanced global optimisation for mission analysis and design* (Tech. Rep. No. 18138/04/NL/MV). ESA. (Cited on p. 387)

Nast, M. (1996). Subdivision of simplices relative to a cutting plane and finite concave minimization. *Journal of Global Optimization*, *9*, 65–93. (Cited on pp. 302, 361)

Nataraj, P. S. V., & Arounassalame, M. (2011). Constrained global optimization of multivariate polynomials using Bernstein branch and prune algorithm. *Journal of Global Optimization*, *49*, 185–212. (Cited on p. 230)

Nataraj, P. S. V., & Kotecha, K. (2002). An algorithm for global optimization using the Taylor-Bernstein form as inclusion function. *Journal of Global Optimization*, *24*, 417–436. (Cited on p. 258)

Nataraj, P. S. V., & Kotecha, K. (2004). Global optimization with higher order inclusion function forms. Part 1: A combined Taylor-Bernstein form. *Reliable Computing*, *10*, 27–44. (Cited on p. 258)

Nataraj, P. S. V., & Kotecha, K. (2005). An improved interval global optimization algorithm using higher-order inclusion function forms. *Journal of Global Optimization*, *32*, 35–63. (Cited on p. 258)

Nedić, A., & Ozdaglar, A. (2008). A geometric framework for nonconvex optimization duality using augmented Lagrangian functions. *Journal of Global Optimization*, *40*, 545–573. (Cited on p. 267)

Nemhauser, G. L., & Wolsey, L. A. (1988). *Integer and combinatorial optimization*. John Wiley & Sons, New York. (Cited on pp. 21, 359)

Nemirovski, A., & Yudin, D. B. (1983). *Problem complexity and method efficiency in optimization*. Wiley-Interscience, New York. (Cited on p. 9)

Nesterov, Y. (1997). *Quality of semidefinite relaxation for nonconvex quadratic optimization* (Tech. Rep. No. 9719). CORE - Université Catholique de Louvain. (Cited on p. 33)

Nesterov, Y. (2003). *Random walk in a simplex and quadratic optimization over convex polytopes* (Tech. Rep. No. 2003/71). CORE-UCL. (Cited on p. 26)

Nesterov, Y., Wolkowicz, H., & Ye, Y. (2000). Semidefinite programming relaxations of nonconvex quadratic optimization. In H. Wolkowicz, R. Saigal, & L. Vandenberghe (Eds.), (Vol. Handbook of Semidefinite Programming, p. 361–419). Kluwer Academic Publishers. (Cited on p. 32)

Neumaier, A. (1990). *Interval methods for systems of equations*. Cambridge University Press, Cambridge. (Cited on p. 258)

Neumaier, A. (2004). Complete search in continuous global optimization and constraint satisfaction. *Acta Numerica*, 271–369. Cambridge University Press. (Cited on pp. 259, 348, 349, 358)

Neumaier, A., & Shcherbina, O. (2004). Safe bounds in linear and mixed-integer programming. *Mathematical Programming*, *99*, 283–296. (Cited on p. 287)

Neumaier, A., Shcherbina, O., Huyer, W., & Vinkó, T. (2005). A comparison of complete global optimization solvers. *Mathematical Programming*, *103*(2), 335–356. doi: 10.1007/s10107-005-0585-4 (Cited on p. 335)

Nie, J., Demmel, J., & Gu, M. (2008). Global minimization of rational functions and the nearest GCDs. *Journal of Global Optimization*, *40*, 697–718. (Cited on p. 230)

Nie, J., Demmel, J., & Sturmfels, B. (2006). Minimizing polynomials via sum of squares over the gradient ideal. *Mathematical Programming*, *106*, 587–606. (Cited on pp. 223, 224, 225)

Nowak, I. (2000). Dual bounds and optimality cuts for all-quadratic programs with convex constraints. *Journal of Global Optimization*, *18*, 337–356. (Cited on p. 358)

Nowak, I. (2005). Lagrangian decomposition of block-separable mixed-integer all-quadratic programs. *Mathematical Programming*, *102*, 295–312. (Cited on p. 265)

Nurmela, K. J., & Oestergard, P. R. J. (1997). Packing up to 50 equal circles in a square. *Discrete Computational Geometry*, *18*, 111–120. (Cited on pp. 383, 385)

Nurmela, K. J., & Oestergard, P. R. J. (1999). More optimal packings of equal circles in a square. *Discrete Computational Geometry*, *22*, 439–457. (Cited on pp. 382, 383)

Olympio, J. T., & Marmorat, J.-P. (2008). *Global trajectory optimization: Can we prune the solution space when considering deep space manoeuvres?* (Final Report). ESA. (Cited on p. 386)

Padberg, M. W. (1989). The boolean quadric polytope: some characteristics, facets and relatives. *Mathematical Programming*, *45*, 139–172. (Cited on p. 204)

Pang, J. S. (1997). Error bounds in mathematical programming. *Mathematical Programming*, *79*, 299–332. (Cited on p. 267)

