Nonconvex All-Quadratic Global Optimization Problems:

Solution Methods, Application and Related Topics

Dissertation

zur Erlangung des akademischen
Grades eines Dr. rer. nat.

Dem Fachbereich IV der Universität Trier
vorgelegt von

Ulrich Raber

Trier, 1999

Eingereicht am 31. Mai 1999

1. Berichterstatter: Prof. Dr. Reiner Horst
2. Berichterstatter: Prof. Dr. Immanuel Bomze

Tag der mündlichen Prüfung: 14. September 1999

Acknowledgments

At this place I would like to thank my family, friends and colleagues, who have contributed to a successful completion of this thesis.

First of all I am very indebted to Prof. Dr. Reiner Horst. He proposed the topic of this thesis to me, and advised and supported me in many ways.

A particular thank is due to Dr. Marco Locatelli from the University of Florence. Significant parts of Chapters 4 and 5 result from a joined work with him.

Furthermore, I would like to thank the people, who helped proofreading parts of the manuscript, especially Dr. habil. Florian Jarre, Pia Leister and Prof. Dr. Nguyen Van Thoai.

Moreover, I like to thank Dr. Michael Nast for his general branch-and-bound framework written in C++, which was used in the implementation of our branch-and-bound algorithms.

Last but not least I would like to thank my family and, especially, my girlfriend Claudia. Their support provided essential motivation to keep up in writing this thesis.

Contents

CHAPTER 1

Introduction

A large part of the mathematical optimization theory deals with the problem of detecting a real n-dimensional point $\bar{x}$ belonging to a set $M \subset \mathbb{R}^n$ such that a real-valued function f attains its minimum over M at this point, i.e., one tries to solve the general problem

$$\begin{aligned} &\min f(x) \\ &x \in M\ . \end{aligned} \tag{GP}$$

The function $f : A \to \mathbb{R}$ is usually defined on a suitable set A satisfying $A \supset M$. In the field of global optimization we are interested in points $\bar{x} \in M$ satisfying $f(\bar{x}) \leq f(x)$, for all $x \in M$, i.e., we are looking for the *global minimum* of Problem (GP). In contrast to this, the local optimization is satisfied if a point $\bar{x} \in M$ with the property $f(\bar{x}) \leq f(x)$, for all $x \in M \cap N$, has been detected, where N is some neighborhood of $\bar{x}$, i.e., it suffices to determine a *local minimum* of (GP).

In general, Problem (GP) is not solvable. In order to obtain practicable solution approaches for this problem we need some knowledge about the structure of the objective function f as well as of the set M. The main interest in Problem (GP) is motivated by real applications and, fortunately, there are a lot of such applications leading to problems of type (GP) with a special usable structure.

In the present thesis we examine minimization problems, where the objective function is a quadratic function and where the feasible region $M \subset \mathbb{R}^n$ is described by a finite set of quadratic and linear constraints. These problems will be called **all-quadratic optimization problems**. They are given in the following way

$$\begin{aligned} \min\ x^T Q^0 x + (d^0)^T x& \\ x^T Q^l x + (d^l)^T x + c^l\ &\leq\ 0 \qquad l = 1, \ldots, p \\ x\ &\in P\ \ , \end{aligned} \tag{QP}$$

where Q^l ($l = 0, \ldots, p$) are real $n \times n$ matrices, d^l ($l = 0, \ldots, p$) are real n-dimensional vectors and c^l ($l = 1, \ldots, p$) are real numbers. The set

$$P = \{x \in \mathrm{I\!R}^n : Ax \leq b\}$$

is a polyhedron described by a real $m \times n$ matrix $A = (a_1, \ldots, a_m)^T$ and a real m-dimensional vector b. We assume that the matrices Q^l ($l = 0, \ldots, p$) are symmetric. This is not a restriction to the generality of the considered problems of type (QP). Indeed, if Q^l ($l \in \{0, \ldots, p\}$) is not symmetric, then we obtain a symmetric matrix by setting $\bar{Q}^l = \frac{1}{2}(Q^l + (Q^l)^T)$ with the property $x^T Q^l x = x^T \bar{Q}^l x$ ($x \in \mathrm{I\!R}^n$). Therefore, we can replace in (QP) the matrix Q^l by the matrix $\bar{Q}^l$ without altering the function values of the corresponding quadratic function. In view of this symmetry assumption we know that the eigenvalues of Q^l ($l = 0, \ldots, p$) are real-valued (see, e.g., [JRA93]). Apart from the symmetry of the matrices Q^l ($l = 0, \ldots, p$) we assume furthermore that the polyhedron P is a non-empty, full-dimensional and bounded set. This is a slight restriction to the generality of the considered problems of type (QP). However, the non-emptiness of the set P can easily be verified. Use, for example, the first phase of the Simplex-Algorithm, which is the well-known solution method developed by Dantzig [DAN63] for linear programs, i.e., for problems of type (GP) where f is a linear function and M is a polyhedron. The assumption that P is full-dimensional is not really needed for the theory in this dissertation, but is nevertheless made in order to reduce the technical effort. The fact that P is a polytope, i.e., that this set is bounded, cannot be guaranteed in general. However, this assumption is satisfied for many applications.

Throughout the present work we denote by

$$F := \{x \in P : x^T Q^l x + (d^l)^T x + c^l \leq 0\,,\, l = 1, \ldots, p\}$$

the feasible region of Problem (QP). Note that this set can be empty since we do not require the existence of a feasible point for (QP).

With respect to the difficulty of detecting global minima of Problem (QP) and the treatment of this problem in the literature we can distinguish some subclasses of (QP). If all quadratic functions in the formulation of (QP) are convex, then it is known that each local minimum of (QP) is a global minimum (see, e.g., [MAN94] or [HPT95, Chapter 1]), i.e., there is no gap between the local and the global minimization of this problem. Moreover, it is known that such problems can be solved in polynomial time up to a certain precision, if some assumptions are fulfilled (see, e.g., [HER94] and references therein). Several solution methods for this particular case of (QP) are available. Apart from the schemes developed only for convex all-quadratic problems (see, for example, [VDP66] for problems with one quadratic constraint, and [BAR72, EN75, PHH82] for arbitrary convex all-quadratic problems) any algorithm for minimizing arbitrary convex functions under convex constraints can be used (see, e.g., [FM68, GMW81]). Among these more general approaches the class of so-called *interior point methods* received a great deal of attention during the last decade. These methods, first developed for linear problems, show numerically an efficient behavior, in particular for large scale problems. Moreover, these efficient methods are applicable to special classes of convex optimization problems, for example in the fully convex all-quadratic case (see, e.g., [NN94, JAR96] and references therein).

The convexity of a quadratic function can be checked easily. It is a known fact [HPT95, Theorem 1.12] that a function $g : C \to \mathbb{R}$, which is twice differentiable on an open convex set $C \subset \mathbb{R}^n$, is convex if and only if its Hessian $\nabla^2 g(x)$ is positive semidefinite at each element x of the set C. In order to verify the convexity of the quadratic functions involved in (QP) we hence have to examine the eigenvalues of the matrices $Q^l \in \mathbb{R}^{n \times n}$ ($l = 0, \ldots, p$). If one of these matrices has at least one negative eigenvalue, the equivalence between the local and the global minima is not guaranteed anymore, and we cannot expect to solve such problems in polynomial time (see [PS88]). Actually it is known that even a problem with a quadratic objective function, whose describing matrix Q^0 has one negative eigenvalue, and with a feasible set determined by linear constraints is $\mathcal{NP}$-hard (see [PV91] or [HPT95, Section 2.4]).

Apart from the fully convex all-quadratic problems there is another subclass of problems of type (QP), which was already treated extensively in the literature. In the so-called *general quadratic programming problem* one is interested in the minimization of an arbitrary quadratic objective function with respect to linear constraints, i.e., problems of type (QP) with $p = 0$ are considered. For information about the theory, algorithms and applications of this type of all-quadratic problems we refer to the survey [FV95] and to more recent works [HPT95, HT96A, DAPT97, BOM97, AT98, YF98] and references therein.

In the present dissertation we will examine the most general case of Problem (QP), which has not been explored as widely in the literature as the fully convex all-quadratic problem or the general quadratic programming problem. We are interested in global minima of all-quadratic optimization problems with an arbitrary, in particular nonconvex, quadratic objective function and with at least one nonconvex quadratic constraint ($p \geq 1$). These problems have at first glance still a nice structure. Only quadratic and affine functions are involved. However, such problems have a nonconvex objective function and a feasible set F, which is in general not convex and, maybe, even not connected. This means that there is a gap between the local and the global optimization of such problems and taking the previous considerations into account we know that these problems can be $\mathcal{NP}$-hard. Nevertheless, nonconvex all-quadratic global optimization problems have a wide variety of applications.

1.1. Applications

Each n-dimensional all-quadratic problem can be easily transformed to a $2n$-dimensional bilinear problem, as it is done, for example, in [AK92, HJ92]. In [HJ92] a strategy for reducing the necessary dimension of the resulting bilinear program is also proposed. However, on the other hand bilinear optimization problems are nothing else than a special instance of Problem (QP). Pooling problems in petrochemistry [FV90A], the modular design problem introduced in [EVA63], in particular the multiple modular design problem [EVA70, AK92] or the more general modularization of product sub-assemblies [RS71], and special classes of structured stochastic games [FS87] are only some examples of the wide range of applications of bilinear programming problems.

Another large class of optimization problems are problems with linear or quadratic functions additionally involving Boolean variables, i.e., variables $x_i \in \mathrm{I\!R}$ with the constraint $x_i \in \{0,1\}$. Since each Boolean variable can be represented by a concave quadratic constraint

$$x_i \in \{0,1\} \quad \Leftrightarrow \quad x_i^2 - x_i \geq 0\,,\, x_i \in [0,1]\,,$$

such integer programming problems can be transformed to (QP). An example of this class of optimization problems is the so-called *synchronization sequence problem* (SSP) resulting from an application in the satellite industry. In this problem one is interested in an n-dimensional integer vector $x \in \{-1,1\}^n$ such that the maximal value of the absolute values of the cyclic autocorrelation functions

$$g^k(x) = \sum_{i=1}^{n} x_i x_{[i+k]}$$

($k = 1, \dots, n-1$) becomes minimal where $[i+k] = i + k (\operatorname{mod} n)$. Problem (SSP) can be formulated as

$$\begin{array}{rcl} & \min & t \\ g^k(x) & \leq & t \\ -g^k(x) & \leq & -t \\ x_i & \in & \{-1, 1\} \end{array} \quad \begin{array}{l} \\ k = 1, \dots, n-1 \\ \\ i = 1, \dots, n\,, \end{array} \tag{SSP}$$

and by using the substitution $x_i = 2y_i - 1$ ($i = 1, \dots, n$) one obtains an integer program with Boolean variables $y \in \{0,1\}^n$.

The problem of packing $n \in \mathbb{N}$ equal circles in a square, which can be transformed to a (QP), is another problem widely explored in the literature. One looks for the maximum radius r of n non-overlapping circles contained in the unit square. This problem is equivalent to an all-quadratic problem with a linear objective function and concave quadratic constraints. It can be formulated as

$$\begin{array}{cl} \max\ t & \\ t - \|x_i - x_j\|_2^2 \leq 0 & 1 \leq i < j \leq n \\ x_i \in [0,1]^2 & i = 1, \dots, n\,. \end{array} \tag{PP}$$

How the optimal value $t^\star$ of (PP) and the optimal radius $r^\star$ are related is discussed in Chapter 5 of the present research study. This chapter will deal extensively with Problem (PP). A related class of global optimization problems are minimax location problems [PHH82], which also lead to quadratic constraints.

Production planning and portfolio optimization are examples where so-called *chance constrained* linear programs occur (see, e.g., [PHH82, WV91, DT92]). These are problems, looking similar to linear programs. However, the matrix describing the linear constraints of such problems is not deterministic, it is a stochastic one. Under certain restrictive assumptions it is possible to transform these stochastic constraints to deterministic quadratic constraints (see again [PHH82, WV91]), such that in general a problem of type (QP) is obtained.

In [AKHP92] it is shown that nonconvex all-quadratic problems can be used for the examination of special instances of nonlinear bilevel programming problems. Other applications of (QP) include the fuel mixture problem encountered in the oil industry [PTA94] and also placement and layout problems in integrated circuit design (see [AKLV95, AKV96] and references therein).

Hence there are many applications of the nonconvex all-quadratic optimization problem (QP). Whether Problem (QP) is in practice applicable for solving, for example, problems resulting from integer programming problems, depends on the numerical efficiency of the solution method for (QP) that is used. Up to now only few methods for solving the considered general case of Problem (QP) were proposed in the literature. Most of them result from methods being developed for other more general problem classes. In Section 1.3 we will shortly discuss some of these solution methods. Before this we will sketch some basic concepts in global optimization. These concepts are used in all solution approaches mentioned in this dissertation.

1.2. Basic Concepts and Notations

In the field of deterministic global optimization there are at least two basic schemes for solving a general problem of type (GP).

1.2.1. Outer Approximation Approaches. Outer approximation (cutting plane) approaches use the following basic concept (see, e.g., [HT96B, Chapter 2]). Determine a superset $\bar{M}$ of M, which has a simple structure, for example a polyhedron, and try to minimize the function f with respect to this bigger set. If the minimization of f with respect to the simpler set $\bar{M}$ is still too complex, determine a simpler function $\bar{f}$, which underestimates f on the set M, and solve the problem

$$\begin{aligned} &\min \bar{f}(x) \\ &x \in \bar{M} \ . \end{aligned} \tag{$\overline{\text{GP}}$}$$

Problem ($\overline{\text{GP}}$) delivers a lower bound for the optimal value of (GP). Such problems are usually called **relaxations** of the original problem. If ($\overline{\text{GP}}$) is a linear program, it is called an **LP-relaxation** of ($\overline{\text{GP}}$). If the detected solution $\bar{x} \in \bar{M}$ of ($\overline{\text{GP}}$) is not contained in the set M, then one tries to determine a function $\ell : \mathbb{R}^n \to \mathbb{R}$ such that the set

$$\hat{M} \ := \ \bar{M} \cap \{x \in \mathbb{R}^n : \ell(x) \leq 0\} \ \supset \ M$$

has still a simple structure, but does not contain the point $\bar{x}$ anymore. If ℓ is an affine function, we call the set $H = \{x \in \mathbb{R}^n : \ell(x) = 0\}$ a **cutting plane**, since the point $\bar{x}$ is cut away by the hyperplane H. By solving the problem $\min_{x \in \hat{M}} \bar{f}(x)$ one obtains hopefully a better lower bound for the optimal value of (GP) and a new solution $\hat{x} \in \hat{M}$. This process is successively applied until a point $\tilde{x} \in M$ has been calculated. If $\bar{f}$ coincides with f at this point, then $\tilde{x}$ is obviously an optimal solution of (GP). Otherwise one has to refine the function $\bar{f}$ and to repeat the described process.

1.2.2. Branch-and-Bound Approaches. Another concept for treating global optimization problems are branch-and-bound methods (see, e.g., [HT96B, Chapter 4]). These schemes start analogously to the outer approximation algorithms with a relaxation $M^0 \supset M$ of the feasible region M of (GP). This relaxation is chosen such that a lower as well as an upper bound for the optimal value of Problem (GP) can be determined. According to a so-called **subdivision rule** one splits in subsequent steps the part of M^0 still of interest into more and more refined sets M^i (*branching*). For these sets new hopefully improved bounds are calculated (*bounding*). If a set M^i considered in the branch-and-bound tree has a lower bound, which exceeds the current best known value for (GP), then this set is eliminated from further considerations (*pruning*). Such sets cannot contain feasible points of Problem (GP) with a smaller objective function value than the best value known so far.

Using these strategies one hopes that the algorithm concentrates the search for a global minimum of Problem (GP) on a small portion of the feasible region M. One expects that a large part of M, which does not contain a global minimum of (GP), is *pruned* from further considerations at an early stage of the examination of the optimization problem by the branch-and-bound algorithm, which is applied for the solution of this problem.

1.2.3. Subdivision Sets. The sets, which are mostly used in branch-and-bound methods, are cones, n-dimensional rectangles or n-simplices. Throughout this dissertation we use only rectangles and simplices. An n-dimensional rectangle R, which we would like to call a **hyperrectangle**, is uniquely determined by two vectors $l, L \in \mathbb{R}^n$

$$R \;=\; \{x \in \mathbb{R}^n : l_i \;\leq\; x \;\leq\; L_i \;,\, i = 1, \ldots, n\} \,.$$

A simplex is the convex hull of an affine independent set of points, which form the vertices of this simplex. Let $\{v_0, \ldots, v_k\} \subset \mathbb{R}^n$ ($k \in \mathbb{N}$) be an arbitrary set. Then we denote by

$$[v_0, \ldots, v_k] \;:=\; \{x \in \mathbb{R}^n : x = \sum_{i=0}^{k} \lambda_i v_i \;,\, \lambda \in \mathbb{R}_+^{k+1} \;,\, \sum_{i=0}^{k} \lambda_i = 1\}$$

the **convex hull** of the points $v_0, \ldots, v_k$, where $\mathbb{R}_+ := \{\lambda \in \mathbb{R} : \lambda \geq 0\}$ denotes the positive orthant. If the points $v_0, \ldots, v_k$ are **affine independent**, i.e., for an arbitrary, but fixed index $i \in \{0, \dots, k\}$, there holds that the set $\{v_j - v_i : j \in \{0, \dots, k\} \setminus \{i\}\}$ is linear independent, then $S = [v_0, \dots, v_k]$ is a k-dimensional simplex, a so-called **k-simplex**. For example, a 2-simplex is a triangle and a 3-simplex is a tetrahedron.

Hyperrectangles and n-simplices are of course polytopes. The **facets** of these sets are easy to determine, where the facet of an n-dimensional polytope P is defined as an $(n-1)$-dimensional intersection of P with a supporting hyperplane, i.e., a $(n-1)$-dimensional **face** of P (see, e.g., [HPT95, Chapter 1]). In the case of an n-simplex $S = [v_0, \dots, v_n]$ there are the $n+1$ facets

$$S_i \;=\; [v_0, \dots, v_{i-1}, v_{i+1}, \dots, v_n] \qquad i = 0, \dots, n\,,$$

which are $(n-1)$-simplices. For a hyperrectangle $R = \{x \in \mathbb{R}^n : l \leq x \leq L\}$ the $2n$ facets are given by

$$\begin{array}{l} R_i^1 = \{x \in \mathbb{R}^n : l \leq x \leq L\,,\; x_i = l_i\} \\ R_i^2 = \{x \in \mathbb{R}^n : l \leq x \leq L\,,\; x_i = L_i\} \end{array} \qquad i = 1, \dots, n\,.$$

In the branch-and-bound methods, which we will consider in this thesis, the used subdivision sets $Z \subset \mathbb{R}^n$ are split into a finite number of subsets Z_i ($i \in I$, I finite index set) forming a *partition* of Z.

DEFINITION 1.2.1.([HPT95, Definition 3.3]) *Let $Z \subset \mathbb{R}^n$ be a polyhedron satisfying $intZ \neq \emptyset$, and let I be a finite set of indices. A family $\{Z_i : i \in I\}$ of subpolyhedra of Z satisfying, for each $i \in I$, $intZ_i \neq \emptyset$ is called a* ***partition*** *of Z, if*

$$\bigcup_{i \in I} Z_i \;=\; Z$$

and, for each $i, j \in I$ with $i \neq j$, there holds

$$intZ_i \cap intZ_j \;=\; \emptyset\,.$$

Simplices are usually subdivided using a so-called *radial subdivision.*

DEFINITION 1.2.2.([HPT95, Definition 3.4]) *Let $S = [v_0, \dots, v_n]$ be an n-simplex and let a point $w \in S \setminus \{v_0, \dots, v_n\}$ be given, which is uniquely represented by its barycentric coordinates, i.e.,*

$$w \;=\; \sum_{i=0}^{n} \lambda_i v_i$$

with $\lambda \in \mathbb{R}_+^{n+1}$, $\sum_{i=0}^{n} \lambda_i = 1$.

Denote, for each $i \in \{j \in \{0, \dots, n\}$ with $\lambda_j > 0\}$, by S_i the n-simplex, which is obtained by replacing the vertex v_i of S by w, i.e.,

$$S_i \ = \ [v_0, \dots, v_{i-1}, w, v_{i+1}, \dots, v_n] \, .$$

*The subdivision of S into the n-simplices S_i ($i \in \{j \in \{0, \dots, n\}$ with $\lambda_j > 0\}$) is called a **radial subdivision** of S with respect to w.*

It is known [HPT95, Proposition 3.7] that the radial subdivision of an n-simplex $S = [v_0, \dots, v_n]$ with respect to an arbitrary point $w \in S \setminus \{v_0, \dots, v_n\}$ forms a partition of S. The choice of the point w depends on the used subdivision (partitioning) rule.

It is not reasonable to apply the concept of radial subdivisions also for the partitioning of a hyperrectangle R, since the resulting polytopes do not necessarily have a rectangular structure anymore. If a point $w \in R$ is given, which does not belong to the set of vertices of R, then a subdivision of R is usually defined via hyperplanes parallel to the facets of R. This strategy leads to a partition of R into up to 2^n hyperrectangles, where the number of the resulting subhyperrectangles depends on the choice of w.

1.2.4. Convex Envelope. In outer approximation as well as in branch-and-bound methods we often need a simpler function $\bar{f}$, which underestimates the examined function f with respect to a given set $\bar{M}$. Since convex functions lead – from a theoretical point of view – to easily solvable problems, the so-called *convex envelope* of an arbitrary function f is a concept frequently used for determining the desired function $\bar{f}$.

DEFINITION 1.2.3. *Let $g : C \to \mathbb{R}$ be a lower-semicontinuous function defined on a non-empty convex set $C \subset \mathbb{R}^n$. The **convex envelope** of g on the set C is a function $\varphi : \mathbb{R}^n \to \mathbb{R}$ with the properties*

(i) *φ is convex on the set C;*
(ii) *$\varphi(x) \ \leq \ g(x)$, for all $x \in C$;*
(iii) *if $\tau : C \to \mathbb{R}$ is a convex function satisfying, for each $x \in C$, $\tau(x) \leq g(x)$, then there holds, for all $x \in C$, $\tau(x) \leq \varphi(x)$.*

Hence, the convex envelope φ of a function g on a set C is the best convex underestimating function for g on the given set. For an overview of the properties of the convex envelope we refer to [HPT95, Section 1.3]. Unfortunately, in general the construction of a convex envelope φ is a problem, which might be harder to solve than the considered optimization problem itself. For some instances, however, the explicit form of the convex envelope is known. For example, if g is a concave function and C is a polytope with given vertex set $V(C) = \{v_1, \dots, v_k\}$, the convex envelope φ of g with respect to C is given by [HPT95, Theorem 1.21]

$$\varphi(x) \;=\; \min\{\sum_{i=1}^{k} \lambda_i g(v_i) : x = \sum_{i=1}^{k} \lambda_i v_i \, , \, \lambda \in \mathbb{R}_+^k \, , \, \sum_{i=1}^{k} \lambda_i = 1\} \, .$$

This implies that the convex envelope of a concave function g with respect to an n-simplex $S = [v_0, \dots, v_n]$ is the uniquely determined affine function, which coincides in the $n+1$ vertices of S with g [HPT95, Theorem 1.22].

In some cases an overestimating function for a given function g with respect to a set C is needed additionally. In this situation the analogous concept of the so-called *concave envelope* ψ can be applied.

DEFINITION 1.2.4. *Let $g : C \to \mathbb{R}$ be an upper-semicontinuous function defined on a non-empty convex set $C \subset \mathbb{R}^n$. The* ***concave envelope*** *of g on the set C is a function $\psi : \mathbb{R}^n \to \mathbb{R}$ such that $-\psi$ is the convex envelope of $-g$ on the set C.*

Hence, the concave envelope ψ of a function g is the best concave overestimating function of g on the set C. Obviously, the concave envelope of a convex function g with respect to an n-simplex S is also the uniquely determined affine function, which coincides in the vertices of S with g.

1.2.5. Further Notations and Conventions. Throughout the present thesis we interpret an n-dimensional vector $x \in \mathbb{R}^n$, as usual, as a column vector, i.e.,

$$x \;=\; \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} .$$

Consequently, a matrix $A \in \mathbb{R}^{m \times n}$ is given as a connection of n m-dimensional vectors, i.e.,

$$A \;=\; (a_1, \dots, a_n) \;=\; \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} .$$

We use the superscript T for identifying the corresponding transposed vectors and matrices, i.e.,

$$x^T = (x_1, \dots, x_n) \in \mathbb{R}^{1\times n} \quad \text{and} \quad A^T = \begin{pmatrix} a_{11} & \cdots & a_{m1} \\ \vdots & & \vdots \\ a_{1n} & \cdots & a_{mn} \end{pmatrix} \in \mathbb{R}^{n\times m} \, .$$

As a measure for the distance of two n-dimensional points we use the *Euclidean norm* $\|\cdot\|_2 : \mathbb{R}^n \to \mathbb{R}$

$$\|x\|_2 \ := \ \left(\sum_{i=1}^{n} |x_i|^2 \right)^{\frac{1}{2}}$$

or the ℓ_∞*-norm* $\|\cdot\|_\infty : \mathbb{R}^n \to \mathbb{R}$

$$\|x\|_\infty \ := \ \max_{i=1,\dots,n} |x_i| \, .$$

The abbreviation

$$\text{int}M \ := \ \{x \in M : \exists\epsilon > 0 \text{ with } B(x,\epsilon) \subset M\}$$

denotes the **interior** of an arbitrary set $M \subset \mathbb{R}^n$, where $B(x,\epsilon) = \{y \in \mathbb{R}^n : \|x-y\|_2 \le \epsilon\}$ describes the sphere centered at x with radius ϵ. The notation

$$\text{cl}M \ := \ \{x \in M : \forall\epsilon > 0 \, \exists y \in B(x,\epsilon) \cap M\}$$

is used for the **closure** of M and

$$\partial M \ := \ \text{cl}M \setminus \text{int}M$$

denotes the **boundary** of M.

Finally, a constraint of the form

$$g(x) \ \le \ 0$$

with a concave function $g : \mathbb{R}^n \to \mathbb{R}$ is called a **reverse convex** constraint (see, e.g., [HPT95, Chapter 4]).

1.3. Solution Approaches

For brevity we define (using $c^0 = 0$), for each $l \in \{0, \dots, p\}$,

$$q^l(x) \ := \ x^T Q^l x + (d^l)^T x + c^l \, .$$

As mentioned before, most of the solution methods in the literature for Problem (QP) were developed for more general problem classes.

1.3.1. D.C. Optimization. Using the fact that the functions q^l ($l = 0, \ldots, p$) can be written as so-called **d.c. functions** (see Section 3.2), i.e., as a difference of two convex functions, Problem (QP) can be interpreted as a general d.c. problem. Therefore, one possible approach for solving (QP) is the application of algorithms developed for solving general d.c. global optimization problems. See, for example, [HPT95, Chapter 4] and the survey [TUY95] for the framework of d.c. optimization. In [PTA94] a special d.c. algorithm is proposed and applied to a quadratically constrained optimization problem resulting from the fuel mixture problem.

1.3.2. Semidefinite Programming. Another class of optimization problems, which can be used for the examination of all-quadratic problems and which has received a great deal of attention in recent times, is the so-called **semidefinite programming problem (SDP)**. This class of problems is a generalization of linear programs and can also be solved in polynomial time. In contrast to a linear program the variable x to optimize in an (SDP) belongs to the space of positive semidefinite symmetric matrices and not to the n-dimensional real space. An (SDP) can be written in the following way (see, e.g., [ALI95])

$$\begin{aligned} \min\ & C \bullet X \\ A_i \bullet X =\ & b_i \qquad i = 1, \ldots, m \\ X \ \succeq\ & 0\,, \end{aligned} \tag{SDP}$$

where $X, C, A_i \in \mathbb{R}^{n \times n}$ ($i = 1, \ldots, m$), X is symmetric, $\bullet$ denotes the inner product of matrices (see Section 2.1) and $X \succeq 0$ means that X is positive semidefinite.

Each all-quadratic problem of type (QP) can be transformed to an (SDP) with an additional rank-one constraint [RAM93]. Omitting this additional constraint one obtains the widely explored SDP-relaxation of (QP) (see, e.g., [SHO87, PRW95, FK97, SHO98]). The properties of this relaxation were examined in the literature (see, e.g., [FK97, NES98]) and improvements of this relaxation were discussed (for example, [QDKRT98]). However, to the author's knowledge there was only one report about the global optimization of (QP) via (SDP). Ramana [RAM93] presented a cutting plane approach using this SDP-relaxation for solving (QP) (see also [HR98] and Chapter 2, respectively, for an extension of this approach). Note that in the fully convex case an all-quadratic problem can be solved by an (SDP) since the rank-one constraint is not necessary in this case (see, e.g., [VB96]).

1.3.3. Bilinear Programming. As mentioned in the context of the applications, each problem of type (QP) can be transformed to a bilinear program. Hence, solution methods developed for bilinear programs can be applied to the nonconvex all-quadratic optimization problem. For example, Floudas and Visweswaran [FV90B, FV93B] propose an algorithm for solving problems belonging to a more general class, which contains in particular general bilinear programs. They solve such problems through a series of primal and relaxed dual problems. The solution of the primal problem provides an upper bound on the global minimum of the considered problem and delivers additionally the corresponding Lagrange multipliers. These multipliers are then used to formulate a Lagrange function that is used in the dual subproblem. Making use of several properties of the considered problem, the proposed algorithm solves the dual problem also through a series of subproblems that, taken together, provide a lower bound on the optimal value. Iterating this process leads to an approach, which is reported to deliver in finite time an approximate solution [FV93B]. In [FV93A] it is shown that it is possible to enhance the computational performance of this algorithm in the case of bilinear programs. The subproblems are considerably more tractable in this special case.

Another method for solving bilinear programs was developed by Sherali and Tuncbilek. In [ST92] (see also [SA99]) they present an algorithm for solving polynomial programming problems, i.e., for optimization problems with a polynomial objective function and polynomial constraints, and hence especially for bilinear programs. Under the assumption that additional box constraints for the variables are known they generate nonlinear implied constraints, which are then included in the original problem. After that they linearize each nonlinear function involved in the resulting problem by defining new variables, one for each distinct nonlinear term (see [SA92] for the reformulation-linearization technique in the bilinear case). The solution of the linear program generated by this reformulation-linearization technique is then a lower bound of the considered problem with respect to the used box constraints. By embedding this reformulation-linearization technique in a rectangular branch-and-bound scheme they obtain a convergent algorithm. Hence, the resulting algorithm for solving polynomial global optimization problems combines a linear outer approximation of the feasible set with a branch-and-bound scheme.

1.3.4. Direct Solution Methods. There exist only a few approaches in the literature, which consider Problem (QP) directly and not as a special instance of a more general class. The first approach mentioned in the literature for solving

(QP) was developed by Reeves [REE75]. However, this approach is restricted to all-quadratic problems, where the matrices Q^l ($l = 0, \dots, p$) are simultaneously diagonalizable, i.e., his algorithm is only able to manage separable quadratic functions. Extending an idea introduced by Falk and Soland [FS69, SOL71] for optimizing problems with nonconvex separable functions, Reeves [REE75] presents a rectangular branch-and-bound method for solving a problem of type (QP) with separable quadratic functions and additional box constraints. For this special type of quadratic functions the convex envelope with respect to a hyperrectangle can be easily derived such that – using the convex envelope concept – lower bounds for (QP) on the considered hyperrectangles can be calculated. Reeves refines the branch-and-bound algorithm by applying additionally a local search procedure in order to obtain feasible points. Moreover, he developed a strategy for identifying neighborhoods of local solutions, where these solutions are even global, such that these neighborhoods can be eliminated from further considerations.

Using the same basic concepts as Reeves, Al-Khayyal et al. [AKLV95], [AKV96] propose a rectangular branch-and-bound scheme for general problems of type (QP) with the additional property that box constraints for the variables are known. By substituting $y^l = Q^l x \in \mathbb{R}^n$ ($l = 0, \dots, p$) each function $q^l(x)$ is first interpreted as a bilinear function $q^l(x, y^l)$. In order to obtain a linearization of the feasible region of the resulting bilinear program, each bilinear term $x_i y_i^l$ ($i = 1, \dots, n$; $l = 0, \dots, p$) is bounded from below by its convex envelope and from above by the corresponding concave envelope. Since the convex envelope of the two-dimensional bilinear function xy on a rectangle is the maximum of two affine functions [AKF83], they obtain by introducing $(p+1)$ auxiliary n-dimensional vectors t^l ($l = 0, \dots, p$) an LP-relaxation of the examined bilinear program in the variables $x, y^0, \dots, y^p, t^0, \dots, t^p$. The resubstitution $Q^l x = y^l$ ($l = 0, \dots, p$) results in an LP-relaxation of the original problem with the variables $x, t^0, \dots, t^p$. This LP-relaxation is then used in a rectangular branch-and-bound scheme for calculating lower bounds for the optimal value of (QP) with respect to the considered hyperrectangle. As in Sherali and Tuncbilek's approach for polynomial programs, Al-Khayyal et al. obtain a solution method for (QP), which is a combination of a successively refined outer approximation of the feasible region with a rectangular branch-and-bound scheme.

1.4. Overview

The main aim of the present dissertation is the development and the theoretical as well as the numerical examination of solution methods for the nonconvex all-quadratic optimization problem (QP).

In **Chapter 2** we discuss an indirect approach for solving (QP). We do not develop an algorithm to determine an optimal solution of Problem (QP). We present several approaches for solving certain so-called *unary problems.* Each problem of type (QP) is equivalent to a unary problem, as we will see in this chapter. Thus, we can use algorithms for solving unary problems in order to detect optimal solutions of quadratic problems. This idea is due to Ramana [RAM93, Chapter 7] and is related to the semidefinite programming approach for all-quadratic problems mentioned before (see Subsection 1.3.2). Since the outer approximation (cutting plane) algorithm introduced by Ramana for solving unary problems cannot be guaranteed to be convergent, we present new approaches overcoming this theoretical deficiency. The resulting algorithms are combinations of linear outer approximations and branch-and-bound like subdivisions of the feasible region of the considered unary problem. In Chapter 2 we give, in particular, an explicit formulation of a so-called *regular* n-simplex with all its vertices on the boundary of the unit sphere $B = \{x \in \mathrm{I\!R}^n : \|x\|_2 \leq 1\}$. The theoretical properties of such an n-simplex were known before, but – to the author's knowledge – such a set has not yet been constructed. Unfortunately, we have to recognize that this indirect solution method for (QP) is not applicable in practice. Only small dimensional all-quadratic problems can be solved with acceptable computational effort via the solution of the equivalent unary problem.

Chapter 3 deals with a direct approach for solving (QP). This method shows a significantly better performance than the foregoing indirect one. The development of the proposed new algorithm was motivated by the work of Al-Khayyal et al. [AKLV95]. The branch-and-bound method for solving problems of type (QP) introduced in [AKLV95] is based on a rectangular subdivision of the feasible region of (QP) and exploits the convex and concave envelopes of the two-dimensional bilinear function xy on a rectangle $R \subset \mathrm{I\!R}^2$, as described in Subsection 1.3.4. By using a simplicial partitioning strategy and the convex envelope of a concave function on an n-simplex (see Subsection 1.2.4), we obtain a simplicial branch-and-bound scheme involving mainly linear programming subproblems. The numerical comparison of our new approach with the rectangular branch-and-bound method by Al-Khayyal et al. shows that the simplex algorithm often outperforms the rectangular algorithm.

In the definition of the simplicial branch-and-bound algorithm in Chapter 3 we use the so-called *bisection* for subdividing an n-simplex. Because of the special property of this subdivision strategy, it is a so-called *exhaustive* subdivision rule, the convergence of the presented approach can be ensured. The convergence is meant in the sense that each accumulation point of a sequence generated by the proposed algorithm is an optimal solution of Problem (QP). Some authors favor another subdivision rule in simplicial branch-and-bound methods, the so-called *ω-subdivision rule*. This strategy is not necessarily exhaustive, and the convergence of an algorithm using this rule was still an open question.

In **Chapter 4** we give an answer to this question. We consider a generalization of Problem (QP). We assume that the nonlinear functions involved in the global optimization problem under examination are d.c., not necessarily quadratic. After presenting an algorithm, which is a generalization of the simplicial branch-and-bound method introduced in Chapter 3 and which is applicable to the generalized problem class, we examine the convergence of this approach with respect to different subdivision rules. The convergence of the simplicial branch-and-bound scheme using the ω-subdivision rule can only be guaranteed for optimization problems with a d.c. objective function and with concave constraints. We present in Chapter 4 a counterexample, which shows that the presented method using this rule does not converge in general. In view of our theoretical results we are non the less able to develop a new convergent subdivision strategy – combining ω-subdivision and bisection. The numerical performance of some variants of this mixed strategy will be examined. The convergence concept, which we use in Chapter 4 in connection with the examination of the ω-subdivision, is – from a theoretical point of view – weaker than the one used in Chapter 3. We will not prove that each accumulation point of a sequence generated by the variant of our approach using ω-subdivisions is optimal. We will only show that this method determines in finite time either an approximate solution or the emptiness of the feasible region of the considered problem. As we will see in Chapter 4 – from a practical point of view – this convergence concept has non the less the same quality as the stronger concept mentioned above.

We conclude the more theoretically oriented Chapter 4 with a finiteness result. We prove that a simplicial branch-and-bound algorithm, which employs only ω-subdivisions and which is applied to the minimization of a concave function with respect to linear constraints, is even finite, if two additional assumptions are fulfilled.

In **Chapter 5** we close our consideration of Problem (QP) by examining an application of this class of global optimization problems. This chapter deals with the problem of packing n equal circles of maximal radius into the unit square, which we will call *packing problem*. Unfortunately, the solution methods, which we developed for general problems of type (QP), are not able to solve the optimization problem resulting from this application. At least they are not able to solve the problem for a high enough number of circles. Therefore, we develop a special global optimization algorithm for solving this problem.

We start in Chapter 5 with a study of the packing problem from a theoretical point of view. Some properties, which have to be satisfied by at least one solution of this problem, are introduced. These properties state the intuitive fact that as many circles as possible should touch the boundary of the unit square. Subsequently we propose a basic rectangular branch-and-bound algorithm and derive special bounds exploiting the structure of the packing problem. We introduce some tools with respect to the subdivision and the possible refinement of the considered hyperrectangles, which again exploit the special structure of the packing problem. They use in particular the theoretical properties of some solutions mentioned above. Applying these tools in the rectangular branch-and-bound algorithm we obtain an efficient algorithm.

In the literature good solutions of the packing problem with up to 50 circles are known. However, the quality of these solutions with respect to their optimality is mostly not known – at least for the packing problem with more than 20 circles. The new approach developed in this thesis is able to guarantee the ϵ-optimality of determined solutions of this problem. We will see, furthermore, that the implementation of our solution method showed a really good numerical performance for the packing problem with up to 27 circles. Moreover, we were also able to solve this problem approximately with up to 31 circles. This means that global optimization problems with a dimension of up to 63 can be solved up to a certain accuracy.

1.5. Test Examples

Throughout this thesis several algorithms are presented, which can be applied for solving nonconvex all-quadratic optimization problems. In order to test the numerical performance of these approaches, particularly to compare the numerical performance of different variants, we used a randomly generated set of test examples. Since the same set of test examples will be used for the examination of the approaches presented in Chapter 2, 3 and 4, we complete the introduction of this dissertation with a short description of these examples. For each combination of the dimension $n \in \{2, \ldots, 8, 10\}$ and the number of quadratic constraints $p \in \{1, \ldots, 2n\}$ we constructed fifty test problems with the general form of (QP) according to the following specifications.

First a polytope P with a non-empty interior was constructed. Starting with a randomly generated dense matrix $\bar{A} \in \mathbb{R}^{2n \times n}$ with integer entries between -10 and 10 we obtained a non-empty polyhedron $\bar{P} = \{x \in \mathbb{R}^n : \bar{A}x \leq \bar{b}\}$ by choosing an appropriate right-hand side vector $\bar{b} \in \mathbb{R}^{2n}$. In order to ensure the boundedness of the set P we intersected the polyhedron $\bar{P}$ with the n-simplex $S_n = [0, ne_1, \ldots, ne_n]$, where e_i $(i = 1, \ldots, n)$ denotes the i-th unit vector. The polytope $P = \bar{P} \cap S_n$ is then described by a $(3n+1) \times n$ matrix A and a $(3n+1)$-dimensional vector b. We iterated the construction of the polyhedron $\bar{P}$ until the interior of the resulting polytope P was not empty, and a point $\bar{x} \in \text{int}P$ $= \{x \in \mathbb{R}^n : Ax < b\}$ was found. In order to avoid in our numerical tests excessive running-times for problems with higher dimensions we used only such polytopes P, which could be circumscribed by an n-simplex with a diameter not bigger than 10.

In the next step dense $n \times n$ matrices Q^l and n-dimensional vectors d^l $(l = 0, \ldots, p)$ were randomly generated also with integer entries between -10 and 10. The coefficients c^l $(l = 1, \ldots, p)$ for the quadratic constraints were chosen such that $q^l(\bar{x}) = \bar{x}^T Q^l \bar{x} + (d^l)^T \bar{x} + c^l \leq -\delta < 0$ holds for the known point $\bar{x} \in \text{int}P$ and a prespecified value δ. This strategy guaranteed that we obtained all-quadratic optimization problems of type (QP) with

$$\text{int}F \neq \emptyset\,.$$

The average values, the standard deviations and sometimes also the medians of the effort, which a proposed solution approach needs for solving the fifty test examples for a combination of the dimension $n \in \{2, \ldots, 8, 10\}$ and the number of quadratic constraints $p \in \{1, \ldots, 2n\}$, will serve as a measure of the numerical performance of this approach.

CHAPTER 2

Convergent Outer Approximation Algorithms for Solving Unary Problems

The first solution method for the all-quadratic Problem (QP), which we propose in detail in the present dissertation, is an indirect one. Instead of solving (QP) directly we determine an optimal solution of a certain so-called unary problem, which is equivalent to (QP). Equivalence between (QP) and this unary problem holds in the sense that each solution of the unary problem yields a unique solution of the (QP) and vice versa.

This chapter deals with solution methods for general unary problems. These approaches are derived from an outer approximation scheme introduced by Ramana [RAM93]. Since the convergence of his approach cannot be guaranteed, it is the purpose of this chapter to develop solution methods which overcome this theoretical deficiency.

2.1. Introduction

In order to introduce the class of unary problems we first have to clarify the concept of unary matrices.