Papadimitriou, C. H., & Steiglitz, K. (1998). *Combinatorial optimization. Algorithms and complexity*. Dover Publications. (Cited on pp. 7, 8, 361)

Pardalos, P. M., & Vavasis, S. A. (1991). Quadratic programming with one negative eigenvalue is NP-hard. *Journal of Global Optimization*, *1*, 15–22. (Cited on p. 17)

Parrillo, P. A. (2000). *Structured semidefinite programs and semi-algebraic geometry methods in robustness and optimization*. Unpublished doctoral dissertation, California Institute of Technology, Pasadena. (Cited on p. 182)

Peña, J., Vera, J., & Zuluaga, L. F. (2007). Computing the stability number of a graph via linear and semidefinite programming. *SIAM Journal on Optimization*, *18*, 87–105. (Cited on p. 182)

Peña, J., Vera, J. C., & Zuluaga, L. F. (2011). *Positive polynomials on unbounded equality-constrained domains.* Retrieved from `http://www.optimization-online.org/DB_FILE/2011/05/3047.pdf` (unpublished) (Cited on p. 180)

Pinter, J. D. (1986). Globally convergent methods for n-dimensional multiextremal optimization. *Optimization*, *17*, 187–202. (Cited on p. 254)

Piyavskii, S. A. (1972). An algorithm for finding the absolute extremum of a function. *USSR Computational Mathematics and Mathematical Physics*, *12*, 57–67. (Cited on p. 253)

Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization. *Swarm Intelligence*, *1*(1), 33–57. doi: 10.1007/s11721-007-0002-0 (Cited on p. 63)

Polya, G. (1974). Über positive Darstellung von Polynomen. *Collected papers - MIT Press*, *2*. (Original paper appeared in 1928) (Cited on p. 182)

Polyak, B. T. (1987). *Introduction to optimization.* Optimization Software. Inc., Publication Division, New York. (Cited on p. 322)

Polyak, B. T. (1998). Convexity of quadratic transformations and its use in control and optimization. *Journal of Optimization Theory and Applications*, *99*, 553–583. (Cited on p. 14)

Pong, T., & Tseng, P. (2011). (Robust) Edge-based semidefinite programming relaxation of sensor network localization. *Mathematical Programming*, *130*(2), 321–358. (Cited on p. 380)

Porembski, M. (2001). Finitely convergent cutting planes for concave minimization. *Journal of Global Optimization*, *20*, 113–136. (Cited on pp. 356, 357)

Povh, J., & Rendl, F. (2007). A copositive programming approach to graph partitioning. *SIAM Journal on Optimization*, *18*, 223–241. (Cited on p. 174)

Povh, J., & Rendl, F. (2009). Copositive and semidefinite relaxations of the quadratic assignment problem. *Discrete Optimization*, *6*, 231–241. (Cited on p. 174)

Prentiss, M. C., Wales, D. J., & Wolynes, P. G. (2008). Protein structure prediction using basin-hopping. *The Journal of Chemical Physics*, *128*(22), 225106. (Cited on p. 371)

Pullan, W. (2005). An unbiased population-based search for the geometry optimization of Lennard–Jones clusters: $2 \leq N \leq 372$. *Journal of Computational Chemistry*, *26*(9), 899–906. doi: 10.1002/jcc.20226 (Cited on p. 62)

Raber, U. (1998). A simplicial branch-and-bound method for solving nonconvex all-quadratic programs. *Journal of Global Optimization*, *13*, 417–432. (Cited on pp. 212, 302)

Ratschek, H., & Rokne, J. (1995). Interval methods. In R. Horst & P. M. Pardalos (Eds.), (Vol. Handbook of Global Optimization, p. 751–828). Dordrecht: Kluwer Academic Publishers. (Cited on pp. 259, 358)

Renpu, G. (1990). A filled function method for finding a global minimizer of a function of several variables. *Mathematical Programming*, *46*, 191–204. (Cited on p. 77)

Resende, M. G., Ribeiro, C. C., Glover, F., & Marti, R. (2010). Scatter search and path-relinking: Fundamentals, advances, and applications. In M. Gendreau & J.-Y. Potvin (Eds.), *Handbook of Metaheuristics* (pp. 87–107). Springer Science+Business Media. (Cited on p. 84)

Rikun, A. D. (1997). A convex envelope formula for multilinear functions. *Journal of Global Optimization*, *10*, 425–437. (Cited on pp. 129, 131, 132, 160, 162)

Rinnooy Kan, A. H. G., & Timmer, G. T. (1987a). Stochastic global optimization methods. Part I: Clustering methods. *Mathematical Programming*, *39*, 27–56. (Cited on p. 52)

Rinnooy Kan, A. H. G., & Timmer, G. T. (1987b). Stochastic global optimization methods. Part II: Multi level methods. *Mathematical Programming*, *39*, 57–78. (Cited on p. 52)