DEFINITION 2.1.1. *A real symmetric matrix $U \in \mathbb{R}^{n\times n}$ is called a* ***unary matrix****, if and only if there exists a vector $v \in \mathbb{R}^n$ with*

$$U \ = \ vv^T \ .$$

Denote by

$$\mathcal{S}_n \ := \ \{ \ S \in \mathbb{R}^{n\times n} : S \text{ symmetric} \ \}$$

the space of real symmetric $n \times n$ matrices and by

$$\mathcal{U}_n \ := \ \{ \ U \in \mathcal{S}_n : U \text{ unary} \ \}$$

the subset of $\mathcal{S}_n$ consisting of all unary matrices. Moreover, let $U^i \in \mathcal{S}_n$ ($i = 0, \dots, d$) be given and let $U : \mathbb{R}^d \to \mathcal{S}_n$ be an affine matrix mapping defined by

$$U(z) \;=\; U^0 + \sum_{i=1}^{d} z_i U^i \,, \tag{2.1.1}$$

A unary problem is then defined as follows.

DEFINITION 2.1.2. *Given $U^i \in \mathcal{S}_n$ ($i = 0, \dots, d$) and $h \in \mathbb{R}^d$, $A = (a_1, \dots, a_m)^T \in \mathbb{R}^{m \times d}$, $b \in \mathbb{R}^m$, the optimization problem*

$$\begin{aligned} \min\, & h^T z \\ Az \le & \; b \\ U(z) \in \mathcal{U}_n \,, & \; z \in \mathbb{R}^d \end{aligned} \tag{UP}$$

is called a ***unary problem****.*

REMARK 2.1.1. It is obvious (see Lemma 2.3.1) that the set $\mathcal{U}_n$ of unary matrices consists of all positive semidefinite matrices $U \in \mathcal{S}_n$ with the additional property

$$\operatorname{rank}(U) \;=\; 1 \,.$$

Therefore, Problem (UP) can also be formulated as a semidefinite program with an additional rank constraint (for related discussion, see again [SHO87, RAM93, PRW95, VB96, FK97] and Subsection 1.3.2).

As we will see in Section 2.2 it is possible to transform an all-quadratic problem of type (QP) to an equivalent unary problem where the polyhedron

$$P \;:=\; \{z \in \mathbb{R}^d : Az \le b\}$$

is bounded, i.e., P is a polytope. Even though we discuss in this chapter solution methods for general problems of type (UP), our interest in Problem (UP) is only motivated by such problems which are equivalent transformations of all-quadratic problems. Regarding the intention of this dissertation it is thus not a restriction to assume that P is always bounded, as we have done in the sequel.

The equivalence between (QP) and a special problem of type (UP) is one of the interesting observations proposed without proof in the dissertation of Ramana

[RAM93, Chapter 7], which was our main motivation for considering unary problems. In Section 2.2 a detailed proof of this equivalence is given. A second observation suggested in Ramana's research study is based on eigenvalue inequalities due to Weyl: given an optimal vertex solution $\bar{z}$ of the LP-relaxation $\min_{z \in P} h^T z$ of (UP) satisfying $U(\bar{z}) \notin \mathcal{U}_n$, and given the eigenvalues of $U(\bar{z})$, a linear constraint $\ell(z) \leq 0$ can be constructed satisfying $\ell(\bar{z}) > 0$ and, for all $z \in \mathbb{R}^d$ with $U(z) \in \mathcal{U}_n$, $\ell(z) \leq 0$. Therefore, by adding successively such valid cuts $\ell(z) \leq 0$ to LP-relaxations of (UP), one obtains an outer approximation (or cutting plane) algorithmic approach for solving (UP). Several variants of this cutting plane approach together with some preliminary numerical results, which are really promising, are proposed in [RAM93]. In Section 2.3 we compile some preliminaries underlying the basic ideas of this outer approximation approach and present Ramana's algorithm.

A serious deficiency of this algorithmic approach, however, consists in the fact that cuts can possibly become very shallow. Therefore, the convergence of the sequence of optimal solutions of the outer approximations to an optimal solution of (UP) cannot be guaranteed. A similar deficiency was observed in other cutting plane methods for certain global optimization problems (see, e.g., [HT96B, Chapter 6]). By proposing alternative outer approximation algorithms for solving (UP), which are convergent in the sense that each accumulation point of the sequence of optimal solutions of the outer approximations is an optimal solution of (UP), we overcome the above deficiency.

As we will see in Section 2.4, it suffices in Problem (UP) with (2.1.1) to consider matrices $U^i \in \mathcal{S}_n$ ($i \in \{1, \ldots, d\}$), which form an orthonormal system with respect to the inner product $\bullet : \mathcal{S}_n \times \mathcal{S}_n \to \mathbb{R}$:

$$B \bullet C \;=\; \operatorname{tr}(B^T C) \;=\; \sum_{i,j=1}^{n} b_{ij} c_{ij} \;, \tag{2.1.2}$$

where $B = (b_{ij})_{1 \leq i,j \leq n}$ and $C = (c_{ij})_{1 \leq i,j \leq n}$, and $\operatorname{tr}(A) = \sum_{i=1}^{n} a_{ii}$ denotes the *trace* of a matrix $A \in \mathbb{R}^{n \times n}$. Using this observation we derive in Section 2.4 a valid quadratic cut. This is a reverse convex constraint. For each optimal solution $\bar{z}$ of an LP-relaxation of (UP) satisfying $U(\bar{z}) \notin \mathcal{U}_n$, it cuts a sufficiently large ball (with respect to the Euclidean norm) centered at $\bar{z}$ out of the feasible region of this LP-relaxation of (UP) without eliminating a feasible point of (UP), i.e., without affecting the unarity.

If this cut is used directly in an outer approximation scheme, the convergence of such a method can be guaranteed. Unfortunately, the direct use of this cut would lead to relaxations of (UP), which are as hard to solve as (UP) itself. If a sufficiently large polytope inscribed in the Euclidean norm ball is known, then we can cut this polytope out of the feasible region instead of the balls. Though the resulting subproblems are still hard to solve, using the fact that a polytope is described by a finite number of linear constraints, we obtain a convergent and practicable algorithm by building up this polytope by successive cutting planes. The basic idea of this approach is presented in Section 2.5. The proposed algorithm is not a pure outer approximation scheme. It is a combination of an outer approximation and a successive subdivision of the feasible region of (UP).

In Section 2.6 we propose three possible ways to construct polytopes containing a sufficiently large part of the intersection of the feasible region of an arbitrary LP-relaxation of (UP) and the relevant Euclidean norm ball. Each one of these types of polytopes can then be used in order to obtain an implementable solution scheme for (UP). In each iteration of these new algorithms we have to split a given polytope into a fixed number of subsets, and then we have to examine each of these subsets – as it is the case in branch-and-bound methods (see, e.g., [HT96B, Chapter 4]). From a numerical point of view this can lead to excessive storage requirements. In order to reduce the number of necessary splits and, thus, in order to reduce the number of generated polytopes, we develop in Section 2.7 a convergent algorithm which does not subdivide each considered polytope. The resulting method combines the cuts introduced by Ramana, a new cut introduced in Section 2.6 and the subdivision strategy developed in Section 2.5. Most of the theoretical results of Section 2.2 up to Section 2.6 were published in [HR98].

In the final Section 2.8 we discuss the numerical performance of the proposed new approaches. Since we are interested in solution methods for all-quadratic problems we tried to solve the unary problems resulting from the equivalent transformation of the problems belonging to our test set (see Section 1.5). Even though a slight modification of the algorithms leads to a significant improvement of their numerical performance, our numerical results in Section 2.8 show that the practical application of the unary problem approach to all-quadratic problems of type (QP) is limited to very small sizes.

2.2. Unary Problems and All-Quadratic Optimization Problems

In this section it is shown that an arbitrary all-quadratic problem of type (QP) in n variables is equivalent to a unary problem in $d = \binom{n+1}{2} + n$ variables. By reasons which will become evident in Section 2.4, we choose a transformation which yields a unary problem, where the matrices U^i ($i = 1, \dots, d$) form an orthonormal system with respect to the inner matrix product (2.1.2).

As usual we have used in the formulation of (QP) as well as in the formulation of (UP) the letters A and b, respectively P for describing the linear constraints. In order to avoid ambiguities we add the superscript Q, if a letter is related to Problem (QP), and the superscript U otherwise.

Consider an arbitrary all-quadratic problem of type (QP), i.e., consider the problem

$$\begin{aligned} &\min\ x^TQ^0x + (d^0)^Tx \\ x^TQ^lx + (d^l)^Tx + c^l \ &\leq\ 0 \qquad l = 1, \dots, p \\ A^Qx\ \leq\ b^Q\ ,\ x\ &\in\ \mathbb{R}^n\ , \end{aligned} \tag{$\overline{\text{QP}}$}$$

where $Q^l = (q^l_{ij})_{1\leq i,j\leq n} \in \mathcal{S}_n$, $d^l \in \mathbb{R}^n$ ($l = 0, \dots, p$), $c^l \in \mathbb{R}$ ($l = 1, \dots, p$), $A^Q = (a^Q_1, \dots, a^Q_m)^T \in \mathbb{R}^{m\times n}$ and $b^Q \in \mathbb{R}^m$. Since we assumed that $P^Q = \{x \in \mathbb{R}^n : A^Qx \leq b^Q\}$ is a polytope we know that there exists a hyperrectangle $R^Q = \{x \in \mathbb{R}^n : l^Q \leq x \leq L^Q\}$ with $l^Q, L^Q \in \mathbb{R}^n$ satisfying

$$P^Q\ \subset R^Q\ .$$

Let $e_i \in \mathbb{R}^{n+1}$ denote the i-th unit vector ($i = 1, \dots, n+1$), and let $E_{ij} \in \mathbb{R}^{(n+1)\times(n+1)}$ be the elementary matrix with entry 1 at position (i, j) and 0 at any other position. The equivalent transformation of Problem ($\overline{\text{QP}}$) leads to the following unary problem

$$\begin{aligned} \min h^Tz& \\ A^Uz\ \leq\ b^U& \\ l^U\ \leq z\ \leq\ L^U& \\ U(z) \in \mathcal{U}_{n+1}\ ,\ z \in \mathbb{R}^{\binom{n+1}{2}+n}& \end{aligned} \tag{$\overline{\text{UP}}$}$$

in the variable $z = (z_{11}, \dots, z_{1n}, z_{1,n+1}, z_{22}, \dots, z_{2,n+1}, \dots, z_{nn}, z_{n,n+1})^T$, where, for $i = 1, \dots, n$,

$h_{i,n+1} = \frac{1}{\sqrt{2}} d_i^0 \ , \ a^U_{l,(i,n+1)} = \frac{1}{\sqrt{2}} d_i^l \ (l = 1, \dots , p),$
$a^U_{p+l,(i,n+1)} = \frac{1}{\sqrt{2}} a^Q_{li} \ (l = 1, \dots , m),$
$l^U_{i,n+1} = \sqrt{2} l_i^Q \ , \ L^U_{i,n+1} = \sqrt{2} L_i^Q,$

$h_{ii} = q^0_{ii} \ , \ a^U_{l,ii} = q^l_{ii} \ (l = 1, \dots , p) \ ,$
$a^U_{p+l,ii} = 0 \ (l = 1, \dots , m),$
$l^U_{ii} = \max\{(\min\{L_i^Q, 0\})^2 \, , \, (\max\{l_i^Q, 0\})^2\} \ , \ L^U_{ii} = \max\{l_i^Q l_i^Q, L_i^Q L_i^Q\},$

and, for $1 \leq i < j \leq n$,
$h_{ij} = \sqrt{2} q^0_{ij} \ , \ a^U_{l,ij} = \sqrt{2} q^l_{ij} \ (l = 1, \dots , p) \ , \ a^U_{p+l,ij} = 0 \ (l = 1, \dots , m),$
$l^U_{ij} = \sqrt{2} \min\{l_i^Q l_j^Q, l_i^Q L_j^Q, L_i^Q l_j^Q, L_i^Q L_j^Q\},$
$L^U_{ij} = \sqrt{2} \max\{l_i^Q l_j^Q, l_i^Q L_j^Q, L_i^Q l_j^Q, L_i^Q L_j^Q\}.$

The right-hand side b^U of the linear constraints is given by

$$b^U_l \ = \ -c^l \ (l = 1, \dots , p) \ , \ b^U_{p+l} \ = \ b^Q_l \ (l = 1, \dots , m) \ ,$$

and the affine matrix mapping in ($\overline{\text{UP}}$) is defined as follows

$$\begin{aligned} U : \mathbb{R}^{\binom{n+1}{2}+n} &\to \mathcal{S}_n :\Leftrightarrow \\ U(z) &= U^0 + \sum_{i=1}^{n} z_{ii} U^{ii} + \sum_{1 \leq i < j \leq n+1} z_{ij} U^{ij} \end{aligned} \tag{2.2.1}$$

with $U^0 = E_{n+1,n+1}$, $U^{ii} = E_{ii}$ $(i = 1, \dots , n)$ and $U^{ij} = \frac{1}{\sqrt{2}}(E_{ij} + E_{ji})$ $(1 \leq i < j \leq n + 1)$.

A quadratic function consists of three different terms of variables. There are linear terms ($x_i, i = 1, \dots , n$), pure quadratic terms ($x_i^2, i = 1, \dots , n$) and bilinear terms ($x_i x_j$, $1 \leq i < j \leq n$). In the formulation of ($\overline{\text{UP}}$) each of these terms is replaced by a new variable such that all functions involved in the formulation of ($\overline{\text{QP}}$) can be transformed to linear functions. The additional unarity condition in ($\overline{\text{UP}}$) guarantees that each feasible point of ($\overline{\text{UP}}$) coincides with a feasible point of ($\overline{\text{QP}}$). For that reason the postulated equivalence between the all-quadratic problem ($\overline{\text{QP}}$) and the unary problem ($\overline{\text{UP}}$) holds in the sense of the following theorem.

THEOREM 2.2.1. *Let $x^\star$ be an optimal solution of Problem ($\overline{\text{QP}}$) and let $z^\star$ be an optimal solution of Problem ($\overline{\text{UP}}$). If we set*

$$\bar{z}_{i,n+1} = \sqrt{2}x_i^\star \ , \ \bar{z}_{ii} = (x_i^\star)^2 \ (i = 1, \dots , n) \ , \ \bar{z}_{ij} = \sqrt{2}x_i^\star x_j^\star \ (1 \le i < j \le n) \ ,$$

and

$$\bar{x}_i = \frac{1}{\sqrt{2}} z^\star_{i,n+1} \ \ (i = 1, \dots , n) \ ,$$

then $\bar{z}$ is a feasible solution of Problem ($\overline{\text{UP}}$), $\bar{x}$ is a feasible solution of Problem ($\overline{\text{QP}}$) and

$$(\bar{x})^T Q^0 \bar{x} + (d^0)^T \bar{x} = (x^\star)^T Q^0 x^\star + (d^0)^T x^\star = h^T \bar{z} = h^T z^\star \ . \tag{2.2.2}$$

PROOF: Straightforward calculation shows that

$$U(\bar{z}) = \begin{pmatrix} x^\star \\ 1 \end{pmatrix} ((x^\star)^T, 1) \ ,$$

and hence $U(\bar{z}) \in \mathcal{U}_{n+1}$. By the definition of l^U and L^U and the fact that $x^\star$ is contained in R^Q it follows immediately

$$l^U \ \le \ \bar{z} \ \le \ L^U \ .$$

For the l-th row a_l^U of the matrix A^U we obtain, for $l = 1, \dots , p$,

$$\begin{aligned} a_l^U \bar{z} \ &= \ \sum_{i=1}^n a^U_{l,(i,n+1)} \bar{z}_{i,n+1} + \sum_{1 \le i \le j \le n} a^U_{l,ij} \bar{z}_{ij} \ = \ \sum_{i=1}^n d_i^l x_i^\star + \sum_{i,j=1}^n q_{ij}^l x_i^\star x_j^\star \\ &= \ (x^\star)^T Q^l x^\star + (d^l)^T x^\star \ \le \ -c^l \ = \ b_l^U \ , \end{aligned}$$

and, for $l = 1, \dots , m$,

$$\begin{aligned} a^U_{p+l} \bar{z} \ &= \sum_{i=1}^n a^U_{p+l,(i,n+1)} \bar{z}_{i,n+1} + \sum_{1 \le i \le j \le n} \underbrace{a^U_{p+l,ij} \bar{z}_{ij}}_{=0} \\ &= \ (a_l^Q)^T x^\star \ \le \ b_l^Q \ = \ b^U_{p+l} \ , \end{aligned}$$

i.e., $\bar{z}$ is a feasible solution of Problem ($\overline{\text{UP}}$). Similar direct calculations show that

$$h^T \bar{z} \ = \ (x^\star)^T Q^0 x^\star + (d^0)^T x^\star \ ,$$

and hence, since $\bar{z}$ satisfies the constraints of ($\overline{\text{UP}}$) and $z^\star$ is an optimal solution of ($\overline{\text{UP}}$), we obtain

$$h^T z^\star \ \le \ (x^\star)^T Q^0 x^\star + (d^0)^T x^\star \ .$$

Analogously one easily obtains that $\bar{x}$ is feasible for ($\overline{\text{QP}}$) and $h^T z^\star = (\bar{x})^T Q^0 \bar{x} + (d^0)^T \bar{x}$, which implies that

$$h^T z^\star \;\geq\; (x^\star)^T Q^0 x^\star + (d^0)^T x^\star \,.$$

■

REMARK 2.2.1. As mentioned in Remark 2.1.1, Problem (UP) can also be interpreted as a special semidefinite program. Using the semidefinite programming notations a short formulation of the previous theorem is available along the lines given, e.g., in [RAM93, PRW95, VB96, FK97]. In order to avoid the introduction of these semidefinite programming notations we decided to use the presented more technical version of the equivalence result.

Example. We conclude this section with a simple example. Consider the one-dimensional all-quadratic problem

$$\begin{aligned} \min\; x^2 + x& \\ -x^2 + 1 \;\leq\; 0& \\ x \in [-2, 2]\,.& \end{aligned} \tag{QPE}$$

The feasible region F^Q of (QPE) is given by the two disjoint intervals $[-2, -1]$ and $[1, 2]$, and the optimal solution $x^\star$ is -1 (see Figure 2.1(a)) with optimal value 0. Using the described transformation we obtain the following unary problem

$$\begin{gathered} \min\; z_{11} + \tfrac{1}{\sqrt{2}} z_{12} \\ -z_{11} \;\leq\; -1 \\ 0 \leq z_{11} \leq 4 \\ -2\sqrt{2} \leq z_{12} \leq 2\sqrt{2} \\ \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} + z_{11} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + z_{12} \begin{pmatrix} 0 & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 0 \end{pmatrix} \;\in\; \mathcal{U}_2\,. \end{gathered} \tag{UPE}$$

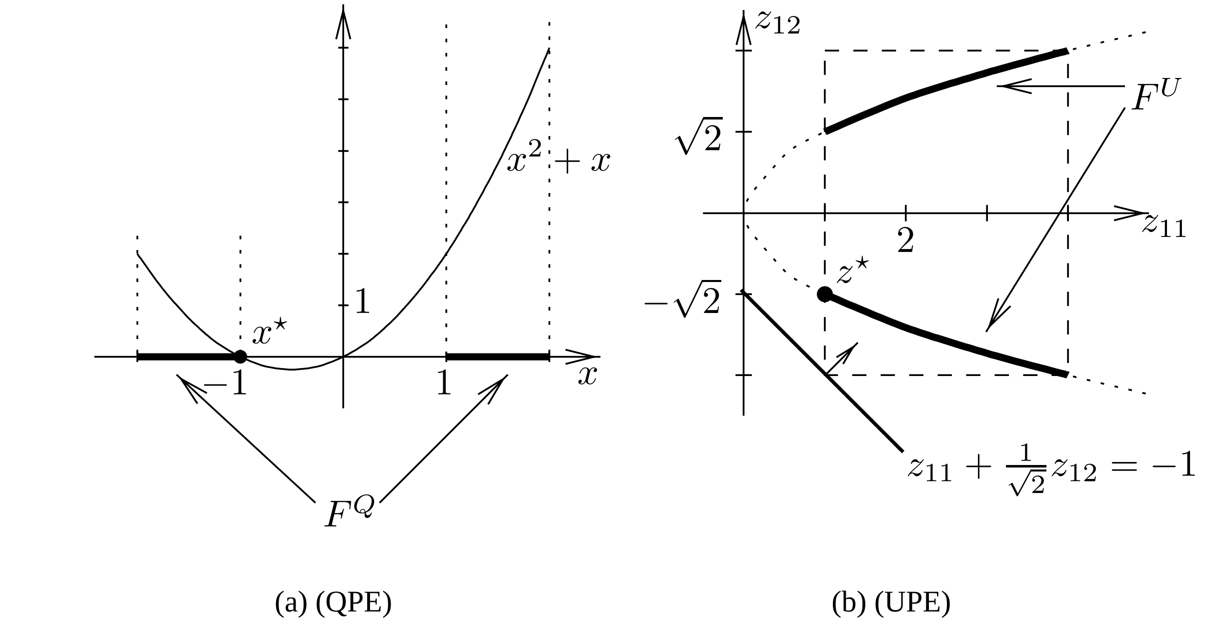
The optimal value of (UPE) is also 0 and is attained at the unique solution point $z^\star = (1, -\sqrt{2})^T$ belonging to the feasible region F^U of (UPE) given by

$$F^U \;=\; \{z \in \mathbb{R}^2 : 1 \leq z_{11} \leq 4\,,\; -2\sqrt{2} \leq z_{12} \leq 2\sqrt{2}\,,\; z_{12}^2 = 2z_{11}\}$$

(see the two disjoint arcs in Figure 2.1(b)). We will use Problem (UPE) throughout this chapter in order to illustrate the proposed solution methods.

Note that in the following sections we consider only unary problems. Therefore, the superscript U is not necessary any more.

FIGURE 2.1. Feasible regions of (QPE) and (UPE)



(a) (QPE) (b) (UPE)

2.3. Preliminaries and Ramana's Approach

The following results taken from [RAM93] are needed for the new cutting plane algorithms discussed in the subsequent sections. Even though the knowledge of Ramana's outer approximation scheme, in particular the knowledge of the cutting planes introduced by Ramana, is not necessary for developing these new approaches we repeat his algorithm in this section. There are at least two reasons for doing that. First of all, the overcome of the theoretical deficiency of the unknown convergence of Ramana's algorithm was the main motivation for developing new algorithms for solving (UP). Another reason is that the combination of the cuts defined by Ramana with our methods results – from a numerical point of view – in a more efficient solution scheme for unary problems, as we will see in Sections 2.7 and 2.8.

In this and the following sections we assume that the dimensions n and d of (UP) are not smaller than 2. The simple example (UPE) in the previous section shows that even the transformation of a one-dimensional (QP) leads to a (UP) with these dimensions.

The following first result characterizes unary matrices by means of their eigenvalues.

LEMMA 2.3.1. *Let $U \in \mathcal{S}_n$, and let $\lambda_i(U)$ ($i = 1, \dots, n$) be the eigenvalues of U indexed in increasing order. Then the following assertions are equivalent:*

(i) $U \in \mathcal{U}_n$;
(ii) $\lambda_i(U) \;=\; 0,\; i = 1, \dots, n-1$;
(iii) $\lambda_1(U) \;\geq\; 0$ *and* $\lambda_{n-1}(U) \;\leq\; 0$;
(iv) $\lambda_1(U) \;\geq\; 0$ *and* $tr(U) \;\leq\; \lambda_n(U)$.

PROOF: The above equivalences follow readily from the well–known facts that a matrix $U \in \mathbb{R}^{n \times n}$ is unary if and only if it is positive semidefinite and $\operatorname{rank}(U) = 1$, and that, for each real $n \times n$ matrix A, there holds $\operatorname{tr}(A) = \sum_{i=1}^{n} \lambda_i(A)$ (see, e.g., [ZUR64, §13]). ■

The second lemma describes now a relation between the eigenvalues of the sum of symmetric matrices and the sum of the eigenvalues of these matrices.

LEMMA 2.3.2. *Let $E, F \in \mathcal{S}_n$ with eigenvalues $\lambda_i(E), \lambda_i(F)$ ($i = 1, \dots, n$) be indexed in the same order as above. Then, for each $k \in \{1, \dots, n\}$, there holds*

$$\lambda_1(E) + \lambda_k(F) \;\leq\; \lambda_k(E + F) \;\leq\; \lambda_k(E) + \lambda_n(F)\,. \tag{2.3.1}$$

PROOF: See, e.g., [HJ85]. ■

This result is due to Hermann Weyl. Therefore, we will denote the inequalities (2.3.1) as **Weyl's inequalities**. Using the result of the last lemma a relation between the eigenvalues of the affine matrix mapping $U(\cdot)$ and the eigenvalues of the matrices U^i ($i = 0, \dots, n$) forming $U(\cdot)$ was derived in [RAM93].

COROLLARY 2.3.3. *Let $U : \mathbb{R}^d \to \mathcal{S}_n$ be an affine matrix mapping defined as in (2.1.1). Then, for every nonnegative $y \in \mathbb{R}^d_+$ and $k \in \{1, \dots, n\}$, there holds*

$$\lambda_k(U(y)) \;\leq\; \lambda_k(U^0) + \sum_{i=1}^{d} y_i \lambda_n(U^i)$$

and

$$\lambda_k(U(y)) \;\geq\; \lambda_k(U^0) + \sum_{i=1}^{d} y_i \lambda_1(U^i)\,,$$

where all eigenvalues $\lambda_i(\cdot)$ ($i = 1, \dots, n$) are indexed in ascending order.

PROOF: The results follow by successive application of Weyl's inequalities (Lemma 2.3.2) and the fact that, for each $U \in \mathcal{S}_n$, $\mu \geq 0$ and $i \in \{1, \ldots, n\}$, there holds $\lambda_i(\mu U) = \mu \lambda_i(U)$. ■

Consider now the LP-relaxation

$$\begin{aligned} \min\ h^T z \\ Az \ \leq\ b \end{aligned} \tag{UPL}$$

of (UP), which arises from (UP) by omitting the unary condition $U(z) \in \mathcal{U}_n$. Given a vertex optimal solution $\bar{z}$ of (UPL) and the affine matrix mapping U defined in (2.1.1), $\lambda_1(U(\bar{z})) = 0$ and $\lambda_{n-1}(U(\bar{z})) = 0$ implies that $\bar{z}$ is an optimal solution of (UP) because of Lemma 2.3.1. Otherwise, one must have $\lambda_1(U(\bar{z})) < 0$ or $\lambda_{n-1}(U(\bar{z})) > 0$ (or both). In this case, however, Corollary 2.3.3 allows one to construct an additional linear constraint $\ell(z) \leq 0$ which, when added to the constraints of (UPL), is violated by $\bar{z}$ but satisfied by all feasible solutions of (UP).

Since $\bar{z}$ is a vertex solution of a linear program it is known that $\bar{z}$ is the unique solution of a nonsingular $d \times d$ system of linear equations binding at $\bar{z}$, which – following the standard terminology in simplex algorithms – will be called a **nonsingular basic system corresponding to z̄**. Simplex-type algorithms provide such a system automatically. In order to derive the linear cuts introduced in [RAM93] let $Bz \leq r$ be the corresponding nonsingular basic system for $\bar{z}$ satisfying $B\bar{z} = r$. By the definition of the corresponding nonsingular basic system we know that each point $z \in P = \{z \in \mathbb{R}^d : Az \leq b\}$ is contained in the cone $C := \{z \in \mathbb{R}^d : Bz \leq r\}$ (C is the smallest of such cones containing P and uniquely determined when $\bar{z}$ is a non-degenerate vertex of P). Choose an arbitrary point $z \in P$ and set

$$y \ :=\ r - Bz\,.$$

The point y is a nonnegative element of $\mathbb{R}^d$, and for the affine matrix mapping $U(\cdot)$ at the point z we obtain

$$U(z) \ =\ U(\underbrace{B^{-1}r}_{=\bar{z}} - B^{-1}y) \ =\ U(\bar{z}) + \sum_{i=1}^{d} y_i \left(U^0 - U(B^{-1}e_i)\right)\ , \tag{2.3.2}$$

where $e_i \in \mathbb{R}^d$ denotes again the i-th unit vector ($i = 1, \ldots, d$). The right-hand side of (2.3.2) is an affine matrix mapping with the form given in (2.1.1). Therefore, Corollary 2.3.3 is applicable, and we obtain

$$\lambda_{n-1}(U(z)) \;\geq\; \lambda_{n-1}(U(\bar{z})) + \sum_{i=1}^{d} y_i \, \lambda_1 \left(U^0 - U(B^{-1}e_i)\right)$$

and

$$\lambda_1(U(z)) \;\leq\; \lambda_1(U(\bar{z})) + \sum_{i=1}^{d} \underbrace{y_i}_{=(r-Bz)_i} \lambda_n \left(U^0 - U(B^{-1}e_i)\right) \;.$$

It follows that, for each point $z \in P$ with $U(z) \in \mathcal{U}_n$, the cut

$$\sum_{i=1}^{d}(r - Bz)_i \, \lambda_1 \left(U^0 - U(B^{-1}e_i)\right) + \lambda_{n-1}(U(\bar{z})) \;\leq\; 0 \qquad (2.3.3)$$

is valid. However, for the point $\bar{z}$ with $\lambda_{n-1}(U(\bar{z})) > 0$, (2.3.3) is violated.

An analogous result is true for the linear constraint

$$\sum_{i=1}^{d}(Bz - r)_i \, \lambda_n \left(U^0 - U(B^{-1}e_i)\right) - \lambda_1(U(\bar{z})) \;\leq\; 0 \,. \qquad (2.3.4)$$

Adding these cuts to the linear constraints describing P we obtain a better outer approximation of the feasible region of (UP) and we can calculate a new, maybe better, vertex solution of this new LP-relaxation of (UP). Continuing in this way, a polyhedral outer approximation (or cutting plane) approach is obtained which, in each iteration, requires only solving linear programs and eigenvalue calculations. Based on the above arguments, Ramana [RAM93] proposed the following approach.

ALGORITHM 2.1 (***Ramana's Algorithm for Solving (UP)***).

```
Initialization
    $P^0 \leftarrow \{z \in \mathrm{I\!R}^d : Az \leq b\}$, STOP ← False, $k \leftarrow 0$

While STOP = False Do
    If $P^k = \emptyset$ Then
        STOP ← True  ($P \cap \{z \in \mathrm{I\!R}^d : U(z) \in \mathcal{U}_n\} = \emptyset$)
    Else
        Solve the linear optimization problem $\min_{z \in P^k} h^T z$ to obtain a vertex
        solution $z^k$ and a corresponding nonsingular basic system $B^k z \leq r^k$
        satisfying $B^k z^k = r^k$.
    Compute the eigenvalues of $U(z^k)$ indexed in increasing order.
    **If** $\lambda_1(U(z^k)) \geq 0$ **AND** $\lambda_{n-1}(U(z^k)) \leq 0$ **Then**
      STOP $\leftarrow$ **True** ($z^k$ is an optimal solution of (UP))
    **Else**
      **If** $\lambda_{n-1}(U(z^k)) > 0$ **Then**
        $(a^1)_i^k \leftarrow -\lambda_1\left(U^0 - U((B^k)^{-1}e_i)\right)\ ,\ i = 1, \ldots, d$
        $(\beta^1)^k \leftarrow -\lambda_{n-1}(U(z^k))$
        $P^k \leftarrow P^k \cap \{z \in \mathrm{I\!R}^d : ((a^1)^k)^T B^k z \ \leq\ ((a^1)^k)^T B^k z^k + (\beta^1)^k\}$
      **EndIf**
      **If** $\lambda_1(U(z^k)) < 0$ **Then**
        $(a^2)_i^k \leftarrow \lambda_n\left(U^0 - U((B^k)^{-1}e_i)\right)\ ,\ i = 1, \ldots, d$
        $(\beta^2)^k \leftarrow \lambda_1(U(z^k))$
        $P^k \leftarrow P^k \cap \{z \in \mathrm{I\!R}^d : ((a^2)^k)^T B^k z \ \leq\ ((a^2)^k)^T B^k z^k + (\beta^2)^k\}$
      **EndIf**
      $P^{k+1} \leftarrow P^k$, $k \leftarrow k + 1$
    **EndIf**
  **EndIf**
**EndWhile**
```

Example. Consider again Problem (UPE). The first vertex solution z^0 is obviously given by $(1, -2\sqrt{2})^T$ (see Figure 2.1(b)). The corresponding nonsingular basic system is

$$\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} z_{11} \\ z_{12} \end{pmatrix} \ \leq\ \begin{pmatrix} -1 \\ 2\sqrt{2} \end{pmatrix} .$$

For the eigenvalues of $U(\cdot)$ at z^0 we obtain

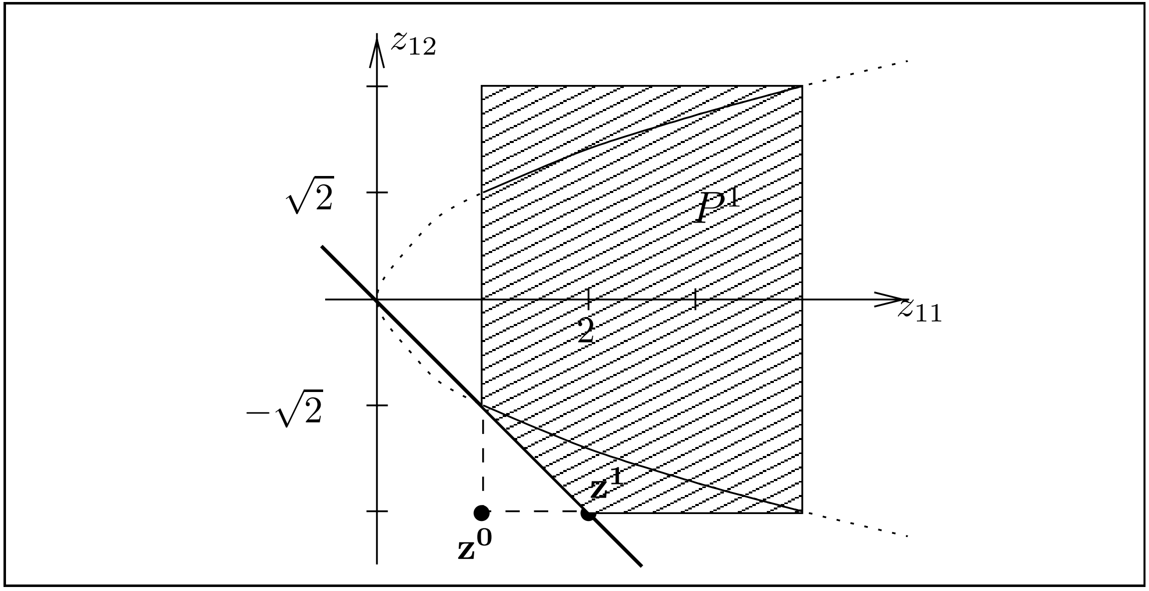
$$\lambda_1(U(z^0)) \ =\ \lambda_{n-1}(U(z^0)) \ =\ -1 \, .$$

The linear cut (2.3.4) is hence defined by

$$-z_{11} - \tfrac{1}{\sqrt{2}} z_{12} \ \leq\ 0 \, ,$$

and for the new outer approximation P^1 of the feasible region of (UPE) it follows $P^1 = \{z \in \mathrm{I\!R}^2 : 1 \leq z_{11} \leq 4\, ,\ -2\sqrt{2} \leq z_{12} \leq 2\sqrt{2}\, ,\ -z_{11} - \tfrac{1}{\sqrt{2}} z_{12} \leq 0\}$ (see Figure 2.2).

FIGURE 2.2. Ramana's cut for (UPE)



If Algorithm 2.1 stops after a finite number of iterations with a point z^k, respectively by detecting the emptiness of P^k, then it is obvious in view of the previous considerations that z^k is an optimal solution of (UP), respectively that the feasible region of (UP) is empty. Up to now it is an open question, whether Algorithm 2.1 is convergent in the sense that each accumulation point $z^\star$ of the sequence $\{z^k\}_{k\in\mathbb{N}}$ satisfies $z^\star \in \{z \in \mathbb{R}^d : Az \leq b, U(z) \in \mathcal{U}_n\}$. Since the sequences $\{((a^j)^k)^T B^k\}_{k\in\mathbb{N}}$ ($j = 1, 2$) might fail to be bounded, it does not seem that the convergence of Algorithm 2.1 can be guaranteed. For a related convergence theory of cutting plane algorithms in global optimization we refer to [HT96B].

REMARK 2.3.1. By applying another cutting plane for the case that the smallest eigenvalue of $U(z^k)$ is smaller than 0, Ramana was able to derive at least a partial convergence result. Let w^k be a normalized eigenvector of $U(z^k)$ corresponding to the smallest eigenvalue of this matrix. The linear cut

$$(w^k)^T U(z) w^k \;=\; \sum_{i=1}^{d} \left((w^k)^T U^i w^k\right) z_i + (w^k)^T U^0 w^k \;\geq\; 0 \tag{2.3.5}$$

is applicable, since there holds $(w^k)^T U(z^k) w^k = \lambda_1(U(z^k)) < 0$, and, for each $z \in \mathbb{R}^d$ with $U(z) \in \mathcal{U}_n$, it follows $(w^k)^T U(z) w^k \geq 0$. Note that each matrix $U \in \mathcal{U}_n$ must be positive semidefinite. If in Algorithm 2.1 the cut (2.3.5) is used instead of (2.3.4) and if the case $\lambda_{n-1}(U(z^k)) > 0$ occurs only a finite number of times, then it is provable (see [RAM93, pages 93f]) that this algorithm is convergent in the required sense.

It is the aim of the subsequent sections to overcome the above theoretical deficiency of Algorithm 2.1 by developing other in each case convergent outer approximation approaches for solving (UP).

2.4. Valid Cuts for Convergent Outer Approximation Algorithms

A first step towards convergent outer approximation algorithms for solving (UP) consists in requiring that in the affine matrix mapping (2.1.1)

$$U : \mathbb{R}^d \to \mathcal{S}_n :\Leftrightarrow U(z) = U^0 + \sum_{i=1}^{d} z_i U^i \,,$$

the matrices U^i ($i = 1, \ldots, d$) form an orthonormal system (ONS) with respect to the inner product (2.1.2). This is not a real restriction for the generality of the considered problems of type (UP). Each unary problem of this type is equivalent to another unary problem which fulfills this additional condition. This is the result of the following lemma.

LEMMA 2.4.1. *Let an arbitrary unary problem*

$$\begin{aligned} \min\ & h^T z \\ Az \le\ & b \\ \bar{U}(z) \in \mathcal{U}_n \,,\ & z \in \mathbb{R}^{\bar{d}} \end{aligned} \tag{UP1}$$

with $h \in \mathbb{R}^{\bar{d}}$, $A \in \mathbb{R}^{m\times\bar{d}}$ and $\bar{U} : \mathbb{R}^{\bar{d}} \to \mathcal{S}_n$, $\bar{U}(z) = U^0 + \sum_{i=1}^{\bar{d}} z_i \bar{U}^i$ be given. Then there exist a dimension $d \le \bar{d}$, vectors $h_1 \in \mathbb{R}^d$, $h_2 \in \mathbb{R}^{\bar{d}-d}$, matrices $A_1 \in \mathbb{R}^{m\times d}$, $A_2 \in \mathbb{R}^{m\times(\bar{d}-d)}$ and an ONS $\{U^i, i = 1, \ldots, d\}$ with respect to the inner product $\bullet$ defined in (2.1.2) such that the optimization problem

$$\begin{gathered} \min\ h_1^T x + h_2^T y \\ A_1 x + A_2 y \le b \\ U(x) = U^0 + \sum_{i=1}^{d} x_i U^i \in \mathcal{U}_n \\ x \in \mathbb{R}^d \,,\ y \in \mathbb{R}^{\bar{d}-d} \end{gathered} \tag{UP2}$$

is equivalent to (UP1).

PROOF: Determine a maximal linearly independent subset

$$\{\bar{U}^{i_j}, j = 1, \ldots, d\} \ \subset \ \{\bar{U}^i, i = 1, \ldots, \bar{d}\}$$

(so that the two linear spaces generated by the $\bar{U}^{i_j}$ respectively the $\bar{U}^i$ have equal dimension). Assume, for ease, that there holds $\{i_1, \ldots, i_d\} = \{1, \ldots, d\}$. The matrices $\bar{U}^j$ ($j \in \{d+1, \ldots, \bar{d}\}$) are contained in the linear space generated by the matrices $\bar{U}^i$ ($i = 1, \ldots, d$). Therefore, there exists, for each $j \in \{1, \ldots, \bar{d}-d\}$, a vector $\lambda^j \in \mathrm{I\!R}^d$ with

$$\bar{U}^{d+j} \ = \ \sum_{i=1}^{d} \lambda_i^j \bar{U}^i \, .$$

Set $L = (\lambda^1, \ldots, \lambda^{\bar{d}-d}) \in \mathrm{I\!R}^{d \times (\bar{d}-d)}$. Use now the Gram-Schmidt procedure (see, e.g., [GVL89, Chapter 5]) in order to generate from $\{\bar{U}^i, i = 1, \ldots, d\}$ a corresponding ONS $\{U^i, i = 1, \ldots, d\}$. Let, for $i \in \{1, \ldots, d\}$, $\mu^i \in \mathrm{I\!R}^d$ be the unique vector satisfying

$$\bar{U}^i \ = \ \sum_{j=1}^{d} \mu_j^i U^j \, .$$

Since the function which maps the $\bar{U}^i$ onto the U^j ($j = 1, \ldots, d$) is a homeomorphism we know that the matrix $M = (\mu^1, \ldots, \mu^d) \in \mathrm{I\!R}^{d \times d}$ is regular. Let $z = (\bar{z}, \hat{z})^T$ with $\bar{z} \in \mathrm{I\!R}^d$ and $\hat{z} \in \mathrm{I\!R}^{\bar{d}-d}$ be an arbitrary element of $\mathrm{I\!R}^{\bar{d}}$. Let, furthermore, the matrix $A \in \mathrm{I\!R}^{m \times \bar{d}}$ be given by $A = (\bar{A}, \hat{A})$ with $\bar{A} \in \mathrm{I\!R}^{m \times d}$ and $\hat{A} \in \mathrm{I\!R}^{m \times (\bar{d}-d)}$, and the vector $h \in \mathrm{I\!R}^{\bar{d}}$ be given by $h = (\bar{h}, \hat{h})^T \in \mathrm{I\!R}^{d+(\bar{d}-d)}$. Set

$$x \ = \ M(\bar{z} + L\hat{z}) \quad , \qquad y \ = \ \hat{z} \quad ,$$

and

$$A_1 \ = \ \bar{A}M^{-1} \quad , \quad A_2 \ = \ \hat{A} - \bar{A}L$$

$$h_1 \ = \ (M^{-1})^T \bar{h} \quad , \quad h_2 \ = \ \hat{h} - L^T \bar{h} \, .$$

Then it follows

$$h^T z \ = \ \bar{h}^T \bar{z} + \hat{h}^T \hat{z} \ = \ \bar{h}^T (M^{-1}x - L\hat{z}) + \hat{h}^T \hat{z} \ = \ h_1^T x + h_2^T y \, ,$$

$$Az \ = \ \bar{A}\bar{z} + \hat{A}\hat{z} \ = \ \bar{A}(M^{-1}x - L\hat{z}) + \hat{A}\hat{z} \ = \ A_1 x + A_2 y \, ,$$

and

$$\begin{aligned}\bar{U}(z) &= U^0 + \sum_{i=1}^{d} \bar{z}_i \bar{U}^i + \sum_{i=d+1}^{\bar{d}} \hat{z}_i \bar{U}^i = U^0 + \sum_{i=1}^{d} \left(\bar{z}_i + \sum_{j=d+1}^{\bar{d}} \hat{z}_j \lambda_i^j \right) \bar{U}^i \\ &= U^0 + \sum_{l=1}^{d} \underbrace{\left(\sum_{i=1}^{d} \mu_l^i (\bar{z}_i + \sum_{j=d+1}^{\bar{d}} \hat{z}_j \lambda_i^j) \right)}_{=(M(\bar{z}+L\hat{z}))_l = x_l} U^l = U(x) \, .\end{aligned}$$

Since the matrix M is regular the previous calculations demonstrate a one-to-one relation between the feasible points of (UP1) and (UP2). This shows the equivalence of both problems. ■

Even though Problem (UP2) has a more general form than Problem (UP) we will develop the following theory and solution methods only for unary problems of type (UP). This is motivated on the one hand by the fact that the transformation presented in Section 2.2, which links the all-quadratic problems of type (QP) to equivalent problems of type (UP), yields an ONS $\{U^{ij}, 1 \leq i \leq j < n+1\}$ in (2.2.1). Since it is the purpose of this research study to develop solution methods for (QP) it is, therefore, sufficient to consider the more restricted form (UP) of unary problems instead of (UP2). On the other hand, the following theory and solution methods can be extended by slight changes to problems of type (UP2). However, this leads to increasing technical effort, what we would like to avoid.