Ripoll, D. R., Liwo, A., & Scheraga, H. A. (2009). Global optimization in protein folding. In C. A. Floudas & P. M. Pardalos (Eds.), (Vol. Encyclopedia of Optimization, pp. 1392–1411). Springer. doi: 10.1007/978-0-387-74759-0_246 (Cited on p. 371)

Roberts, C., Johnston, R. L., & Wilson, N. T. (2000). A genetic algorithm for the structural optimization of Morse clusters. *Theoretical Chemistry Accounts: Theory, Computation, and Modeling (Theoretica Chimica Acta)*, *104*(2), 123–130. (Cited on p. 73)

Rockafellar, R. T. (1970). *Convex analysis*. Princeton University Press, Princeton, NJ. (Cited on pp. 11, 128, 260, 392, 396)

Rockafellar, R. T. (1971). Integrals which are convex functionals. *Pacific Journal of Mathematics*, *39*, 439–469. (Cited on p. 343)

Rote, G. (1992). The convergence rate of the sandwich algorithm for approximating convex functions. *Computing*, *48*, 337–361. (Cited on p. 272)

Rubinov, A., Glover, B., & Yang, X. (1999). Decreasing functions with applications to penalization. *SIAM Journal on Optimization*, *10*, 289–313. (Cited on p. 267)

Rubinov, A., & Wu, Z. (2009). Optimality conditions in global optimization and their applications. *Mathematical Programming B*, *120*, 101–123. (Cited on p. 4)

Rubinstein, R. Y. (1983). Smoothed functionals in stochastic optimization. *Mathematics of Operations Research*, *8*(1), 26–33. (Cited on p. 79)

Ryoo, H., & Sahinidis, N. V. (1996). A branch-and-reduce approach to global optimization. *Journal of Global Optimization*, *8*, 107–139. (Cited on p. 335)

Ryoo, H., & Sahinidis, N. V. (2001). Analysis of bounds for multilinear functions. *Journal of Global Optimization*, *19*, 403–424. (Cited on p. 286)

Sacks, J., Welch, W. J., Mitchell, T. J., & Wynn, H. P. (1989). Design and analysis of computer experiments. *Statistical Science*, *4*, 409–435. (Cited on pp. 103, 106)

Sahinidis, N. V. (1996). BARON: A general purpose global optimization software package. *Journal of Global Optimization*, *8*, 201–205. (Cited on p. 335)

Sahni, S. (1974). Computationally related problems. *SIAM Journal on Computing*, *3*, 262–279. (Cited on p. 16)

Saxena, A., Bonami, P., & Lee, J. (2010). Convex relaxations of non-convex mixed integer quadratically constrained programs: Extended formulations. *Mathematical Programming*, *124*, 383–411. (Cited on p. 286)

Schaback, R. (1993). Comparison of radial basis function interpolants. In K. Jetter & F. Utreras (Eds.), *Multivariate approximations: From CAGD to wavelets* (pp. 293–305). Singapore: World Scientific. (Cited on p. 101)

Schachinger, W., Addis, B., Bomze, I. M., & Schoen, F. (2007). New results for molecular formation under pairwise potential minimization. *Computational Optimization and Applications*, *38*(3), 329–349. (Cited on p. 374)

Schaer, J. (1965). The densest packing of nine circles in a square. *Canadian Mathematical Bulletin*, *8*, 273–277. (Cited on p. 382)

Schaer, J., & Meir, A. (1965). On a geometric extremum problem. *Canadian Mathematical Bulletin*, *8*, 21–27. (Cited on p. 382)

Schaible, S. (1995). Fractional programming. In R. Horst & P. M. Pardalos (Eds.), (Vol. Handbook of Global Optimization, p. 495–608). Kluwer Academic Publishers. (Cited on p. 286)

Schaible, S., & Ibaraki, T. (1983). Fractional programming. *European Journal of Operational Research*, *12*, 325–338. (Cited on p. 18)

Schaible, S., & Shi, J. (2003a). Fractional programming : The sum-of-ratios case. *Optimization Methods and Software*, *18*, 219–229. (Cited on p. 18)

Schaible, S., & Shi, J. (2003b). Fractional programming—the sum of ratios case. *Optimization Methods and Software*, *18*, 219–229. (Cited on p. 286)

Schectman, J. P., & Sahinidis, N. V. (1998). A finite algorithm for global minimization of separable concave programs. *Journal of Global Optimization*, *12*, 1–36. (Cited on pp. 298, 361)

Schoen, F. (1982). On a sequential search strategy in global optimization problems. *Calcolo*, *XIX*, 321–334. (Cited on p. 253)

Schoen, F. (1993). A wide class of test functions for global optimization. *Journal of Global Optimization*, *3*, 133–138. (Cited on p. 369)

Scholz, D. (2012). Theoretical rate of convergence for interval inclusion functions. *Journal of Global Optimization*, *53*, 749–767. (Cited on p. 258)

Schön, J. (1997). Preferential trapping on energy landscapes in regions containing deep-lying minima: The reason for the success of simulated annealing? *Journal of Physics A: Mathematical and General*, *30*, 2367–2389. (Cited on p. 75)