The following lemma shows the postulated fact that the matrices U^{ij} $(1 \leq i \leq j < n+1)$ defined in (2.2.1) form an ONS with respect to the inner product given by (2.1.2).

LEMMA 2.4.2. *Let* $E_{ij} = e_i e_j^T \in \mathbb{R}^{(n+1)\times(n+1)}$ $(i, j = 1, \ldots, n+1)$ *be given as in Section 2.2. Then the matrices*

$$\begin{aligned} U^{ii} &= E_{ii} \quad , \; i = 1, \ldots, n \\ U^{ij} &= \tfrac{1}{\sqrt{2}}(E_{ij} + E_{ji}) \quad , \; 1 \leq i < j \leq n+1 \end{aligned}$$

form an ONS with respect to the inner product $\bullet$ *defined in (2.1.2).*

PROOF: This result can be verified by straightforward calculations. ■

With the orthonormal property of the set $\{U^i, i = 1, \ldots, d\}$ we are now able to derive a relation between the Euclidean distance of two points $z, \bar{z} \in \mathbb{R}^d$ and the *distance* between the two corresponding matrices $U(z)$ and $U(\bar{z})$. In order to measure the *distance* between two matrices we use a suitable matrix norm. Let $\|A\|_F = \sqrt{A \bullet A}$ ($A \in \mathcal{S}_n$) denote the norm induced by the inner product (2.1.2) – the so-called **Frobenius-norm**.

LEMMA 2.4.3. *Let $\{U^i, i = 1, \ldots, d\} \subset \mathcal{S}_n$ form an ONS with respect to the inner product $\bullet$ defined in (2.1.2). Then, for each $z, \bar{z} \in \mathbb{R}^d$, there holds*

$$\|\sum_{i=1}^{d}(z-\bar{z})_i U^i\|_F \ = \ \|z-\bar{z}\|_2 \,. \tag{2.4.1}$$

PROOF: By the orthonormality of $\{U^i, i = 1, \ldots, d\}$ we know that, for each $i, j \in \{1, \ldots, d\}$, there holds

$$\operatorname{tr}\left((U^i)^T U^j\right) \ = \ U^i \bullet U^j \ = \ \begin{cases} 1 & \text{, if } i = j \\ 0 & \text{, otherwise} \end{cases} .$$

Thus, for each $z, \bar{z} \in \mathbb{R}^d$, it follows

$$\begin{aligned} \|\sum_{i=1}^{d}(z-\bar{z})_i U^i\|_F^2 \ &= \ \operatorname{tr}\left((\sum_{i=1}^{d}(z-\bar{z})_i U^i)^T(\sum_{i=1}^{d}(z-\bar{z})_i U^i)\right) \\ &= \ \sum_{i,j=1}^{d}(z-\bar{z})_i(z-\bar{z})_j \operatorname{tr}\left((U^i)^T U^j\right) \\ &= \ \sum_{i=1}^{d}(z-\bar{z})_i^2 \ = \ \|z-\bar{z}\|_2^2 \,. \end{aligned}$$

■

The combination of (2.4.1) with Weyl's inequalities (2.3.1) allows us to prove that for arbitrary points $z, \bar{z} \in \mathbb{R}^d$ the distance between the eigenvalues of $U(z)$ and $U(\bar{z})$ is at least as big as the Euclidean distance between these points. With this result of the following theorem we will develop a valid cut for a convergent outer approximation algorithm.

THEOREM 2.4.4. *Let $\{U^i, i = 1, \ldots, d\} \subset \mathcal{S}_n$ form an ONS with respect to the inner product $\bullet$ defined in (2.1.2), and let $U : \mathbb{R}^d \to \mathcal{S}_n$ be an affine matrix* *mapping of the form*

$$z \;\to\; U(z) \;=\; U^0 + \sum_{i=1}^{d} z_i U^i$$

with $U^0 \in \mathcal{S}_n$. Assume that the eigenvalues of the matrices involved are indexed in an increasing order. Then, for each $z, \bar{z} \in \mathbb{R}^d$, there holds

$$\lambda_{n-1}\left(U(z)\right) \;\geq\; \lambda_{n-1}\left(U(\bar{z})\right) - \|z - \bar{z}\|_2 \tag{2.4.2}$$

and

$$\lambda_1\left(U(z)\right) \;\leq\; \lambda_1\left(U(\bar{z})\right) + \|z - \bar{z}\|_2 \,. \tag{2.4.3}$$

PROOF: Since the Frobenius norm is an upper bound for the spectral radius $\rho(S) = \max\{|\lambda|, \lambda \text{ eigenvalue of } S\}$ ($S \in \mathcal{S}_n$) (see, e.g., [ZUR64]), one obtains by means of Lemma 2.3.2

$$\begin{aligned}
\lambda_{n-1}(U(z)) \;&=\; \lambda_{n-1}(U(z-\bar{z}) + U(\bar{z}) - U^0) \;=\; \lambda_{n-1}(\sum_{i=1}^{d}(z-\bar{z})_i U^i + U(\bar{z})) \\
&\geq\; \lambda_{n-1}(U(\bar{z})) + \lambda_1(\sum_{i=1}^{d}(z-\bar{z})_i U^i) \\
&\geq\; \lambda_{n-1}(U(\bar{z})) - \|\sum_{i=1}^{d}(z-\bar{z})_i U^i\|_F \;=\; \lambda_{n-1}(U(\bar{z})) - \|z-\bar{z}\|_2 \,.
\end{aligned}$$

Similarly, inequality (2.4.3) follows from

$$\begin{aligned}
\lambda_1(U(z)) \;&=\; \lambda_1(U(z-\bar{z}) + U(\bar{z}) - U^0) \;=\; \lambda_1(\sum_{i=1}^{d}(z-\bar{z})_i U^i + U(\bar{z})) \\
&\leq\; \lambda_1(U(\bar{z})) + \lambda_n(\sum_{i=1}^{d}(z-\bar{z})_i U^i) \\
&\leq\; \lambda_1(U(\bar{z})) + \|\sum_{i=1}^{d}(z-\bar{z})_i U^i\|_F \;=\; \lambda_1(U(\bar{z})) + \|z-\bar{z}\|_2 \,.
\end{aligned}$$

■

REMARK 2.4.1. The result of Theorem 2.4.4 can also be derived by a combination of Lemma 2.4.3 and the Hoffman-Wielandt inequality given in [HW53]. Indeed, let $A, B \in \mathcal{S}_n$ be two arbitrary matrices with eigenvalues $\alpha_1, \ldots, \alpha_n$ and $\beta_1, \ldots, \beta_n$ indexed in increasing order. The Hoffman-Wielandt inequality in [HW53] says that there is a permutation $\pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$ satisfying

$$\sum_{i=1}^{n} |\alpha_i - \beta_{\pi(i)}|^2 \;\leq\; \|A - B\|_F^2 \,. \tag{2.4.4}$$

If we denote by Π the set of all permutations of $\{1, \ldots, n\}$, then (2.4.4) is equivalent to

$$\min_{\pi \in \Pi} \sum_{i=1}^{n} |\alpha_i - \beta_{\pi(i)}|^2 \;\leq\; \|A - B\|_F^2 \,.$$

Set $\alpha = (\alpha_1, \ldots, \alpha_n)^T$ and $\beta = (\beta_1, \ldots, \beta_n)^T$. It can be proven by an induction with respect to the dimension n that there holds

$$\max_{\pi \in \Pi} \sum_{i=1}^{n} \alpha_i \beta_{\pi(i)} \;=\; \alpha^T \beta \,.$$

Using this fact we obtain

$$\begin{aligned} \min_{\pi \in \Pi} \sum_{i=1}^{n} |\alpha_i - \beta_{\pi(i)}|^2 \;&=\; \|\alpha\|_2^2 + \|\beta\|_2^2 - 2 \max_{\pi \in \Pi} \sum_{i=1}^{n} \alpha_i \beta_{\pi(i)} \\ &=\; \|\alpha - \beta\|_2^2 \,, \end{aligned}$$

and in view of (2.4.4) it follows, for each $i \in \{1, \ldots, n\}$,

$$|\alpha_i - \beta_i| \;\leq\; \|A - B\|_F \,. \tag{2.4.5}$$

If we apply this relation to the situation of Theorem 2.4.4, the use of Lemma 2.4.3 yields the inequalities (2.4.2) and (2.4.3).

As in the description of Ramana's cuts introduced in the previous section, let $\bar{z} \in \mathbb{R}^d$ be an optimal solution of an LP-relaxation of (UP) satisfying $U(\bar{z}) \notin \mathcal{U}_n$. In view of Lemma 2.3.1(iii) we know that

$$\epsilon(\bar{z}) \;:=\; \max\left\{\lambda_{n-1}(U(\bar{z})), -\lambda_1(U(\bar{z}))\right\}$$

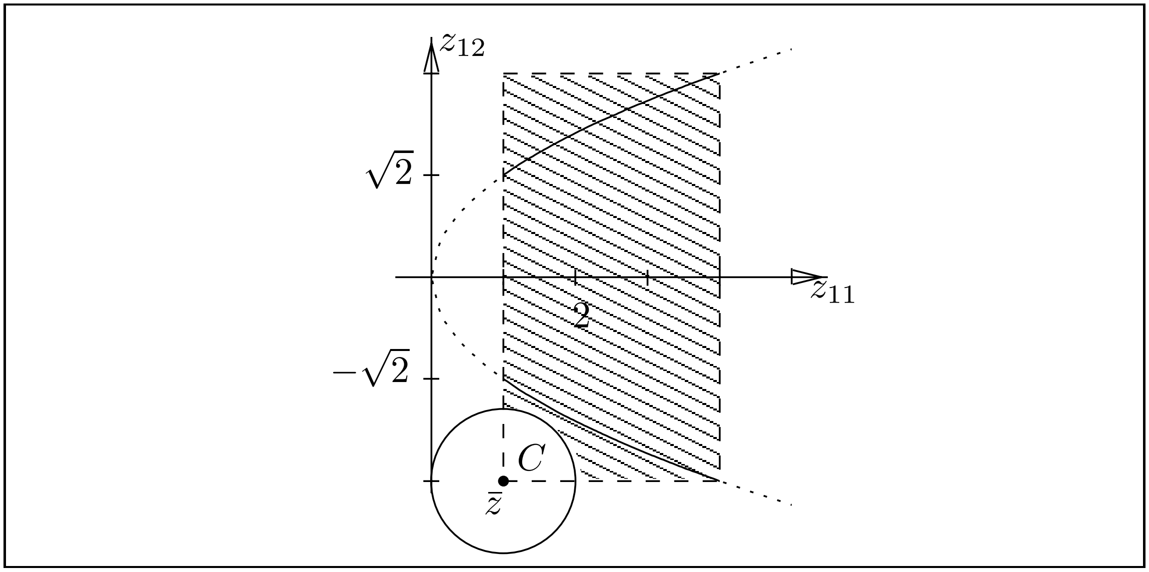
must be greater than 0. From Theorem 2.4.4 it follows that each point $z \in \mathbb{R}^d$ contained in a ball (with respect to the Euclidean norm), which has a radius equal to $\epsilon(\bar{z})$ and is centered at $\bar{z}$, cannot be feasible for (UP). Therefore, we see that

$$\ell_{\bar{z}}(z) \;:=\; \epsilon(\bar{z}) - \|z - \bar{z}\|_2 \;\leq\; 0 \tag{2.4.6}$$

is a valid cut, i.e., we know $\ell_{\bar{z}}(\bar{z}) > 0$, and, for each $z \in \mathbb{R}^d$ with $U(z) \in \mathcal{U}_n$, there holds $\ell_{\bar{z}}(z) \leq 0$.

Example. In the situation of Problem (UPE) we know that $\bar{z} = (1, -2\sqrt{2})^T$ is an optimal solution of an LP-relaxation of this problem with $\epsilon(\bar{z}) = 1$ (see page 31). In view of the above arguments it follows that each point contained in the

FIGURE 2.3. First quadratic cut for (UPE)



circle C centered at $\bar{z}$ with radius 1 (see Figure 2.3) is not feasible for (UPE).

If we replace in Ramana's Algorithm 2.1 the linear cuts used there by $\ell_{z^k}(z) \leq 0$, then we obtain a convergent outer approximation algorithm for solving (UP), as the following theorem shows.

THEOREM 2.4.5. *Let $\{z^k\}_{k\in\mathbb{N}}$ be a sequence of points in the polytope $P = \{z \in \mathbb{R}^d : Az \leq b\}$ satisfying, for each $k, i \in \mathbb{N}$ with $k < i$,*

$$\ell_{z^k}(z^i) \;\leq\; 0\,. \tag{2.4.7}$$

Then every accumulation point $z^\star$ of $\{z^k\}_{k\in\mathbb{N}}$ satisfies $U(z^\star) \in \mathcal{U}_n$.

PROOF: Let $z^\star$ be an accumulation point of the sequence $\{z^k\}_{k\in\mathbb{N}}$ and let $\{z^{k_q}\}_{q\in\mathbb{N}}$ be a subsequence converging to $z^\star$. From (2.4.7) it follows that, for each $q \in \mathbb{N}$, we know that

$$\ell_{z^{k_q}}(z^{k_{q+1}}) \;\leq\; 0\,.$$

Since $\|z^{k_{q+1}} - z^{k_q}\|_2 \to 0$ ($q \to \infty$), this relation implies – in view of (2.4.6) and because of $\max\{\lambda_{n-1}(U(z^{k_q})), -\lambda_1(U(z^{k_q}))\} \geq 0$ ($q \in \mathbb{N}$) – that there holds

$$\max\left\{\lambda_{n-1}(U(z^{k_q})), -\lambda_1(U(z^{k_q}))\right\} \;\to\; 0 \quad (q \to \infty)\,.$$

From this ensues

$$\lambda_1(U(z^\star)) \;=\; \lambda_{n-1}(U(z^\star)) \;=\; 0$$

by the continuity of the eigenvalue functionals $\lambda_1, \lambda_{n-1} : \mathcal{S}_n \to \mathrm{I\!R}$. This is equivalent to $U(z^\star) \in \mathcal{U}_n$ because of Lemma 2.3.1 and completes the proof. ■

We have now a convergent outer approximation approach for solving (UP). However, the possible cut is nonlinear, in particular reverse convex, such that an algorithm using this cut directly induces difficult subproblems. In the next three sections we will discuss ways to overcome this practical difficulty.

2.5. Basic Idea for Convergent Implementable Algorithms

In order to apply the results of the previous section we assume in this and in the subsequent sections that the matrices $\{U^i, i = 1, \ldots, d\}$ defining the matrix mapping in (UP) form an orthonormal system with respect to the inner product $\bullet$ defined in (2.1.2). We assume furthermore that the polytope $P = \{z \in \mathrm{I\!R}^d : Az \leq b\}$ is not empty, what can be tested by the first phase of the Simplex-Algorithm.

Let $\bar{P}$ be the feasible set of an arbitrary LP-relaxation of (UP). If a point $\bar{z} \in \bar{P}$ satisfying $U(\bar{z}) \notin \mathcal{U}_n$ is given, then we have seen in Section 2.4 that it is possible to cut an Euclidean norm ball $B_{\bar{z}}$ centered at $\bar{z}$ with radius $\epsilon(\bar{z}) = \max\{\lambda_{n-1}(U(\bar{z})), -\lambda_1(U(\bar{z}))\}$ out of the polytope $\bar{P}$ without affecting the unarity.

Let $Q_{\bar{z}} = \{z \in \mathrm{I\!R}^d : \bar{q}_i^T z \leq \bar{c}_i \,,\, i = 1, \ldots, l\}$ be a polyhedron ($\bar{q}_i \in \mathrm{I\!R}^d$, $\bar{c}_i \in \mathrm{I\!R}$, $i = 1, \ldots, l$) with the properties

$$\bar{P} \cap Q_{\bar{z}} \;\subset\; \bar{P} \cap B_{\bar{z}} \tag{2.5.1}$$

and, for each $i \in \{1, \ldots, l\}$,

$$d(\bar{z}, H(\bar{q}_i, \bar{c}_i)) \;\geq\; \rho\epsilon(\bar{z})\,, \tag{2.5.2}$$

where $d(\bar{z}, H(\bar{q}_i, \bar{c}_i))$ denotes the Euclidean distance of the hyperplane $H(\bar{q}_i, \bar{c}_i) = \{z \in \mathrm{I\!R}^d : \bar{q}_i^T z = \bar{c}_i\}$ to the point $\bar{z}$, and $\rho \in (0, 1]$ is a positive real number. In view of (2.5.1) we see that $Q_{\bar{z}}$ can be cut out of the polytope $\bar{P}$ without eliminating a feasible point of (UP). Actually, the set $\bar{P} \cap Q_{\bar{z}}$ is an inner approximation polytope of the part of $B_{\bar{z}}$ belonging to $\bar{P}$ and contains no element of $\bar{P}$ lying outside the ball $B_{\bar{z}}$. Property (2.5.2) guarantees, furthermore, that each point located within $\bar{P} \setminus Q_{\bar{z}}$ has a distance greater than $\rho\epsilon(\bar{z})$ to the point $\bar{z}$. If it is possible to construct such a polyhedron for each infeasible point $\bar{z}$, then we are able to develop a convergent algorithm for solving (UP). How this can be done is the content of the present section. In the next section we will propose three different possibilities for constructing appropriate polyhedra.

Assume now that for each point $\bar{z}$ belonging to a polytope $\bar{P} \subset P$ and satisfying $U(\bar{z}) \notin \mathcal{U}_n$ a polyhedron $Q_{\bar{z}}$ with Properties (2.5.1) and (2.5.2) is known. Of course we cannot cut the set $Q_{\bar{z}}$ out of $\bar{P}$ in one step. The closure of $\bar{P} \setminus Q_{\bar{z}}$ is not necessarily a polytope and, thus, an algorithm doing this would induce difficult subproblems, as it is the case by using the quadratic cut directly. However, in contrast to the Euclidean norm ball $B_{\bar{z}}$ the polyhedron $Q_{\bar{z}}$ is described by a finite number of linear constraints. If we construct l new polytopes $\bar{P}_i$ ($i = 1, \dots, l$) by adding one of the constraints describing $Q_{\bar{z}}$ to the constraints describing $\bar{P}$, then we know that the union of the $\bar{P}_i$'s ($i = 1, \dots, l$) contains no point of the interior of $Q_{\bar{z}}$, but all feasible elements of $\bar{P}$. Applying this strategy the algorithm is as follows.

Algorithm 2.2 (***Basic Convergent Algorithm for Solving (UP)***).

```
Initialization
    Choose $\rho \in (0,1]$ and $l \in \mathbb{N}$, and set $P^0 \leftarrow \{z \in \mathbb{R}^d : Az \leq b\}$.
    Solve the linear optimization problem (LP) $\min_{z \in P^0} h^T z$, and let $z^0$ be an
    optimal solution with optimal value $\mu_{P^0} = h^T z^0$.
    $\mu^0 \leftarrow \mu_{P^0}$, $\mathcal{P} \leftarrow \{P^0\}$, STOP $\leftarrow$ False, $k \leftarrow 0$

While STOP = False Do
    Compute the eigenvalues of $U(z^k)$ indexed in increasing order.
    If $\lambda_1(U(z^k)) \geq 0$ AND $\lambda_{n-1}(U(z^k)) \leq 0$ Then                (SC1)
        STOP $\leftarrow$ True ($z^k$ is an optimal solution of (UP))
    Else
        $\epsilon(z^k) \leftarrow \max\{\lambda_{n-1}(U(z^k)), -\lambda_1(U(z^k))\}$
        Construct a polyhedron $Q^k = \{z \in \mathbb{R}^d : (q_i^k)^T z \leq c_i^k\ ,\ i = 1, \dots, l\}$
        satisfying
            $P^k \cap Q^k \subset P^k \cap \{z \in \mathbb{R}^d : \|z - z^k\|_2 \leq \epsilon(z^k)\}$       (PR1)
        and, for each $i \in \{1, \dots, l\}$,
            $d(z^k, H(q_i^k, c_i^k)) = \frac{|(q_i^k)^T z^k - c_i^k|}{\|q_i^k\|_2} \geq \rho\epsilon(z^k)$.       (PR2)
        For $i = 1$ To $l$ Do
            $P_i^k \leftarrow P^k \cap \{z \in \mathbb{R}^d : (q_i^k)^T z \geq c_i^k\}$
            If $P_i^k \neq \emptyset$ Then
            Solve the LP $\min_{z\in P_i^k} h^T z$, and let $z_i^k$ be an optimal solution
            with optimal value $\mu_{P_i^k} = h^T z_i^k$.
            $\mathcal{P} \leftarrow \mathcal{P} \cup \{P_i^k\}$
          **EndIf**
        **EndFor**
        $\mathcal{P} \leftarrow \mathcal{P} \setminus \{P^k\}$
        **If** $\mathcal{P} = \emptyset$ **Then**                                        (SC2)
          STOP ← **True** ($P^0 \cap \{z \in \mathbb{R}^d : U(z) \in \mathcal{U}_n\} = \emptyset$)
        **Else**
          $\mu^{k+1} \leftarrow \min_{P\in\mathcal{P}} \mu_P$
          Choose $P^{k+1} \in \mathcal{P}$ and $z^{k+1} \in P^{k+1}$ with $\mu^{k+1} = \mu_{P^{k+1}} = h^T z^{k+1}$.
        **EndIf**
      **EndIf**
      $k \leftarrow k+1$
    **EndWhile**
```

REMARK 2.5.1.

(a) It is known that the Euclidean distance $d(\bar z, H)$ of an arbitrary hyperplane $H = \{z \in \mathbb{R}^d : q^T z = c\}$ ($q \in \mathbb{R}^d$, $c \in \mathbb{R}$) to a point $\bar z \in \mathbb{R}^d$ is given by

$$d(\bar z, H) \;=\; \frac{|q^T \bar z - c|}{\|q\|_2}\,. \tag{2.5.3}$$

(b) The choice of $\rho \in (0,1]$ and $l \in \mathbb{N}$ depends on the the used polyhedra, as we will see in the next section.

(c) Algorithm 2.2 is not a pure outer approximation scheme – in contrast to Algorithm 2.1. In each iteration we combine a better outer approximation of the feasible region of (UP) with a subdivision of this feasible set. Notice that – from a numerical point of view – this subdivision process can lead to excessive storage requirements, since in each iteration we eliminate only one polytope from the collection $\mathcal{P}$, but we add up to l new sets.

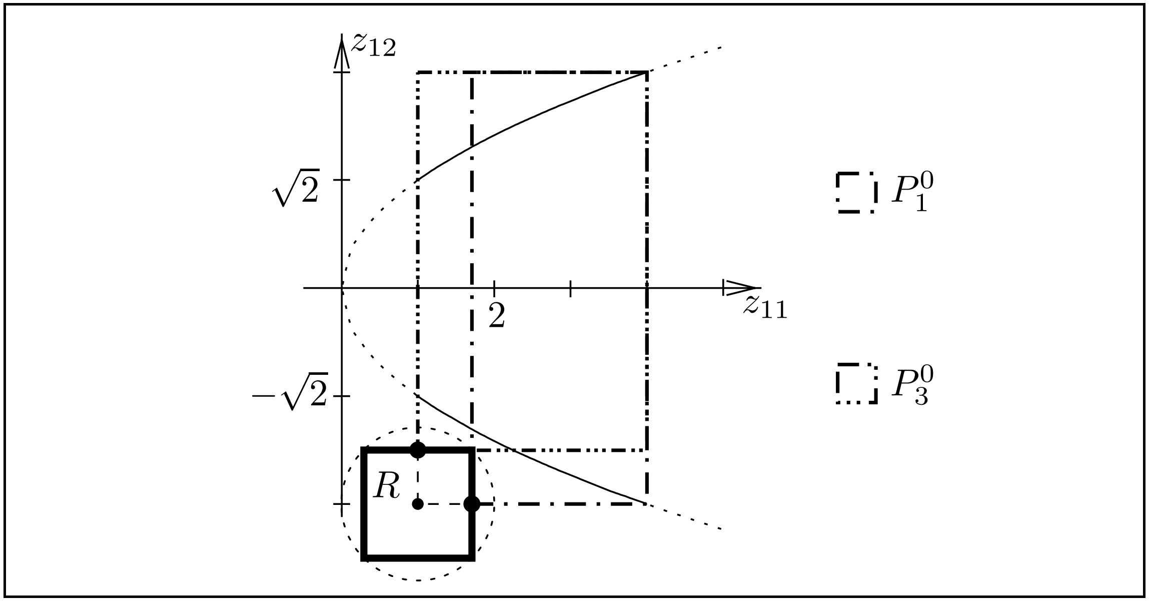
Example. In order to illustrate Algorithm 2.2 let us consider again Problem (UPE). The initialization polytope P^0 is given by the set $\{z \in \mathbb{R}^2 : 1 \le z_{11} \le 4\,,\, -2\sqrt{2} \le z_{12} \le 2\sqrt{2}\}$ and the first optimal solution is $z^0 = (1, -2\sqrt{2})^T$ with $\epsilon(z^0) = 1$ and $\mu^0 = -1$. Since the square R with edge-length $\sqrt{2}$ and centered at z^0 is contained in the circle C with radius 1 (compare with

Figure 2.3), we can use R as the necessary polytope Q^0. Thus, the first subdivision of the feasible region of (UPE) leads to the polytopes

$$\begin{aligned}
P_1^0 &= P^0 \cap \{z \in \mathbb{R}^2 : z_{11} \geq 1 + 0.5\sqrt{2}\} \\
P_2^0 &= P^0 \cap \{z \in \mathbb{R}^2 : z_{11} \leq 1 - 0.5\sqrt{2}\} = \emptyset \\
P_3^0 &= P^0 \cap \{z \in \mathbb{R}^2 : z_{12} \geq -1.5\sqrt{2}\} \\
P_4^0 &= P^0 \cap \{z \in \mathbb{R}^2 : z_{12} \leq -2.5\sqrt{2}\} = \emptyset
\end{aligned}$$

(see Figure 2.4). We obtain the new solutions $z_1^0 = (1 + 0.5\sqrt{2}, -2\sqrt{2})^T$ with

FIGURE 2.4. First iteration of Algorithm 2.2 applied for (UPE)



objective function value $-1 + 0.5\sqrt{2}$ and $z_3^0 = (1, -1.5\sqrt{2})^T$ with value -0.5. Hence, the new polytope for iteration 1 is $P^1 = P_3^0$ with $\mu^1 = -0.5$.

In order to guarantee the correctness of Algorithm 2.2 we first prove that in iteration $k \in \mathbb{N}$ each feasible point of Problem (UP) is contained in at least one of the polytopes belonging to the current collection $\mathcal{P}$.

LEMMA 2.5.1. *Let $\mathcal{P}$ be the collection of polytopes at iteration $k \in \mathbb{N}$ of Algorithm 2.2 and denote by $F = \{z \in \mathbb{R}^d : Az \leq b, U(z) \in \mathcal{U}_n\}$ the feasible set of (UP). Then there holds*

$$\bigcup_{P \in \mathcal{P}} P \supset F\,. \tag{2.5.4}$$

PROOF: We show this result by an induction with respect to the iteration counter k.

For $k = 0$, there holds $\mathcal{P} = \{P^0\}$ with $P^0 = \{z \in \mathbb{R}^d : Az \leq b\}$, and hence (2.5.4) is fulfilled. Assume that (2.5.4) holds at the beginning of iteration k . Then it suffices to show that

$$\bigcup_{i=1}^{l} P_i^k \ \supset \ F \cap P^k \ . \tag{2.5.5}$$

Let $\hat{z}$ be an element of $F \cap P^k$. From Theorem 2.4.4 we know that

$$\lambda_{n-1}(U(\hat{z})) \ \geq \ \lambda_{n-1}(U(z^k)) - \|\hat{z} - z^k\|_2$$

and

$$\lambda_1(U(\hat{z})) \ \leq \ \lambda_1(U(z^k)) + \|\hat{z} - z^k\|_2 \ .$$

Since $\hat{z}$ is a feasible point of (UP), Lemma 2.3.1 tells us that there holds $\lambda_{n-1}(U(\hat{z})) = \lambda_1(U(\hat{z})) = 0$, and hence

$$\|\hat{z} - z^k\|_2 \ \geq \ \max\{\lambda_{n-1}(U(z^k)), -\lambda_1(U(z^k))\} = \epsilon(z^k) \ . \tag{2.5.6}$$

The polytopes P_i^k ($i = 1, \ldots, l$) are constructed such that

$$\bigcup_{i=1}^{l} P_i^k \ = \ P^k \setminus \{z \in \mathbb{R}^d : (q_i^k)^T z < c_i^k \ , \ i = 1, \ldots, l\} =: \hat{P}^k \ ,$$

and regarding Property (PR1) of the polyhedron Q^k we know, furthermore, that

$$\hat{P}^k \ \supset \ P^k \cap \{z \in \mathbb{R}^d : \|z - z^k\|_2 \geq \epsilon(z^k)\} \ .$$

The point $\hat{z}$ is an element of P^k. Therefore, we obtain in view of (2.5.6) that

$$\hat{z} \ \in \ P^k \cap \{z \in \mathbb{R}^d : \|z - z^k\|_2 \geq \epsilon(z^k)\} \ \subset \ \bigcup_{i=1}^{l} P_i^k \ ,$$

which proves (2.5.5). ■

If Algorithm 2.2 stops with $\mathcal{P} = \emptyset$, it follows immediately by (2.5.4) that the feasible region of (UP) is empty. Moreover, Relation (2.5.4) implies that μ^k is at each iteration $k \in \mathbb{N}$ a lower bound for the optimal value of (UP), i.e., for each $k \in \mathbb{N}$, there holds

$$\mu^k \ \leq \ \min_{z \in F} h^T z \ . \tag{2.5.7}$$

Therefore, we know that, if Algorithm 2.2 terminates with a point z^k, then z^k is an optimal solution of Problem (UP). Indeed, in view of the stopping criterion (SC1) the point z^k must be feasible for (UP) (see Lemma 2.3.1) and with (2.5.7) we obtain

$$\mu^k \;=\; h^T z^k \;\le\; \min_{z\in F} h^T z \;\le\; h^T z^k\,, \tag{2.5.8}$$

which shows the optimality of z^k.

For the case that Algorithm 2.2 does not stop after a finite number of iterations, the following theorem guarantees the convergence of our approach in the required sense.

THEOREM 2.5.2. *If Algorithm 2.2 generates an infinite point sequence $\{z^k\}_{k\in\mathbb{N}}$, then each accumulation point $z^\star$ of this sequence is an optimal solution of Problem (UP).*

PROOF: Let $z^\star$ be an accumulation point of the sequence $\{z^k\}_{k\in\mathbb{N}}$ and let $\{z^{k_q}\}_{q\in\mathbb{N}}$ be a subsequence converging to $z^\star$. By passing to a subsequence, if necessary, we can assume that the corresponding sequence $\{P^{k_q}\}_{q\in\mathbb{N}}$ of polytopes is decreasing, i.e., for each $q\in\mathbb{N}$, there holds

$$P^{k_{q+1}} \;\subset\; P^{k_q}\,, \tag{2.5.9}$$

and, moreover, that $P^{k_{q+1}}$ has been generated by adding constraints to the set of inequalities describing P^{k_q}. In view of Relation (2.5.7) it suffices to show that $z^\star$ is a feasible point of (UP), i.e., $z^\star \in F$ (see also Relation (2.5.8)). Because of (2.5.9) we know that, for each $q\in\mathbb{N}$, there is an index $i\in\{1,\ldots,l\}$ with

$$P^{k_{q+1}} \;\subset\; P^{k_q} \cap \{z\in\mathbb{R}^d : (q_i^{k_q})^T z \ge c_i^{k_q}\}\,.$$

Using Property (PR2) of the hyperplanes describing the polyhedra Q^{k_q} ($q\in\mathbb{N}$) we see that, for each $q\in\mathbb{N}$,

$$\|z^{k_{q+1}} - z^{k_q}\|_2 \;\ge\; \rho\epsilon(z^{k_q}) \;>0\,. \tag{2.5.10}$$

With the definition of $\epsilon(z^{k_q})$ ($q\in\mathbb{N}$) and the continuity of the eigenvalue functionals it follows

$$\begin{array}{ccccccc}
0 \;\le\; \max\{\lambda_{n-1}(U(z^{k_q}))\,, & -\lambda_1(U(z^{k_q}))\} & \le & \frac{1}{\rho}\| z^{k_q} & - & z^{k_{q+1}} & \|_2 \\
\downarrow & \downarrow & & \downarrow & & \downarrow & (q\to\infty) \\
0 \;\le\; \max\{\lambda_{n-1}(U(z^\star))\,, & -\lambda_1(U(z^\star))\;\} & \le & \frac{1}{\rho}\| z^\star & - & z^\star & \|_2 \;=\; 0.
\end{array}$$

This implies in view of Lemma 2.3.1 the feasibility of $z^\star$. ■

REMARK 2.5.2. As the previous considerations show, it is not necessary that in the formulation of Algorithm 2.2 the number $l \in \mathbb{N}$ and the positive real value ρ are chosen independent of the iteration counter k. As long as there is a number $L \in \mathbb{N}$ with $l^k \leq L$ ($k \in \mathbb{N}$) and a constant $c > 0$ with $\rho^k \geq c$ ($k \in \mathbb{N}$) the correctness of this solution method for (UP) can be proven.

Under the assumption that appropriate polyhedra Q^k ($k \in \mathbb{N}$) can be constructed we have now a convergent algorithm with linear subproblems for solving unary problems of type (UP). In order to obtain implementable algorithms we still have to specify, how such polyhedra can be determined. In the next section we present three possibilities for the choice of such sets.

2.6. Appropriate Polyhedra for Algorithm 2.2

Let z^k ($k \in \mathbb{N}$) be the current point at iteration k of Algorithm 2.2 with $\epsilon(z^k) = \max\{\lambda_{n-1}(U(z^k)), -\lambda_1(U(z^k))\} > 0$, and let B_{z^k} be the corresponding Euclidean norm ball with radius $\epsilon(z^k)$ centered at z^k. There exists of course an infinite number of polyhedra $Q^k \subset \mathbb{R}^d$ satisfying the required properties, if $\rho \in (0, 1]$ and $l \in \mathbb{N}$ are chosen accordingly. In order to obtain an efficient algorithm such polyhedra should satisfy some criteria apart from the necessary properties (PR1) and (PR2). First of all these sets should be easy to construct. Moreover, such a polyhedron should have as few describing hyperplanes as possible in order to reduce the storage requirements (see Remark 2.5.1(c)). And, a third criterion is, that the intersection of this polyhedron with the eliminable ball should have the biggest possible volume. Unfortunately, these criteria are conflictive. For example, the less hyperplanes we use to describe the polyhedra the less volume of the resulting intersection sets we can expect.

The first type of polyhedra, which we present in this section, is a hypercube. These sets are really easy to construct and are a relative good choice with respect to the third criterion. However, they do not pay so much attention to our second criterion. Therefore, we propose furthermore two possible polyhedra which base on d-simplices and are described by only $d+1$ respectively d hyperplanes, instead of the $2d$ hyperplanes in the case of the hypercubes. The first simplex, which we propose in Subsection 2.6.2, is also easy to construct. In order to obtain a better set with respect to the volume criterion we modify this simplex in Subsection 2.6.3. However, the construction of this modified d-simplex will need more effort.

2.6.1. Hypercubes. Using the fact that, for each $z \in \mathbb{R}^d$, there holds

$$\|z\|_2^2 \;=\; \sum_{i=1}^{d} |z_i|^2 \;\leq\; \sum_{i=1}^{d} \|z\|_\infty^2 \;=\; d\|z\|_\infty^2 \,, \tag{2.6.1}$$

we immediately see, that the ℓ_∞-norm ball centered at z^k with radius $\frac{\epsilon(z^k)}{\sqrt{d}}$ is contained in the Euclidean norm ball with radius $\epsilon(z^k)$. This ℓ_∞-norm ball is a hypercube centered at z^k with edge-length $2\frac{\epsilon(z^k)}{\sqrt{d}}$ and can be described by

$$R^k \;=\; \{z \in \mathbb{R}^d : (q_i^k)^T z \;\leq\; c_i^k \,,\, i = 1, \ldots, 2d\} \tag{2.6.2}$$

where, for $i = 1, \ldots, d$,

$$(q_i^k)^T z = z_i \qquad \text{and} \qquad c_i^k = z_i^k + \frac{\epsilon(z^k)}{\sqrt{d}} \,,$$

and, for $i = d+1, \ldots, 2d$,

$$(q_i^k)^T z = -z_{i-d} \qquad \text{and} \qquad c_i^k = -z_{i-d}^k + \frac{\epsilon(z^k)}{\sqrt{d}} \,.$$

The hypercubes R^k ($k \in \mathbb{N}$) fulfill Property (PR1) (see (2.6.1)) and in view of the definition of the hyperplanes $H(q_i^k, c_i^k)$ ($i = 1, \ldots, 2d$; $k \in \mathbb{N}$) we know

$$d(z^k, H(q_i^k, c_i^k)) \;=\; \frac{|(q_i^k)^T z^k - c_i^k|}{\|q_i^k\|_2} \;=\; \frac{1}{\sqrt{d}}\epsilon(z^k) \,.$$

Choosing $l = 2d$ and $\rho = \frac{1}{\sqrt{d}}$ in the initialization of Algorithm 2.2 the hypercube R^k is an appropriate choice for the necessary polyhedron Q^k ($k \in \mathbb{N}$). If we apply Algorithm 2.2 using these hypercubes for solving our example problem, then the first iteration of this approach looks like it is described on page 43 (see, in particular, Figure 2.4).

REMARK 2.6.1. If the hypercubes R^k are used in Algorithm 2.2 for Q^k ($k \in \mathbb{N}$), the number of inequalities describing a polytope $P \in \mathcal{P}$ can be bounded by $m + 2d$. Note that the normals q_i^k ($i = 1, \ldots, 2d$; $k \in \mathbb{N}$) of the constraints describing R^k do not depend on the iteration counter, and, thus, only the right-hand sides c_i^k ($i = 1, \ldots, 2d$; $k \in \mathbb{N}$) of the constraints change.

The hypercubes R^k ($k \in \mathbb{N}$) are really easy to construct and fulfill thus the postulated first criterion. However, the number $2d$ of generated new polytopes in each iteration of the algorithm is already rather large. In order to reduce this number we develop now an inner approximation polytope for the ball B_{z^k}, which can be described by $d+1$ hyperplanes. This choice is hence better – regarding our second criterion.

2.6.2. Regular d-Simplices. A d-simplex is the set among all d-dimensional polytopes, which can be described by the least number of linear constraints. We present now a d-simplex contained in the Euclidean norm ball B_{z^k}, whose vertices lie on the boundary of this ball. It is known that among all d-simplices contained in such a ball the so-called **regular** simplices, i.e., the simplices where the distance between each pair of vertices is equal, are the largest ones with respect to the volume (see [SLE69] for a proof). In view of the third criterion we choose, therefore, a regular d-simplex contained in B_{z^k}.

In order to simplify the presentation we start with the description of a regular d-simplex centered at the origin and with vertices on the boundary of the unit ball $B = \{z \in \mathbb{R}^d : \|z\|_2 \leq 1\}$. This simplex can later be easily transformed to the required d-simplex lying in the relevant ball B_{z^k}.

Assume, at first, that a regular d-simplex $S = [v_0, \ldots, v_d]$ centered at the origin and with all its vertices on the boundary of B is given. Then it is known from the literature that the edge-length of S, i.e., the Euclidean distance between each pair of vertices, is given by

$$\|v_i - v_j\|_2 \;=\; \sqrt{\frac{2(d+1)}{d}}\;,\; i,j \in \{0,\ldots,d\} \text{ with } i \neq j \tag{2.6.3}$$

(see, e.g., [SOM29, GKL95]). Moreover, it is elementary to show that $0 = \frac{1}{d+1}\sum_{i=0}^{d} v_i$, i.e., the origin is the barycenter of S, and that the radius of the largest Euclidean ball, which can be inscribed into S, is

$$r \;=\; \frac{1}{d}\,. \tag{2.6.4}$$

The number r is also the distance of each facet of S to the origin. Furthermore, we can use the fact that, for each $j \in \{0,\ldots,d\}$, the vertex v_j is orthogonal to the facet $S_j = [v_0,\ldots,v_{j-1},v_{j+1},\ldots,v_d]$ of S, and hence the hyperplanes H_{S_j} generated by S_j can be described by

$$H_{S_j} \;=\; \{z \in \mathbb{R}^d : v_j^T(v_i - z) = 0\} \tag{2.6.5}$$

with an arbitrary, but fixed index $i \in \{0,\ldots,d\} \setminus \{j\}$.

These are known results about the properties of a regular d-simplex centered at the origin and with all its vertices on the boundary of the unit ball B. To the author's knowledge there is, unfortunately, no explicit construction of such a simplex in the literature – except of [HR98]. In order to derive an implementable algorithm we need an explicit formulation of the hyperplanes describing such a simplex and, thus, in view of (2.6.5) we need an explicit formulation of its vertices. This will be done in the following. For reasons which will become evident later in this section we construct a regular r-simplex with $r \in \mathbb{N}$. Set

$$\boxed{\begin{aligned} v_0 &= \sqrt{a_0} e_r \,, \\ v_i &= \sqrt{a_{2i}} e_{r-i} - \sum_{j=1}^{i} \sqrt{a_{2j-1}} e_{r-(j-1)} \,, \; i = 1, \dots, r-1 \,, \\ v_r &= -\sqrt{a_{2(r-1)}} e_1 - \sum_{j=1}^{r-1} \sqrt{a_{2j-1}} e_{r-(j-1)} \,, \end{aligned}} \tag{2.6.6}$$

where

$$\boxed{\begin{aligned} a_0 &= 1 \,, \\ a_i &= \begin{cases} a_{i-1} / \left(r - \frac{i-1}{2} \right)^2 & \text{, if } i \text{ odd} \\ a_{i-2} - a_{i-1} & \text{, if } i \text{ even} \end{cases} \quad , \, i = 1, \dots, 2(r-1) \,, \end{aligned}} \tag{2.6.7}$$

and $e_i \in \mathbb{R}^r$ is the i-th unit vector. The r-simplex $S = [v_0, \dots, v_r]$, which is generated by these vertices, is a regular simplex with the edge-length (2.6.3), and all its vertices belong to the boundary of the unit ball $B \subset \mathbb{R}^r$. This will be the result of Theorem 2.6.2. At first, however, a technical lemma is needed in order to establish this theorem.

LEMMA 2.6.1. *Let a_i ($i \in \{0, \dots, 2(r-1)\}$) be defined as in (2.6.7). Then, for each $i = 1, \dots, r-1$, there holds*

$$\frac{r-i+1}{r-i} a_{2i} = \frac{r+1}{r} \,. \tag{2.6.8}$$

PROOF: We prove this result by an induction with respect to i. The assertion is obviously correct for $i = 0$. Assume that it holds for $i = j-1$ with $j \geq 1$. Then it follows by definition of a_l ($l \in \{0, \ldots, 2(r-1)\}$)

$$\begin{aligned}\frac{r-j+1}{r-j}\, a_{2j} &= \frac{r-j+1}{r-j}\,(a_{2j-2} - a_{2j-1}) \\ &= \frac{r-j+1}{r-j}\left(a_{2j-2} - \frac{a_{2j-2}}{(r-j+1)^2}\right) \\ &= \frac{r-j+1}{r-j}\,\frac{(r-j+1)^2-1}{(r-j+1)^2}\, a_{2j-2} \\ &= \frac{r-j+2}{r-j+1}\, a_{2j-2} \;=\; \frac{r+1}{r}\,,\end{aligned}$$

which is the required result for $i = j$. ■

With the technical result of Lemma 2.6.1 the postulated properties of the simplex generated by the vertices defined in (2.6.6) can now be shown.

THEOREM 2.6.2. *Let $S = [v_0, \ldots, v_r]$ be the r-simplex with the vertices v_i ($i = 0, \ldots, r$) constructed as in (2.6.6). Then the following assertions are true.*

(i) *Each vertex of S belongs to the boundary of the r-dimensional unit ball $B = \{z \in \mathbb{R}^r : \|z\|_2 \leq 1\}$, i.e., for each $i \in \{0, \ldots, r\}$, there holds*

$$\|v_i\|_2 \;=\; 1\,.$$

(ii) *The distance between each pair of vertices is equal. Moreover, for each $i, j \in \{0, \ldots, r\}$ with $i \neq j$, there holds*

$$\|v_i - v_j\|_2 \;=\; \sqrt{\tfrac{2(r+1)}{r}}$$

(compare with (2.6.3)).

PROOF: In view of the definition of a_l for $l \in \{0, \ldots, 2(r-1)\}$ even we obtain, for each $i \in \{0, \ldots, r-1\}$,

$$a_{2i} \;=\; 1 - \sum_{j=1}^{i} a_{2j-1}\,. \tag{2.6.9}$$

Hence, for each $i \in \{0, \ldots, r-1\}$, it follows

$$\|v_i\|_2^2 \;=\; a_{2i} + \sum_{j=1}^{i} a_{2j-1} \;=\; 1\,.$$

Using the fact that v_r and v_{r-1} have by definition the same distance to the origin, assertion (i) is proven.

Lemma 2.6.1 yields

$$\|v_{r-1} - v_r\|_2^2 \;=\; 4a_{2(r-1)} \;=\; 2\,\frac{r-(r-1)+1}{r-(r-1)}\,a_{2(r-1)} \;=\; \frac{2(r+1)}{r}\;,$$

and by using additionally (2.6.9) we obtain, for $i, j \in \{0, \dots, r\}$ with $i < j$ and $i < r-1$,

$$\begin{aligned} \|v_i - v_j\|_2^2 \;&=\; a_{2j} + \sum_{l=i+2}^{j} a_{2l-1} + \left(\sqrt{a_{2i}} + \sqrt{a_{2i+1}}\right)^2 \\ &=\; 1 - \sum_{l=1}^{j} a_{2l-1} + \sum_{l=i+2}^{j} a_{2l-1} + \left(\sqrt{a_{2i}} + \sqrt{a_{2i+1}}\right)^2 \\ &=\; 1 - \sum_{l=1}^{i+1} a_{2l-1} + a_{2i} + a_{2i+1} + 2\sqrt{a_{2i}}\sqrt{a_{2i+1}} \\ &=\; 2a_{2i} + 2\sqrt{a_{2i}}\sqrt{\tfrac{a_{2i}}{(r-i)^2}} \;=\; 2a_{2i} + 2a_{2i}\frac{1}{r-i} \\ &=\; 2\,\frac{r-i+1}{r-i}a_{2i} \;=\; \frac{2(r+1)}{r}\;, \end{aligned}$$

which shows assertion (ii) and completes the proof. ■

As a direct consequence of the previous theorem, we obtain that the inner product of each pair of vertices of the simplex $S = [v_0, \dots, v_r]$ is equal $-\frac{1}{r}$.

COROLLARY 2.6.3. *Under the assumptions of Theorem 2.6.2 there holds, for each* $i, j \in \{0, \dots, r\}$ *with* $i \neq j$*,*

$$v_i^T v_j \;=\; -\tfrac{1}{r}\;. \tag{2.6.10}$$

PROOF: From result (ii) of Theorem 2.6.2 we know that, for each $i, j \in \{0, \dots, r\}$ with $i \neq j$, there holds

$$2\,\tfrac{r+1}{r} \;=\; \|v_i - v_j\|_2^2 \;=\; \|v_i\|_2^2 + \|v_j\|_2^2 - 2v_i^T v_j\;.$$

Using assertion (i) of this theorem we obtain

$$2\,\tfrac{r+1}{r} \;=\; 2 - 2v_i^T v_j\;,$$

which implies (2.6.10). ■

In view of the previous results the construction (2.6.6) with $r = d$ yields the needed explicit formulation of a regular d-simplex $S = [v_0, \dots, v_d]$ centered at the origin, whose vertices lie on the boundary of the unit ball. Assume now that we are again in the situation of Algorithm 2.2 and that a point $z^k \in P = \{z \in \mathbb{R}^d : Az \leq b\}$ is given satisfying $U(z^k) \notin \mathcal{U}_n$, i.e.,

$$\epsilon(z^k) \;=\; \max\{\lambda_{n-1}(U(z^k)), -\lambda_1(U(z^k))\} \;>\; 0\,.$$

It could be verified by straightforward calculations that the polyhedron

$$S^k \;:=\; \{z \in \mathbb{R}^d : -v_i^T z \;\leq\; \tfrac{\epsilon(z^k)}{d} - v_i^T z^k\,,\, i = 0, \ldots, d\} \tag{2.6.11}$$

with v_i $(i = 0, \ldots, d)$ defined as in (2.6.6) is a regular d-simplex centered at z^k (compare with (2.6.5)). The vertices of S^k are $\epsilon(z^k)v_i + z^k$ $(i = 0, \ldots, d)$, which lie on the boundary of the ball B_{z^k}. For the Euclidean distance of the point z^k to the hyperplanes describing S^k we obtain regarding (2.5.3), for each $i \in \{0, \ldots, d\}$,

$$d(z^k, H(-v_i, \tfrac{\epsilon(z^k)}{d} - v_i^T z^k)) \;=\; \frac{\epsilon(z^k)}{d} \tag{2.6.12}$$

(compare with (2.6.4)). Thus, choosing $l = d + 1$ and $\rho = \frac{1}{d}$ in the initialization of Algorithm 2.2, the regular d-simplices S^k are also an appropriate choice for the polyhedra Q^k $(k \in \mathbb{N})$ needed in this approach.

Remark 2.6.2. If the regular d-simplices defined in (2.6.11) are used in Algorithm 2.2 for Q^k $(k \in \mathbb{N})$, the number of inequalities describing a polytope $P \in \mathcal{P}$ can be bounded by $m + d + 1$. As in the case of the hypercubes (see Remark 2.6.1), the normals $q_i^k = -v_i$ $(i = 0, \ldots, d;\ k \in \mathbb{N})$ do not depend on the iteration counter k.

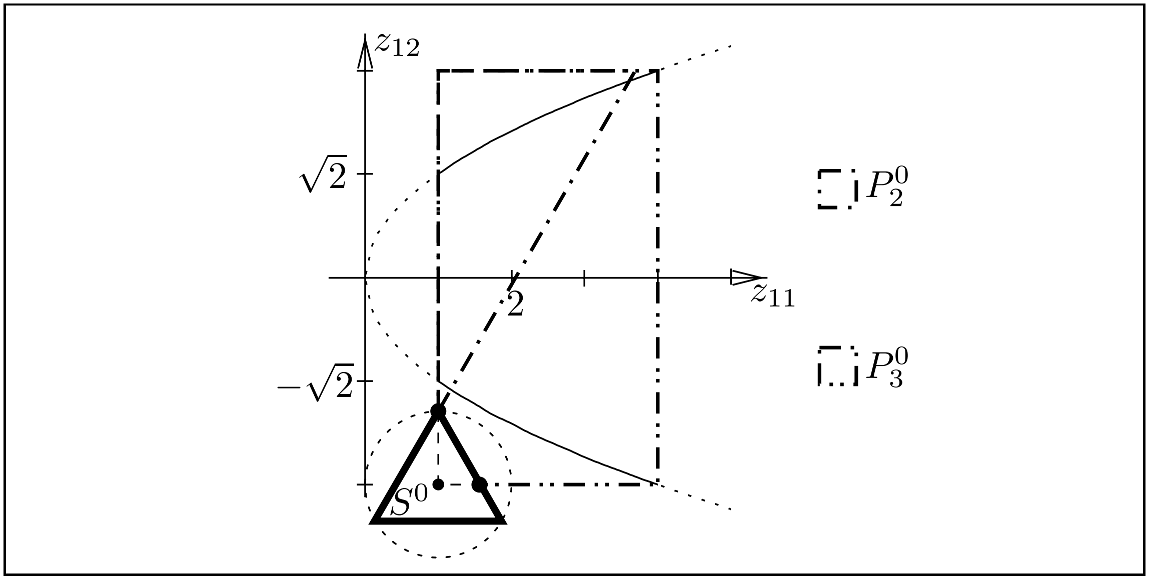
Example. If we choose in Algorithm 2.2 this regular d-simplex for subdividing the feasible region of Problem (UPE), then we obtain in the first iteration the following polytopes (see also page 43).

$$\begin{aligned}
P_1^0 \;&=\; P^0 \cap \{z \in \mathbb{R}^2 : -z_{12} \;\geq\; \tfrac{1}{2} + 2\sqrt{2}\} \;=\; \emptyset \\
P_2^0 \;&=\; P^0 \cap \{z \in \mathbb{R}^2 : \tfrac{1}{2}(-\sqrt{3}z_{11} + z_{12}) \geq \tfrac{1}{2}(1 - \sqrt{3} - 2\sqrt{2})\} \\
P_3^0 \;&=\; P^0 \cap \{z \in \mathbb{R}^2 : \tfrac{1}{2}(\sqrt{3}z_{11} + z_{12}) \geq \tfrac{1}{2}(1 + \sqrt{3} - 2\sqrt{2})\}
\end{aligned}$$

This situation is illustrated in Figure 2.5. The new solutions are given by $z_2^0 = (1, 1 - 2\sqrt{2})^T$ with optimal value $\frac{1}{\sqrt{2}} - 1$ and $z_3^0 = (1 + \frac{1}{\sqrt{3}}, -2\sqrt{2})^T$ with value $\frac{1}{\sqrt{3}} - 1$. The polytope P^1 for iteration 1 is hence P_3^0 with $\mu^1 = -0.4226$.

The presented d-simplex S^k is an inner approximation polytope for the whole ball B_{z^k}, which can be cut out of the relevant feasible set P^k. In Section 2.5 we

FIGURE 2.5. First iteration of Algorithm 2.2 with a regular simplex applied for (UPE)



only require that the intersection of the polyhedron Q^k with P^k is an inner approximation of the intersection of P^k with the ball B_{z^k}. Therefore, by constructing a set based on another d-simplex, which contains a bigger part of $P^k \cap B_{z^k}$, i.e., a bigger part of the set which can really be eliminated, we obtain – taking our third criterion for appropriate polyhedra Q^k ($k \in \mathbb{N}$) into account – a better choice. Note, in particular, that all points of P^k belonging to B_{z^k} must lie in a half-ball of B_{z^k}.

2.6.3. A Better Polyhedron Based on a Modified d-Simplex. The regular d-simplex S^k defined in (2.6.11) does not depend on the current polytope P^k. The construction of these sets only use the point z^k and the corresponding value $\epsilon(z^k)$. In the following we present a polyhedron derived from a d-simplex, which also recognize the bearing of the polytope P^k with respect to the point z^k. For this aim we need, as in Ramana's approach (see Section 2.3), that z^k is a vertex of the current polytope P^k. This is always satisfied, if we use the Simplex-Algorithm for solving the linear subproblems in Algorithm 2.2.

Let z^k be a vertex of P^k ($k \in \mathbb{N}$) and let $B^k z \leq r^k$, with $B^k = (b_1^k, \dots , b_d^k)^T$ regular $d \times d$ matrix, be the nonsingular basic system corresponding to z^k (compare with Section 2.3, in particular page 29). Let, furthermore,

$$C^k \;=\; \{z \in \mathbb{R}^d : B^k z \leq r^k\}$$

be the cone defined by this system. Each of the d extremal directions $w_i^k \in \mathbb{R}^d$ ($i = 1, \dots, d$) of C^k is a nontrivial solution of the system

$$\begin{aligned} (b_j^k)^T w_i^k &= 0 \qquad j = 1, \dots, i-1, i+1, \dots, d \\ (b_i^k)^T w_i^k &\leq 0\,. \end{aligned}$$

Let, for $i \in \{1, \dots, d\}$, the vector $\bar{w}_i^k \in \mathbb{R}^d$ denote the intersection point of the ray

$$\{z \in \mathbb{R}^d : z = z^k + \beta w_i^k\,,\ \beta \geq 0\}$$

with the boundary of the ball B_{z^k}, i.e.,

$$\bar{w}_i^k \;=\; z^k + \epsilon(z^k) \frac{w_i^k}{\|w_i^k\|_2}$$

(see Figure 2.6). Let, furthermore,

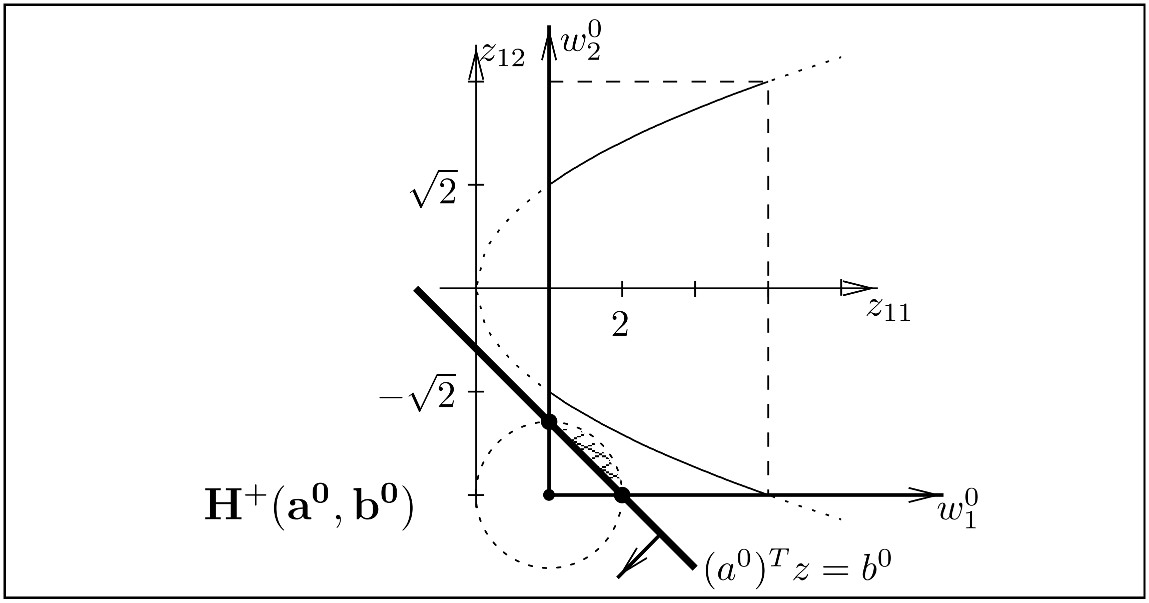
$$H^k \;=\; H(a^k, b^k) \;=\; \{z \in \mathbb{R}^d : (a^k)^T z = b^k\} \tag{2.6.13}$$

with $a^k \in \mathbb{R}^d$, $b^k \in \mathbb{R}$ be the uniquely determined hyperplane containing each of these intersection points $\bar{w}_i^k$ ($i = 1, \dots, d$) and satisfying $(a^k)^T z^k > b^k$. Since P^k is a subset of C^k and in view of the quadratic cut (2.4.6) we know that no feasible point of (UP) belongs to the set

$$H^+(a^k, b^k) \;=\; \{z \in \mathbb{R}^d : (a^k)^T z \geq b^k\}$$

(see again Figure 2.6). This means that the linear constraint

FIGURE 2.6. The hyperplane H^0 in the case of Problem (UPE)



$$(a^k)^T z \ \leq \ b^k \tag{2.6.14}$$

is a valid cut for (UP).

REMARK 2.6.3. The cut (2.6.14) could be used in order to derive an outer approximation method for solving (UP), as we did in Section 2.3 with the cuts introduced by Ramana (see Algorithm 2.1). However, since the definition of a^k ($k \in \mathbb{N}$) depends on the current nonsingular basic system $B^k z \leq r^k$ corresponding to z^k, such an algorithm can – similar to Ramana's original approach – fail to converge. Nevertheless, as we will see in Section 2.7, each known valid cut can be used for accelerating the convergence of our solution scheme for (UP).

If we take a d-simplex $\bar{S}^k$, which is the convex hull of the intersection point $\bar{a}^k$ of the ray $\{z \in \mathbb{R}^d : z = z^k - \beta a^k \ , \ \beta \geq 0\}$ with the boundary of B_{z^k} and a regular $(d-1)$-simplex contained in the intersection of H^k with the ball B_{z^k}, then we obtain $\bar{S}^k \subset B_{z^k} \cap \{z \in \mathbb{R}^d : (a^k)^T z \leq b^k\}$ ($\bar{S}^k$ is contained in the shaded region in Figure 2.6). The polyhedron $\bar{Q}^k$ described by the d hyperplanes, which are induced by just the facets of $\bar{S}^k$ containing $\bar{a}^k$, obviously fulfills Property (PR1). And, moreover, we can expect that the Euclidean distance of the hyperplanes describing $\bar{Q}^k$ to the point z^k is bigger than the distance of the facets of the regular d-simplex introduced in the previous subsection (see (2.6.11)). The two possible choices of Q^k in Algorithm 2.2 proposed until now are fully contained in the ball B_{z^k}. The polyhedron $\bar{Q}^k$, which we present below, does not have this property. Only the intersection of $\bar{Q}^k$ with the current polytope P^k will be contained in this ball. Therefore, we can hope, that a bigger part of P^k is cut out of this set by applying the polyhedron $\bar{Q}^k$ instead of S^k or maybe even instead of R^k.

As mentioned before, the construction of the new polyhedron $\bar{Q}^k$ is based on a d-simplex. Let us first describe the construction of this d-simplex. In order to simplify the presentation we assume again that B_{z^k} is the unit ball B and that H^k is a hyperplane parallel to $\{z \in \mathbb{R}^d : z_d = 0\}$, i.e., $H^k = H = \{z \in \mathbb{R}^d : -e_d^T z = -\delta\}$, where $\delta \in [0, 1)$ denotes the Euclidean distance of H to the origin. After the derivation of the required d-simplex for this situation we describe, how this "*standard*" simplex can be transformed to the general case of B_{z^k} and H^k defined as in (2.6.13).

The intersection of H with the unit ball is a (d-1)-dimensional sphere with radius $\bar{\epsilon} = \sqrt{1 - \delta^2}$ and centered at δe_d. Let $v_0, \ldots, v_{d-1}$ be the vertices of a regular $(d-1)$-simplex constructed as in (2.6.6). Assume that these vertices are imbedded in the space $\mathbb{R}^d$ by adding one dimension. Set now, for $i = 0, \dots, d-1$,

$$\bar{v}_i \ := \ \bar{\epsilon} v_i + \delta e_d$$

and

$$\bar{v}_d \ := \ e_d \, .$$

It follows immediately that the vertices $\bar{v}_i$ ($i = 1, \dots d-1$) are contained in the hyperplane H. From Theorem 2.6.2 and the construction of the points $\bar{v}_i$ ($i = 0, \dots d$) we see that

$$\begin{aligned} \|\bar{v}_i\|_2 \ &= \ 1 && , i \in \{1, \dots, d\} \, , \\ \|\bar{v}_i - \bar{v}_j\|_2 \ &= \ \bar{\epsilon} \sqrt{\tfrac{2d}{d-1}} && , i, j \in \{0, \dots, d-1\} \text{ with } i \neq j \, , \\ \|\bar{v}_i - \bar{v}_d\|_2 \ &= \ \sqrt{1 - \delta^2 + (1-\delta)^2} && , i \in \{0, \dots, d-1\} \, . \end{aligned}$$

In order to use the simplex $\bar{S} = [\bar{v}_0, \dots, \bar{v}_d]$ for the construction of an appropriate polyhedron Q^k ($k \in \mathbb{N}$) for Algorithm 2.2, we have to derive, for each $i \in \{0, \dots, d-1\}$, a representation of the hyperplanes $H_{\bar{S}_i}$ generated by the facets

$$\bar{S}_i \ = \ [\bar{v}_0, \dots, \bar{v}_{i-1}, \bar{v}_{i+1}, \dots, \bar{v}_d]$$

of $\bar{S}$. Note that H is the hyperplane induced by the facet $\bar{S}_d$. The following lemma delivers this representation.

LEMMA 2.6.4. *Let $v_0, \dots, v_{d-1} \in \mathbb{R}^d$ be the vertices of a regular ($d-1$)-simplex defined as in (2.6.6). Set, for each $i \in \{0, \dots, d-1\}$,*

$$\hat{v}_i \ := \ v_i - \frac{\tau}{d-1} \, e_d$$

with

$$\tau \ = \ \frac{\sqrt{1-\delta^2}}{1-\delta} \ \geq \ 1 \, . \tag{2.6.15}$$

Then, for each $i \in \{0, \dots, d-1\}$, the hyperplane $H_{\bar{S}_i}$ generated by the facet $\bar{S}_i$ of the d-simplex $\bar{S}$ can be described by

$$H_{\bar{S}_i} \ = \ \{z \in \mathbb{R}^d : \hat{v}_i^T z = \hat{v}_i^T \bar{v}_d\} \, . \tag{2.6.16}$$

PROOF: Since, for each $i \in \{0, \dots, d-1\}$, we know

$$H_{\bar{S}_i} \ = \ \{z \in \mathbb{R}^d : z = \bar{v}_d + \sum_{j=0, j \neq i}^{d-1} \gamma_j (\bar{v}_j - \bar{v}_d) \, , \, \gamma_j \in \mathbb{R}_+\} \, ,$$

it suffices to show that $\hat{v}_i$ is orthogonal to each direction $(\bar{v}_j - \bar{v}_d)$ ($j = 0, \ldots, d-1$; $j \neq i$) of $H_{\bar{S}_i}$ and, thus, orthogonal to $H_{\bar{S}_i}$ itself. I.e., we have to prove, for each $j \in \{0, \ldots, d-1\} \setminus \{i\}$,

$$\hat{v}_i^T(\bar{v}_j - \bar{v}_d) \; = \; 0 \, . \tag{2.6.17}$$

Choose an arbitrary, but fixed index $j \in \{0, \ldots, d-1\} \setminus \{i\}$. Applying Corollary 2.6.3 and the fact that, for each $l \in \{0, \ldots, d-1\}$, the d-th component of v_l is zero we obtain

$$\begin{aligned} \hat{v}_i^T(\bar{v}_j - \bar{v}_d) \; &= \; \left(v_i - \tfrac{\tau}{d-1} e_d\right)^T (\bar{\epsilon} v_j + \delta e_d - e_d) \\ &= \; \bar{\epsilon} \underbrace{v_i^T v_j}_{=-\frac{1}{d-1}} + (\delta - 1) \underbrace{v_i^T e_d}_{=0} - \tfrac{\bar{\epsilon}\tau}{d-1} \underbrace{e_d^T v_j}_{=0} + \tfrac{\tau(1-\delta)}{d-1} \underbrace{e_d^T e_d}_{=1} \\ &= \; -\tfrac{\bar{\epsilon}}{d-1} + \tfrac{\tau(1-\delta)}{d-1} \; = \; 0 \, , \end{aligned}$$

which shows (2.6.17) and finishes the proof. ■

The polyhedron, which we derive from the simplex $\bar{S}$, will be determined by the d hyperplanes described in the last lemma. By construction we know that this polyhedron fulfills Property (PR1). In order to guarantee that this polyhedron also satisfies Property (PR2) we need the Euclidean distance of the hyperplanes $H_{\bar{S}_i}$ ($i = 0, \ldots, d-1$) to the point z^k, i.e., in the considered situation to the origin. Moreover, we have postulated that the polyhedron, which we develop in this subsection, cuts a bigger part out of the unit ball B than the regular d-simplex S derived in Subsection 2.6.2. This would be satisfied, if the distance of the hyperplanes $H_{\bar{S}_i}$ ($i = 0, \ldots, d-1$) is bigger than $\frac{1}{d}$ (compare with (2.6.4)).

THEOREM 2.6.5. *Let $H_{\bar{S}_i}$ ($i = 0, \ldots, d-1$) be the hyperplanes defined in Lemma 2.6.4. Then, for each $i \in \{0, \ldots, d-1\}$, the Euclidean distance $d(0, H_{\bar{S}_i})$ of these hyperplanes to the origin is*

$$d(0, H_{\bar{S}_i}) \; = \; \frac{\tau}{\sqrt{(d-1)^2 + \tau^2}} \; > \; \frac{1}{d} \tag{2.6.18}$$

with τ given as in (2.6.15).

PROOF: From

$$\|\hat{v}_i\|_2^2 \; = \; 1 + \frac{\tau^2}{(d-1)^2} \; = \; \frac{(d-1)^2 + \tau^2}{(d-1)^2}$$

we obtain by using (2.5.3), for each $i \in \{0, \dots, d-1\}$,

$$d(0, H_{\bar{S}_i}) \;=\; \frac{|\hat{v}_i^T \bar{v}_d|}{\|\hat{v}_i\|_2} = \frac{\tau}{d-1} \frac{d-1}{\sqrt{(d-1)^2 + \tau^2}} \;=\; \frac{\tau}{\sqrt{(d-1)^2 + \tau^2}} \,.$$

The function $\varrho : \mathrm{I\!R} \to \mathrm{I\!R}$, $\varrho(\tau) = \frac{\tau}{\sqrt{(d-1)^2+\tau^2}}$ is monotonously decreasing in $\mathrm{I\!R}_+$ and, additionally, there holds $\varrho(1) > \frac{1}{d}$. Therefore, it follows that $\varrho(\tau)$ is bigger than $\frac{1}{d}$ for each $\tau \geq 1$, which shows in view of (2.6.15) the right-hand side of Relation (2.6.18). ■

In view of the previous result we know that the polyhedron

$$\bar{Q} \;=\; \{z \in \mathrm{I\!R}^d : -\hat{v}_i^T z \;\leq\; \tfrac{\tau}{d-1}\}$$

cuts a bigger part out of the unit ball B than the regular d-simplex introduced in the previous subsection. Note that $\hat{v}_i^T \bar{v}_d$ coincides with $-\frac{\tau}{d-1}$ $(i = 0, \dots, d-1)$. The construction of $\bar{Q}$ and $\bar{S}$, respectively, depends on the hyperplane H. Therefore, we cannot transform $\bar{S}$ to the interesting situation of B_{z^k} and H^k by simply multiplying the relevant values with $\epsilon(z^k)$, as it was the case for the previous two choices of the polyhedron Q^k. We will need more effort.

Let $\{y_1^k, \dots, y_{d-1}^k\}$ be an orthonormal basis of the linear subspace $H^k - \{z^k\}$. Such a basis could be developed by applying the Gram-Schmidt method or another orthonormalization procedure (see, again, [GVL89]) to the set $\{\bar{w}_i^k - \bar{w}_1^k,$ $i = 2, \dots, d\}$, which forms by construction a basis of $H^k - \{z^k\}$ (see page 54). Let

$$A^k \;=\; (y_1^k, \dots, y_{d-1}^k, -a^k)$$

be the $d \times d$ matrix with the columns $y_1^k, \dots, y_{d-1}^k$ and $-a^k$. If a^k is normalized, it is obvious that this matrix is orthogonal, i.e., there holds $(A^k)^T A^k = E$, where E denotes the d-dimensional identity matrix. In view of this property we see that the transformation

$$T^k : \mathrm{I\!R}^d \to \mathrm{I\!R}^d \quad :\Leftrightarrow \quad T^k(z) \;=\; \epsilon(z^k) A^k z + z^k$$

yields, for any z, $\hat{z} \in \mathrm{I\!R}^d$,

$$\|T^k(z) - z^k\|_2 \;=\; \epsilon(z^k)\|z\|_2 \quad \text{and} \quad \|T^k(z) - T^k(\hat{z})\|_2 \;=\; \epsilon(z^k)\|z - \hat{z}\|_2 \,.$$

The affine function T^k maps the unit ball B and the hyperplane $H = \{z \in \mathrm{I\!R}^d : -e_d^T z = -\delta\}$ to the current ball B_{z^k} and the current hyperplane H^k. Applying the inverse function $(T^k)^{-1}$ we can hence transform the situation of the current iteration k to the just examined *standard* situation. In order to construct the simplex $\bar{S}$ in the *standard* situation we need the Euclidean distance of the resulting hyperplane

$$H \;=\; \{z \in \mathrm{I\!R}^d : T^k(z) \in H^k\} \;=\; \{z \in \mathrm{I\!R}^d : (a^k)^T T^k(z) = b^k\}$$

to the origin. This is given by

$$\delta^k \;=\; \frac{|(a^k)^T z^k - b^k|}{\epsilon(z^k)} \tag{2.6.19}$$

(compare with (2.5.3) and note that $(a^k)^T A^k = -e_d$). Transforming the simplex $\bar{S} = [\bar{v}_0, \dots, \bar{v}_d]$ of the *standard* situation to the current situation in iteration k we obtain with

$$\bar{S}^k \;=\; [T^k(\bar{v}_0), \dots, T^k(\bar{v}_d)]$$

a d-simplex contained in the set

$$B_{z^k} \cap \{z \in \mathrm{I\!R}^d : (a^k)^T z \le b^k\}\ .$$

It can be verified by straightforward calculations that the hyperplanes induced by the facets $\bar{S}^k_i$ ($i = 0, \dots, d-1$) of the simplex $\bar{S}^k$ containing the point $\bar{a}^k = z^k - \epsilon(z^k) a^k$ are given by

$$H_{\bar{S}^k_i} \;=\; \left\{ z \in \mathrm{I\!R}^d : \left(A^k \left(\tfrac{\tau^k}{d-1} e_d - v_i \right) \right)^T (z - z^k) \;=\; \tfrac{\epsilon(z^k)\tau^k}{d-1} \right\}$$

with $\tau^k = \frac{\sqrt{1-(\delta^k)^2}}{1-\delta^k}$ and with $v_0, \dots, v_{d-1}$ defined as in (2.6.6) for $r = d-1$. Moreover, it follows that, for each $i \in \{0, \dots, d-1\}$, the Euclidean distance of these hyperplanes to the point z^k is

$$\begin{aligned} d(z^k, H_{\bar{S}^k_i}) \;&=\; \epsilon(z^k) \frac{\tau^k}{\sqrt{(d-1)^2 + (\tau^k)^2}} \qquad (2.6.20) \\ &\ge\; \epsilon(z^k) \frac{1}{\sqrt{(d-1)^2 + 1}}\ . \end{aligned}$$

Choosing $l = d$ and $\rho = \frac{1}{\sqrt{(d-1)^2+1}}$ in the initialization of Algorithm 2.2 the polyhedra

$$\begin{aligned} \bar{Q}^k \;=\; \{z \in \mathrm{I\!R}^d : \left(A^k \left(\tfrac{\tau^k}{d-1} e_d - v_i \right) \right)^T (z - z^k) \;\le\; \tfrac{\epsilon(z^k)\tau^k}{d-1}\ , \\ i = 0, \dots, d-1\} \qquad (2.6.21) \end{aligned}$$

are the third possible choice for the sets Q^k ($k \in \mathbb{N}$) needed in this approach.

Example. Consider once again Problem (UPE). The nonsingular basic system corresponding to $z^0 = (1, -2\sqrt{2})^T$ is given by $B^0 = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$ and $r^0 = \binom{-1}{2\sqrt{2}}$ (see page 31). For the hyperplane H^0 we obtain

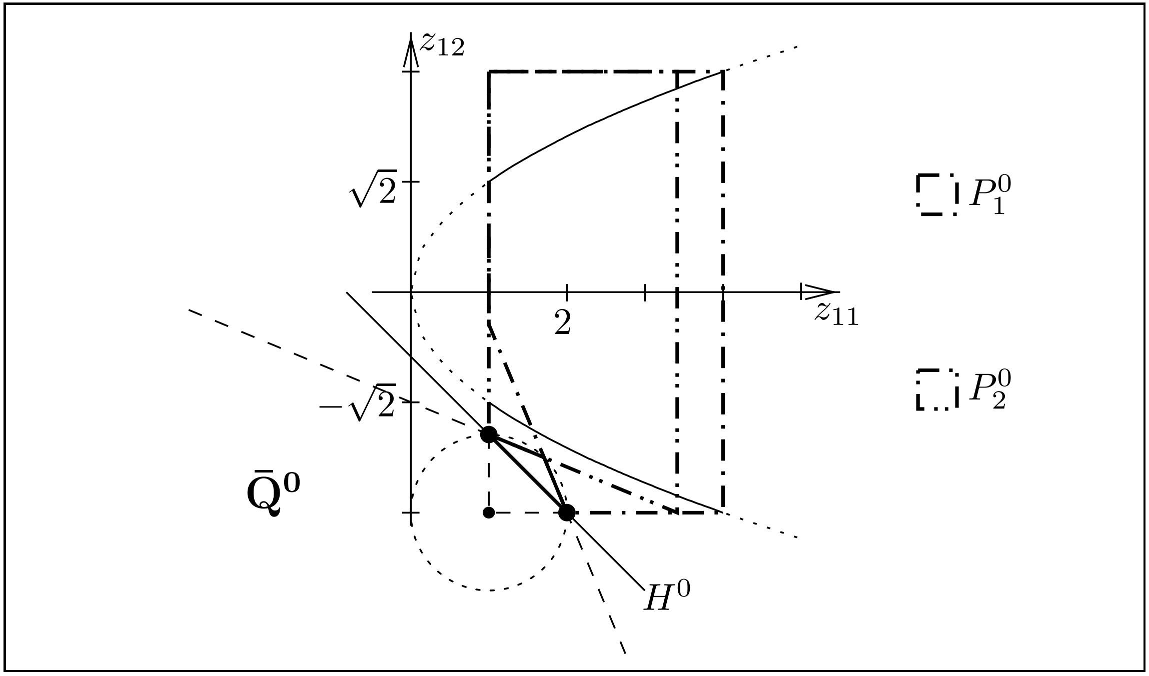
$$H^0 \ = \ \{z \in \mathbb{R}^2 : -\tfrac{1}{\sqrt{2}} z_{11} - \tfrac{1}{\sqrt{2}} z_{12} \ = \ 2 - \sqrt{2}\} \, .$$

The point $\bar{a}^0$ is $(\frac{1}{\sqrt{2}} + 1, -\frac{3}{\sqrt{2}})^T$, and the distance d^0 of H^0 to the point z^0 is given by $\frac{1}{\sqrt{2}}$. Thus, we have $\tau^0 = 1 + \sqrt{2}$ and using the matrix $A^0 = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix}$ the subdivision of P^0 leads to the two polytopes

$$\begin{aligned} P_1^0 \ &= \ P^0 \cap \{z \in \mathbb{R}^d : (1 + \sqrt{2}) z_{11} + z_{12} \ \geq \ 2\} \\ P_2^0 \ &= \ P^0 \cap \{z \in \mathbb{R}^d : z_{11} + (1 + \sqrt{2}) z_{12} \ \geq \ -2 - \sqrt{2}\} \end{aligned}$$

(see Figure 2.7). The new solutions are $z_1^0 = (2, -2\sqrt{2})^T$ with optimal value 0 and $z_2^0 = (1, 1 - 2\sqrt{2})^T$ with value $\frac{1}{\sqrt{2}} - 1$. The polytope P^1 for the next iteration of Algorithm 2.2 is therefore P_2^0 with $\mu^1 = -0.2929$.

FIGURE 2.7. Subdivision of P^0 with the polyhedron $\bar{Q}^0$ in Algorithm 2.2 applied for Problem (UPE)



Among the presented possibilities for the construction of the polyhedra Q^k ($k \in \mathbb{N}$) for Algorithm 2.2 the last one leads to the least number of new polytopes in each iteration. However, the construction of these sets is, on the other hand, the most expensive one. Moreover, in contrast to the other two possibilities (see Remark 2.6.1 and 2.6.2), the number of the constraints describing an element P of the collection $\mathcal{P}$ cannot be bounded. Note that the normals of the linear constraints determining $\bar{Q}^k$ ($k \in \mathbb{N}$) depend on the iteration counter k. Therefore, even though the last approach leads to deeper cuts, at least in comparison to the regular d-simplex introduced in Subsection 2.6.2, it is not definitely clear, which approach leads to a more efficient algorithm for solving unary problems of type (UP). Before discussing the numerical performance of these three possibilities we propose in the next section a still convergent variant of Algorithm 2.2, which does not need a subdivision of the current polytope P^k ($k \in \mathbb{N}$) in each iteration.

2.7. A Variant of Algorithm 2.2

Throughout the previous sections we proposed four possible valid linear cuts (see (2.3.3), (2.3.4), (2.3.5) and (2.6.14)) for the considered unary problem. For an algorithm using only these cuts the convergence cannot be guaranteed. Nevertheless, the use of any valid cut can accelerate the convergence of Algorithm 2.2. If we use in Algorithm 2.2 for the definition of the subdivision polytopes P_i^k ($i = 1, \ldots, l$) also some of these cuts, then the resulting approach is of course still convergent. And, moreover, we can hope that this method needs less iterations for solving (UP). For example, in Problem (UPE) the additional use of cut (2.3.5) leads to a termination of Algorithm 2.2 after one step.

For the convergence of Algorithm 2.2 it is essential that for a decreasing sequence $\{P^k\}_{k\in\mathbb{N}}$ of polytopes we know that the corresponding point sequence $\{z^k\}_{k\in\mathbb{N}}$ satisfies

$$\|z^{k+1} - z^k\|_2 \ \geq \ \rho\epsilon(z^k)\,. \tag{2.7.1}$$

The use of the subdivision process guarantees this property (see the necessary Property (PR2) of the hyperplanes describing the polyhedra Q^k). As long as this relation holds also if P^{k+1} results from P^k by adding some other cuts, the convergence can be ensured even without the subdivision of P^k.

The following algorithm uses this consideration. As long as a relation similar to (2.7.1) holds by adding only valid cuts we do not subdivide the current set P^k. If this relation fails, we enforce (2.7.1) by splitting P^k.

ALGORITHM 2.3 (***Another Convergent Algorithm for Solving (UP)***).

```
Initialization
    Choose $\rho \in (0,1]$ and $l \in \mathbb{N}$, and set $P^0 \leftarrow \{z \in \mathbb{R}^d : Az \le b\}$.
    Solve the linear optimization problem (LP) $\min_{z \in P^0} h^T z$, and let $z^0$ be an
    optimal solution with optimal value $\mu_{P^0} = h^T z^0$.
    $V_{P^0} \leftarrow \{z^0\}$, $z_{P^0} \leftarrow z^0$,
    $\mu^0 \leftarrow \mu_{P^0}$, $\mathcal{P} \leftarrow \{P^0\}$, STOP $\leftarrow$ False, $k \leftarrow 0$

While STOP = False Do
    Compute the eigenvalues of $U(z^k)$ indexed in increasing order.
    If $\lambda_1(U(z^k)) \ge 0$ AND $\lambda_{n-1}(U(z^k)) \le 0$ Then                    (SC1)
        STOP $\leftarrow$ True ($z^k$ is an optimal solution of (UP))
    Else
        $\epsilon(z^k) \leftarrow \max\{\lambda_{n-1}(U(z^k)), -\lambda_1(U(z^k))\}$
        Determine affine functions $\ell_i : \mathbb{R}^d \to \mathbb{R}$ ($i = 1, \ldots, q^k \in \mathbb{N}$) satisfying
                $\ell_i(z^k) > 0$ and, for each $z \in P^k$ with $U(z) \in \mathcal{U}_n$, $\ell_i(z) \le 0$        (VCP)
        $P^k \leftarrow P^k \cap \{z \in \mathbb{R}^d : l_i(z) \le 0\,,\ i = 1, \ldots, q^k\}$
        If $P^k = \emptyset$ Then
            $\mathcal{P} \leftarrow \mathcal{P} \setminus \{P^k\}$
        Else
            Solve the LP $\min_{z \in P^k} h^T z$ and let $\bar{z}^k$ be an optimal solution.
            If $\min_{z \in V_{P^k}} \{\|\bar{z}^k - z\| - \rho\epsilon(z)\} < 0$ Then            (SDC)
                Choose $\hat{z} \in V_{P^k}$ satisfying $\|\bar{z}^k - \hat{z}\|_2 - \rho\epsilon(\hat{z}) < 0$.
                Construct a polyhedron $Q^k = \{z \in \mathbb{R}^d : (q_i^k)^T z \le c_i^k\,,\ i = 1, \ldots, l\}$
                satisfying
                        $P^k \cap Q^k \subset P^k \cap \{z \in \mathbb{R}^d : \|z - \hat{z}\|_2 \le \epsilon(\hat{z})\}$        (PR1)
                and, for each $i \in \{1, \ldots, l\}$,
                        $d(\hat{z}, H(q_i^k, c_i^k)) = \frac{|(q_i^k)^T \hat{z} - c_i^k|}{\|q_i^k\|_2} \ge \rho\epsilon(\hat{z})$.        (PR2)
                For $i = 1$ To $l$ Do
                    $P_i^k \leftarrow P^k \cap \{z \in \mathbb{R}^d : (q_i^k)^T z \ge c_i^k\}$
                    $V_{P_i^k} \leftarrow V_{P^k} \setminus \{\hat{z}\}$, $z_{P_i^k} \leftarrow z_{P^k}$
                    If $P_i^k \ne \emptyset$ Then
                Solve the LP $\min_{z \in P_i^k} h^T z$, and let $z_i^k$ be an optimal solution
                with optimal value $\mu_{P_i^k} = h^T z_i^k$.
                $\mathcal{P} \leftarrow \mathcal{P} \cup \{P_i^k\}$
              **EndIf**
            **EndFor**
            $\mathcal{P} \leftarrow \mathcal{P} \setminus \{P^k\}$
          **Else**
            $V_{P^k} \leftarrow V_{P^k} \cup \{\bar{z}^k\}$, $z_{P^k} \leftarrow \bar{z}^k$, $\mu_{P^k} \leftarrow h^T \bar{z}^k$
          **EndIf**
        **EndIf**
        **If** $\mathcal{P} = \emptyset$ **Then**                                                    (SC2)
          STOP ← **True** ($P^0 \cap \{z \in \mathbb{R}^d : U(z) \in \mathcal{U}_n\} = \emptyset$)
        **Else**
          $\mu^{k+1} \leftarrow \min_{P \in \mathcal{P}} \mu_P$
          Choose $P^{k+1} \in \mathcal{P}$ and $z^{k+1} \in P^{k+1}$ with $\mu^{k+1} = \mu_{P^{k+1}} = h^T z^{k+1}$.
        **EndIf**
      **EndIf**
      $k \leftarrow k + 1$
**EndWhile**
```

REMARK 2.7.1.