Schwartz, B. L. (1970). Separating points in a square. *Journal of Recreational Mathematics*, *3*, 195–204. (Cited on p. 382)

Schweighofer, M. (2006). Global optimization of polynomials using gradient tentacles and sums of squares. *SIAM Journal on Optimization*, *17*(3), 920–942. (Cited on p. 225)

Scott, J. K., Stuber, M. D., & Barton, P. I. (2011). Generalized McCormick relaxations. *Journal of Global Optimization*, *51*, 569–606. (Cited on p. 267)

Sergeyev, Y. D. (1995). An information global optimization algorithm with local tuning. *SIAM Journal on Optimization*, *5*, 858–870. (Cited on p. 255)

Sergeyev, Y. D., Famularo, D., & Pugliese, P. (2001). Index branch-and-bound algorithm for Lipschitz univariate global optimization with multiextremal constraints. *Journal of Global Optimization*, *21*, 317–341. (Cited on p. 253)

Sergeyev, Y. D., & Kvasov, D. E. (2006). Global search based on efficient diagonal partitions and a set of lipschitz constants. *SIAM Journal on Optimization*, *16*, 910–937. (Cited on p. 255)

Sharkey, T., Romeijn, H., & Geunes, J. (2011). A class of nonlinear nonseparable continuous knapsack and multiple-choice knapsack problems. *Mathematical Programming*, *126*, 69–96. (Cited on p. 40)

Shelokar, P. S., Siarry, P., Jayaraman, V. K., & Kulkarni, B. D. (2007). Particle swarm and ant colony algorithms hybridized for improved continuous optimization. *Applied Mathematics and Computation*, *188*(1), 129–142. (Cited on p. 84)

Shen, P., Ma, Y., & Chen, Y. (2011). Global optimization for the generalized polynomial sum of ratios problem. *Journal of Global Optimization*, *50*, 439–455. (Cited on p. 346)

Sherali, H. D. (1997). Convex envelopes of multilinear functions over a unit hypercube and over special discrete sets. *Acta Mathematica Vietnamica*, *22*, 245–270. (Cited on p. 162)

Sherali, H. D. (2002). Tight relaxations for nonconvex optimization problems using the reformulation-linearization/convexification technique. In P. M. Pardalos & H. E. Romeijn (Eds.), (Vol. Handbook of Global Optimization - Vol. 2, p. 1–64). Kluwer Academic Publishers. (Cited on pp. 163, 165, 170)

Sherali, H. D., & Adams, W. P. (1990). A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, *3*, 411–430. (Cited on p. 170)

Sherali, H. D., & Adams, W. P. (1994). A hierarchy of relaxations and convex hull characterizations for mixed-integer zero-one programming problems. *Discrete Applied Mathematics*, *52*, 83–106. (Cited on p. 170)

Sherali, H. D., & Alameddine, A. (1992). An explicit characterization of the convex envelope of a bivariate bilinear function over special polytopes. *Annals of Operations Research*, *27*, 197–210. (Cited on p. 161)

Sherali, H. D., & Fraticelli, B. M. (2002). Enhancing RLT relaxations via a new class of semidefinite cuts. *Journal of Global Optimization*, *22*, 233–261. (Cited on pp. 163, 166)

Sherali, H. D., Lee, Y., & Adams, W. P. (1995). A simultaneous lifting strategy for identifying new classes of facets for the boolean quadric polytope. *Operations Research Letters*, *17*, 19–26. (Cited on p. 205)

Sherali, H. D., & Tuncbilek, C. H. (1992). A global optimization algorithm for polynomial programming problems using a reformulation-linearization technique. *Journal of Global Optimization*, *2*, 101–112. (Cited on pp. 163, 164)

Sherali, H. D., & Tuncbilek, C. H. (1995). A reformulation-convexification approach for solving nonconvex quadratic programming problems. *Journal of Global Optimization*, *7*, 1–31. (Cited on pp. 163, 169)

Sherali, H. D., & Wang, H. (2001). Global optimization of nonconvex factorable programming problems. *Mathematical Programming*, *89*, 459–478. (Cited on p. 271)

Shor, N. (1987). Quadratic optimization problems. *Soviet Journal of Computer and Systems Sciences*, *25*, 1–11. (Cited on p. 201)

Shor, N. Z. (1977). Cut-off method with space extension in convex programming problems. *Cybernetics*, *13*, 94–96. (Cited on p. 322)

Shor, N. Z. (1992). Dual estimates in multiextremal problems. *Journal of Global Optimization*, *2*, 411–418. (Cited on p. 262)

Shubert, B. O. (1972). A sequential method seeking the global maximum of a function. *SIAM Journal on Numerical Analysis*, *9*, 379–388. (Cited on p. 253)

Sima, D., Van Huffel, S., & Golub, G. H. (2004). Regularized total least squares based on quadratic eigenvalue problem solvers. *BIT Numerical Mathematics*, *44*, 793–812. (Cited on p. 14)

Skelboe, S. (1974). Computation of rational interval functions. *BIT*, *14*, 87–95. (Cited on p. 296)

Smith, E. M. B., & Pantelides, C. C. (1999). A symbolic reformulation/spatial branch-and-bound algorithm for the global optimization of nonconvex MINLPs. *Computers and Chemical Engineering*, *23*, 457–478. (Cited on p. 270)

So, A. M. C. (2011). Deterministic approximation algorithms for sphere constrained homogeneous polynomial optimization problems. *Mathematical Programming B*, *129*, 357–382. (Cited on p. 37)

Socha, K., & Dorigo, M. (2008). Ant colony optimization for continuous domains. *European Journal of Operational Research*, *185*(3), 1155–1173. (Cited on p. 84)

Storn, R., & Price, K. (1997). Differential evolution. A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, *11*(4), 341–359. (Cited on p. 66)

Strongin, R. G. (1973). On the convergence of an algorithm for finding a global extremum. *Engineering Cybernetics*, *11*, 549–555. (Cited on p. 253)

Sun, X. L., & Li, J. L. (2006). A branch-and-bound based method for solving monotone optimization problems. *Journal of Global Optimization*, *35*, 367–385. (Cited on p. 325)

Szabó, P. G., Markót, M. C., & Csendes, T. (2005). Global optimization in geometry - Circle packing into the square. In C. Audet, P. Hansen, & G. Savard (Eds.), *Essays and Surveys in Global Optimization* (pp. 233–266). Kluwer Academic Publishers. (Cited on p. 383)

Szabó, P. G., Markót, M. C., Csendes, T., Specht, E., Casado, L. G., & Garcia, I. (2007). *New approaches to circle packing in a square with program codes*. Springer. (Cited on p. 383)

Tam, B. T., & Ban, V. T. (1985). Minimization of a concave function under linear constraints. *Economika i Mathematicheskie Metody*, *21*, 709–714. (Cited on pp. 302, 361)

Tang, K., Li, X., Suganthan, P. N., Yang, Z., & Weise, T. (2010). *Benchmark functions for the CEC'2010 special session and competition on large-scale global optimization* (Tech. Rep.). Nature Inspired Computation and Applications Laboratory (NICAL), School of Computer Science and Technology, University of Science and Technology of China (USTC). (Cited on p. 369)

Tardella, F. (2003). On the existence of polyhedral convex envelopes. In C. A. Floudas & P. M. Pardalos (Eds.), (Vol. Frontiers in Global Optimization, p. 563–574). Kluwer Academic Publishers. (Cited on p. 132)

Tardella, F. (2008). Existence and sum decomposition of vertex polyhedral convex envelopes. *Optimization Letters*, *2*, 363–375. doi: 10.1007/s11590-007-0065-2 (Cited on pp. 132, 158, 160)

Tardos, G. (1994). Multi-prover encoding schemes and three prover proof systems. In *Proceedings of the ninth annual conference on structure in complexity theory, IEEE.* (Cited on p. 29)

Tawarmalani, M., Richard, J. P. P., & Chung, K. (2010). Strong valid inequalities for orthogonal disjunctions and bilinear covering sets. *Mathematical Programming*, *124*, 481–512. (Cited on pp. 275, 276, 277, 278, 281, 284, 285)

Tawarmalani, M., Richard, J. P. P., & Xiong, C. (2012). Explicit convex and concave envelopes through polyhedral subdivisions. *Mathematical Programming*, to appear. (Cited on pp. 138, 155, 156)

Tawarmalani, M., & Sahinidis, N. V. (2001). Semidefinite relaxations of fractional programs via novel convexification techniques. *Journal of Global Optimization*, *20*, 137–158. (Cited on pp. 139, 161)

Tawarmalani, M., & Sahinidis, N. V. (2002a). Convex extensions and envelopes of lower semicontinuous functions. *Mathematical Programming*, *93*, 247–263. (Cited on p. 129)

Tawarmalani, M., & Sahinidis, N. V. (2002b). *Convexification and global optimization in continuous and mixed-integer nonlinear programming: Theory, algorithms, software and applications*. Kluwer Academic Publishers, Dordrecht. (Cited on p. 346)

Tawarmalani, M., & Sahinidis, N. V. (2004). Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. *Mathematical Programming*, *99*(3), 563–591. (Cited on pp. 271, 273, 338)

Tawarmalani, M., & Sahinidis, N. V. (2005). A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, *103*, 225–249. (Cited on pp. 273, 274)

Thoai, N. V. (2000a). Conical algorithm in global optimization for optimizing over efficient sets. *Journal of Global Optimization*, *18*, 321–336. (Cited on p. 309)

Thoai, N. V. (2000b). Duality bound method for the general quadratic programming problem with quadratic constraints. *Journal of Optimization Theory and Applications*, *107*, 331–354. (Cited on p. 265)

Thoai, N. V. (2002). Convergence and application of a decomposition method using duality bounds for nonconvex global optimization. *Journal of Optimization Theory and Applications*, *113*, 165–193. (Cited on p. 265)