(a) If $\lambda_1(U(z^k))$ is smaller than 0, we can use the cuts (2.3.4) and (2.3.5) introduced by Ramana. If $\lambda_{n-1}(U(z^k))$ is greater than 0, the cut (2.3.3) fulfills the valid cut property (VCP), and in both cases the new cut (2.6.14) presented in the previous section is usable.

(b) If additional cuts satisfying (VCP) are used, then the number of inequalities describing a set $P \in \mathcal{P}$ cannot be bounded anymore. This does not depend on the used polyhedra Q^k ($k \in \mathbb{N}$) (compare with Remarks 2.6.1 and 2.6.2).

(c) If the set V_{P^k} is empty, then the subdivision criterion (SDC) is not fulfilled. By convention, there holds $\min_{z \in \emptyset} f(z) = \infty$.

(d) The values $\epsilon(z) = \max\{\lambda_{n-1}(U(z)), -\lambda_1(U(z))\}$ (see (SDC)) have been calculated for each element of V_{P^k} ($k \in \mathbb{N}$) at an earlier stage of the algorithm.

(e) The Algorithm 2.3 does not coincide with Algorithm 3 presented in [HR98].

If Algorithm 2.3 terminates after a finite number of iterations either by detecting the emptiness of the feasible region of (UP) or by yielding an optimal solution z^k of this problem, then the correctness of these results follows by the same argumentation as in the case of Algorithm 2.2. The result of Lemma 2.5.1 is obviously true also for this approach. Thus, we know that, for each $k \in \mathbb{N}$, μ^k is a lower bound for the optimal value of Problem (UP) (compare with (2.5.7)).

In order to guarantee the convergence of Algorithm 2.3 without a subdivision of P^k in each iteration $k \in \mathbb{N}$ we have introduced the new sets V_{P^k} ($k \in \mathbb{N}$) and the points z_{P^k} ($k \in \mathbb{N}$). The subdivision criterion (SDC) shows that only such points $\bar{z}^k \in P^k$ are added to the set V_{P^k}, which fulfill, for each $z \in V_{P^k}$,

$$\|z - \bar{z}^k\|_2 \ \geq \rho\epsilon(z) \, . \tag{2.7.2}$$

There holds furthermore that in each iteration either a point is added to V_{P^k} or one point is eliminated from this set and that the elimination of $\hat{z} \in V_{P^k}$ leads to a subdivision of P^k. In the elimination case it follows, moreover, that each point contained in a polytope $P \in \mathcal{P}$ – $\mathcal{P}$ be the collection of the relevant polytopes in an iteration $\bar{k} \geq k$ –, which is a subset of P^k must have a distance greater than $\rho\epsilon(\hat{z})$ to the point $\hat{z}$, i.e., fulfills (2.7.2) for $\hat{z}$. These special properties of the set V_{P^k} enable us to prove the convergence of Algorithm 2.3. However, we will not show the convergence of Algorithm 2.3 in the sense of Theorem 2.5.2. We prove that each accumulation point $z^\star$ of the sequence $\{z_{P^k}\}_{k\in\mathbb{N}}$, which is a special subsequence of $\{z^k\}_{k\in\mathbb{N}}$, leads to a unary matrix $U(z^\star)$ and is hence optimal for (UP).

For this purpose we first have to show that the elements of the sequence $\{z_{P^k}\}_{k\in\mathbb{N}}$ change infinitely often.

LEMMA 2.7.1. *Assume that Algorithm 2.3 generates an infinite sequence $\{P^k\}_{k\in\mathbb{N}}$ of polytopes. Let $\{P^{k_q}\}_{q\in\mathbb{N}}$ be a subsequence of $\{P^k\}_{k\in\mathbb{N}}$ with the properties that, for each $q \in \mathbb{N}$, $P^{k_{q+1}}$ is a subset of P^{k_q} and, moreover, that $P^{k_{q+1}}$ is generated by adding linear inequalities to the list of constraints describing P^{k_q}. Denote by $I = \{q \in \mathbb{N} : z_{P^{k_{q+1}}} \neq z_{P^{k_q}}\}$ the set of all indices $q \in \mathbb{N}$, where $z_{P^{k_q}}$ is different from its successor in the sequence $\{z_{P^{k_q}}\}_{q\in\mathbb{N}}$. Then the following assertions are true.*

(i) *The set I contains an infinite number of elements, i.e., $|I| = \infty$.*

(ii) *For each $q \in I$, there holds*

$$\|z_{P^{k_{q+1}}} - z_{P^{k_q}}\|_2 \ \geq \ \rho\epsilon(z_{P^{k_q}}) \, . \tag{2.7.3}$$

PROOF: Assume, first, that the decreasing sequence $\{P^{k_q}\}_{q\in\mathbb{N}}$ has an additional property. Let, for each $q \in \mathbb{N}$, $P^{k_{q+1}}$ be a *direct child* of P^{k_q}, i.e., assume that, for each $q \in \mathbb{N}$, there holds

$$P^{k_{q+1}} \;=\; P^{k_q} \cap \{z \in \mathbb{R}^d : \ell_i(z) \le 0\,,\, i = 1, \dots, q^{k_q}\} \tag{2.7.4}$$

or

$$\begin{aligned} P^{k_{q+1}} \;&=\; P^{k_q} \cap \{z \in \mathbb{R}^d : \ell_i(z) \le 0\,,\, i = 1, \dots, q^{k_q}\} \\ &\qquad \cap \{z \in \mathbb{R}^d : (q_j^{k_q})^T z \ge c_j^{k_q}\}\,, \end{aligned} \tag{2.7.5}$$

where $\ell_i : \mathbb{R}^d \to \mathbb{R}$ $(i = 1, \dots, q^{k_q})$ are affine functions satisfying (VCP) with respect to z^{k_q} and j is an element of $\{1, \dots, l\}$. In view of the definition of Algorithm 2.3 we know that the point $z_{P^{k_{q+1}}}$ is only different to $z_{P^{k_q}}$, if (2.7.4) holds.

In order to prove assertion (i) assume, by contradiction, that I contains only a finite number of elements, i.e., there is an index $q_0 \in \mathbb{N}$ such that, for each $q \ge q_0$,

$$z_{P^{k_q}} \;=\; z_{P^{k_{q_0}}}\;.$$

It follows that $P^{k_{q+1}}$ results from P^{k_q} $(q \ge q_0)$ by adding valid cuts and executing a subdivision (see (2.7.5)). Then we obtain, for each $q \ge q_0$,

$$|V_{P^{k_{q+1}}}| \;=\; |V_{P^{k_q}}| - 1\;.$$

Since $V_{P^{k_q}}$ $(q \in \mathbb{N})$ contains only a finite number of points, this is a contradiction and proves (i), in particular, it shows that I is not empty.

Choose next an arbitrary, but fixed index $q \in I$. It follows that $P^{k_{q+1}}$ is given by (2.7.4), and, moreover, that $z_{P^{k_{q+1}}} = \bar{z}^{k_q}$ and (SDC) is not fulfilled for $\bar{z}^{k_q}$. We prove now that, for each $r \in \{0, \dots, q\}$, there holds

$$\|z_{P^{k_r}} - z_{P^{k_{q+1}}}\|_2 \;\ge\; \rho\epsilon(z_{P^{k_r}})\,, \tag{2.7.6}$$

which is a stronger result than (2.7.3). Choose $r \in \{0, \dots, q\}$ and let $\bar{r} \in \mathbb{N}$, $\bar{r} < r$ be the index such that $z_{P^{k_r}} = \bar{z}^{k_{\bar{r}}}$, i.e., $z_{P^{k_r}}$ was set in iteration $k_{\bar{r}}$. Each point used for updating z_{P^k} is added to the set V_{P^k} and, thus, we have $z_{P^{k_r}} \in V_{P^{k_{\bar{r}+1}}}$. We distinguish two cases.

If $z_{P^{k_r}}$ is still an element of $V_{P^{k_q}}$, then we obtain

$$0 \;\le\; \min_{z\in V_{P^{k_q}}} \{\|z - \bar{z}^{k_q}\|_2 - \rho\epsilon(z)\} \;\le\; \|z_{P^{k_r}} - z_{P^{k_{q+1}}}\|_2 - \rho\epsilon(z_{P^{k_r}})\,,$$

since the subdivision criterion (SDC) is not fulfilled for $\bar{z}^{k_q}$. This shows (2.7.6) in this case.

If $z_{P^{k_r}}$ is not an element of $V_{P^{k_q}}$, then we know that there is an index $l \in \mathbb{N}$, $\bar{r} < l < q$ such that $z_{P^{k_r}} \in V_{P^{k_l}}$ and $z_{P^{k_r}} \notin V_{P^{k_{l+1}}}$. This implies that in iteration k_l a polyhedron Q^{k_l} satisfying (PR1) and (PR2) with respect to $\hat{z} = z_{P^{k_r}}$ was constructed. In view of (PR2) it follows that, for each $z \in P^{k_{l+1}}$, there holds

$$\|z - z_{P^{k_r}}\|_2 \;\geq\; \rho\epsilon(z_{P^{k_r}}) \; .$$

$P^{k_{q+1}}$ is by assumption a subset of $P^{k_{l+1}}$. Thus, (2.7.6) follows also in this case.

Let now $\{P^{k_q}\}_{q\in\mathbb{N}}$ be an arbitrary sequence of polytopes with the properties given in the formulation of the lemma. In view of (2.7.6) we know that each update of $z_{P^{k_{q+1}}}$ by $\bar{z}^{k_q}$ leads to a point, which is different from all $z_{P^{k_l}}$ $(l \leq q)$. Therefore, assertion (i) follows immediately by the facts that the special sequence considered first in the present proof has this property and that $\{P^{k_q}\}_{q\in\mathbb{N}}$ is a subsequence of such a special sequence. We obtain, furthermore, that (2.7.6) implies (2.7.3). ■

With the results of the previous lemma we are now able to prove the postulated convergence of Algorithm 2.3.

THEOREM 2.7.2. *Assume that Algorithm 2.3 does not terminate after a finite number of iterations. Then there holds that each accumulation point $z^\star$ of the sequence $\{z_{P^k}\}_{k\in\mathbb{N}}$ is an optimal solution of Problem (UP).*

PROOF: Let $z^\star$ be an accumulation point of the sequence $\{z_{P^k}\}_{k\in\mathbb{N}}$ and let $\{z_{P^{k_q}}\}_{q\in\mathbb{N}}$ be a subsequence converging to $z^\star$. By passing to a subsequence, if necessary, we can assume that the corresponding sequence $\{P^{k_q}\}_{q\in\mathbb{N}}$ of polytopes is decreasing and, moreover, that $P^{k_{q+1}}$ is generated by adding linear constraints to the list of constraints describing P^{k_q} $(q \in \mathbb{N})$. In view of Lemma 2.7.1(i) we can, in addition, assume that each element of the sequence $\{z_{P^{k_q}}\}_{q\in\mathbb{N}}$ is different from its successor, i.e., there holds, for each $q \in \mathbb{N}$,

$$P^{k_{q+1}} \;\subset\; P^{k_q}$$

and

$$z_{P^{k_{q+1}}} \;\neq\; z_{P^{k_q}} \; .$$

From Relation (2.7.3) (Lemma 2.7.1(ii)) we obtain, for each $q \in \mathbb{N}$,

$$\|z_{P^{k_{q+1}}} - z_{P^{k_q}}\|_2 \;\geq\; \rho\epsilon(z_{P^{k_q}}) \; .$$

Using the definition of $\epsilon(z_{P^{k_q}})$ $(q \in \mathbb{N})$ and the continuity of the eigenvalue functionals this relation implies, as in the proof of Theorem 2.5.2, the feasibility of $z^\star$

for (UP). Furthermore, for each $q \in \mathbb{N}$, we know that

$$h^T z_{P^{k_q}} \;\leq\; h^T z^{k_q} \;=\; \mu_{P^{k_q}} \;=\; \mu^{k_q} \;\leq\; \min_{z \in F} h^T z \;\leq\; h^T z^\star \,,$$

where F denotes the feasible region of(UP). This shows the optimality of $z^\star$. Note that $z_{P^{k_q}}$ is the optimal solution of $\min_{z \in \bar{P}} h^T z$ for a polytope $\bar{P} \supset P^{k_q}$. ■

At first glance this convergence result is weaker than the one obtained for Algorithm 2.2 (see Theorem 2.5.2). We only prove the convergence of a subsequence of $\{z^k\}_{k \in \mathbb{N}}$. However, a direct consequence of Lemma 2.7.1 is that at the beginning of an infinite number of iterations we have the situation that the current point z^k coincides with the point z_{P^k}. In view of Theorem 2.7.2 this implies that the values $|\lambda_1(U(z^k))|$ and $|\lambda_{n-1}(U(z^k))|$ ($k \in \mathbb{N}$) become arbitrarily small. Thus, Algorithm 2.3 is also well defined.

We cannot expect that either Algorithm 2.2 or Algorithm 2.3 stop with an optimal solution of Problem (UP) after a finite number of iterations. In order to obtain finite algorithms we have to be satisfied with ϵ-approximate solutions of this problem, i.e., with points $\bar{z} \in P$ satisfying

$$\max\{\lambda_{n-1}(U(\bar{z})), -\lambda_1(U(\bar{z}))\} \;\leq\; \epsilon \tag{2.7.7}$$

for a given tolerance $\epsilon > 0$. If we replace the stopping criterion (SC1) in Algorithm 2.2 and in Algorithm 2.3 by

$$\textbf{If } \lambda_1(U(z^k)) \geq -\epsilon \textbf{ AND } \lambda_{n-1}(U(z^k)) \leq \epsilon \textbf{ Then} \,, \tag{2.7.8}$$

then we obtain by considering Theorem 2.5.2, respectively by taking the previous considerations into account, in both cases a finite approach. From this point of view, both convergence results – Theorem 2.5.2 as well as Theorem 2.7.2 – have a comparable quality.

We finish the discussion of solution methods for unary problems of type (UP) with some numerical results. In the next section we examine, in particular, the numerical applicability of the presented algorithms for solving all-quadratic problems of type (QP), since this is the main scope of this dissertation.

2.8. Computational Results

Algorithm 2.3 was encoded in C++ with management of the collection $\mathcal{P}$ of relevant polytopes by so-called AVL-trees. The linear subproblems were solved by using the Simplex-Algorithm based *CPLEX-5.0* code. After solving the first LP-relaxation of (UP) in the initialization phase of Algorithm 2.3 each new subproblem results from a previous one by adding some new constraints or by changing some right-hand sides. For that reason we solved only the initial problem by applying the primal Simplex-Algorithm. The solution of each subsequent subproblem was determined with the dual Simplex-Algorithm, which is supported by the *CPLEX-5.0* code. This strategy reduced the running-time for solving the subproblems. However, on the other hand, we needed more storage, since all necessary information about the current solution, like the dual variables, the slacks and so on, had to be stored for each polytope $P \in \mathcal{P}$. Otherwise, we would not be able to start the dual version of the Simplex-Algorithm without additional effort.

Apart from the solution of linear optimization problems, other classical problems can occur in Algorithm 2.3. First of all, we have to calculate eigenvalues of different matrices. For the construction of the cuts introduced by Ramana we have to determine the inverse of a matrix (see Section 2.3). In order to obtain a representation of the polyhedra $\bar{Q}^k$ ($k \in \mathbb{N}$) (see Subsection 2.6.3) we need solutions of linear equations and we need an orthonormal basis describing the linear space $H^k - \{z^k\}$. In the implementation of the algorithm all these problems were solved by applying appropriate routines from the *NAG* C-library.

With respect to the choice of possible linear constraints satisfying the valid cut property (VCP) and in view of the three types of polyhedra Q^k ($k \in \mathbb{N}$) proposed in Section 2.6 there is a large number of implementable variants of Algorithm 2.3. Before discussing the numerical performance of some selected variants we present a slight modification of the subdivision process, which can lead to a substantial improvement of the numerical performance of our approach.

2.8.1. A Slight Modification of the Subdivision Process. In the subdivision process in Algorithm 2.3 we construct each new polytope P_i^k ($i = 1, \ldots, l; k \in \mathbb{N}$) by adding one of the constraints describing Q^k to the list of constraints describing P^k. Independent of the choice of the polyhedron Q^k this strategy can lead to overlapping regions, i.e., there can hold

$$\mathrm{int} P_i^k \cap \mathrm{int} P_j^k \;\neq\; \emptyset$$

for some $i, j \in \{1, \ldots, l\}$ (see the Figures 2.4, 2.5 and 2.7). This is not reasonable, since parts of the feasible region of (UP) are examined more than once by using this strategy. For the correctness of Algorithm 2.3 and Algorithm 2.2, respectively, it is sufficient, if the new polytopes P_i^k $(i = 1, \dots, l)$ form a partition of the set $P^k \setminus \{z \in \mathbb{R}^d : (q_i^k)^T z < c_i^k\ ,\ i = 1, \dots, l\}$, i.e., if there holds

$$\bigcup_{i=1}^{l} P_i^k \;=\; P^k \setminus \{z \in \mathbb{R}^d : (q_i^k)^T z < c_i^k\ ,\ i = 1, \dots, l\} \;\supset\; P^k \setminus B_{z^k}$$

and, for each $i, j \in \{1, \dots, l\}$ with $i \neq j$,

$$\mathrm{int} P_i^k \cap \mathrm{int} P_j^k \;=\; \emptyset \tag{2.8.1}$$

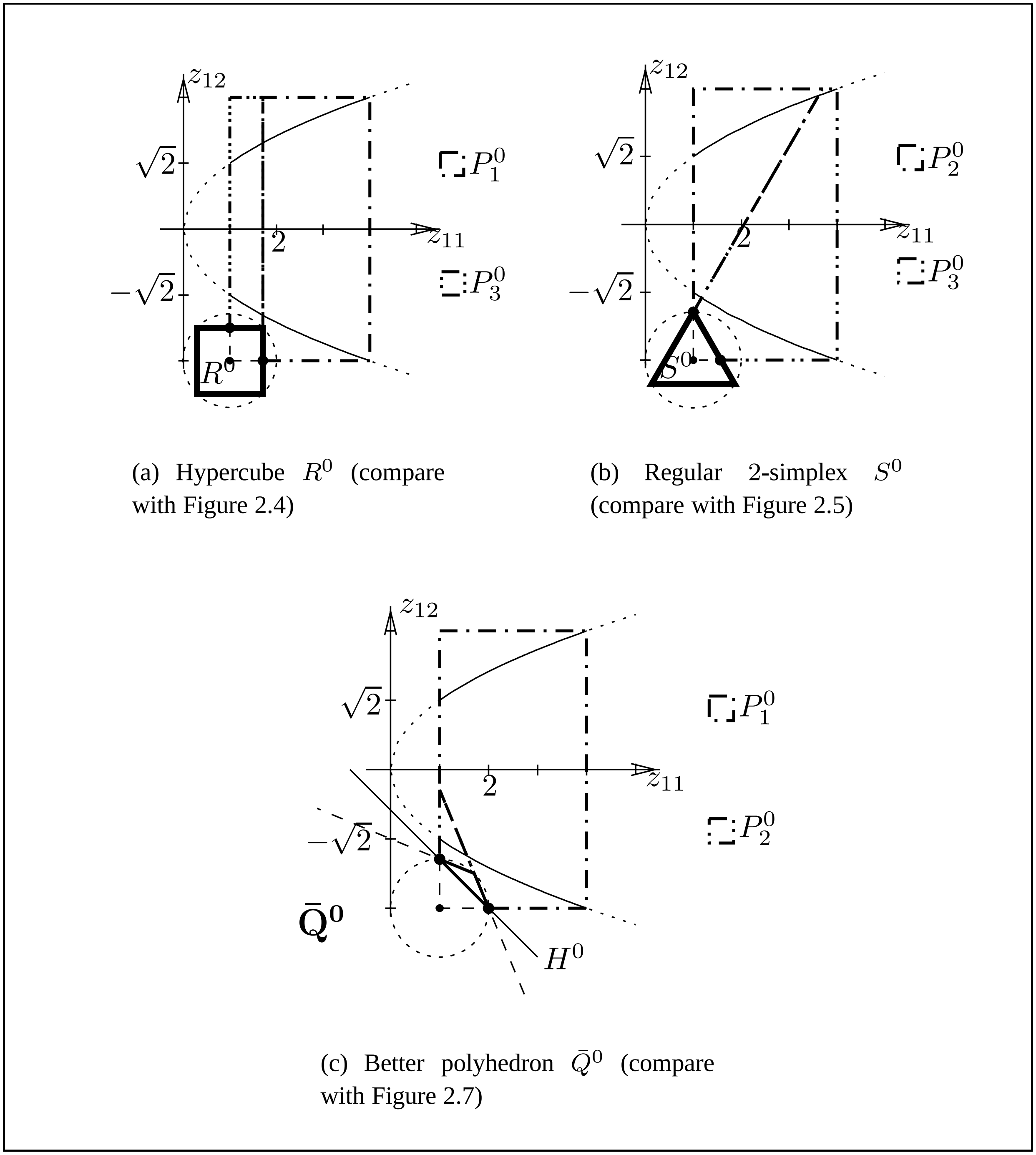
(see Definition 1.2.1). Property (2.8.1) can be achieved by a slight modification of the definition of the polytopes P_i^k. If we set, for each index $i \in \{1, \dots, l\}$,

$$P_i^k \leftarrow P^k \cap \{z \in \mathbb{R}^d : (q_i^k)^T z \geq c_i^k\ ,\ (q_j^k)^T z \leq c_j^k\ ,\ j = 1, \dots, i-1\}\ ,$$

then we obtain that the union of the sets P_i^k $(i = 1, \dots, l)$ is the same set as by only adding the constraint $(q_i^k)^T z \geq c_i^k$. And, moreover, these sets fulfill the additional property (2.8.1) (see Figure 2.8). In Remark 2.6.1 and Remark 2.6.2 we pointed out that the normals of the hyperplanes describing the hypercubes R^k and the regular d-simplices S^k, respectively, do not depend on the iteration counter k. Thus, if one of these two sets is used in Algorithm 2.3 for the polyhedron Q^k, the subdivision of P^k leads only to a change of the right-hand sides of some constraints. Therefore, the proposed modification is – from a numerical point of view – not expensive and does not lead to new storage requirements. It does not really matter whether one right-hand side is changed or up to $2d$. In the case of the third presented polyhedron this new subdivision strategy leads to growing storage requirements and is numerically more expensive, since the number of constraints describing a polytope $P \in \mathcal{P}$ is growing faster. However, in each case we can expect that the elimination of the overlapping parts results in a more efficient approach for solving (UP).

We applied Algorithm 2.2, i.e., Algorithm 2.3 without additional cuts, for solving our example problem, where we used the subdivision process with and without the modification. If we used the hypercubes R^k or the polyhedron $\bar{Q}^k$ based on the modified d-simplex, then in both cases the algorithm needed the same number of iterations and the same number of linear subproblems had to be solved. For these two cases the modification led only to a slight increase in the running-time, especially in the case of $\bar{Q}^k$. Note that by adding more than one constraint in an iteration the effort for solving the resulting linear subproblems increases faster than by adding only one constraint. Table 2.1 shows the effort for solving (UPE) with these two choices of the polyhedron $\bar{Q}^k$. The execution of Algorithm 2.2 was terminated,

FIGURE 2.8. Modification of the subdivision process applied for (UPE)



(a) Hypercube R^0 (compare with Figure 2.4)

(b) Regular 2-simplex S^0 (compare with Figure 2.5)

(c) Better polyhedron $\bar{Q}^0$ (compare with Figure 2.7)

if the ϵ-approximate stopping criterion (2.7.8) with $\epsilon = 10^{-4}$ was satisfied. The fourth column of this table showing the maximal number of polytopes, which had to be stored at an iteration of Algorithm 2.2 in the set $\mathcal{P}$, illustrates the storage requirements of the different approaches.

TABLE 2.1. Effort for solving (UPE) with Algorithm 2.2

Polyhedron Q^k	Number of iterations	Number of solved LP's	Maximal number of elements in $\mathcal{P}$	Time [a] [b] (in sec.)
R^k	27	71	28	0.18 (0.18)
$\bar{Q}^k$	16	33	17	0.17 (0.18)

[a] run on a *SUN SPARC 20* workstation
[b] running-time for Algorithm 2.2 with modification is given in brackets

That the modification of the subdivision process in Algorithm 2.2 does not result in an improvement, if we use the hypercube R^k or the polyhedron $\bar{Q}^k$, depends on the special structure of Problem (UPE). An examination of the iterations of Algorithm 2.2 without the modification shows that each optimal solution z^k of a linear subproblem does not belong to a part of the current polytope P^k, which could be eliminated at an earlier stage of the method by applying the modification. Therefore, as well with as without the modification, the same work has to be done in order to solve Problem (UPE).

If we apply the regular d-simplex S^k, the numerical performance depends significantly on the subdivision strategy used in Algorithm 2.2, as it is displayed in Table 2.2. In view of the first iteration of Algorithm 2.2 with the polyhedron S^k this result is not surprising. In Figure 2.5 we see that the optimal solution $z^\star = (1, -\sqrt{2})^T$ belongs to the two non-empty polytopes P_2^0 and P_3^0. Therefore, we know that Algorithm 2.2 without the modified subdivision strategy must generate at least two sequences of polytopes $\{P^{k_q}\}_{q\in\mathbb{N}}$, one starting with P_2^0 and one starting with P_3^0, such that the corresponding point sequences $\{z_{P^{k_q}}\}_{q\in\mathbb{N}}$ converge to $z^\star$. If we apply the modification, $z^\star$ is contained in only one polytope (see Figure 2.8(b)).

TABLE 2.2. Effort for solving (UPE) by applying S^k

Subdivision strategy	Number of iterations	Number of solved LP's	Maximal number of elements in $\mathcal{P}$	Time [a] (in sec.)
no modification	1708	5125	1709	16.09
modification	68	183	47	0.56

[a] run on a *SUN SPARC 20* workstation

Taking the large performance difference of Algorithm 2.2 with and without the modification in the above case into account, we can expect that on average the modification of the subdivision strategy results in an improvement of the numerical performance. The extra work we have to do, especially in the case of $\bar{Q}^k$, can lead to a substantial reduction of the number of iterations and, hence, of the total time for solving unary problems. For that reason we examine in the next subsection the numerical performance only of variants of Algorithm 2.3, which apply the described modification of the subdivision strategy.

2.8.2. Applicability to All-Quadratic Problems. In the following we discuss the numerical applicability of Algorithm 2.3 to all-quadratic problems of type (QP). We saw in Section 2.2 that for each problem of type (QP) there is an equivalent unary problem of type (UP), and, thus, we can solve arbitrary all-quadratic problems by applying the approaches presented so far. We tried to solve the unary problems, which result from the previously described transformation (see Section 2.2) of the all-quadratic problems belonging to our randomly generated test set (see Section 1.5).

At the end of Section 2.7 we pointed out that we have to be satisfied with ϵ-approximate solutions in order to obtain a finite algorithm. The ϵ-approximate stopping criterion (2.7.8) is usable for arbitrary unary problems. However, if we apply this stopping criterion, we know nothing about the quality of the determined solution z^k, in particular, we do not know how far away from the optimal value lies the calculated value $\mu^k = h^T z^k$. If we solve the transformations of all-quadratic problems, we are able to formulate a stopping criterion such that this quality of the determined solution with respect to the original quadratic problem can be estimated. Before discussing the numerical performance of our approaches, we propose first this special stopping criterion.

Assume that an all-quadratic problem of type (QP), i.e., a problem with the form

$$\begin{aligned} \min\ x^T Q^0 x + (d^0)^T x & \\ x^T Q^l x + (d^l)^T x + c^l \ \leq \ 0 & \qquad l = 1, \dots, p \\ A^Q x \ \leq \ b^Q & \\ l^Q \leq x \leq L^Q & \\ x \in \mathbb{R}^n & \end{aligned} \tag{$\overline{\text{QP}}$}$$

is given.

Let, furthermore,

$$\begin{aligned} &\min h^T z \\ A^U z &\leq b^U \\ l^U \leq z &\leq L^U \\ U(z) \in \mathcal{U}_{n+1}\ ,\ & z \in \mathbb{R}^{\binom{n+1}{2}+n} \end{aligned} \tag{$\overline{\text{UP}}$}$$

be the equivalent unary problem resulting from the transformation of ($\overline{\text{QP}}$) described in Section 2.2. The superscripts U and Q respectively are used in the same way as in Section 2.2, and the dimensions of all involved matrices and vectors are the same as there (see, in particular, pages 23f.).

Set

$$\delta \ := \ \min\{\frac{1}{\|h\|_2}\,,\ \frac{1}{\|a_l^U\|_2}\,,\ l = 1, \ldots, p\}\,\epsilon\,, \tag{2.8.2}$$

where ϵ is a given tolerance greater than 0, and a_l^U $(l = 1, \ldots, p)$ denotes the l-th row of the matrix A^U. Let z^k $(k \in \mathbb{N})$ be the current point at the beginning of iteration k of Algorithm 2.3. Determine an n-dimensional point x^k by setting

$$x_i^k \ := \ \tfrac{1}{\sqrt{2}} z_{i,n+1}^k \tag{2.8.3}$$

(compare with the definition of $\bar{x}$ in Theorem 2.2.1). If z^k is feasible for ($\overline{\text{UP}}$), then we know by the same arguments as in the proof of Theorem 2.2.1 that x^k must be feasible for ($\overline{\text{QP}}$). Determine, furthermore, a ($d = \binom{n+1}{2} + n$)-dimensional point $\hat{z}^k$ indexed in the same manner as z^k by setting

$$\begin{aligned} &\hat{z}_{i,n+1}^k := \sqrt{2} x_i^k\ ,\ \hat{z}_{ii}^k := (x_i^k)^2\ (i = 1, \ldots, n)\ , \\ &\hat{z}_{ij}^k := \sqrt{2} x_i^k x_j^k\ (1 \leq i < j \leq n) \end{aligned} \tag{2.8.4}$$

(compare with the definition of $\bar{z}$ in Theorem 2.2.1). If x^k is feasible for ($\overline{\text{QP}}$), we know (see again the proof of Theorem 2.2.1) that $\hat{z}^k$ is feasible for ($\overline{\text{UP}}$), and, moreover, that the points z^k and $\hat{z}^k$ coincide. If we replace the ϵ-approximate stopping criterion (2.7.8) by the following

$$\textbf{If}\ \|z^k - \hat{z}^k\| \ \leq \ \delta\ \textbf{Then} \tag{2.8.5}$$

with δ defined as in (2.8.2), then we obtain a solution method for all-quadratic problems, which detects in finite time either the emptiness of the feasible region of ($\overline{\text{QP}}$), or delivers a point z^k such that the corresponding point x^k defined as in (2.8.3) has the properties

$$\begin{aligned} (x^k)^T Q^l x^k + (d^l)^T x^k + c^l \ &\leq \ \epsilon \qquad l = 1, \dots, p \\ A^Q x \ &\leq \ b^Q \\ l^Q \ \leq x &\leq L^Q \ . \end{aligned} \tag{2.8.6}$$

If this point x^k is additionally feasible for ($\overline{\text{QP}}$), it even follows, that the calculated value μ^k and the optimal value of ($\overline{\text{QP}}$) have a distance not bigger than ϵ.

Indeed, by replacing the stopping criterion (2.7.8) with (2.8.5) the resulting algorithm is, first of all, still well defined. In view of Theorem 2.7.2 we know that each accumulation point $z^\star$ of the sequence $\{z_{P^k}\}_{k\in\mathbb{N}}$ is a feasible point for ($\overline{\text{UP}}$). Thus, $\hat{z}^\star$ defined as in (2.8.4) is equal to $z^\star$. As mentioned at the end of the previous section we know, furthermore, that in an infinite number of iterations there holds that the points z^k and z_{P^k} coincide. Therefore, we achieve that the Euclidean distance between z^k and $\hat{z}^k$ becomes arbitrarily small, i.e, (2.8.5) will be fulfilled after a finite number of iterations.

If Algorithm 2.2 or Algorithm 2.3 detects the emptiness of the feasible region of ($\overline{\text{UP}}$), then the emptiness of the feasible set of ($\overline{\text{QP}}$) follows by the equivalence between both problems. If one of these algorithms terminates with a point z^k satisfying (2.8.5), it is clear that x^k defined as in (2.8.3) fulfills the linear constraints of ($\overline{\text{QP}}$). This follows by the construction of A^U, b^U, l^U and L^U. Moreover, by the special definition of $\hat{z}^k$ we achieve, as in the proof of Theorem 2.2.1, that there holds

$$(x^k)^T Q^l x^k + (d^l)^T x^k + c^l \ = \ (a_l^U)^T \hat{z}^k - b_l^U \qquad l = 1, \dots, p \tag{2.8.7}$$

and

$$(x^k)^T Q^0 x^k + (d^0)^T x^k \ = \ h^T \hat{z}^k \ . \tag{2.8.8}$$

The relation (2.8.7) and the feasibility of z^k with respect to the linear constraints of ($\overline{\text{UP}}$) imply, for each $l \in \{1, \dots, p\}$,

$$(x^k)^T Q^l x^k + (d^l)^T x^k + c^l \ \leq \ (a_l^U)^T (\hat{z}^k - z^k) \ \leq \ \|a_l^U\|_2 \|z^k - \hat{z}^k\|_2 \ \leq \ \epsilon \ .$$

Hence, x^k fulfills (2.8.6). If x^k is additionally feasible for ($\overline{\text{QP}}$), it follows with (2.8.8)

$$|h^T z^k - (x^k)^T Q^0 x^k - (d^0)^T x^k| \ \leq \ \|h\|_2 \|z^k - \hat{z}^k\|_2 \ \leq \ \epsilon$$

and

$$h^T z^k \;=\; \mu^k \;\leq\; \min_{z \in F^U} h^T z \;\leq\; h^T \hat{z}^k \;=\; (x^k)^T Q^0 x^k + (d^0)^T x^k \;,$$

where F^U denotes the feasible region of ($\overline{\text{UP}}$). This means that x^k is ϵ-optimal for Problem ($\overline{\text{QP}}$). With the foregoing considerations we have shown that the stopping criterion (2.8.5) is a more reasonable criterion than (2.7.8), when we solve all-quadratic problems of type (QP) via unary problems. If we use (2.8.5), then we know something about the quality of the calculated point x^k with respect to the quadratic problem, which we would like to solve.

REMARK 2.8.1. The point $\hat{z}^k$ ($k \in \mathbb{N}$) defined as in (2.8.4) leads to a unary matrix $U(\hat{z}^k)$. Therefore, we know taking Lemma 2.3.1 and Theorem 2.4.4 into account that there holds

$$\max\{\lambda_{n-1}(U(z^k)), -\lambda_1(U(z^k))\} \;\leq\; \|z^k - \hat{z}^k\|_2 \,.$$

This shows that with respect to the definition of δ we need, in comparison with the stopping criterion (2.7.8), a higher accuracy for the values $\lambda_{n-1}(U(z^k))$ and $\lambda_1(U(z^k))$ in order to satisfy (2.8.5).

Our main motivation for considering unary problems were the results of Ramana's dissertation [RAM93, Chapter 7], in particular, his really promising preliminary numerical results. He solved with Algorithm 2.1 large unary problems with acceptable running-times. However, the affine function $U : \mathbb{R}^d \to \mathcal{S}_n$, which he used, had a simple structure. By applying Algorithm 2.1 for solving unary problems, which result from the transformation of all-quadratic problems and which, thus, have a complex affine function, this pure outer approximation approach showed a really bad performance in our computational tests. Even small unary problems resulting from 2-dimensional quadratic problems could not be solved in acceptable times. Moreover, this approach induced numerical problems. In many test problems the algorithm seemed to stick in a point away from an ϵ-approximate solution. Since the hyperplanes used in this scheme became too *flat* the algorithm made small progress and the numerical problems increased. Note that too *flat* hyperplanes can lead to ill-conditioned matrices B^k such that we can obtain increasing numerical errors, if we do not invest additional effort.

Even though an algorithm based only on the cuts introduced by Ramana showed a bad performance, his linear constraints can be used in Algorithm 2.3 in order to accelerate the convergence of this approach. The fact that the use of additional cuts in Algorithm 2.3 led to a more efficient solution method for unary problems, was the first result of our numerical tests. We compared different combinations of the four valid cuts presented in this chapter. The new cut (2.6.14) did not accelerate the convergence of Algorithm 2.3 in the most cases, when we used the cuts (2.3.3) and (2.3.5). The cut (2.3.5) was mostly better than (2.3.4). Consequently, the most efficient combination of the possible four cuts in our numerical tests was the cut (2.3.3) for the case $\lambda_{n-1}(U(z^k)) > 0$ and (2.3.5) for $\lambda_1(U(z^k)) < 0$.

In the following we compare the numerical performance of Algorithm 2.3 applying these two cuts with the numerical performance of Algorithm 2.2, i.e., of Algorithm 2.3 without any additional valid cut. In both approaches we used the hypercubes R^k developed in Subsection 2.6.1 for subdividing the set P^k, if necessary. The execution of the algorithms was terminated, if the appropriate stopping criterion (2.8.5) was satisfied with δ defined as in (2.8.2) for a prespecified tolerance $\epsilon > 0$. Remember that the all-quadratic problems belonging to our test set have always a non-empty feasible region (see Section 1.5). In order to avoid excessive storage requirements, and, thus, also in order to avoid excessive running-times we restricted the maximal number of polytopes P, which had to be stored at an iteration in the collection $\mathcal{P}$. In the case of Algorithm 2.2 this maximal number was $100,000$. Since the storage requirements increase, when additional cuts are used, we reduced this number to $50,000$ in the case of Algorithm 2.3.

TABLE 2.3. Comparison of the numerical effort for solving 2-dimensional all-quadratic problems with the accuracy $\epsilon = 0.1$

Algorithm	NuP	ANuLP	MNuLP	ATime	MTime	ACol	MCol
$p = 1$							
2.2	42	142,377	52,224	103.5	38.7	24,853	18,499
2.3	50	26,914	1,304	96.9	2.63	2,931	404
$p = 2$							
2.2	42	148,956	66,168	123.9	51.7	24,708	18,901
2.3	50	14,015	497.5	51.9	1.08	1,235	145
$p = 3$							
2.2	42	98,574	95,566	83.9	79.1	23,724	19,034
2.3	50	4,787	746.5	12.2	1.54	789	199.5
$p = 4$							
2.2	41	121,551	72,285	102.8	61.1	25,688	17,744
2.3	50	7,423	1,398	19.5	2.96	1,038	294

Table 2.3 and Table 2.4 display the numerical effort, which the two described approaches needed in order to solve the 50 5-dimensional unary problems resulting from the transformation of our 2-dimensional quadratic test problems. We use the abbreviations NuP for the number of test problems, which could be solved by the two methods within the given storage capacities. ANuLP is used for the average number of linear problems, which had to be solved during the execution of each algorithm. ATime stands for the average running-time in seconds, and in the column ACol we display the average maximal number of elements, which had to be stored in the collection $\mathcal{P}$. The three columns with MNuLP, MTime and MCol show the corresponding values of the medians. Note that in the calculation of the average values and of the medians we considered only the problems, which could be solved within the given storage capacities. All numerical test discussed here, were run on a *SUN ULTRA 60* workstation.

TABLE 2.4. Comparison of the numerical effort for solving 2-dimensional all-quadratic problems with the accuracy $\epsilon = 0.01$

Algorithm	NuP	ANuLP	MNuLP	ATime	MTime	ACol	MCol
$p = 1$							
2.2	34	225,822	108,570	157.2	86.4	34,896	34,749
2.3	49	56,341	2,841	197.5	6.6	5,555	768
$p = 2$							
2.2	36	173,958	138,446	143.5	109.8	34,656	34,607
2.3	49	14,699	1,007	49.0	2.11	1,473	260
$p = 3$							
2.2	32	204,578	163,942	175.2	139.3	40,597	40,361
2.3	50	11,920	1,752	35.2	3.51	1,954	401.5
$p = 4$							
2.2	30	150,128	126,474	131.1	110.4	32,695	32,762
2.3	50	17,351	2,725	53.8	6.10	2,505	532

It is obvious that Algorithm 2.3 with the additional cuts is the more efficient approach for determining ϵ-approximate solutions for our test problems. In almost all cases this approach was significantly faster and, moreover, with this algorithm we were able to solve the most problems within the given storage capacities. Algorithm 2.2 did not terminate with a solution in one third of the test problems, if an accuracy of $\epsilon = 0.01$ was required. In both approaches there is a great difference between the average values and the medians. This makes clear – even though these approaches, in particular Algorithm 2.3, showed a rather good performance in at least 50% of the solved test examples (see the medians) – there were some examples, where we needed a huge effort in order to determine a solution. Thus, our approach did not show a good performance on average, particularly in comparison with the simplicial branch-and-bound method for all-quadratic problems, which we will develop in the next chapter.

An advantage of the presented approach is that the solution effort does not depend on the number p of quadratic constraints, as it will be the case for the method described in the next chapter. This is due to the fact that the effort for solving a unary problem does not depend on the number of linear constraints. The structure of the affine matrix mapping is decisive.