Thoai, N. V., & Tuy, H. (1980). Convergent algorithm for minimizing a concave function. *Mathematics of Operations Research*, *5*, 556–566. (Cited on p. 314)

Thoai, N. V., Yamamoto, Y., & Yoshise, A. (2005). Global optimization method for solving mathematical programs with linear complementarity constraints. *Journal of Optimization Theory and Applications*, *124*, 467–490. (Cited on p. 286)

Timonov, L. N. (1977). An algorithm for search of a global extremum. *Engineering Cybernetics*, *15*, 38–44. (Cited on p. 253)

Tóth, B., & Casado, L. (2007). Multi-dimensional pruning from the Baumann point in an interval global optimization algorithm. *Journal of Global Optimization*, *38*, 215–236. (Cited on p. 346)

Tseng, P. (2007). Second-order cone programming relaxation of sensor network localization. *SIAM Journal on Optimization*, *18*(1), 156–185. (Cited on p. 380)

Turkay, M., & Grossmann, I. E. (1996). Disjunctive programming techniques for the optimization of process systems with discontinuous investment costs-multiple size regions. *Industrial & Engineering Chemistry Research*, *35*, 2611–2623. (Cited on p. 286)

Tuy, H. (1964). Concave programming under linear constraints. *Soviet Mathematics*, *5*, 1437–1440. (Cited on pp. 299, 309, 314)

Tuy, H. (1991a). Effect of the subdivision strategy on convergence and efficiency of some global optimization algorithms. *Journal of Global Optimization*, *1*, 23–36. (Cited on p. 298)

Tuy, H. (1991b). Normal conical algorithm for concave minimization over polytopes. *Mathematical Programming*, *51*, 229–245. (Cited on pp. 309, 319, 331)

Tuy, H. (1995). D.C. optimization: Theory, methods and algorithms. In R. Horst & P. M. Pardalos (Eds.), *Handbook of Global Optimization* (pp. 149–216). Dordrecht: Kluwer. (Cited on p. 242)

Tuy, H. (2000). Monotonic optimization: Problems and solution approaches. *SIAM Journal on Optimization*, *11*(2), 464–494. (Cited on pp. 252, 324)

Tuy, H. (2005a). On solving nonconvex optimization problems by reducing the duality gap. *Journal of Global Optimization*, *32*, 349–365. (Cited on pp. 265, 331)

Tuy, H. (2005b). Robust solution of nonconvex global optimization problems. *Journal of Global Optimization*, *32*, 307–323. (Cited on p. 334)

Tuy, H. (2010). D(C)-optimization and robust global optimization. *Journal of Global Optimization*, *47*, 485–501. (Cited on p. 334)

Tuy, H., & Hoai-Phuong, N. T. (2007). A robust algorithm for quadratic optimization under quadratic constraints. *Journal of Global Optimization*, *37*, 557–569. (Cited on p. 334)

Tuy, H., Khachaturov, V., & Utkin, S. (1987). A class of exhaustive cone splitting procedures in conical algorithms for concave minimization. *Optimization*, *18*, 791–808. (Cited on p. 309)

Tuy, H., & Luc, L. T. (2000). A new approach to optimization under monotonic constraint. *Journal of Global Optimization*, *18*, 1–15. (Cited on pp. 252, 324)

Vandenbussche, D., & Nemhauser, G. L. (2005a). A branch-and-cut algorithm for nonconvex quadratic programs with box constraints. *Mathematical Programming*, *102*, 559–575. (Cited on pp. 187, 191, 192, 201, 360, 371)

Vandenbussche, D., & Nemhauser, G. L. (2005b). A polyhedral study of nonconvex quadratic programs with box constraints. *Mathematical Programming*, *102*, 531–557. (Cited on pp. 187, 188, 189, 360)

Van Der Spoel, D., Lindahl, E., Hess, B., Groenhof, G., Mark, A. E., & Berendsen, H. J. C. (2005). Gromacs: Fast, flexible, and free. *Journal of Computational Chemistry*, *26*(16), 1701–1718. doi: 10.1002/jcc.20291 (Cited on p. 372)

Van Voorhis, T. (2002). A global optimization algorithm using Lagrangian underestimates and the interval Newton method. *Journal of Global Optimization*, *24*, 349–370. (Cited on p. 265)

Vasile, M., Minisci, E., & Locatelli, M. (2011). An inflationary differential evolution algorithm for space trajectory optimization. *IEEE Transactions on Evolutionary Computation*, *15*(2), 267–281. (Cited on pp. 68, 386)

Vavasis, S. A. (1992). Approximation algorithms for indefinite quadratic programming. *Mathematical Programming*, *57*, 279–311. (Cited on pp. 16, 17, 40)

Vavasis, S. A. (1995). Complexity issues in global optimization : A survey. In R. Horst & P. M. Pardalos (Eds.), (Vol. Handbook of Global Optimization, p. 27–41). Kluwer Academic Publishers. (Cited on p. 11)