Another interesting result of our numerical tests was that the subdivision process used in Algorithm 2.3 had a regularization effect in the following sense. We have mentioned that Algorithm 2.1 can lead to numerical problems, if the hyperplanes used there get too *flat*. In Algorithm 2.3 we used the same construction rule for the additional cuts, but the subdivision of the current polytope P^k, which was enforced, if the additional cuts became too *shallow*, avoided such numerical problems. From this point of view, Algorithm 2.3 was numerically more stable.

We have seen that the additional use of valid cuts in Algorithm 2.3 is reasonable, since we obtain a significant speedup of our solution method. Our numerical experience also showed, that on average the additional cuts (2.3.4) and (2.6.14) only increased the running-time of Algorithm 2.3. It is hence not cogent that each affine function satisfying (VCP) accelerate the convergence of this approach. An appropriate combination of valid cuts is decisive. This should be considered, when new cuts are developed in order to improve the performance of Algorithm 2.3.

We still have to examine, which choice of the polyhedron Q^k leads to the most efficient algorithm. For this aim we also tried to solve the 2-dimensional all-quadratic test problems using the regular d-simplex S^k and using the polyhedron $\bar{Q}^k$. The corresponding results together with the effort of Algorithm 2.3 using the hypercubes R^k are presented in Table 2.5 and Table 2.6. We use the same abbreviations as in the foregoing tables. The additional columns ACon and MCon display the average and the median of the maximal number of linear constraints, which were needed for describing an element P of $\mathcal{P}$. These facts together with the columns corresponding to the maximal number of elements contained in $\mathcal{P}$ give us more insight into the real storage requirements. The more constraints we

TABLE 2.5. Comparison of the numerical effort for solving 2-dimensional all-quadratic problems with the accuracy $\epsilon = 0.1$

Q^k	NuP	ANuLP	MNuLP	ATime	MTime	ACol	MCol	ACon	MCon
$p = 1$									
R^k	50	26,914	1,340	96.9	2.63	2,931	404	61	54
S^k	45	37,721	2,2237	266	9.36	2,578	323	151	164
$\bar{Q}^k$	46	27,273	2,980	226	13.42	3,226	643	159	156
$p = 2$									
R^k	50	14,015	497.5	51.9	1.08	1,235	145	54	43.5
S^k	48	12,612	814.5	89.5	3.09	1,034	147.5	132.6	113.5
$\bar{Q}^k$	49	15,421	1,176	133	4.11	1,631	300	146	111
$p = 3$									
R^k	50	4,787	746.5	12.2	1.54	789	199.5	49	50.5
S^k	50	13,622	1530	98.8	5.74	1,343	193.5	134	126.5
$\bar{Q}^k$	50	11,756	1,319	80.3	5.28	1,689	299	124	126
$p = 4$									
R^k	50	7,423	1,398	19.5	2.96	1,038	294	58	56
S^k	49	14,796	1,804	93.8	8.08	1,522	330	154	149
$\bar{Q}^k$	50	14,668	1,915	96.4	8.44	1,963	341	155	161

need for the description of a polytope P the more storage is used by this set. As in the runs of Algorithm 2.3 using the hypercubes R^k, we restricted the maximal number of elements belonging to $\mathcal{P}$. By applying S^k or $\bar{Q}^k$ we use $\rho = \frac{1}{d}$ or $\rho = \frac{1}{\sqrt{(d-1)^2+1}}$ in the subdivision criterion (SDC). These numbers are smaller than $\frac{1}{\sqrt{d}}$, which is used for ρ in the case of R^k. Thus, we know that subdivisions are more rarely enforced and that the number of constraints describing an element of $\mathcal{P}$ and consequently the storage size of such an element can increase faster. For that reason we restricted the maximal number of polytopes in $\mathcal{P}$ to $20,000$, when using the regular d-simplex S^k or the polyhedron $\bar{Q}^k$ in Algorithm 2.3.

The numerical results presented in the Tables 2.5 and 2.6 definitely show that Algorithm 2.3 using the hypercubes R^k is more efficient than the same approach using S^k or $\bar{Q}^k$, at least with respect to our test problems. This seems to depend on the fact that by using R^k a bigger part of the current polytope P^k can be eliminated. Note that the volume of $R^k \subset \mathbb{R}^d$ is given by

$$V(R^k) = \left(\frac{\epsilon(z^k)2}{\sqrt{d}}\right)^d = \left(\frac{\epsilon(z^k)}{\sqrt{d}}\right)^d 2^d ,$$

TABLE 2.6. Comparison of the numerical effort for solving 2-dimensional all-quadratic problems with the accuracy $\epsilon = 0.01$

Q^k	NuP	ANuLP	MNuLP	ATime	MTime	ACol	MCol	ACon	MCon
$p = 1$									
R^k	49	56,341	2,841	197.5	6.60	5,555	768	74	71
S^k	40	45,923	2,600	372.4	13.5	2,632	458.5	167	170.5
$\bar{Q}^k$	40	28,680	3,501	261.6	17.3	3,178	807	174	174.5
$p = 2$									
R^k	49	14,699	1,007	49.0	2.11	1,473	260	64	55
S^k	48	26,494	2,209	221.6	8.56	2,010	343	168	144
$\bar{Q}^k$	48	25,035	2,645	149.8	10.94	2,578	491	191	150.5
$p = 3$									
R^k	50	11,920	1,752	35.2	3.51	1,954	401.5	64	62
S^k	48	22,820	2,930	177.2	12.0	2,441	393.5	160	165.5
$\bar{Q}^k$	48	18,217	2,088	135.9	9.54	2,764	457	163	173.5
$p = 4$									
R^k	50	17,351	2,725	53.8	6.10	2,505	532	75	72.5
S^k	47	25,994	3,801	207.7	16.2	2,805	554	192	195
$\bar{Q}^k$	47	22,226	4,382	191.7	25.5	3,114	809	201	213

whereas the volume of the regular d-simplex S^k is

$$V(S^k) \; = \; \frac{\sqrt{d+1}}{d!}\left(\epsilon(z^k)\sqrt{\frac{d+1}{d}}\right)^d \; = \; \left(\frac{\epsilon(z^k)}{\sqrt{d}}\right)^d \; \frac{(\sqrt{d+1})^{d+1}}{d!}$$

(see, e.g., [GKL95]). This implies that the volume of S^k is smaller than $V(R^k)$ and, moreover, that $V(S^k)$ is decreasing faster with respect to the dimension d than $V(R^k)$. The advantage of the larger volume of R^k seems to be greater than the disadvantage of the higher number of hyperplanes, which are necessary for describing R^k.

Whether the use of the regular simplex S^k or the use of the theoretically better polyhedron $\bar{Q}^k$ (see Theorem 2.6.5) leads to a more efficient approach cannot be answered definitely. Even though Algorithm 2.3 using $\bar{Q}^k$ was always faster on average – except for $p = 2$ and $\epsilon = 0.1$ – a comparison of the corresponding medians does not show a unique result. The same is true for the number of subproblems, which had to be solved during the execution of our method. Note that the average values as well as the medians were calculated with respect to the number of solved problems. Thus, these values are not directly comparable, when different numbers of problems were solved. For example, in the case $p = 4$ and $\epsilon = 0.1$ (see Table

2.5) we obtain for $\bar{Q}^k$ an average number of $11,743$ LP's and an average running-time of 76.13, considering only the 49 test problems, which were also solved with S^k.

With respect to the storage requirements we see that Algorithm 2.3 using S^k is a better solution scheme. We needed less polytopes P and we needed additionally less constraints for describing these sets. Note that by using S^k and the corresponding value for ρ the subdivision criterion (SDC) is more seldomly satisfied, such that less splittings of P^k are necessary. Note, furthermore, that by using $\bar{Q}^k$ the number of constraints determining a polytope P^k increases also if P^k is subdivided. By using R^k and S^k this number only grows, when the additional cuts are used (see Remark 2.6.1 and Remark 2.6.2).

Using S^k and $\bar{Q}^k$ the numerical results show again a high difference between the average values and the medians. The reason is the same as in the case of R^k. In at least 50% of the test problems both approaches showed an acceptable performance. However, there were numerical outliers, which destroyed the average performance of our algorithm. In view of the presented computational results we have to recognize that the use of the polyhedra $\bar{Q}^k$ did not have the expected success. The extra work for determining a better inner approximation polytope for the eliminable part of P^k did not result in a substantial improvement of the numerical performance of Algorithm 2.3. The easiest set, i.e., the hypercube R^k, showed the best numerical results.

Comparing the presented results with the numerical performance of the solution method for (QP), which we develop in the next chapter, Algorithm 2.3 is – even with R^k – not a good approach for solving all-quadratic problems. For an accuracy of $\epsilon = 0.01$ and quadratic problems of size $n = 2$ and $p = 4$ we needed on average 53.8 seconds. This bad performance boosted, if we tried to solve higher dimensional problems. In Figure 2.9 the numbers of the 3-dimensional all-quadratic problems are displayed, which could not be solved within the given storage capacities by Algorithm 2.3 using the three discussed possibilities for Q^k and the poor accuracy $\epsilon = 0.5$. The transformed unary problems had the dimension 9. Therefore, we reduced the maximal number of polytopes, which could belong to the set $\mathcal{P}$. When using R^k, we allowed $20,000$ elements. In the cases of S^k and $\bar{Q}^k$ we restricted this number to $10,000$. The corresponding minimal running-times, i.e., the fastest time after which Algorithm 2.3 was terminated since the storage capacity was exceeded, are given in seconds in Table 2.7. Considering this table it is not reasonable to increase the storage capacities in order to solve more problems.

FIGURE 2.9. Number of 3-dimensional all-quadratic test problems where Algorithm 2.3 exceeded the given storage capacity

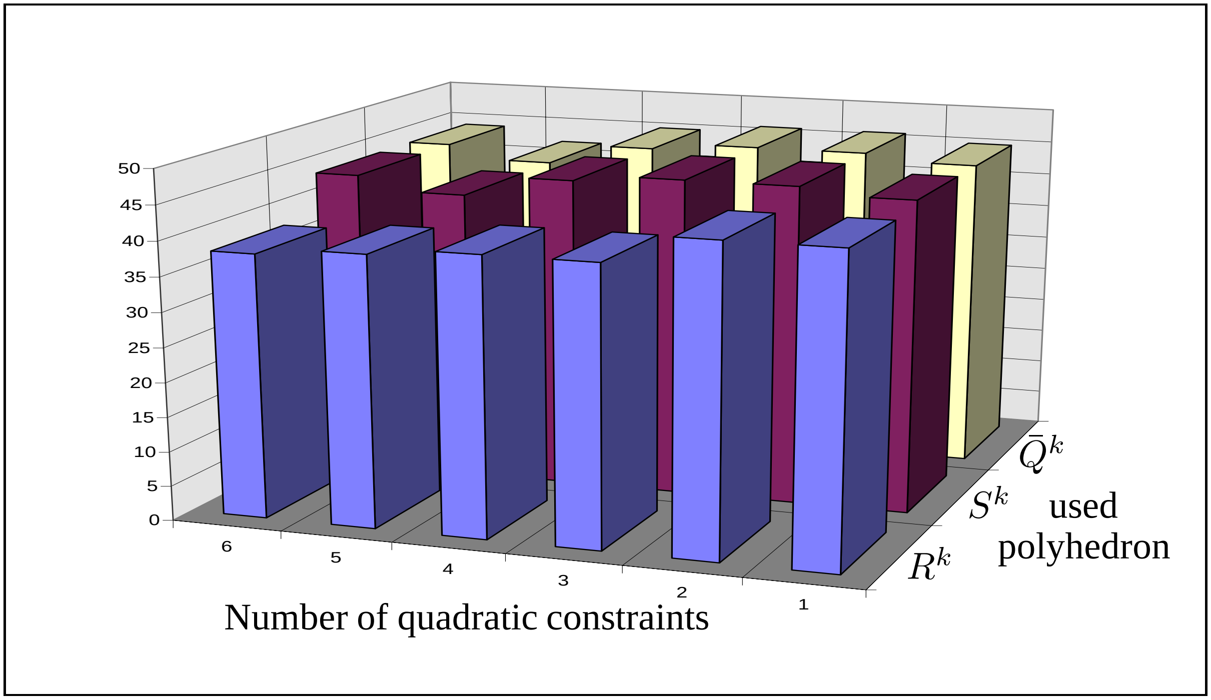


TABLE 2.7. Minimal running-times of unsolved 3-dimensional all-quadratic problems with $\epsilon = 0.5$

	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$	$p = 6$
R^k	125.9	122.2	139.9	147.2	303.8	433.7
S^k	503.6	473.1	440.4	431.8	549.4	586.9
$\bar{Q}^k$	280.1	334.5	301.2	321.7	348.3	414.5

A running-time of at least 2 minutes for one of the still unsolved 3-dimensional quadratic test problems is indeed not acceptable.

The last computational results demonstrate the, maybe, biggest disadvantage of the attempt to solve all-quadratic problems of type (QP) via unary problems. The transformation of the quadratic problems leads to an *explosion* of the dimension of the resulting (UP). Even for a 3-dimensional (QP) we obtain a 9-dimensional unary problem. If we recognize, furthermore, that the numerical applicability of general global optimization methods based on cutting planes or on branch-and-bound techniques is limited to problems in small spaces, it is not surprising that Algorithm 2.3 is not able to solve all-quadratic problems in dimensions higher than

3, at least that Algorithm 2.3 is not able to solve such problems with acceptable effort.

Algorithm 2.3 has still a lot of features, which could be changed. We could try to develop new valid cuts. We could use other values of ρ in (SDC) (see Remark 2.5.2) in order to change the number of subdivisions or instead we could look for other polyhedra. Nevertheless, in view of the previous considerations, it is unlikely that the solution of all-quadratic problems by using unary problems is a practicable way. In the next chapter we will see that a direct solution method for all-quadratic problems can have a significantly better numerical performance.

CHAPTER 3

A Simplicial Branch-and-Bound Method for Solving Nonconvex All-Quadratic Problems

In this chapter we will discuss a direct approach for solving nonconvex all-quadratic problems of type (QP). In the introduction (see Section 1.3) we pointed out that the most solution approaches for Problem (QP) proposed in the literature were developed for more general problem classes containing (QP) as a special instance. To the author's knowledge there is up to now only one approach considering directly the general nonconvex all-quadratic problem. This approach presented by Al-Khayyal et al. [AKLV95] is a rectangular branch-and-bound scheme.

The simplicial branch-and-bound method for solving (QP), which we will introduce and examine throughout the present chapter, use the same basic concepts as this rectangular scheme. This new solution method shows a significantly better computational performance than the indirect scheme presented in the foregoing chapter. Moreover, this simplicial branch-and-bound algorithm often also outperforms the rectangular approach by Al-Khayyal et al.

3.1. Introduction

As in the introduction of this thesis we define (using $c^0 = 0$), for each $l \in \{0, \ldots, p\}$ and $x \in \mathbb{R}^n$,

$$q^l(x) \; := \; x^T Q^l x + (d^l)^T x + c^l \; ,$$

such that (QP) can be written as

$$\begin{aligned} &\min \; q^0(x) \\ &q^l(x) \; \leq \; 0 \quad\;\; l = 1, \ldots, p \\ &\; x \; \in P \; . \end{aligned} \tag{QP}$$

Apart from the general assumptions for Problem (QP), like the symmetry of Q^l ($l = 0, \dots , p$) and the boundedness of P, we assume in this chapter that, for each $l \in \{0, \dots , p\}$, real $n \times n$ matrices C^l and D^l are known with the following properties

$$C^l \text{ is positive semidefinite,}$$

$$D^l \text{ is negative semidefinite}$$

and

$$Q^l \;=\; C^l + D^l \,.$$

If we denote by

$$\rho(B) \;=\; \max\{|\lambda| \,,\, \lambda \text{ eigenvalue of } B\}$$

the *spectral radius* of a real $n \times n$ matrix B, then it is easy to see that $C^l := \rho^l E$ and $D^l := Q^l - \rho^l E$ ($l \in \{0, \dots , p\}$) is a possible choice for these matrices, where E is the n-dimensional identity matrix and ρ^l is a real value not smaller than $\rho(Q^l)$. Note that matrix norms like the Frobenius norm (see Section 2.4 or [ZUR64]) are upper bounds for the spectral radius, and hence we can use such norms for the calculation of ρ^l ($l \in \{0, \dots , p\}$). Another possible way in order to obtain matrices C^l and D^l with the required properties is the spectral decomposition (see, e.g., [JRA93]).

As mentioned before, the simplicial branch-and-bound algorithm to be introduced in this chapter uses the same basic concepts as the rectangular approach proposed in [AKLV95]. For a given hyperrectangle Al-Khayyal et al. construct an LP-relaxation of (QP) by applying the known convex envelope [AKF83] of the two-dimensional bilinear function xy on a rectangle (for details we refer to [AKLV95], see also Subsection 1.3.4). The resulting relaxations are linear programs with $n + (p + 1)n$ variables and $4(p + 1)n + p + m$ constraints.

If an n-simplex is used instead of a hyperrectangle, it is possible to construct an LP-relaxation of (QP) with respect to this simplex having only n variables and $p+m+n+1$ constraints. How this can be done, is described in Section 3.2. Using this LP-relaxation of (QP) we derive in Section 3.3 a simplicial branch-and-bound method for solving (QP). This approach has the same theoretical properties as Al-Khayyal et al.'s rectangular scheme. In Section 3.4 we show that our method stops after a finite number of steps, if no feasible point exists. For the case $F \neq \emptyset$ the subsequent convergence theorem guarantees that each accumulation point of the point sequence generated by our approach is an optimal solution of Problem

(QP). By accepting approximate solutions for Problem (QP) this convergence result enables us to ensure finiteness of our simplicial branch-and-bound approach. We complete the examination of our new method in Section 3.5 by reporting on results on a computational comparison of our simplicial algorithm with the rectangular algorithm of Al-Khayyal et al. The content of the present chapter was published in [RAB98], except the numerical results and the new feature in Subsection 3.5.3.

3.2. A Linear Programming Relaxation over an n-Simplex

Let $S = [v_0, \dots, v_n] \subset \mathbb{R}^n$ be an n-simplex with the property that the intersection of this simplex with the polytope P of Problem (QP) is not empty. Consider now the all-quadratic problem (QP) with the additional constraint that each feasible point belongs to S, i.e., consider the problem

$$\begin{aligned} &\min q^0(x) \\ &q^l(x) \;\le\; 0 \quad l = 1, \dots, p \\ &x \;\in P \cap S \,. \end{aligned} \tag{QP^S}$$

Denote by W_S the $n \times n$ matrix with the columns $(v_i - v_0)$ $(i = 1, \dots, n)$ and let $B^n := \{\lambda \in \mathbb{R}^n_+ : \sum_{i=1}^n \lambda_i \le 1\}$ be a standard n-simplex. For each $x \in S$ there is a uniquely determined element $\lambda \in B^n$ such that x can be represented by

$$x \;=\; v_0 + W_S\lambda \,. \tag{3.2.1}$$

Using this substitution for $x \in S$ we can rewrite Problem (QP^S) as

$$\begin{aligned} &\min\, (W_S\lambda)^T Q^0 W_S\lambda + (d_S^0)^T W_S\lambda + c_S^0 \\ &(W_S\lambda)^T Q^l W_S\lambda + (d_S^l)^T W_S\lambda + c_S^l \;\le\; 0 \quad l = 1, \dots, p \\ &\qquad AW_S\lambda \;\le\; b - Av_0 \\ &\qquad\quad \lambda \;\in\; B^n \,, \end{aligned} \tag{$\overline{\text{QP}}^S$}$$

where, for $l \in \{0, \dots, p\}$,

$$d_S^l \;=\; d^l + 2Q^l v_0 \;\in\; \mathbb{R}^n$$

and

$$c_S^l \;=\; c^l + v_0^T Q^l v_0 + (d^l)^T v_0 \;\in\; \mathbb{R} \,.$$

In view of the properties of the matrices C^l and D^l ($l \in \{0, \ldots, p\}$) we know that, for each $l \in \{0, \ldots, p\}$, the function $\bar{q}^l_S : B^n \to \mathrm{I\!R}$

$$\bar{q}^l_S(\lambda) \;:=\; (W_S\lambda)^T Q^l W_S\lambda + (d^l_S)^T W_S\lambda + c^l_S$$

can be split into a convex and a concave part

$$\bar{q}^l_S(\lambda) \;=\; \underbrace{(W_S\lambda)^T D^l W_S\lambda + (d^l_S)^T W_S\lambda + c^l_S}_{\text{concave on } B^n} + \underbrace{(W_S\lambda)^T C^l W_S\lambda}_{\text{convex on } B^n} .$$

We are interested in an affine function $\bar{\ell}^l_S : B^n \to \mathrm{I\!R}$ ($l \in \{0, \ldots, p\}$), which underestimates $\bar{q}^l_S$ on the n-simplex B^n. As in the rectangular branch-and-bound algorithm in [AKLV95] we use the concept of the convex envelope. It is known (see Subsection 1.2.4 or [HPT95, Theorem 1.22]) that the convex envelope of a concave function g on an n-simplex S is the uniquely determined affine function, which coincides in the vertices of S with g. Therefore, we obtain, for each $l \in \{0, \ldots, p\}$, that the linear function $\varphi^l_S : B^n \to \mathrm{I\!R}$

$$\varphi^l_S(\lambda) \;:=\; \sum_{i=1}^{n} \lambda_i (v_i - v_0)^T D^l (v_i - v_0)$$

is the convex envelope of the concave function $(W_S\lambda)^T D^l W_S\lambda$ on the n-simplex B^n. Using the properties of the convex envelope (see Definition 1.2.3) and the positive semidefiniteness of the matrices C^l ($l = 0, \ldots, p$) it follows, for each $\lambda \in B^n$ and $l \in \{0, \ldots, p\}$, that

$$\begin{aligned} \bar{q}^l_S(\lambda) = \;& (W_S\lambda)^T D^l W_S\lambda + (d^l_S)^T W_S\lambda + c^l_S \; + (W_S\lambda)^T C^l W_S\lambda \\ \geq \;& \qquad \varphi^l_S(\lambda) \qquad + (d^l_S)^T W_S\lambda + c^l_S \; + \qquad 0 \qquad =: \; \bar{\ell}^l_S(\lambda) . \end{aligned}$$

I.e., neglecting the convex part of $\bar{q}^l_S$ and underestimating its concave part with the convex envelope we obtain the required affine function $\bar{\ell}^l_S$ ($l = 0, \ldots, p$). Using these affine underestimating functions we obtain an LP-relaxation of Problem ($\overline{\mathrm{QP}}^S$)

$$\begin{aligned} \min \; & \bar{\ell}^0_S(\lambda) \\ \bar{\ell}^l_S(\lambda) \; & \leq \; 0 \qquad l = 1, \ldots, p \\ AW_S\lambda \; & \leq b - Av_0 \\ \lambda \; & \in B^n . \end{aligned} \qquad (\overline{\mathrm{LP}}^S)$$

REMARK 3.2.1. If we do not omit the convex part of the functions $\bar{q}^l_S$ ($l = 0, \dots, p$), then we obtain, for each $l \in \{0, \dots, p\}$, with

$$\bar{g}^l_S(\lambda) \ := \ \varphi^l_S(\lambda) + (d^l_S)^T W_S \lambda + c^l_S + (W_S \lambda)^T C^l W_S \lambda$$

a convex quadratic function, which also underestimates $\bar{q}^l_S$ on the set B^n. The use of these functions would lead to a convex relaxation of Problem ($\overline{\text{QP}}^S$). Simplicial branch-and-bound algorithms using convex relaxations instead of LP-relaxations will be considered in Chapter 4.

The matrix W_S is regular, by construction. Using the resubstitution

$$\lambda \ = \ W_S^{-1}(x - v_0)$$

we see that Problem ($\overline{\text{LP}}^S$) is equivalent to

$$\begin{aligned} &\min\ \ell^0_S(x) \\ &\ell^l_S(x) \ \leq \ 0 \quad l = 1, \dots, p \\ &x \in P \cap S\,, \end{aligned} \tag{LP^S}$$

where, for each $l \in \{0, \dots, p\}$, the function $\ell^l_S : \mathrm{I\!R}^n \to \mathrm{I\!R}$

$$\ell^l_S(x) \ = \ \sum_{i=1}^n \left(W_S^{-1}(x - v_0)\right)_i (v_i - v_0)^T D^l (v_i - v_0) + (d^l_S)^T (x - v_0) + c^l_S$$

is the convex envelope of the concave quadratic function

$$q^l(x) - (x - v_0)^T C^l (x - v_0)\,.$$

Note that the convex envelope of the sum of an arbitrary function g and an affine function ℓ on a convex set C is just $\varphi + \ell$, where φ is the convex envelope of g with respect to the set C.

REMARK 3.2.2.

(a) From an implementational point of view the previous resubstitution is not reasonable. Problem ($\overline{\text{LP}}^S$) is easier to solve, since we do not need to calculate the inverse of W_S and the constraints describing B^n are explicitly given, whereas S is only described by its vertices. Therefore, in the implementation of the algorithm presented in Section 3.3 we used Problem ($\overline{\text{LP}}^S$) in order to determine a lower bound for the optimal value of (QP^S). Problem (LP^S), i.e., a formulation of the LP-relaxation of (QP^S) in the x-space, is only needed for the subsequent theoretical analysis.

(b) The LP-relaxation (LP^S) of (QP^S) is not uniquely determined, since it depends on the numbering of the vertices of the n-simplex S. Note that the function q^l and the affine underestimating function ℓ_S^l ($l \in \{0, \ldots, p\}$) coincide in the vertex v_0 of S.

(c) Let $\hat{S} = [\hat{v}_0, \ldots, \hat{v}_n]$ be an n-simplex contained in the n-simplex $S = [v_0, \ldots, v_n]$. It is a known fact [HPT95, Theorem 1.23] that the function values of the convex envelope $\varphi_{\hat{S}}$ of an arbitrary function g on the set $\hat{S}$ must be greater than or equal to the function values of the convex envelope φ_S of g with respect to the larger set S. If there holds $\hat{v}_0 = v_0$, then, for each $l \in \{0, \ldots, p\}$, we know that $\ell_{\hat{S}}^l$ and ℓ_S^l are convex envelopes of the function $q^l(x) - (x - v_0)^T C^l (x - v_0)$ and thus it follows, for each $x \in \hat{S}$,

$$\ell_{\hat{S}}^l(x) \;\geq\; \ell_S^l(x)\,. \tag{3.2.2}$$

In this case we know that the optimal value of (LP$^{\hat{S}}$) is not smaller than the optimal value of (LP^S). If the vertex $\hat{v}_0$ does not coincide with v_0, Relation (3.2.2) is no longer guaranteed, and we do not know how the optimal values of (LP$^{\hat{S}}$) and (LP^S) are related.

In order to prove the convergence of the simplicial branch-and-bound method introduced in the next section we will need a relation between the *size* of a given n-simplex S and the maximal distance between the function q^l and the underestimating function ℓ_S^l ($l \in \{0, \ldots, p\}$) on this simplex. The subsequent lemma shows that this maximal distance is bounded from above by a term depending on the diameter of the simplex S.

LEMMA 3.2.1. *Let $d^2(S)$ denote the squared diameter of the n-simplex $S = [v_0, \ldots, v_n]$, i.e., $d^2(S) = \max\{\|v_i - v_j\|_2^2 : i, j \in \{0, \ldots, n\}\}$, and let $\rho(C^l)$ and $\rho(D^l)$ be the spectral radius of C^l and D^l, respectively ($l \in \{0, \ldots, p\}$). Then, for each $l \in \{0, \ldots, p\}$, there holds*

$$\max_{x \in S} |q^l(x) - \ell_S^l(x)| \;\leq\; d^2(S)\left(\rho(C^l) + \rho(D^l)\right)\,. \tag{3.2.3}$$

PROOF: Choose an arbitrary, but fixed index $l \in \{0, \ldots, p\}$ and an arbitrary, but fixed element x of S. Then there exists a uniquely determined $\lambda^x \in B^n$ (see (3.2.1)) with $q^l(x) = \bar{q}_S^l(\lambda^x)$ and $\ell_S^l(x) = \bar{\ell}_S^l(\lambda^x)$. In Subsection 1.2.4 we pointed out that the concave envelope of a convex function on an n-simplex S is the uniquely defined affine function coinciding with this convex function in the vertices of S. Therefore, we know that the linear function $\psi_S^l : B^n \to \mathrm{I\!R}$, $\psi_S^l(\lambda) = \sum_{i=1}^n \lambda_i (v_i - v_0)^T C^l (v_i - v_0)$ is the concave envelope of $(W_S\lambda)^T C^l W_S \lambda$ on the n-simplex B^n, and hence there holds that ψ_S^l is an overestimator for $(W_S\lambda)^T C^l W_S \lambda$ on the set B^n. Using the negative semidefiniteness of D^l it follows

$$
\begin{aligned}
|q^l(x) - \ell_S^l(x)| &= \bar{q}_S^l(\lambda^x) - \bar{\ell}_S^l(\lambda^x) \\
&= \underbrace{(W_S\lambda^x)^T C^l W_S \lambda^x}_{\leq \psi_S^l(\lambda^x)} + \underbrace{(W_S\lambda^x)^T D^l W_S \lambda^x}_{\leq 0} - \varphi_S^l(\lambda^x) \\
&\leq \sum_{i=1}^n \lambda_i^x (v_i - v_0)^T (C^l - D^l)(v_i - v_0) \, .
\end{aligned}
$$

The spectral radius is a matrix norm on the space $\mathcal{S}_n$ of symmetric real $n \times n$ matrices. Moreover, this spectral radius norm is compatible with the Euclidean vector norm. Using these facts we, furthermore, obtain

$$
\begin{aligned}
|q^l(x) - \ell_S^l(x)| &\leq \sum_{i=1}^n \lambda_i^x \|v_i - v_0\|_2 \rho(C^l - D^l) \|v_i - v_0\|_2 \\
&\leq d^2(S) \rho(C^l - D^l) \underbrace{\sum_{i=1}^n \lambda_i^x}_{\leq 1} \\
&\leq d^2(S) \left(\rho(C^l) + \rho(D^l) \right) \, .
\end{aligned}
$$

Since x is an arbitrary element of S, Relation (3.2.3) follows readily. ■

As a direct consequence of this lemma we know that the maximal distance between q^l and ℓ_S^l ($l \in \{0, \ldots, p\}$) tends to 0, if the simplex S shrinks to a singleton. This is not surprising since q^l and ℓ_S^l coincide by construction at least in the vertex v_0 of S (see Remark 3.2.2(b)).

REMARK 3.2.3. If the matrices C^l and D^l ($l = 0, \ldots, p$) were constructed by a spectral decomposition of Q^l, then it is possible to prove that, for each $l \in \{0, \ldots, p\}$, there holds

$$\rho(C^l - D^l) \; = \; \rho(Q^l) \, .$$

In this special case we can replace, for each $l \in \{0, \ldots, p\}$, the right-hand side of (3.2.3) by $d^2(S)\rho(Q^l)$.

The simplicial branch-and-bound algorithm, which we present in the next section, will use the LP-relaxation (LP^S) of (QP^S) in order to calculate a lower bound for the optimal value of (QP) with respect to a given n-simplex S.

3.3. A Simplicial Branch-and-Bound Algorithm

In the introduction of this thesis (see especially Subsection 1.2.2) we pointed out that we need a relaxation $S^0 \supset F$ in order to start a branch-and-bound approach. Of course we would like to start with an n-simplex $S^0 \supset F$. Since we assumed that P is a non-empty full-dimensional polytope, we know that there always exists an n-simplex $S^0 \supset P$ (see, e.g., [HPT95, pages 145f.] for the construction of such sets), which we can use as a start relaxation of $F \subset P$.

In the previous section we have seen, how it is possible to calculate a lower bound $\mu(S)$ for the optimal value of (QP), at least if the feasible region of (QP) is additionally restricted to an n-simplex S. Upper bounds for the optimal value can be obtained as usual by considering feasible points $\bar{x} \in F$, which were generated, for example, during the solution of the LP-relaxation (LP^S). The function value of q^0 at each feasible point $\bar{x} \in F$ is obviously an upper bound for the optimal value of (QP).

Apart from the start relaxation $S^0 \supset F$ and the knowledge of the construction of lower and upper bounds with respect to the used subdivision sets, we need finally in order to formulate a branch-and-bound scheme (see again Subsection 1.2.2) a rule for refining a considered n-simplex. We use the so-called **bisection**, where an n-simplex S is split into two subsimplices S_1, $S_2 \subset S$ by a radial subdivision with respect to the midpoint of the longest edge of S, as we will see in the formulation of the algorithm (see also Definition 1.2.2). This subdivision rule was introduced in [HOR76] for branch-and-bound algorithms based on simplices and will ensure in connection with the result of Lemma 3.2.1 the convergence of the presented approach. The following algorithm is formulated according to the guidelines of a basic branch-and-bound scheme given in [HPT95, Algorithm 3.5].

ALGORITHM 3.1 (***Simplicial Branch-and-Bound Algorithm for (QP)***).

```
Initialization
    Determine an $n$-simplex $S^0 = [v_0^0, \ldots, v_n^0]$ with $S^0 \supset P$.
    $FLP_{S^0} \leftarrow \{x \in S^0 \cap P : \ell^l_{S^0}(x) \leq 0\,,\, l = 1, \ldots, p\}$
    If $FLP_{S^0} = \emptyset$ Then
        STOP ← True ($F = \emptyset$)
```

```
  Else
    Solve the linear optimization problem (LP) $\min_{x \in FLP_{S^0}} \ell^0_{S^0}(x)$. Let $\omega(S^0)$
    be an optimal solution and $\mu(S^0) = \ell^0_{S^0}(\omega(S^0))$ be the optimal value.
    $\mu^0 \leftarrow \mu(S^0)$, $\mathcal{P} \leftarrow \{S^0\}$
    If $\omega(S^0) \in F$ Then
      $Q \leftarrow \{\omega(S^0)\}$, $\eta^0 \leftarrow q^0(\omega(S^0))$, $x_f \leftarrow \omega(S^0)$
    Else
      $Q \leftarrow \emptyset$, $\eta^0 \leftarrow \infty$
    EndIf
    STOP $\leftarrow$ False, $k \leftarrow 0$
  EndIf

While STOP $=$ False Do
  If $\eta^k = \mu^k$ Then                                                    (SC)
    STOP $\leftarrow$ True ($x_f$ is an optimal solution of (QP))
  Else
    Determine indices $i_0$, $i_1 \in \{0, \dots, n\}$ satisfying
        $\|v^k_{i_0} - v^k_{i_1}\|_2^2 = \max_{i,j=0,\dots,n} \|v^k_i - v^k_j\|_2^2$
    and set
        $S^k_1 = [v^k_0, \dots, v^k_{i_0-1}, m^k, v^k_{i_0+1}, \dots, v^k_n]$,
        $S^k_2 = [v^k_0, \dots, v^k_{i_1-1}, m^k, v^k_{i_1+1}, \dots, v^k_n]$
    with $m^k = \frac{1}{2}(v^k_{i_0} + v^k_{i_1})$, i.e., split $S^k$ into $S^k_1$ and $S^k_2$ by bisection.
    For $j = 1$ To $2$ Do
      $FLP_{S^k_j} \leftarrow \{x \in S^k_j \cap P : \ell^l_{S^k_j}(x) \leq 0\,,\ l = 1, \dots, p\}$
      If $FLP_{S^k_j} \neq \emptyset$ Then
        Solve the LP $\min_{x \in FLP_{S^k_j}} \ell^0_{S^k_j}(x)$. Let $\omega(S^k_j)$ be an optimal solution
        and $\bar{\mu}(S^k_j) = \ell^0_{S^k_j}(\omega(S^k_j))$ be the optimal value.
        $\mu(S^k_j) \leftarrow \max\{\mu(S^k), \bar{\mu}(S^k_j)\}$                    (LBR)
        If $\omega(S^k_j) \in F$ Then $Q \leftarrow Q \cup \{\omega(S^k_j)\}$
        $\mathcal{P} \leftarrow \mathcal{P} \cup \{S^k_j\}$
      EndIf
    EndFor
```

```
        $\mathcal{P} \leftarrow \mathcal{P} \setminus \{S^k\}$
        If $Q \neq \emptyset$ Then
          $\eta^{k+1} \leftarrow \min_{x \in Q} q^0(x)$, choose $x_f \in Q$ with $\eta^{k+1} = q^0(x_f)$
        Else
          $\eta^{k+1} \leftarrow \eta^k$
        EndIf
        $\mathcal{P} \leftarrow \mathcal{P} \setminus \{S \in \mathcal{P} : \mu(S) \geq \eta^{k+1}\}$                    (PR)
        If $\mathcal{P} \neq \emptyset$ Then
          $\mu^{k+1} \leftarrow \min_{S \in \mathcal{P}} \mu(S)$, choose $S^{k+1} \in \mathcal{P}$ with $\mu^{k+1} = \mu(S^{k+1})$
        Else
          If $Q \neq \emptyset$ Then
            $\mu^{k+1} \leftarrow \eta^{k+1}$
          Else
            STOP $\leftarrow$ True ($F = \emptyset$)
          EndIf
        EndIf
        $k \leftarrow k + 1$
    EndIf
EndWhile
```

REMARK 3.3.1.

(a) We know by construction that $\mu(S)$ is a lower bound for the minimal value of q^0 on the set $F \cap S$. η^k ($k \in \mathbb{N}$) is constructed such that this value is an upper bound for q^0 on the whole feasible set F. Therefore, there holds that a simplex $S \in \mathcal{P}$ with the property $\mu(S) \geq \eta^{k+1}$ cannot contain a feasible point $\bar{x} \in F$ satisfying $q^0(\bar{x}) < q^0(x_f)$, and hence we can eliminate each of these simplices in the *pruning rule* (PR).

(b) The pruning rule (PR) can only be successful, if the set Q is not empty, since otherwise we would have $\eta^{k+1} = \infty > \mu(S)$ ($S \in \mathcal{P}$). Note that it is possible that after a finite number of steps Algorithm 3.1 never detects a feasible point, what means that Q could always be empty.

(c) If the partition $\mathcal{P}$ is empty after the execution of (PR) and if Q is not empty, then it is obvious that the upper bound $\eta^{k+1} < \infty$ is also a lower bound for the optimal value of (QP).

(d) Because of the formulation of the pruning rule (PR) with "$\geq$" instead of "$>$" there holds at the beginning of iteration k, for each $S \in \mathcal{P}$, $\mu(S) < \eta^k$ and hence $\mu^k < \eta^k$. This implies that the stopping criterion (SC) can only be fulfilled in iteration $k \geq 2$, if $\mathcal{P}$ is empty and Q is not empty at the end of iteration $k-1$.

(e) In view of Remark 3.2.2(c) we do not know whether the optimal value $\bar{\mu}(S_j^k) = \ell^0_{S_j^k}(\omega(S_j^k))$ ($k \in \mathbb{N}$; $j = 1, 2$) of Problem (LP$^{S_j^k}$) is in each case not smaller than the lower bound $\mu(S^k)$. However, by setting

$$\mu(S_j^k) \;=\; \max\{\, \mu(S^k)\;,\; \bar{\mu}(S_j^k)\,\}$$

in the lower bounding rule (LBR) in Algorithm 3.1 we obtain a value, which is of course also a lower bound for (QP$^{S_j^k}$) ($k \in \mathbb{N}$; $j = 1, 2$). Moreover, these values satisfy, for each $k \in \mathbb{N}$,

$$\min\{\, \mu(S_1^k)\;,\; \mu(S_2^k)\,\} \;\geq\; \mu(S^k)\;. \tag{3.3.1}$$

This guarantees that the sequence $\{\mu^k\}_{k\in\mathbb{N}}$ is non-decreasing.

The polytopes FLP_S are relaxations of the portion of the feasible set F of (QP) contained in the simplex S. Algorithm 3.1 can stop by detecting the emptiness of F only, if all considered simplices S lead to empty relaxations FLP_S. Thus we know that F is really empty in this case, since we start with an n-simplex $S^0 \supset F$. The construction of μ^k ($k \in \mathbb{N}$) as the minimal value of the lower bounds $\mu(S)$ of all n-simplices $S \in \mathcal{P}$, which were not pruned till iteration $k-1$, guarantees that this value is a lower bound for the optimal value of q^0 with respect to the whole feasible region F. If Algorithm 3.1 stops after a finite number of steps with a solution x_f, we obtain hence

$$q^0(x_f) \;=\; \eta^k \;=\; \mu^k \;\leq\; \min_{x\in F} q^0(x) \;\leq\; q^0(x_f)\,,$$

showing the optimality of x_f for Problem (QP). It follows that Algorithm 3.1 is well defined, as long as this approach terminates after a finite number of iterations. The proof of the correctness of our method in the infinite case is the content of the next section.

3.4. Convergence

In Algorithm 3.1 we used bisection as a subdivision rule for the current simplex S^k at iteration $k \in \mathbb{N}$. This rule has the property that, for each infinite nested sequence $\{S^q\}_{q\in\mathbb{N}}$ of simplices generated by using this rule, there holds

$$d^2(S^q) \;\to\; 0 \qquad (q\to\infty) \tag{3.4.1}$$

(see, e.g., [HOR76, KEA78]). This special property of the bisection in connection with the result of Lemma 3.2.1 enables us to prove that in the infinite case each accumulation point of the sequence $\{\omega(S^k)\}_{k\in\mathbb{N}}$ generated by Algorithm 3.1 is an optimal solution of Problem (QP). This will be the result of the Convergence Theorem 3.4.2. At first, however, we need an additional lemma in order to establish this convergence result. In this lemma we show that the feasible region F of (QP) cannot be empty, if Algorithm 3.1 does not stop after a finite number of iterations.

LEMMA 3.4.1. *Algorithm 3.1 stops after a finite number of iterations, if no feasible point for Problem (QP) exists, i.e., if $F=\emptyset$.*

PROOF: Assume that F is empty and define the function $\hat{F}:\mathbb{R}^n\to\mathbb{R}$ by

$$\hat{F}(x) \;:=\; \max_{l=1,\dots,p} q^l(x)\,.$$

$\hat{F}$ is a continuous function and hence attains its minimum over the compact set P. Since $F=\{x\in P:\hat{F}(x)\le 0\}$ is empty we know that there exists a positive real value δ satisfying

$$\min_{x\in P}\hat{F}(x) \;\ge\; \delta\,. \tag{3.4.2}$$

Assume now, by contradiction, that Algorithm 3.1 generates an infinite sequence $\{S^k\}_{k\in\mathbb{N}}$ of n-simplices. It follows that there must exist an infinite subsequence $\{S^{k_q}\}_{q\in\mathbb{N}}$ of $\{S^k\}_{k\in\mathbb{N}}$ with the properties that, for each $q\in\mathbb{N}$, there holds

$$S^{k_{q+1}} \;\subset\; S^{k_q} \tag{3.4.3}$$

and

$$FLP_{S^{k_q}} \;\neq\; \emptyset\,.$$

In view of Property (3.4.1) of the bisection, we obtain from (3.4.3)

$$d^2(S^{k_q}) \;\to\; 0 \qquad (q\to\infty)\,. \tag{3.4.4}$$

Choose a real value $\bar{\delta}$ with

$$0 \;<\; \bar{\delta} \;<\; \delta\frac{1}{\max\limits_{l=1,\dots,p}(\rho(C^l)+\rho(D^l))}\,.$$

From (3.4.4) we see that there must be an index $q_0 \in \mathbb{N}$ such that, for each $q \geq q_0$, there holds

$$d^2(S^{k_q}) \;\leq\; \bar{\delta}\,.$$

Due to Lemma 3.2.1 we hence obtain, for each $q \geq q_0$, $l \in \{1, \ldots, p\}$ and $x \in FLP_{S^{k_q}}$,

$$\begin{aligned} q^l(x) \;&=\; q^l(x) - \ell^l_{S^{k_q}}(x) + \underbrace{\ell^l_{S^{k_q}}(x)}_{\leq\, 0} \\ &\leq\; d^2(S^{k_q})\left(\rho(C^l) + \rho(D^l)\right) \;\leq\; \bar{\delta}\left(\rho(C^l) + \rho(D^l)\right) \;<\; \delta\,. \end{aligned}$$

We know that $FLP_{S^{k_q}}$ ($q \geq q_0$) is not empty and, moreover, that each element of this set belongs to P. Thus, from the previous relation it follows, for each $q \geq q_0$ and $x \in FLP_{S^{k_q}}$,

$$\hat{F}(x) \;<\; \delta\,,$$

contradicting – in view of (3.4.2) – the emptiness assumption for F. ■

If Algorithm 3.1 does not stop after a finite number of iterations, then we know in view of the previous lemma that the feasible region F of (QP) is not empty and hence that a finite optimal value of Problem (QP) exists. With this result we are now able to prove the convergence result mentioned before.

THEOREM 3.4.2. *If Algorithm 3.1 generates an infinite sequence $\{S^k\}_{k\in\mathbb{N}}$ of simplices, then every accumulation point $\omega^\star$ of the corresponding point sequence $\{\omega(S^k)\}_{k\in\mathbb{N}}$ is an optimal solution of Problem (QP).*

PROOF: Due to Lemma 3.4.1 we know that there exists an optimal solution $x^\star$ of Problem (QP) with optimal value $q^0(x^\star)$. Since the current simplex S^k ($k \in \mathbb{N}$) is chosen such that $\mu^k = \mu(S^k)$ holds, and since we know that μ^k ($k \in \mathbb{N}$) is by construction a lower bound for $q^0(x)$ ($x \in F$) and, moreover, that $\{\mu^k\}_{k\in\mathbb{N}}$ is a non-decreasing sequence (see Remark 3.3.1(e)), there holds that the non-decreasing sequence $\{\mu(S^k)\}_{k\in\mathbb{N}}$ is bounded from above by $q^0(x^\star)$, and hence convergent.

Let $\omega^\star$ be an accumulation point of $\{\omega(S^k)\}_{k\in\mathbb{N}}$ and let $\{\omega(S^{k_q})\}_{q\in\mathbb{N}}$ be a subsequence converging to $\omega^\star$. By passing to a further subsequence, if necessary, we can assume that the corresponding simplex sequence $\{S^{k_q}\}_{q\in\mathbb{N}}$ is decreasing. At first we prove that $\omega^\star$ is a feasible point of (QP).

Taking the result of Lemma 3.2.1 into account it follows from (3.4.1), for each $l \in \{1, \dots, p\}$,

$$\begin{aligned} 0 \ \le \ & q^l(\omega(S^{k_q})) - \ell^l_{S^{k_q}}(\omega(S^{k_q})) \\ & \le \ d^2(S^{k_q})\left(\rho(C^l) + \rho(D^l)\right) \ \to \ 0 \quad (q \to \infty) \, . \end{aligned} \tag{3.4.5}$$

Note that, for each $q \in \mathbb{N}$ and $l \in \{1, \dots, p\}$, $\ell^l_{S^{k_q}}$ is an underestimating function for q^l on the set S^{k_q} and that $\omega(S^{k_q})$ is an element of S^{k_q}. Since the functions q^l ($l = 1, \dots, p$) are continuous we obtain from (3.4.5), for each $l \in \{1, \dots, p\}$,

$$0 \ \ge \ \ell^l_{S^{k_q}}(\omega(S^{k_q})) \ \to \ q^l(\omega^\star) \qquad (q \to \infty) \, ,$$

showing the feasibility of $\omega^\star$, i.e., $\omega^\star \in F$.

Relation (3.4.5) is obviously fulfilled also for the functions q^0 and $\ell^0_{S^{k_q}}$ ($q \in \mathbb{N}$). By continuity of q^0 and the mentioned boundedness of $\{\mu(S^k)\}_{k\in\mathbb{N}}$ it follows

$$q^0(x^\star) \ \ge \ \mu(S^{k_q}) \ \ge \ \ell^0_{S^{k_q}}(\omega(S^{k_q})) \ \to \ q^0(\omega^\star) \quad (q \to \infty) \, . \tag{3.4.6}$$

This implies with respect to the feasibility of $\omega^\star$ that

$$q^0(\omega^\star) \ \le \ q^0(x^\star) \ = \ \min_{x \in F} q^0(x) \ \le \ q^0(\omega^\star) \, ,$$

and hence $q^0(\omega^\star) = q^0(x^\star)$, which proves the optimality of $\omega^\star$. ■

REMARK 3.4.1.

(a) Property (3.4.1) of the bisection is essential for the proof of Lemma 3.4.1 as well as for the proof of the previous convergence theorem. Therefore, each subdivision rule, which has this property, can be used in Algorithm 3.1 without altering the theoretical properties of this approach. Subdivision rules satisfying (3.4.1) belong to the class of so-called *exhaustive* subdivision rules (see Definition 4.3.1), which will be considered in more detail in the next chapter (see, in particular, Section 4.3).

(b) In order to guarantee the convergence of Algorithm 3.1 in the sense of Theorem 3.4.2 we have not proved that for an infinite decreasing sequence $\{S^{k_q}\}_{q\in\mathbb{N}}$ of simplices there holds

$$\eta^{k_q} - \mu^{k_q} \ \to \ 0 \qquad (q \to \infty) \, .$$

> Therefore, the used bounding procedure in Algorithm 3.1 does not belong to the class of so-called *consistent* bounding operations (see [HT96B, Section 4.2]). Hence the general convergence theory for branch-and-bound methods proposed, for example, in [HPT95, HT96B] is not applicable. Note that we are only able to prove the convergence of the sequence $\{\mu^k\}_{k\in\mathbb{N}}$ of lower bounds towards the optimal value of (QP). We do not know, how the sequence $\{\eta^k\}_{k\in\mathbb{N}}$ of upper bounds behave.

Similar to the case of Algorithms 2.2 and 2.3 in the previous chapter we cannot expect that Algorithm 3.1 detects in a finite number of steps an optimal solution of Problem (QP). However, for the applicability of a solution method for (QP) in practice we need a finite approach. The finiteness of Algorithm 3.1 can be achieved, if we are satisfied with an approximate solution, where approximate solution is meant in the sense of feasibility as well as of optimality. Let ϵ, $\delta > 0$ be two prespecified tolerances. If we add in Algorithm 3.1 each solution $\omega(S)$ of a linear subproblem satisfying, for each $l \in \{1, \ldots, p\}$,

$$q^l(\omega(S)) \;\le\; \delta\,, \tag{3.4.7}$$

to the set Q, then we obtain a finite method by replacing the stopping criterion (SC) with

$$\textbf{If} \quad \eta^k - \mu^k \;\le\; \epsilon \quad \textbf{Then} \;\; \text{STOP} \leftarrow \; \textbf{True}\,. \tag{$\overline{\text{SC}}$}$$

Indeed, in view of Lemma 3.4.1 we know that Algorithm 3.1 is always finite, if F is empty. If the feasible region is not empty, then we have seen in the proof of Theorem 3.4.2 that this method generates a point sequence $\{\omega(S^q)\}_{q\in\mathbb{N}}$ converging to an optimal solution $\omega^\star$ of (QP). Since this optimal solution is feasible, we know by continuity of the quadratic functions q^l ($l \in \{1, \ldots, p\}$) that there is an index $q_0 \in \mathbb{N}$ such that, for each $q \ge q_0$,

$$q^l(\omega(S^q)) \;\le\; \delta \qquad l = 1, \ldots, p\,.$$

This means that $\omega(S^q)$ ($q \ge q_0$) is added to the set Q and hence used for updating the upper bounds η^q ($q \ge q_0$). It follows, for each $q \ge q_0$,

$$\mu^q \;=\; \mu(S^q) \;=\; \ell^0_{S^q}(\omega(S^q)) \;\le\; q^0(\omega(S^q))$$

and

$$\eta^q \;\le\; q^0(\omega(S^q))\,.$$

This implies – in view of (3.4.6) – that the stopping criterion ($\overline{\text{SC}}$) must be satisfied after a finite number of steps.

REMARK 3.4.2.

(a) If points $x \in P$ satisfying (3.4.7) are added to the set Q, then there does not hold anymore that η^k ($k \in \mathbb{N}$) is an upper bound for the optimal value of (QP). We only know that η^k ($k \in \mathbb{N}$) is an upper bound for the function values of q^0 on the set

$$F_\delta \;=\; \{x \in P : q^l(x) \leq \delta\,,\, l = 1, \ldots, p\}\,.$$

(b) If Algorithm 3.1 using the stopping criterion ($\overline{\text{SC}}$) terminates at iteration k with a solution $x_f \in Q$, we obtain a point satisfying

$$q^l(x_f) \;\leq\; \delta \qquad l = 1, \ldots, p$$

and

$$q^0(x_f) - \mu^k \;\leq\; \epsilon \qquad \Leftrightarrow \qquad q^0(x_f) - \epsilon \;\leq\; \mu^k\,.$$

We do not know anything about the optimality of this point. Note that it is even possible that there holds $F = \emptyset$. We only know that $q^0(x_f) - \epsilon$ is a lower bound for the optimal value of (QP), which is by convention ∞ in the empty case. Only in the case that x_f is additionally feasible, we obtain also the ϵ-optimality of this point in the sense that the optimal value of (QP) and $q^0(x_f)$ have a distance not bigger than ϵ.

(c) The used concept of approximate feasible points and approximate optimal solutions will be discussed in more detail in the next chapter, where we examine a generalization of Algorithm 3.1.

We complete the discussion of Algorithm 3.1 with an examination of its numerical performance. In the next section we will demonstrate the better performance of our simplicial branch-and-bound method in comparison with the performance of the rectangular method by Al-Khayyal et al. [AKLV95].

3.5. Computational Results

The presented simplicial branch-and-bound Algorithm 3.1 and the rectangular algorithm of Al-Khayyal et al. were encoded in C++. As in the implementation of Algorithm 2.3 (see Section 2.8) the partition sets, which had to be stored in $\mathcal{P}$, were managed by AVL-trees. In order to test and to compare the computational performance of both algorithms we used the set of randomly generated test examples introduced in Section 1.5. Before presenting the numerical results we give some notes on the implementation of both methods.

3.5.1. Implementational Details. The implementation of the rectangular algorithm followed closely the formulation given in [AKLV95]. As a subdivision rule for a considered hyperrectangle R^k ($k \in \mathbb{N}$) we used the special rule given in [AKV96], which was also used in the numerical tests in [AKLV95]. In this rule the hyperrectangle $R^k = \{x \in \mathbb{R}^n : l^k \leq x \leq L^k\}$ ($l^k, L^k \in \mathbb{R}^n$) is subdivided into two hyperrectangles R_1^k and R_2^k in the following way. Let $\omega(R^k)$ be a solution of the LP-relaxation used by Al-Khayyal et al. with respect to the hyperrectangle R^k (see [AKLV95] for details). Let $i_0 \in \{1, \dots, n\}$ be an index, where the following maximum

$$\max_{i=1,\dots,n} \frac{\max\{\omega(R^k)_i - l_i^k\,,\, L_i^k - \omega(R^k)_i\}}{\max\{1.0\,,\, L_i^k - l_i^k\}}$$

is attained. Then the new hyperrectangles are given by

$$R_1^k \;=\; \{x \in R^k : l_{i_0}^k \;\leq\; x_{i_0} \;\leq\; \frac{l_{i_0}^k + L_{i_0}^k}{2}\}$$

and

$$R_2^k \;=\; \{x \in R^k : \frac{l_{i_0}^k + L_{i_0}^k}{2} \;\leq\; x_{i_0} \;\leq\; L_{i_0}^k\}\;.$$

REMARK 3.5.1. We also tested *bisection* in the rectangular algorithm, where the above index $i_0 \in \{1, \dots, n\}$ is chosen such that

$$L_{i_0}^k - l_{i_0}^k \;=\; \max_{i=1,\dots,n} \left[L_i^k - l_i^k\right]$$

holds. The average numerical performance in our computational tests was nearly the same. Therefore, we restrict the subsequent presentation of the numerical results to those obtained by using the described special subdivision rule, and not by using bisection.

The construction of the necessary initial set $S^0 \supset P$, respectively $R^0 \supset P$, was done according to the following specifications. In the case of the rectangular algorithm we obtained a hyperrectangle $R^0 = \{x \in \mathbb{R}^n : l^0 \leq L^0\}$ by solving the $2n$ linear programs

$$l_i^0 \;:=\; \min_{x \in P} x_i \quad , \quad L_i^0 \;:=\; \max_{x \in P} x_i \qquad i = 1, \dots, n\,. \tag{3.5.1}$$

In order to construct an initial simplex S^0 we used one of the possibilities described in [HPT95, pages 145f.]. Note that the test examples were generated such that the polytope P is full-dimensional (see again Section 1.5). Let $v_0 \in \mathbb{R}^n$ be a vertex solution of one of the $2n$ linear problems in (3.5.1). Let, furthermore, $\{a_{i_1}, \dots, a_{i_n}\}$

be a linear independent subset of $\{a_i : i \in \{1,\dots,m\}$ and $a_i^T v_0 = b_i\}$, i.e., a subset of the set of constraints describing P, which are binding at v_0. If $\bar{\gamma}$ is the optimal value of

$$\max_{x\in P}\left(-\sum_{j=1}^{n} a_{i_j}^T x\right) ,$$

then it is provable that

$$S^0 \;=\; \{x \in \mathrm{I\!R}^n : a_{i_j}^T x \;\le\; b_{i_j} \;,\; j=1,\dots,n \;,\; -\sum_{j=1}^{n} a_{i_j}^T x \;\le\; \bar{\gamma}\}$$

is an n-simplex satisfying $S^0 \supset P$. In the implementation of Algorithm 3.1 we needed the vertices of S^0. These could be obtained by solving n linear equation systems. From (3.5.1) we had $2n$ possibilities in order to generate simplices $S^0 \supset P$. We constructed with each nondegenerate vertex solution of a problem in (3.5.1) a simplex in the described way and chose among these up to $2n$ possibilities the one with the smallest diameter.

The necessary positive semidefinite matrices C^l and the negative semidefinite matrices D^l with $Q^l = C^l + D^l$ $(l = 0,\dots,p)$ for our simplicial branch-and-bound method were determined by spectral decomposition. The eigenvalues and the eigenvectors of each matrix Q^l $(l \in \{0,\dots,p\})$ were calculated by applying an appropriate routine from the *NAG*-library.

In the implementation of Algorithm 3.1 we added the subsequent cheap test in order to decide whether $F \cap S = \emptyset$ holds for a given n-simplex $S = [v_0,\dots,v_n]$

$$\max_{l=1,\dots,p}\;\min_{i=1,\dots,n}(v_i - v_0)^T D^l (v_i - v_0) + (d_S^l)^T (v_i - v_0) + c_S^l \;>\; 0 \qquad (3.5.2)$$
$$\Rightarrow \quad F \cap S \;=\; \emptyset \,.$$

By using the fact that a concave function attains its minimum on a polytope in a vertex of this polytope [HPT95, Theorem 1.19], it is easy to verify that the left-hand side of (3.5.2) is a lower bound for $\max_{l=1,\dots,p} q^l(x)$ on the simplex S.

In both algorithms we have to solve linear subproblems. Since the LP-relaxations in Al-Khayyal et al.'s approach have a sparse structure we applied *MINOS 5.4* for solving these subproblems. This code is able to exploit sparsity. The LP-relaxation $(\overline{\mathrm{LP}}^S)$ used in Algorithm 3.1 has a dense constraint matrix. Even though the application of a code exploiting sparse structure for solving the linear subproblems in Algorithm 3.1 leads thus to unnecessary effort, we decided to use also in this approach the *MINOS 5.4* code. By doing this we could guarantee that in both algorithms the linear subproblems were solved with the same linear optimization algorithm.

REMARK 3.5.2.

(a) We also tested both algorithms using the LP-subroutine *E04NFF* of the *NAG*-library. This code is not able to manage sparsity. As it was to be expected, the running-times of Algorithm 3.1 decreased (see also the numerical results in Subsection 4.6.1), whereas the running-times of the rectangular method increased significantly, especially for problems with higher dimensions and higher number of quadratic constraints.

(b) As noted in Remark 3.2.1 it is possible to construct a convex relaxation of the restricted Problem (QP^S). We implemented a variant of Algorithm 3.1 using these convex relaxations, where the subproblems were solved with the *MINOS 5.4* convex solver. Even though the necessary number of subproblems, which had to be considered, decreased, the running-times increased so much that the version with linear subproblems was substantially faster. The *MINOS 5.4* convex solver use a *projected augmented Lagrangian algorithm*. In the computational results in the next chapter (see again Subsection 4.6.1) we will see that the use of convex subproblems can also lead to decreasing running-times, if another code is used for solving the convex quadratic relaxations, which is at least for our test problem more efficient.

In both algorithms the branching is stopped, if the *relative* difference between η^k and μ^k ($k \in \mathbb{N}$) is smaller than the tolerance value $\epsilon = 10^{-4}$, i.e., if there holds

$$\eta^k - \mu^k \ \le\ \epsilon \max\{1.0\,,\, |\eta^k|\} \tag{$\overline{\overline{\text{SC}}}$}$$

(compare with the stopping criterion ($\overline{\text{SC}}$)). Note that the rectangular algorithm by Al-Khayyal et al. generates sequences $\{\mu^k\}_{k\in\mathbb{N}}$ and $\{\eta^k\}_{k\in\mathbb{N}}$ with the same properties as the corresponding sequences in Algorithm 3.1 such that the above stopping criterion is also applicable in this approach. As mentioned at the end of the previous section, we have to be satisfied with approximate feasible points in order to obtain a finite method. In both algorithms each generated point satisfying (3.4.7) with an accuracy $\delta = 10^{-8}$ was interpreted as feasible and hence used for updating η^k ($k \in \mathbb{N}$). In the application of the *MINOS 5.4* code we chose the accuracy 10^{-9}.

3.5.2. Numerical Comparison. Tables 3.1 and 3.2 show some numerical results for the solved test problems. We use the abbreviation NuP S<R for the number of problems, where the simplicial algorithm was faster with respect to the running-time than the rectangular one. Note (see Section 1.5) that there are 50 test problems for each pair (n, p) of the dimension n and the number of quadratic constraints p. The abbreviation AvgNuLP is used for the average number of linear subproblems solved for each test problem with the simplicial Algorithm 3.1 (S) or the rectangular algorithm (R). StdLP stands for the standard deviation of the number of linear subproblems. In the column AvgTime the average running-times in seconds are displayed and the column StdTime shows the corresponding values of the standard deviation. Finally, the abbreviation Su is used for the speedup

TABLE 3.1. All test results for $n = 2, 3, 4$

p	NuP	AvgNuLP		StdLP		AvgTime		**Su**	StdTime	
	S<R	S	R	S	R	S	R		S	R
$n = 2$										
1	18	47.2	24.4	20.9	20.7	0.27	0.23	**0.84**	0.09	0.14
2	32	45.8	29.3	21.5	21.5	0.27	0.32	**1.18**	0.10	0.20
3	45	77.8	52.0	45.8	23.8	0.40	0.62	**1.54**	0.19	0.28
4	46	69.5	48.6	35.1	18.5	0.39	0.68	**1.74**	0.15	0.25
$n = 3$										
1	18	129.6	56.9	88.1	32.2	0.78	0.62	**0.80**	0.50	0.31
2	36	163.1	75.4	200.8	55.8	0.96	1.12	**1.17**	1.07	0.86
3	43	215.8	99.0	199.7	63.5	1.30	1.96	**1.51**	1.12	1.28
4	48	158.9	83.9	83.4	29.7	0.95	2.05	**2.17**	0.44	0.73
5	47	181.0	90.8	101.0	30.5	1.19	2.77	**2.34**	0.61	1.07
6	50	195.9	98.4	93.8	27.4	1.27	4.02	**3.17**	0.57	1.48
$n = 4$										
1	15	333.0	104.8	371.3	96.3	2.19	1.51	**0.69**	2.25	1.35
2	35	364.6	109.4	383.6	73.2	2.48	2.38	**0.96**	2.51	1.61
3	43	354.4	129.2	456.2	71.8	2.38	4.19	**1.69**	2.84	2.70
4	46	652.5	195.2	973.5	172.3	4.87	8.06	**1.65**	6.63	7.48
5	48	376.7	141.2	271.2	49.8	2.83	7.52	**2.66**	1.90	3.61
6	49	750.5	201.5	1,644	194.9	6.26	13.6	**2.17**	15.2	12.8
7	50	470.3	185.7	352.9	94.7	3.83	14.9	**3.89**	2.79	8.88
8	50	431.6	156.7	406.2	69.2	3.68	14.6	**3.96**	3.28	8.16

TABLE 3.2. Some test results for $n = 5, 6, 7, 8$

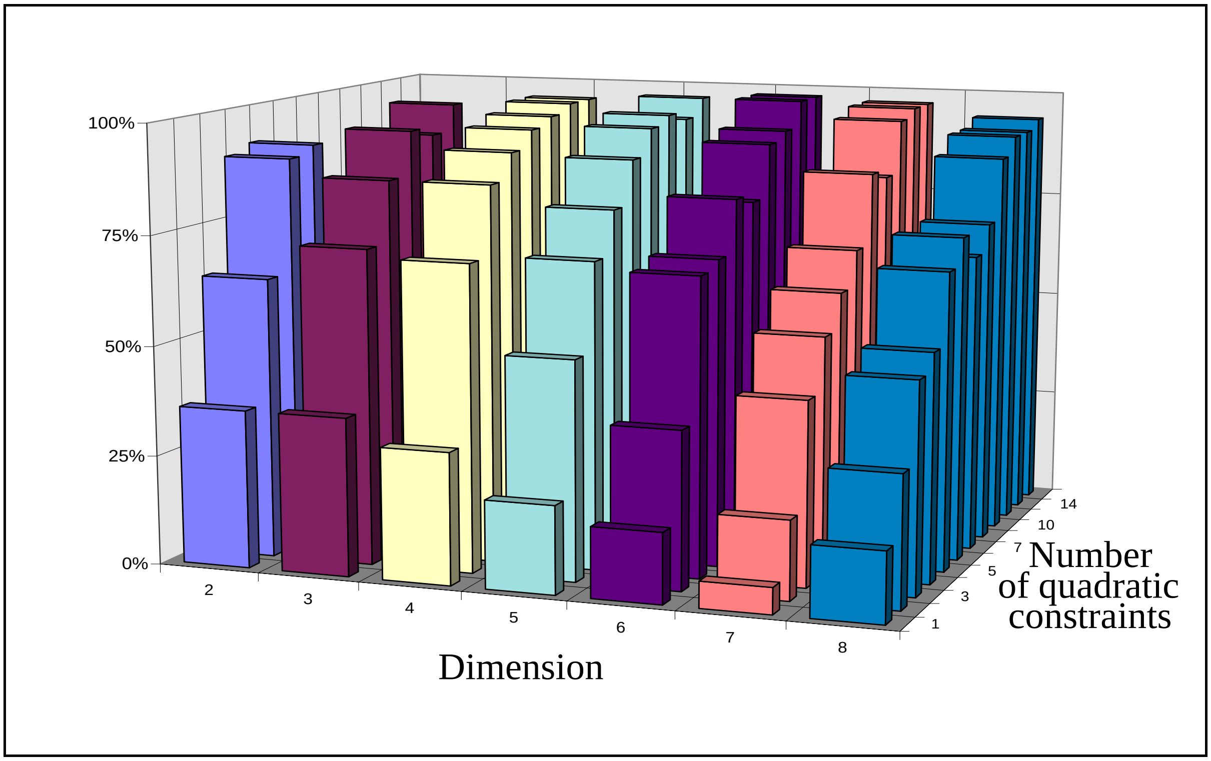
p	NuP	AvgNuLP		StdLP		AvgTime		**Su**	StdTime	
	S<R	S	R	S	R	S	R		S	R
$n = 5$										
2	25	955.6	185.6	1,451	151.6	8.32	5.96	**0.72**	11.4	5.02
4	40	1,110	224.8	1,492	192.9	10.6	14.7	**1.38**	13.7	14.2
6	48	1,070	267.3	1,103	192.7	10.7	28.0	**2.62**	10.5	22.5
8	48	1,386	312.8	1,430	246.5	17.4	50.6	**2.91**	21.2	37.4
10	50	905.8	250.9	928.3	123.5	11.05	52.2	**4.72**	10.1	23.6
$n = 6$										
2	18	2,623	325.2	2,808	238.1	28.7	16.2	**0.57**	32.0	12.7
4	35	5,425	394.4	14,913	371.1	58.2	40.1	**0.69**	147.3	42.7
6	40	5,421	505.4	12,106	558.1	69.6	83.9	**1.20**	154.8	101.7
8	47	4,366	483.0	6,286	329.2	61.5	115.2	**1.87**	82.2	83.0
10	50	2,680	450.8	3,185	300.4	41.2	161.2	**3.92**	48.3	108.3
12	50	3,649	489.2	3,943	256.3	60.6	219.6	**3.62**	63.9	122.4
$n = 7$										
2	9	11,710	521.0	26,545	465.9	46.9	10.7	**0.23**	111.0	9.41
4	27	13,039	634.4	39,539	958.0	55.0	27.0	**0.49**	165.5	41.1
6	35	7,233	526.6	9,164	346.0	35.8	39.3	**1.10**	46.1	27.7
8	42	9,901	714.7	12,297	456.4	52.3	80.3	**1.53**	63.5	55.3
10	48	9,280	682.7	11,811	393.6	54.0	104.5	**1.93**	67.4	57.5
12	49	8,902	686.2	11,321	497.8	58.4	141.9	**2.43**	75.8	99.7
14	49	10,368	688.8	13,496	396.7	73.0	181.4	**2.49**	95.8	107.4
$n = 8$										
2	15	18,718	766.1	26,154	788.2	89.5	26.9	**0.30**	122.0	25.6
4	26	17,465	707.2	33,566	404.1	100.3	45.3	**0.45**	197.3	27.8
6	37	20,929	1,053	45,667	1,008	127.3	110.7	**0.87**	282.4	109.8
8	37	33,094	1,379	59,485	1,345	225.8	223.0	**0.99**	402.0	218.9
10	44	22,382	1,100	44,348	901.5	161.4	236.0	**1.46**	316.5	182.9
12	46	20,920	1,157	24,312	730.5	169.1	338.1	**2.00**	187.0	209.7
14	46	22,618	987.6	40,649	599.8	200.6	372.0	**1.85**	358.6	242.9
16	47	22,023	1,239	28,888	932.4	214.0	573.0	**2.68**	284.4	427.5

between the simplicial and the rectangular version, which is the quotient of the average running-time needed by the rectangular version and the average running-time needed by Algorithm 3.1. The problems with dimension $n \leq 6$ were run on a *SUN SPARCserver 1000* workstation. For problems with dimension $n \in \{7, 8\}$ we used

SUN ULTRA 60 workstations, which are – with our code – on average at least 3 times faster.

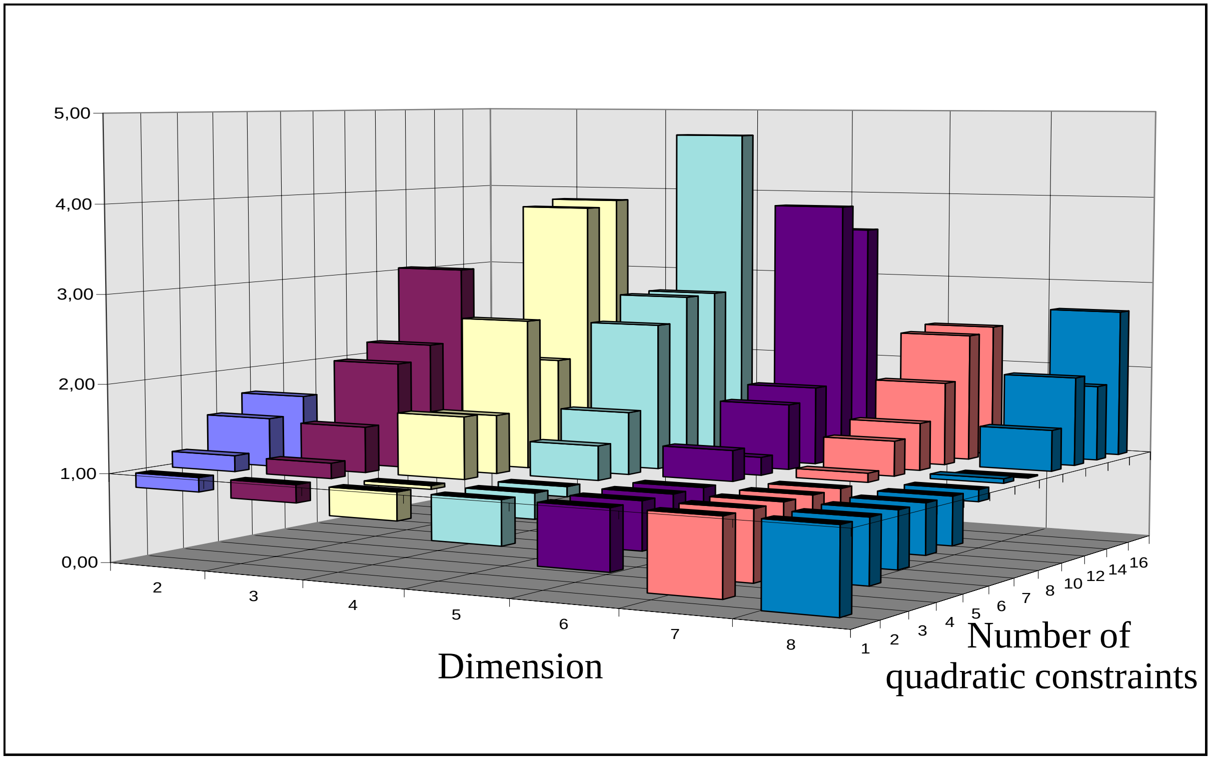
The numerical results show that for $p \geq n$ Algorithm 3.1 was, with respect to the average running-times, almost always faster than the rectangular algorithm. Only for couples (n, p) with $p \leq \max\{1, \lceil \frac{n}{2} \rceil\}$ the rectangular approach needed less time in more than 50% of the test examples. For fixed n the relative performance of the simplicial method improved with growing p (consider the bold-printed speedup columns in Table 3.1 and Table 3.2). In the cases $p = 2n$ Algorithm 3.1 outperformed the rectangular version. It was then up to 4.7 times faster.

FIGURE 3.1. Number of test problems in percent where Algorithm 3.1 is faster than Al-Khayyal et al.'s rectangular approach



The simplicial approach had many more linear subproblems to solve in order to detect an approximate solution of our test problems than the rectangular one. Moreover, this rate increased with growing dimension. There is at least one reason for this effect. In the derivation of the LP-relaxation ($\overline{\text{LP}}^S$) of (QP^S) in Section 3.2 we neglect the convex information contained in the transformed problem ($\overline{\text{QP}}^S$). In contrast to this Al-Khayyal et al. use all available information in order to generate their lower bounds. They do not omit anything. Therefore it is not surprising that

FIGURE 3.2. Speedup



the lower bounds used in the rectangular algorithm are better than those in our method. Nevertheless, even though the number of linear subproblems increased, the reduction in the complexity of each subproblem (smaller dimension and fewer constraints) led to a decrease in the running-time.

Let us illustrate the presented computational results with two figures. In Figure 3.1 the number of test problems, where Algorithm 3.1 was faster with respect to the running-time than the rectangular algorithm, are displayed in percent for all tested combinations of the dimension n and the number of quadratic constraints p. Figure 3.2 shows the corresponding speedup coefficients. Both graphics show on the one hand that for growing dimension and smaller number of quadratic constraints the relative performance of the rectangular algorithm in comparison with the simplicial approach improved. On the other hand, they emphasize that for a higher number of quadratic constraints ($p \geq n$) the numerical performance of the simplicial algorithm was much better.

The numerical performance of the method for solving (QP), which we discussed in the previous chapter, does not depend on the number of quadratic constraints. This is not the case for Algorithm 3.1. The computational results show that the effort for solving an all-quadratic program depends on the dimension as well as on the number of quadratic constraints. It is interesting to note that the average running-time of the simplicial method was by far less sensitive to the number of quadratic constraints than the running-time of Al-Khayyal et al.'s approach. For example, in the test problems with dimension $n = 8$ the average running-time of Algorithm 3.1 grew by a factor of almost 3, whereas the average running-time of the rectangular method grew by a factor of almost 45.

In Section 1.5 we described the construction of all-quadratic problems with dimension $n \in \{1, \dots, 8, 10\}$, but we do not present the numerical results for the ten-dimensional test problems. The reason is that both algorithms do not seem to be attractive for solving problems of type (QP) with a dense structure and a dimension higher than 8. For such problems they required excessive running-times.

TABLE 3.3. A comparison of the medians of the running-times

$p =$	1	2	3	4	5	6	7	8	10	12	14	16
$n = 2$												
S	0.26	0.26	0.36	0.38								
R	0.21	0.34	0.63	0.71								
MSu	**0.79**	**1.31**	**1.76**	**1.87**								
$n = 3$												
S	0.63	0.66	0.80	0.89	21.03	1.23						
R	0.58	1.04	1.68	2.11	2.92	3.80						
MSu	**0.93**	**1.58**	**2.09**	**2.37**	**2.83**	**3.01**						
$n = 4$												
S	1.33	1.59	1.94	3.11	2.50	2.48	2.99	2.40				
R	1.07	2.17	3.86	6.30	6.96	10.2	12.7	12.5				
MSu	**0.81**	**1.36**	**1.99**	**2.03**	**2.79**	**4.09**	**4.26**	**5.21**				
$n = 5$												
S	5.48	4.81	5.53	6.31	7.61	8.14	9.45	8.55	9.07			
R	2.58	4.89	10.7	12.1	19.6	19.8	31.0	39.1	52.2			
MSu	**0.47**	**1.02**	**1.94**	**2.58**	**2.43**	**3.28**	**4.57**	**4.22**	**5.76**			
$n = 6$												
S	10.2	19.7	21.0	24.0	22.8	21.5	26.7	36.5	24.9	41.9		
R	4.13	11.6	21.6	29.9	46.6	57.0	68.6	91.6	131.4	190.0		
MSu	**0.40**	**0.59**	**1.03**	**1.25**	**2.04**	**2.65**	**2.57**	**2.51**	**5.27**	**4.53**		
$n = 7$												
S	10.2	14.9	13.5	11.7	21.9	16.7	21.5	32.1	29.0	38.3	31.9	
R	2.63	7.76	12.9	15.0	32.6	32.8	51.8	61.3	89.8	127.0	153.2	
MSu	**0.26**	**0.52**	**0.96**	**1.28**	**1.49**	**1.97**	**2.40**	**1.91**	**3.09**	**3.31**	**4.80**	
$n = 8$												
S	28.9	20.8	27.6	39.1	31.0	42.5	83.2	86.7	59.3	120.7	77.1	88.1
R	6.58	19.7	30.7	45.5	59.3	79.4	114.3	151.0	181.5	294.0	297.6	454.5
MSu	**0.23**	**0.95**	**1.11**	**1.17**	**1.91**	**1.87**	**1.37**	**1.74**	**3.06**	**2.44**	**3.86**	**5.16**

A look at the standard deviation values in Tables 3.1 and 3.2 shows that the simplicial algorithm had, unfortunately, a significantly higher variation of the numerical effort. The standard deviation of the running-time is for problems with

$n \geq 5$ almost always higher than the corresponding average value (see Table 3.2). We even have combinations of n and p, where the standard deviation is more than 3 times larger than the average value. The rectangular algorithm did not have this property. The standard deviation of running-times is mostly smaller than the average value. This shows that the rectangular algorithm had a more robust behavior than the simplicial one in the sense that the effort for solving problems with the same dimension and the same number of quadratic constraints did not vary so keen. In spite of that higher variation of the effort for solving all-quadratic problems, Algorithm 3.1 was almost always faster and led on average to a substantial speedup, at least for problems with higher number of quadratic constraints. If we neglect the numerical outliers leading to the high standard deviation values, then there holds that Algorithm 3.1 showed an even better relative performance. In Table 3.3 the medians of the running-times for all tested combinations (n, p) together with the median speedup coefficients (MSu) are displayed, where this speedup coefficient is again the quotient of the median of the running-time for the rectangular algorithm (R) and the corresponding value for Algorithm 3.1 (S). This table shows that, with respect to the medians, Algorithm 3.1 was also for higher dimensional problems and $p = 2n$ about 5 times faster than Al-Khayyal et al.'s rectangular approach (compare with the speedup coefficients in Table 3.2).

3.5.3. A Modification of Algorithm 3.1. We finish the numerical examination of Algorithm 3.1 by considering a slight modification of this approach. In Remark 3.2.2(b) we pointed out that the LP-relaxation (LP^S) of (QP^S) depends on the numbering of the vertices of $S = [v_0, \dots , v_n]$, in particular on the choice of v_0. In the numerical tests we described till now, we did not care about the choice of this vertex. Since we did not apply any rule, it was somehow randomly which vertex of the considered simplex S_j^k ($k \in \mathbb{N}$; $j = 1, 2$) was the first one.

In the sequel we will see that a special choice of this vertex can lead to numerical improvements of our approach. There are of course many possible decision rules for choosing v_0^k, which could depend on the function values of q^0 at the vertices, or on the function values of the $n + 1$ possible affine underestimating functions, or also on the behavior of the quadratic constraints q^l ($l \in \{1, \dots , p\}$) and on the behavior of their underestimators. We tested several rules and would like to present only the one, which showed the best performance in our numerical tests.

Let $S = [v_0, \dots , v_n]$ be an arbitrary n-simplex and let, for $j \in \{0, \dots , n\}$, $\ell_S^{0,j} : \mathbb{R}^n \to \mathbb{R}$ be the affine underestimating function for q^0 on S constructed with respect to the vertex v_j, i.e.,

$$\begin{aligned}\ell_S^{0,j}(x) \;=\; &\sum_{i=0,i\neq j}^{n} \Big((W_S^j)^{-1}(x-v_j)\Big)_i (v_i-v_j)^T D^0 (v_i-v_j) \\ &\qquad + \left(d^0+2Q^0 v_j\right)^T (x-v_j) + (d^0)^T v_j + v_j^T Q^0 v_j \;,\end{aligned}$$

where W_S^j denotes the real $n \times n$ matrix with the columns $(v_i - v_j)$ $(i \in \{0,\dots,n\} \setminus \{j\})$. Consider the maximal distance of the objective function q^0 and the function $\ell_S^{0,j}$ $(j \in \{0,\dots,n\})$ at the vertices of S. It can be verified by straightforward calculation that there holds

$$\max_{i=0,\dots,n}\Big[q^0(v_i) - \ell_S^{0,j}(v_i)\Big] \;=\; \max_{i=0,\dots,n}(v_i-v_j)^T C^0 (v_i-v_j)\;.$$

Among these $n+1$ possibilities for the function ℓ_S^0 we choose one, where the minimum of these maximal distances is attained, i.e., we choose $j_0 \in \{0,\dots,n\}$ satisfying

$$\max_{i=0,\dots,n}(v_i-v_{j_0})^T C^0 (v_i-v_{j_0}) \;=\; \min_{j=0,\dots,n}\left[\max_{i=0,\dots,n}(v_i-v_j)^T C^0(v_i-v_j)\right]\;.$$

In Algorithm 3.1 this means that we interpret v_{j_0} as the first vertex of S, i.e., $S = [v_{j_0}, v_0, \dots, v_{j_0-1}, v_{j_0+1}, \dots, v_n]$, and construct the affine functions ℓ_S^l $(l \in \{0,\dots,p\})$ with respect to this vertex.

This decision rule for the choice of the first vertex of S_j^k ($k \in \mathbb{N}$, $j = 1, 2$) led to an improvement of the numerical performance of our approach. By applying this rule we could reduce the effort for solving our test problems. In Table 3.4 the proportional reductions of the average number of linear subproblems are displayed for

TABLE 3.4. Proportional reduction of the average number of LP's by applying a special selection rule for the first vertex v_0^k

n	p	1	2	3	4	5	6	7	8	10	12	14	16
2		9.92	9.96	5.30	6.67								
3		13.1	18.1	14.8	10.9	14.7	15.0						
4		16.0	21.7	19.3	21.9	16.5	32.5	16.8	15.3				
5		17.4	21.6	30.2	27.1	28.1	16.5	22.3	23.3	16.8			
6		23.4	25.7	33.5	34.9	25.4	34.4	25.4	31.4	24.0	26.1		
7		2.22	37.8	36.2	37.6	37.2	34.4	34.6	32.2	29.4	31.6	31.2	
8		55.6	28.1	46.5	24.1	45.6	32.4	35.2	34.4	43.9	42.3	41.1	39.9

all combinations of the dimension n and the number of quadratic constraints p examined in our numerical tests. This proportional reduction is calculated according to

$$\frac{\text{AvgNuLP(A)} - \text{AvgNuLP(B)}}{\text{AvgNuLP(A)}} \cdot 100\% \, .$$

AvgNuLP(A) denotes the average number of linear subproblems needed by Algorithm 3.1 without any rule for the choice of the first vertex. AvgNuLP(B) is the corresponding value, when the above decision rule was applied. This table shows that the application of the proposed selection rule was able to reduce the average number of LP's by up to 50%. Since this selection rule is not time-consuming – from a computational point of view – we obtained almost the same reduction in the average running-times of Algorithm 3.1.

It is interesting to note that the numerical improvement increased with growing dimension. A reason for this effect might be that with growing dimension the number of possibilities for choosing the first vertex increases. For small dimensional problems we have a high probability that the choice of v_0^k according to our special rule and the random choice of v_0^k coincide. Hence, we obtain only a slight difference in the numerical effort. However, for higher dimensional problems this probability decreases and the positive results in Table 3.4 corroborate that our selection rule for the first vertex was a good choice in the sense that the lower bounds, which we obtained by applying this rule, were mostly better than the one obtained without using any rule. Another remarkable result of Table 3.4 is that the numerical improvement did not depend on the number of quadratic constraints. Nevertheless, it can be possible that an additional consideration of the quadratic constraints in the selection rule for the first vertex leads to a further improvement of Algorithm 3.1.