Vavasis, S. A., & Zippel, R. (1990). *Proving polynomial-time for sphere constrained quadratic programming* (Tech. Rep. No. 90-1182). Department of Computer Science - Cornell University, Ithaca, New York. (Cited on p. 12)

Vaz, I. F., & Vicente, L. N. (2007). A particle swarm pattern search method for bound constrained global optimization. *Journal of Global Optimization*, *39*, 197–219. doi: 10.1007/s10898-007-9133-5 (Cited on pp. 63, 66, 68)

Vielma, J. P., Ahmed, S., & Nemhauser, G. (2010). Mixed-integer models for nonseparable piecewise linear optimization: unifying framework and extensions. *Operations Research*, *58*, 303–315. (Cited on p. 287)

Vielma, J. P., & Nemhauser, G. (2011). Modeling disjunctive constraints with a logarithmic number of binary variables and constraints. *Mathematical Programming*, *128*, 49–72. (Cited on p. 287)

Vui, H. H., & So'n, P. T. (2008). Global optimization of polynomials using the truncated tangency variety and sums of squares. *SIAM Journal on Optimization*, *19*(2), 941–951. (Cited on p. 226)

Wales, D. J. (2003). *Energy landscapes*. Cambridge, U.K.: Cambridge University Press. (Cited on p. 375)

Wales, D. J., & Doye, J. P. K. (1997). Global optimization by Basin-Hopping and the lowest energy structures of Lennard–Jones clusters containing up to 110 atoms. *Journal of Physical Chemistry A*, *101*(28), 5111–5116. (Cited on pp. 45, 57, 376)

Wales, D. J., & Scheraga, H. A. (1999). Global optimization of clusters, crystals, and biomolecules. *Science*, *285*, 1368–1372. (Cited on p. 57)

Wang, C. Y., & Li, D. (2009). Unified theory of augmented Lagrangian methods for constrained global optimization. *Journal of Global Optimization*, *44*, 433–458. (Cited on p. 266)

Wang, H., Huang, W., Zhang, Q., & Xu, D. (2002). An improved algorithm for the packing of unequal circles within a larger containing circle. *European Journal of Operational Research*, *141*, 440–453. (Cited on p. 383)

Wang, Y., Zhang, K., & Gao, Y. (2004). Global optimization of generalized geometric programming. *Computers & Mathematics with Applications*, *48*, 1505–1516. (Cited on p. 286)

Watson, L. T. (1986). Numerical linear algebra aspects of globally convergent homotopy methods. *SIAM Review*, *28*(4), 529–545. (Cited on p. 83)

Watson, L. T. (2001). Theory of globally convergent probability-one homotopies for nonlinear programming. *SIAM Journal on Optimization*, *11*(3), 761–780. (Cited on p. 83)

Wendland, H. (2005). *Scattered data approximation*. Cambridge, U.K.: Cambridge University Press. (Cited on pp. 89, 93)

Wengerodt, G. (1983). Die dichteste Packung von 16 Kreisen in einem Quadrat. *Beiträge Algebra Geometrie*, *16*, 173–190. (Cited on p. 382)

Wengerodt, G. (1987a). Die dichteste Packung von 14 Kreisen in einem Quadrat. *Beiträge Algebra Geometrie*, *25*, 25–46. (Cited on p. 382)

Wengerodt, G. (1987b). Die dichteste Packung von 25 Kreisen in einem Quadrat. *Ann. Univ. Sci. Budapest Eötvös Sect. Math.*, *30*, 3–15. (Cited on p. 382)

Wengerodt, G., & Kirchner, K. (1987). Die dichteste Packung von 36 Kreisen in einem Quadrat. *Beiträge Algebra Geometrie*, *25*, 147–159. (Cited on p. 382)

Wets, R. J.-B. (1974). On inf-compact mathematical programs. *Lecture Notes in Computer Science*, *5*, 426–436. (Cited on p. 343)

Wicaksono, D. S., & Karimi, I. A. (2008). Piecewise MILP under- and overestimators for global optimization of bilinear programs. *AIChE J.*, *54*, 991–1008. (Cited on p. 287)

Wood, G. R. (1991). Multidimensional bisection applied to global optimization. *Computers and Mathematics with Applications*, *21*, 161–172. (Cited on pp. 254, 302)

Wood, G. R., & Zabinsky, Z. B. (2002). Stochastic adaptive search. In P. M. Pardalos & H. E. Romeijn (Eds.), *Handbook of Global Optimization - Vol. 2* (pp. 231–249). The Netherlands: Kluwer Academic Publishers. (Cited on p. 78)

Wu, Z., Bai, F., Lee, H., & Yang, Y. (2007). A filled function method for constrained global optimization. *Journal of Global Optimization*, *39*, 495–507. (Cited on p. 77)

Wu, Z., Lee, H., Zhang, L., & Yang, X. (2006). A novel filled function method and quasi-filled function method for global optimization. *Computational Optimization and Applications*, *34*, 249–272. (Cited on p. 77)

Xia, Y., Sun, X., Li, D., & Zheng, X. (2011). On the reduction of duality gap in box constrained nonconvex quadratic program. *SIAM Journal on Optimization*, *21*(3), 706–729. (Cited on p. 264)

Xue, G. L. (1994). Improvements on the northby algorithm for molecular conformation: Better solutions. *Journal of Global Optimization*, *4*(4), 425–440. (Cited on p. 374)

Xue, G. L. (1997). Minimum inter-particle distance at global minimizers of Lennard-Jones clusters. *Journal of Global Optimization*, *11*(1), 83–90. (Cited on p. 374)

Yajima, Y., & Fujie, T. (1998). A polyhedral approach for nonconvex quadratic programming problems with box constraints. *Journal of Global Optimization*, *13*, 151–170. (Cited on pp. 204, 205)

Yamada, S., Tanino, T., & Inuiguchi, M. (2000). Inner approximation method for a reverse convex programming problem. *Journal of Optimization Theory and Applications*, *107*, 355–389. (Cited on p. 243)

Yang, X., & Huang, X. (2001). A nonlinear Lagrangian approach to constraint optimization problems. *SIAM Journal on Optimization*, *11*, 1119Ű1144. (Cited on p. 267)

Yang, X., & Sun, M. (2007). Theoretical convergence analysis of a general division-deletion algorithm for solving global search problems. *Journal of Global Optimization*, *37*, 27–45. (Cited on p. 331)

Yang, Y., & Yang, Q. (2012). On solving biquadratic optimization via semidefinite relaxation. *Computational Optimization and Applications*, *53*, 845–867. (Cited on p. 37)

Ye, Y. (1999). Approximating quadratic programming with bound and quadratic constraints. *Mathematical Programming*, *84*, 219–226. (Cited on pp. 30, 38)

Ye, Y., & Zhang, S. (2003). New results on quadratic minimization. *SIAM Journal on Optimization*, *14*, 245–267. (Cited on pp. 16, 267)

Zabinsky, Z. B., & Smith, R. L. (1992). Pure adaptive search in global optimization. *Mathematical Programming*, *53*, 323–338. (Cited on p. 78)

Zabinsky, Z. B., & Wood, G. R. (2002). Implementation of stochastic adaptive search with Hit-and-Run as a generator. In P. M. Pardalos & H. E. Romeijn (Eds.), *Handbook of Global Optimization - Vol. 2* (pp. 251–273). The Netherlands: Kluwer Academic Publihers. (Cited on p. 79)

Zaharie, D. (2002). Critical values fo the control parameters of differential evolution. In R. Matoušek & P. Ošmera (Eds.), *Proceedings of Mendel 2002, 8th International Conference on Soft Computing* (pp. 62–67). (Cited on p. 68)

Zamora, J. M., & Grossmann, I. E. (1999). A branch and contract algorithm for problems with concave univariate, bilinear and linear fractional terms. *Journal of Global Optimization*, *14*, 217–249. (Cited on pp. 161, 335, 338)

Zeng, G., & Xiao, S. (2012). Global minimization of multivariate polynomials using nonstandard methods. *Journal of Global Optimization*, *53*, 391–415. (Cited on p. 230)

Zhang, J. F., & Kwong, C. P. (2005). Some applications of a polynomial inequality to global optimization. *Journal of Optimization Theory and Applications*, *127*, 193–205. (Cited on p. 230)

Zhang, S. (2000). Quadratic maximization and semidefinite relaxation. *Mathematical Programming*, *87*, 453–465. (Cited on p. 267)

Zhang, X., Ling, C., & Qi, L. (2011). Semidefinite relaxation bounds for bi-quadratic optimization problems with quadratic constraints. *Journal of Global Optimization*, *49*, 293–311. (Cited on p. 37)

Zheng, X. J., Sun, M., Li, D., & Xu, Y. F. (2012). On zero duality gap in nonconvex quadratic programming problems. *Journal of Global Optimization*, *52*, 229–242. (Cited on pp. 265, 267)

Zheng, X. J., Sun, X. L., & Li, D. (2011a). Convex relaxations for nonconvex quadratically constrained quadratic programming: matrix cone decomposition and polyhedral approximation. *Mathematical Programming B*, *129*, 301–329. (Cited on pp. 213, 214, 216, 217, 249)

Zheng, X. J., Sun, X. L., & Li, D. (2011b). Nonconvex quadratically constrained quadratic programming: Best D.C. decompositions and their SDP representations. *Journal of Global Optimization*, *50*, 695–712. (Cited on pp. 213, 249, 250)

Zhu, Y., & Kuno, T. (2005). A global optimization method, QBB, for twice-differentiable nonconvex optimization problem. *Journal of Global Optimization*, *33*, 435–464. (Cited on pp. 231, 302)

Zwart, P. B. (1974). Global maximization of a convex function with linear inequality constraints. *Operations Research*, *22*, 602–609. (Cited on pp. 299, 309)

Index