The simplicial branch-and-bound method presented in this chapter used bisection as a subdivision rule for simplices. In Remark 3.4.1(a) we pointed out that Property (3.4.1) of this rule is substantial for the convergence of our approach. Some authors favor another subdivision rule. In this so-called *ω-subdivision* rule an n-simplex S^k is subdivided into up to $n+1$ subsimplices by using a radial subdivision with respect to the optimal solution $\omega(S^k)$ of the current LP-relaxation on S^k (see Definition 1.2.2). One hopes that this point bears some information of the original problem (QP), and hence one expects that this rule leads to better numerical results. However, this subdivision rule has not Property (3.4.1) such that the convergence of a simplicial branch-and-bound method, which is based on this rule, is still an open question. In the next chapter we will give an answer to this theoretical problem. We will consider a generalization of Algorithm 3.1, which uses convex subproblems and is able to deal with a more general problem class.

CHAPTER 4

On the Convergence of Simplicial Branch-and-Bound Methods

In this chapter we are interested in a generalization of the all-quadratic optimization problem studied in this thesis. We treat problems of the form

$$\begin{aligned} &\min g^0(x) + f^0(x) \\ &g^l(x)+f^l(x) \ \leq \ 0 \qquad l = 1, \dots , p \\ &\qquad x \in P \ , \end{aligned} \tag{DCP}$$

where $g^l : \mathbb{R}^n \to \mathbb{R}$ $(l = 0, \dots , p)$ are convex functions, $f^l : \mathbb{R}^n \to \mathbb{R}$ $(l = 0, \dots , p)$ are concave functions, and $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ with $A = (a_1, \dots , a_m)^T \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ is a non-empty and full-dimensional polytope. Problems of this form belong to the class of **general d.c. problems** since the objective function and a part of the constraint functions can be written as a **d**ifference of two **c**onvex functions (see, e.g., [HPT95, TUY95, HT96B] for the framework of d.c. problems). We denote by

$$F := \{x \in \mathbb{R}^n : Ax \leq b \ , \ g^l(x) + f^l(x) \leq 0 \ , l = 1, \dots , p\}$$

the feasible region of Problem (DCP).

4.1. Introduction

In the following we distinguish three subclasses of (DCP):

(DCP$_1$) $g^0 \equiv 0$, $p = 0$, i.e., minimization of a concave function over a polytope;

(DCP$_2$) $g^l \equiv 0$ $(l = 1, \dots , p)$, i.e., minimization of a d.c. function over a feasible region described by a polytope and by reverse convex constraints;

(DCP$_3$) $\exists l \in \{1, \dots , p\}$: $g^l \neq 0$, i.e., minimization of a d.c. function over a non-polyhedral set.

It is well-known that the optimal value $f^\star$ of a problem of type (DCP$_1$) is attained in at least one vertex of P [HPT95, Theorem 1.19]. Nevertheless, this problem class was proven to be NP-hard. NP-hardness holds even in very special cases, such as problems whose objective function is concave quadratic and whose feasible region is a hypercube (see, e.g., [PS88]).

This class of global optimization problems encompasses a wide variety of applications. Among them we recall problems with *economies of scale*, where concave costs arise because of decreasing marginal costs, and *minimum concave cost flow problems,* i.e., flow problems in which the cost functions of the arcs are concave. Moreover, some well-known mathematical problems can be reformulated as concave optimization problems, for instance *integer programming* [PS76] and *linear complementarity problems* (see, for example, [HPT95, pages 69-70]).

Different approaches to the solution of (DCP$_1$) were proposed in the literature. They can be subdivided in three classes: enumerative methods, successive approximation methods and branch-and-bound methods. The first two classes of algorithms are mostly guaranteed to return an optimal solution in finite time by exploring, in the worst case, all the vertices of P. Branch-and-bound methods can generally only be guaranteed to be convergent. On the other hand, algorithms from this class are often efficient in practice.

The class of branch-and-bound methods can be further subdivided in three subclasses according to the kind of partition sets they employ (see also Subsections 1.2.2 and 1.2.3): *conical* algorithms, first introduced in [TUY64], *rectangular* algorithms, first introduced in [FS69], and *simplicial* algorithms, first introduced in [HOR76]. For further, more detailed information about theory, algorithms and applications in the field of concave optimization over a polytope we refer to the quite extensive literature on the subject, including, in particular, [HOR84, PR86, BEN95, HPT95, HT96B].

If the objective function of a problem of type (DCP$_2$) is linear and there is only one concave constraint, i.e., $p = 1$, then this problem is a **canonical d.c. problem**

$$\begin{aligned} &\min\, c^T x \\ &\bar{f}(x) \;\geq\; 0 \\ &x \in P \subset \mathbb{R}^n \end{aligned} \qquad (\overline{\text{DCP}}_2)$$

with $\bar{f}(x) := -f^1(x)$ (see, e.g., [HPT95] or [TUY95] for the definition of the canonical d.c. problem). It is known that, under mild regularity conditions, the optimal value $f^\star$ of ($\overline{\text{DCP}}_2$) is attained at a point $x^\star$ on an edge of P satisfying $\bar{f}(x^\star) = 0$ [HPT95, Theorem 4.4] .

The class (DCP_2) of global optimization problems is not so widely explored in the literature. The articles on d.c. programming consider mostly the canonical d.c. problem or a general d.c. problem of the form (DCP_3). Our differentiation between (DCP_2) and (DCP_3) is theoretically motivated, as we will see in the next sections. An application of problem class (DCP_2) is the *packing problem* (see Section 1.3 and particularly Chapter 5 for details). Problems with a d.c. objective function, linear constraints and Boolean variables can also be transformed to problems of type (DCP_2) ($x_i \in \{0, 1\} \;\Leftrightarrow\; x_i^2 - x_i \geq 0, x_i \in [0, 1]$).

Using the fact that an optimal solution of the canonical d.c. problem is attained on an edge of the polytope P a special finite solution method for this problem class was developed in [TT85] (modified in [HPT95]). If the objective function of (DCP_2) is nonlinear or if there is more than one concave constraint then, extending the ideas used for solving problems of type (DCP_1), Problem (DCP_2) can be solved by branch-and-bound methods, as we will see in subsequent sections.

The class of d.c. functions defined on a compact convex set of $\mathbb{R}^n$ is dense in the set of continuous functions [HPT95, Corollary 4.2]. Furthermore, the set of d.c. functions is closed with respect to arbitrary linear combinations, finite maximizations, finite minimizations and multiplications [HPT95, Theorem 4.1]. Therefore, the class of general d.c. problems encompasses a wide variety of problem classes and applications. However, finding a representation of a d.c. function as a difference of two convex functions, which we assume to be given in the formulation of problem type (DCP_3), is in general a hard, still open problem. For the all-quadratic problems such a representation is easy to generate as we saw in Chapter 3. They are also known for many interesting function classes. Among the various applications of d.c. problems of type (DCP_3) we recall the *bridge location problem* (see [HT99]), the *general location problem* (see, for example, [ILM88]) and the *design centering problem* [VS82, THA88, NS92]. Similar to problems of type (DCP_2) general d.c. problems of type (DCP_3) can be solved by branch-and-bound methods. In order to avoid convex subproblems in such approaches, the branch-and-bound methods are often combined with an outer approximation scheme (see [HT99] and, in particular, [TUY95] and references therein).

In this chapter simplicial branch-and-bound methods for solving all types of (DCP) are considered. In particular, simplicial algorithms based on the so-called *ω-subdivision*, a special subdivision rule, first introduced in [TUY64] for conical branch-and-bound algorithms and applied for simplicial approaches, for example, in [HT96B], are studied. As long as a so-called *exhaustive* subdivision rule is used, the convergence of the simplicial branch-and-bound method suggested in the next section can be ensured. The convergence of this algorithm using ω-subdivision, which is not necessarily exhaustive, was an open theoretical question for a long time. Recently, a similar question was answered for conical branch-and-bound algorithms used for solving (DCP$_1$) independently and by different techniques in [JM98] and [LOC97]. However, to the author's knowledge, no proof of convergence for simplicial algorithms, which base only on ω-subdivisions, has been published yet.

In the next section the general scheme of the studied simplicial branch-and-bound algorithm is given, and in Section 4.3 the convergence of this approach based on an exhaustive subdivision rule is proved. This convergence result corresponds to the one obtained in the previous chapter for Algorithm 3.1 (see Theorem 3.4.2). If we use ω-subdivision, then a similar convergence result for the proposed algorithm, used for solving problems of type (DCP$_1$), can also be proven. In order to show, additionally, a convergence result for this approach, applied for problems of type (DCP$_2$), we need an assumption with respect to the concave part f^0 of the objective function. This assumption is not a restriction, as we will see in Section 4.4. Though the obtained convergence result for this case will be theoretically weaker than the one before, – from a practical point of view – all convergence results derived in this chapter have the same quality. They imply that the proposed approach detects in finite time either the emptiness of F or an approximate solution of the considered problem. The proofs of the statements yielding these convergence results for our approach using ω-subdivisions are given together in Section 4.4 for both subclasses of (DCP). Even though the convergence results are slightly different, their derivation is non the less connected. The part of the proofs relating to Problem (DCP$_1$) is equivalent to the convergence proofs in [LR97B]. In Section 4.5 we propose an example, which shows that the simplicial branch-and-bound method using ω-subdivisions for solving problems of type (DCP$_3$) is not necessarily convergent. We conclude the consideration of the convergence in Section 4.6 with a numerical comparison of the simplicial branch-and-bound algorithm with different subdivision rules for solving all-quadratic optimization problems of type (DCP$_3$). In particular, we suggest in this section a modification of the ω-subdivision, which leads to a convergent algorithm also for problems of the general class (DCP$_3$). In the last Section 4.7 a partial answer to the theoretical problem of finiteness of the simplicial branch-and-bound algorithm with ω-subdivision, applied for solving problems of type (DCP$_1$), is presented. This finiteness result is also given in [LR97A].

4.2. Simplicial Algorithms for (DCP)

In this section we describe a simplicial branch-and-bound algorithm for solving Problem (DCP). The formulation is a generalization of the one given in Chapter 3 (see Algorithm 3.1). Before giving the exact description we recall some notations, which will be extensively used in the following.

- Let, for $r \in \mathbb{N}$, $r \leq n$,

$$B_r := \{\lambda \in \mathbb{R}^{r+1} : \sum_{i=0}^{r} \lambda_i = 1 \ , \ \lambda_i \geq 0 \ , \ i = 0, \ldots, r\}$$

  be the standard r-simplex in $\mathbb{R}^{r+1}$.

- Let $S = [v_0, \ldots, v_n]$ be an n-simplex and $l \in \{0, \ldots, p\}$. The affine function $\varphi_S^l : \mathbb{R}^n \to \mathbb{R}$,

$$\varphi_S^l(x) = \sum_{i=0}^{n} \lambda(x)_i f^l(v_i) \qquad (4.2.1)$$

  is the convex envelope of f^l over S (see, e.g., [HPT95] or Subsection 1.2.4). The vector $\lambda(x) \in \{\lambda \in \mathbb{R}^{n+1} : \sum_{i=0}^{n} \lambda_i = 1\}$ denotes the uniquely determined barycentric coordinates of $x \in \mathbb{R}^n$ with respect to the vertex set $\{v_0, \ldots, v_n\}$ of S, i.e., $x = \sum_{i=0}^{n} \lambda(x)_i v_i$. Let $\zeta_S^l \in \mathbb{R}^n$ be the unique solution of the following system of equations

$$(\zeta_S^l)^T (v_i - v_0) \ = \ f^l(v_i) - f^l(v_0) \qquad i = 1, \ldots, n \,.$$

  Then there holds, for $x \in \mathbb{R}^n$,

$$\varphi_S^l(x) \ = \ (\zeta_S^l)^T (x - v_0) + f^l(v_0) \,. \qquad (4.2.2)$$

- Let $S = [v_0, \ldots, v_n]$ be an n-simplex. For each $j \in \{0, \ldots, n\}$, choose $i(j) \in \{0, \ldots, n\} \setminus \{j\}$ and let $\bar{v}_j^S \in \mathbb{R}^n$ be the unique solution of the system of linear equations

$$(\bar{v}_j^S)^T (v_i - v_{i(j)}) \ = \ 0 \qquad i \in \{0, \ldots, n\} \setminus \{j, i(j)\} \qquad (4.2.3.a)$$

$$(\bar{v}_j^S)^T (v_j - v_{i(j)}) \ = \ -1 \,. \qquad (4.2.3.b)$$

  Then, with $c_j^S := (\bar{v}_j^S)^T v_{i(j)}$, the n-simplex S can be represented by the system of linear inequalities

$$(\bar{v}_j^S)^T x \ \leq \ c_j^S \qquad j = 0, \ldots, n \,. \qquad (4.2.4)$$

The vector $\bar{v}_j^S$ is the normal of the facet $[v_0, \ldots, v_{j-1}, v_{j+1}, \ldots, v_n]$ of S.

In Chapter 3 (see, in particular, Section 3.4) we pointed out that it is necessary to be satisfied with approximate solutions of all-quadratic problems of type (QP), if we are interested in a finite solution approach. The finiteness of the simplicial branch-and-bound method for Problem (DCP) to be discussed in the present chapter, can also only be obtained, if approximate feasible respectively optimal solutions are sufficient. Therefore, we introduce the concept of *(δ,ρ)-feasible* points and of *(ϵ,δ,ρ)-solutions* of Problem (DCP).

DEFINITION 4.2.1. *A point $\bar{x} \in \mathrm{I\!R}^n$ is called* ***(δ,ρ)-feasible*** *for Problem (DCP) with real numbers $\delta, \rho \geq 0$, if there holds*

$$a_j^T \bar{x} - b_j \;\leq\; \rho \qquad j = 1, \ldots, m \tag{4.2.5.a}$$

and

$$g^l(\bar{x}) + f^l(\bar{x}) \;\leq\; \delta \qquad l = 1, \ldots, p\,. \tag{4.2.5.b}$$

A point $\bar{x} \in \mathrm{I\!R}^n$ is called an ***(ϵ,δ,ρ)-solution*** *for Problem (DCP) with real numbers ϵ,δ,$\rho \geq 0$, if there holds*

$$\bar{x} \text{ is } (\delta,\rho)\text{-feasible} \tag{4.2.6.a}$$

and

$$g^0(\bar{x}) + f^0(\bar{x}) - \epsilon \;\leq\; \min_{x \in F} \left[g^0(x) + f^0(x)\right] \;\;, \tag{4.2.6.b}$$

where the right-hand side of (4.2.6.b) is defined as ∞, if F is empty.

For $\rho = 0$ we say that $\bar{x}$ is a δ-feasible point, for $\delta = \rho = 0$ $\bar{x}$ is a feasible point and for $\epsilon = \delta = \rho = 0$ $\bar{x}$ is an optimal solution of (DCP). Note that for an (ϵ,δ,ρ)-solution $\bar{x} \in \mathrm{I\!R}^n$ it is not necessary that there holds

$$g^0(x^\star) + f^0(x^\star) \;\leq\; g^0(\bar{x}) + f^0(\bar{x})\,,$$

where $x^\star$ denotes the optimal solution of (DCP), if $F \neq \emptyset$. Therefore, Condition (4.2.6.b) does not guarantee that the objective function value of $\bar{x}$ has a distance smaller than ϵ to the optimal one (see, in particular, the case $F = \emptyset$). This distance depends on the choice of δ and ρ and on the behavior of the objective function of (DCP) on the set

$$\begin{aligned} F_{\delta,\rho} := \{x \in \mathrm{I\!R}^n : a_j^T x - b_j \leq \rho\,,\; j = 1, \ldots, m, \\ g^l(x) + f^l(x) \leq \delta\,,\; l = 1, \ldots, p\}\,. \end{aligned}$$

If, however, $\bar{x}$ is feasible, then $\bar{x}$ is ϵ-optimal in the sense of

$$|g^0(\bar{x}) + f^0(\bar{x}) - g^0(x^\star) - f^0(x^\star)| \;\leq\; \epsilon\,.$$

In order to formulate the simplicial branch-and-bound method for Problem (DCP) we need a convex relaxation of (DCP) with respect to a given simplex. Let $S = [v_0, \ldots, v_n]$ be an n-simplex. The set

$$F_S \ := \ \{x \in S : \ Ax \leq b \ , \ g^l(x) + \varphi_S^l(x) \leq 0 \ , \ l = 1, \ldots, p\} \qquad (4.2.7)$$

is convex with $F_S \ \supset \ F \cap S$. If F_S is non-empty, we know that the optimal solution $\mu^\star(S)$ of the convex optimization problem

$$\begin{aligned} \min \ g^0(x) + \varphi_S^0(x) \\ x \in F_S \end{aligned} \qquad (\text{DCP}^S)$$

is a lower bound for $\min\limits_{x \in F \cap S} \left[g^0(x) + f^0(x)\right]$, i.e., (DCP^S) is a convex relaxation of (DCP). In the following we denote by (DCP$_i^S$) ($i = 1, 2, 3$) the proposed convex relaxation of Problem (DCP$_i$) ($i = 1, 2, 3$) with respect to the simplex S. Note that (DCP$_1^S$) is a linear optimization problem, (DCP$_2^S$) is a convex optimization problem with linear constraints and (DCP$_3^S$) is an optimization problem with a convex objective function as well as convex constraints.

In general, it is not possible to solve exactly the convex optimization problem (DCP^S) in finite time. We are only able to assume that a solution method for (DCP^S) is known, which solves the problem in finite time with arbitrary accuracies $\bar{\epsilon}$, $\bar{\delta}$, $\bar{\rho} > 0$. This means, if such a method does not detect the emptiness of F_S, it generates a point $\bar{x}$ with the properties

$$a_j^T \bar{x} - b_j \ \leq \ \bar{\rho} \qquad j = 1, \ldots, m \ , \qquad (4.2.8.a)$$

$$(\bar{v}_i^S)^T \bar{x} - c_i^S \ \leq \ \bar{\rho} \qquad i = 0, \ldots, n \ , \qquad (4.2.8.b)$$

$$g^l(\bar{x}) + \varphi_S^l(\bar{x}) \ \leq \ \bar{\delta} \qquad l = 1, \ldots, p \qquad (4.2.8.c)$$

and

$$g^0(\bar{x}) + \varphi_S^0(\bar{x}) - \bar{\epsilon} \ \leq \ \min_{x \in F_S} \left[g^0(x) + \varphi_S^0(x)\right] \ = \ \mu^\star(S) \ . \qquad (4.2.8.d)$$

According to Definition 4.2.1, $\bar{x}$ is an ($\bar{\epsilon}$, $\bar{\delta}$, $\bar{\rho}$)-solution of Problem (DCP^S). Note that it is possible that the used method stops with a ($\bar{\delta}$, $\bar{\rho}$)-feasible point, even though the feasible region F_S is empty. In each case

$$\mu(S) \ := \ g^0(\bar{x}) + \varphi_S^0(\bar{x}) - \bar{\epsilon} \qquad (4.2.9)$$

is a lower bound for $\mu^\star(S)$, since $\mu^\star(S)$ is defined as ∞ in the empty case. In the following we denote by CONVEXSOLVER$_{\bar{\epsilon},\bar{\delta},\bar{\rho}}$ ($\bar{\epsilon}$, $\bar{\delta}$, $\bar{\rho} \geq 0$) a solution method for Problem (DCP^S), which detects after finite time either the emptiness of F_S or a point $\bar{x}$ with Properties (4.2.8.a)-(4.2.8.d).

REMARK 4.2.1.

(a) Since (DCP$^S_1$) is a linear optimization problem we can use the Simplex-Algorithm as a CONVEXSOLVER$_{0,0,0}$.

(b) If the objective function of (DCP$^S_2$) is quadratic with a positive definite Hessian, then several solution methods exist. For example, this problem is equivalent to a linear complementarity problem (LCP) and, therefore, there exists also a CONVEXSOLVER$_{0,0,0}$. For an overview on the relation between convex quadratic optimization problems and LCP‘s and for solution methods for LCP‘s we refer to [CPS92].

(c) In Section B.1 a CONVEXSOLVER$_{\bar{\epsilon},\bar{\delta},0}$, which bases on the KCG-cutting-plane-method [CG59, KEL60] and needs no additional assumptions on the involved nonlinear functions beside of their convexity, is presented. For extensions of this method we refer to [HTT87, HT96B] and references therein.

(d) If an ($\bar{\epsilon}$, $\bar{\delta}$, 0)-solution of Problem (DCP^S) is required, it is possible to adjust an arbitrary CONVEXSOLVER$_{\bar{\epsilon},\bar{\delta},\bar{\rho}}$ in such a way that the calculated point $\bar{x} \in \mathrm{I\!R}^n$ is feasible with respect to the linear constraints. If a CONVEXSOLVER$_{\tilde{\epsilon},\tilde{\delta},\tilde{\rho}}$ delivers a point $\tilde{x} \notin P \cap S$ and the accuracies are chosen sufficiently small, then we can use the orthogonal projection point $\bar{x}$ of $\tilde{x}$ on the polytope $P \cap S$ as an ($\bar{\epsilon}$, $\bar{\delta}$, 0)-solution of Problem (DCP^S). How this can be done and, in particular, how the accuracies $\tilde{\epsilon}$, $\tilde{\delta}$ and $\tilde{\rho}$ have to be chosen, is described in Section B.2.

For the formulation of the algorithm we assume further that a real number $\bar{\eta} < \infty$ with the property

$$\bar{\eta} \;\geq\; \begin{cases} \min\limits_{x \in F} \;\left[g^0(x) + f^0(x)\right] & \text{, if } F \neq \emptyset \\ \max\limits_{x \in P} \;\left[g^0(x) + f^0(x)\right] & \text{, otherwise} \end{cases} \quad , \tag{4.2.10}$$

is known. If a feasible point $\bar{x} \in F$ is known in advance, then we can use $\bar{\eta} := g^0(\bar{x}) + f^0(\bar{x})$. Otherwise by setting

$$\bar{\eta} \;:=\; \max_{x \in P} \left[\psi^0_S(x) + f^0(x)\right] \;, \tag{4.2.11}$$

where $\psi^0_S : \mathrm{I\!R}^n \to \mathrm{I\!R}$ denotes the concave envelope of g^0 with respect to an arbitrary n-simplex $S \supset P$, we get the required value. Note that (4.2.11) is a concave maximization problem, which can be solved by an arbitrary CONVEXSOLVER$_{\bar{\epsilon},\bar{\delta},\bar{\rho}}$.

If the accuracy $\bar{\epsilon}$ is greater than 0, we have to adjust the calculated *optimal* value of (4.2.11) with $\bar{\epsilon}$ in order to obtain a real number with Property (4.2.10).

The description of the algorithm follows the guidelines of a basic branch-and-bound algorithm given in [HPT95, Algorithm 3.5] and is similar to Algorithm 3.1 discussed in Chapter 3.

ALGORITHM 4.1 (***Simplicial Branch-and-Bound Algorithm for (DCP)***).

```
Initialization
    Choose real numbers $\epsilon$, $\delta$, $\rho \geq 0$ and sequences $\{\bar{\epsilon}^k\}_{k\in\mathbb{N}_0}$, $\{\bar{\delta}^k\}_{k\in\mathbb{N}_0}$,
    $\{\bar{\rho}^k\}_{k\in\mathbb{N}_0}$ with $\bar{\epsilon}^k$, $\bar{\delta}^k$, $\bar{\rho}^k \geq 0$ ($k \in \mathbb{N}_0$) and $\bar{\epsilon}^k$, $\bar{\delta}^k$, $\bar{\rho}^k \to 0$ ($k \to \infty$).
    Determine an $n$-simplex $S^0 = [v_0^0, \dots, v_n^0]$ with $S^0 \supset P$ .
    $Q \leftarrow \{v_i^0 \, : \, i = 0, \dots, n$ with $v_i^0$ is ($\delta$,$\rho$)-feasible$\}$
    If CONVEXSOLVER$_{\bar{\epsilon}^0,\bar{\delta}^0,\bar{\rho}^0}$ detects $F_{S^0} = \emptyset$ Then
        STOP $\leftarrow$ True ($F = \emptyset$)
    Else
        Let $\omega(S^0)$ be an ($\bar{\epsilon}^0$, $\bar{\delta}^0$, $\bar{\rho}^0$)-solution of (DCP$^{S^0}$) and
        $\mu(S^0) = g^0(\omega(S^0)) + \varphi^0_{S^0}(\omega(S^0))$ be the corresponding function value.
        $\mu(S^0) \leftarrow \mu(S^0) - \bar{\epsilon}^0, \; \mu^0 \leftarrow \mu(S^0)$ , $\mathcal{P} \leftarrow \{S^0\}$
        If $\omega(S^0)$ is ($\delta$, $\rho$)-feasible Then $Q \leftarrow Q \cup \{\omega(S^0)\}$
        If $Q \neq \emptyset$ Then
            $\eta^0 \leftarrow \min_{x\in Q} \left[g^0(x) + f^0(x)\right]$
            Choose $x_f \in Q$ with $\eta^0 = g^0(x_f) + f^0(x_f)$ .
        Else
            $\eta^0 \leftarrow \bar{\eta} + \epsilon + \tau$     ($\tau > 0$ arbitrary)
        EndIf
        STOP $\leftarrow$ False , $k \leftarrow 0$
    EndIf

While STOP $=$ False Do
    If $\eta^k - \mu^k \leq \epsilon$ Then                                        (SC)
        STOP $\leftarrow$ True ($x_f$ is an ($\epsilon$, $\delta$, $\rho$)-solution of (DCP) )
    Else
        Choose $w^k \in S^k \backslash \{v_0^k, \dots, v_n^k\}$ and set                          (PSR)
        $I^k \leftarrow \{i \in \{0, \dots, n\} : \lambda_i^k > 0$ with $\lambda^k \in B_n$ , $\sum_{j=0}^n \lambda_j^k v_j^k = w^k$ $\}$
    If $w^k$ is ($\delta$, $\rho$)-feasible Then $Q \leftarrow Q \cup \{w^k\}$
    For $j \in I^k$ Do
      $S_j^k \leftarrow [v_0^k, \dots, v_{j-1}^k, w^k, v_{j+1}^k, \dots, v_n^k] \subset S^k$
      If CONVEXSOLVER$_{\bar{\epsilon}^k, \bar{\delta}^k, \bar{\rho}^k}$ does not detect $F_{S_j^k} = \emptyset$ Then
        Let $\omega(S_j^k)$ be an ($\bar{\epsilon}^k$, $\bar{\delta}^k$, $\bar{\rho}^k$)-solution of (DCP$^{S_j^k}$) and $\mu(S_j^k) =$
        $g^0(\omega(S_j^k)) + \varphi^0_{S_j^k}(\omega(S_j^k))$ be the corresponding function value.
        $\mu(S_j^k) \leftarrow \max\{\mu(S_j^k) - \bar{\epsilon}^k, \mu^k\}$                    (LBR)
        If $\omega(S_j^k)$ is ($\delta$, $\rho$)-feasible Then $Q \leftarrow Q \cup \{\omega(S_j^k)\}$
        $\mathcal{P} \leftarrow \mathcal{P} \cup \{S_j^k\}$
      EndIf
    EndFor
    $\mathcal{P} \leftarrow \mathcal{P} \backslash \{S^k\}$
    If $Q \neq \emptyset$ Then
      $\eta^{k+1} \leftarrow \min_{x \in Q} \left[g^0(x) + f^0(x)\right]$
      Choose $x_f \in Q$ with $\eta^{k+1} = g^0(x_f) + f^0(x_f)$ .
    Else
      $\eta^{k+1} \leftarrow \eta^k$
    EndIf
    $\mathcal{P} \leftarrow \mathcal{P} \backslash \{S \in \mathcal{P} : \mu(S) \geq \eta^{k+1} - \epsilon\}$                    (PR)
    If $\mathcal{P} \neq \emptyset$ Then
      $\mu^{k+1} \leftarrow \min_{S \in \mathcal{P}} \mu(S)$
      Choose $S^{k+1} \in \mathcal{P}$ with $\mu^{k+1} = \mu(S^{k+1})$.                    (SSR)
    Else
      If $Q \neq \emptyset$ Then
        $\mu^{k+1} \leftarrow \eta^{k+1} - \epsilon$
      Else
        STOP $\leftarrow$ True ($F = \emptyset$)
      EndIf
    EndIf
    $k \leftarrow k + 1$
  EndIf
EndWhile
```

REMARK 4.2.2.

(a) P is assumed to be a non-empty full-dimensional polytope. Therefore, there exists always an n-simplex S^0 with $S^0 \supset P$. For the construction possibilities we refer to [HPT95, pp. 145f].

(b) Since we are in general – as mentioned before – not able to solve the convex subproblems exactly, we cannot guarantee that there holds

$$\mu(S_j^k) \;\geq\; \mu(S^k) = \mu^k \;,$$

even though we know that $S_j^k \subset S^k$. In order to generate a non-decreasing sequence of lower bounds we use the lower bounding rule (LBR).

(c) The deletion of the simplices in the classical *pruning rule* (PR) is the consequence of the fact that $\mu(S)$ is a lower bound of $\min_{x \in F \cap S} \left[g^0(x) + f^0(x) \right]$ (see (4.2.9)). If $\mathcal{P}$ is empty after executing this rule and Q is not empty, then it is obvious that $\eta^{k+1} - \epsilon$ is a lower bound for the optimal value of (DCP).

(d) Each (δ, ρ)-feasible point for (DCP) generated during the solution of the convex subproblems should be affiliated to the set Q in order to possibly improve the upper bound η^k.

(e) The algorithm has to generate at least one (δ, ρ)-feasible point for Problem (DCP), such that the stopping criterion (SC) can be satisfied. If no feasible point is known in advance, it follows immediately from Property (4.2.10) of $\bar{\eta}$ that, for each simplex S generated during the execution of the algorithm, there holds

$$\mu(S) \;\leq\; \bar{\eta} \;=\; \eta^0 - \epsilon - \tau \;.$$

Therefore, without an update of η^{k+1} different from η^k the stopping criterion (SC) cannot be fulfilled. Moreover, taking the pruning rule into account, we know that (SC) can only be satisfied in iteration $k \geq 2$, if there holds $\mathcal{P} = \emptyset$ and $Q \neq \emptyset$ at the end of the previous iteration (compare with Remark 3.3.1(d)).

(f) It is possible that Algorithm 4.1 generates an (ϵ, δ, ρ)-solution of Problem (DCP) with ϵ, δ, $\rho > 0$, even if $F = \emptyset$.

(g) Since we choose $w^k \notin \{v_0^k, \ldots, v_n^k\}$ in the point selection rule (PSR) it is ensured that there holds $I^k \neq \emptyset$. Note that the used subdivision of S^k is a radial subdivision (see Definition 1.2.2), which forms a partition of S^k (see Definition 1.2.1 and [HPT95, Proposition 3.7]).

(h) If there holds, for an iteration $k \in \mathbb{N}$, $\epsilon \geq \bar{\epsilon}^k$, $\delta \geq \bar{\delta}^k$ and $\rho \geq \bar{\rho}^k$ and, additionally, if $\omega(S_j^k)$ is a vertex of the simplex S_j^k ($j \in I^k$), then we know that this simplex S_j^k is fathomed in the pruning rule (PR). Indeed, let $\omega(S_j^k)$ be a vertex of S_j^k. It follows that the function value of each concave function f^l ($l \in \{0, \dots, p\}$) at the point $\omega(S_j^k)$ coincides with the function value of the corresponding convex envelope $\varphi^l_{S_j^k}$. The point $\omega(S_j^k)$ is at least (δ, ρ)-feasible for Problem (DCP$^{S_j^k}$) and, consequently, also (δ, ρ)-feasible for Problem (DCP), i.e., $\omega(S_j^k)$ is used for updating the upper bound η^{k+1}. It follows that

$$\mu(S_j^k) \;\geq\; g^0(\omega(S_j^k)) + \underbrace{\varphi^0_{S_j^k}(\omega(S_j^k))}_{=f^0(\omega(S_j^k))} - \epsilon \;\geq\; \eta^{k+1} - \epsilon\,,$$

i.e., S_j^k must be fathomed in the pruning rule (PR).

The construction of μ^k ($k \in \mathbb{N}$) guarantees that this value is always a lower bound for $g^0(x) + f^0(x)$ with respect to the whole feasible region F. Therefore, it is obvious that, when finite, the algorithm will determine either an (ϵ, δ, ρ)-solution ($\epsilon, \delta, \rho \geq 0$) of (DCP) or the emptiness of F. How far it is possible to prove the finiteness of Algorithm 4.1 with ϵ, δ, $\rho > 0$, respectively the convergence for $\epsilon = \delta = \rho = 0$, depends on the choice of the rule to split the current simplex S^k. The subdivision rule applied in Algorithm 4.1, which is determined by the selection of the point w^k (see the point selection rule (PSR) in the formulation of Algorithm 4.1), is also a critical one with respect to the efficiency of the presented approach (see, e.g., [TUY91A] or the numerical results in Section 4.6).

There exist two classical subdivision rules. In the so-called **bisection**, which was first introduced in [HOR76] for simplicial algorithms, w^k is chosen as the midpoint of the longest edge of the current simplex $S^k = [v_0^k, \dots, v_n^k]$. Let $i_0, i_1 \in \{0, \dots, n\}$ be two indices corresponding to vertices of S^k with the longest Euclidean distance, i.e.,

$$\|v_{i_0}^k - v_{i_1}^k\|_2 \;=\; \max_{i,j=0,\dots,n} \|v_i^k - v_j^k\|_2\,. \tag{4.2.12}$$

Then we choose

$$w^k \;:=\; \frac{v_{i_0}^k + v_{i_1}^k}{2}\,.$$

This is the subdivision rule, we chose in the formulation of Algorithm 3.1 (see Section 3.3).

If we apply the so-called ω**-subdivision** (see, e.g., [TUY64, HT96B]), then we subdivide the current simplex S^k with respect to the calculated solution $\omega(S^k)$ of the subproblem (DCP$^{S^k}$). In the classical ω-subdivision rule, as it is described, e.g., in [HT96B], the point w^k is chosen as $\omega(S^k)$. Since it is assumed there that $\omega(S^k)$ is feasible with respect to the relevant simplex S^k and, additionally, feasible with respect to the original problem, i.e., $\omega(S^k) \in F \cap S^k$, it is clear (see Remark 4.2.2(h)) that in this situation there holds $\omega(S^k) \in S^k \setminus \{v_0^k, \ldots, v_n^k\}$.

In the formulation of Algorithm 4.1 we use a CONVEXSOLVER$_{\bar{\epsilon}^k, \bar{\delta}^k, \bar{\rho}^k}$ with $\bar{\epsilon}^k$, $\bar{\delta}^k$, $\bar{\rho}^k \geq 0$ ($k \in \mathbb{N}$). Therefore, in general we cannot expect that $\omega(S^k)$ is contained in S^k, and, in particular, we do not know anything about the feasibility of $\omega(S^k)$ with respect to the original Problem (DCP). For that reason, we cannot choose $w^k = \omega(S^k)$ in each case and we need, thus, a generalization of the classical ω-subdivision rule. A possible choice of w^k is the following, which we would like to call the **generalized ω-subdivision rule (GWSR)**:

```
Choose $\bar{\lambda}^k \in \{\lambda \in \mathbb{R}^{n+1} : \sum_{i=0}^n \lambda_i = 1\}$ with $\omega(S^k) = \sum_{i=0}^n \bar{\lambda}_i^k v_i^k$.
$\bar{I}^k \leftarrow \{i \in \{0, \ldots, n\} : \bar{\lambda}_i^k > 0\}$
If $|\bar{I}^k| = 1$ Then
   $w^k \leftarrow \frac{1}{2}\left(v_{i_0}^k + v_{i_1}^k\right)$     (i.e., choose a bisection)
Else
   Determine $\gamma^k := \sum_{i \in \bar{I}^k} \bar{\lambda}_i^k$
   For $i = 0$ To $n$ Do
      If $i \in \bar{I}^k$ Then
         $\lambda_i^k \leftarrow \frac{\bar{\lambda}_i^k}{\gamma^k}$
      Else
         $\lambda_i^k \leftarrow 0$
      EndIf
   EndFor
   $w^k \leftarrow \sum_{i=0}^n \lambda_i^k v_i^k$
EndIf
```

The ω-subdivision rule, which is connected to the information returned by the algorithm inside the selected simplex, seems to be the more natural choice than bisection, at least for problems of type (DCP$_1$). On the other hand, while Algorithm 4.1 based on bisections can be proven to be convergent (see Section 4.3), the same is in general not true for the variant of Algorithm 4.1, which employs only ω-subdivisions, as the counterexample in Section 4.5 shows.

If we solve problems of type (DCP$_1$), we can choose $\bar{\epsilon}^k = \bar{\delta}^k = \bar{\rho}^k = 0$ ($k \in \mathbb{N}_0$) in the initialization of Algorithm 4.1 since (DCP$_1^S$) is a linear optimization problem, i.e., we can assume that (DCP^S) is solvable exactly in finite time (see Remark 4.2.1 (a)). Furthermore, it follows in this situation that (GWSR) coincides with the classical ω-subdivision rule, i.e., $w^k = \omega(S^k)$ for any $k \in \mathbb{N}$. Even for this case the convergence of the presented approach based on ω-subdivisions was an open question.

Some mixed approaches for solving problems of type (DCP$_1$) were proposed in the literature, in which both bisection and ω-subdivision are used. These are the so-called normal algorithms, first introduced and proven to be convergent in [TUY91B] for conical algorithms and extended in [HT96B] to simplicial algorithms. These normal algorithms could be further extended in a straightforward way to a convergent solution method for Problem (DCP) by combining Algorithm 4.1 with the special subdivision strategy used in these approaches.

Nevertheless, as already mentioned in the introduction of this chapter, we will prove the convergence of the proposed approach only employing ω-subdivisions, applied for solving problems of type (DCP$_1$). Under some assumptions, which are particularly fulfilled for all-quadratic problems, we are also able to prove a slightly weaker convergence result for problems of type (DCP$_2$). Before discussing these convergence results in Section 4.4 we show, first of all, the convergence of Algorithm 4.1 based on an exhaustive subdivision rule.

4.3. Convergence with Exhaustive Subdivision Rules

If we set $\epsilon = \delta = \rho = 0$ in the initialization of Algorithm 4.1 and if we use an exhaustive subdivision rule, then we are able to prove the convergence of the approach presented in the previous section. This is the content of this section. Proving the convergence we obtain finiteness of Algorithm 4.1 for ϵ, δ, $\rho > 0$. First we recall the definition of an exhaustive subdivision rule in a simplicial branch-and-bound algorithm (see [HPT95, Definition 3.5] and [HT96B, Definition 4.10]).

DEFINITION 4.3.1. *A nested sequence of simplices $\{S^k\}_{k\in\mathbb{N}}$, $S^{k+1} \subset S^k$ ($k \in \mathbb{N}$) is called **exhaustive**, if S^k shrinks to a unique point $s \in \mathbb{R}^n$ as $k \to \infty$, i.e.,*

$$\lim_{k\to\infty} S^k \;=\; \bigcap_{k=1}^{\infty} S^k \;=\; \{s\}\,. \tag{4.3.1}$$

*Within a simplicial branch-and-bound algorithm, a subdivision rule is called **exhaustive**, if every nested subsequence of simplices generated throughout the algorithm is exhaustive.*

The bisection defined in the previous section is an exhaustive subdivision rule for simplices (see, e.g., [HOR76, KEA78]), as already pointed out in Chapter 3 (see Remark 3.4.1(a)). A – still exhaustive – generalization of the classical bisection is given in [HPT95, HOR97]. In this **generalized bisection** w^k is chosen as

$$w^k \;=\; \lambda^k v^k_{i_0} + (1-\lambda^k)v_{i_1} \;\; (k \in \mathbb{N}) \tag{4.3.2}$$

with $\lambda^k \in [c, 0.5]$, $c > 0$ ($k \in \mathbb{N}$) and $i_0, i_1 \in \{0, \dots, n\}$ defined as in (4.2.12).

If Algorithm 4.1 employs only an exhaustive subdivision rule, the convergence of the presented solution method for Problem (DCP) can be shown.

THEOREM 4.3.1. *Assume that $\epsilon = \delta = \rho = 0$ and that an exhaustive subdivision rule is used. Then Algorithm 4.1 is convergent in the following sense: If Algorithm 4.1 generates an infinite sequence $\{S^k\}_{k\in\mathbb{N}}$ of simplices, then every accumulation point $\omega^\star$ of the corresponding point sequence $\{\omega(S^k)\}_{k\in\mathbb{N}}$ is an optimal solution of Problem (DCP).*

PROOF: Let $\omega^\star$ be an accumulation point of the sequence $\{\omega(S^k)\}_{k\in\mathbb{N}}$ and let $\{\omega(S^{k_q})\}_{q\in\mathbb{N}}$ be a subsequence converging to $\omega^\star$. Without loss of generality we assume that $\{S^{k_q}\}_{q\in\mathbb{N}}$ is an infinite nested sequence of simplices. Since $\{S^k\}_{k\in\mathbb{N}}$ is generated by an exhaustive subdivision rule there exists a point $s \in \mathbb{R}^n$ with

$$\lim_{q\to\infty} S^{k_q} \;=\; \bigcap_{q=1}^{\infty} S^{k_q} \;=\; \{s\}\,. \tag{4.3.3}$$

From the calculation of $\omega(S^{k_q})$ ($q \in \mathbb{N}$) we know that, for each $q \in \mathbb{N}$, there exists a number $k(q) \in \mathbb{N}$ satisfying

$$k_{q-1} \;\le\; k(q) \;<\; k_q\,, \tag{4.3.4.a}$$

$$a_j^T\omega(S^{k_q}) - b_j \;\le\; \bar{\rho}^{k(q)} \qquad j = 1, \dots, m\,, \tag{4.3.4.b}$$

$$(\bar{v}_i^{S^{k_q}})^T \omega(S^{k_q}) - c_i^{S^{k_q}} \;\le\; \bar{\rho}^{k(q)} \qquad i = 0, \dots, n\,, \tag{4.3.4.c}$$

$$g^l(\omega(S^{k_q})) + \varphi^l_{S^{k_q}}(\omega(S^{k_q})) \;\le\; \bar{\delta}^{k(q)} \qquad l = 1, \dots, p \tag{4.3.4.d}$$

and

$$g^0(\omega(S^{k_q})) + \varphi^0_{S^{k_q}}(\omega(S^{k_q})) \;\le\; \mu(S^{k_q}) + \bar{\epsilon}^{k(q)} \tag{4.3.4.e}$$

($k(q)$ is the iteration in which S^{k_q} has been generated). Because of the affine independence of $v_0^{k_q}, \dots, v_n^{k_q}$ we know further that, for each $q \in \mathbb{N}$, there exists a unique $\lambda^q \in \{\lambda \in \mathbb{R}^{n+1} : \sum_{i=0}^n \lambda_i = 1\}$ with

$$\omega(S^{k_q}) \;=\; \sum_{i=0}^n \lambda_i^q v_i^{k_q}\,. \tag{4.3.5}$$

With (4.2.3.a), (4.2.3.b) and (4.3.4.c) we obtain, for each $j \in \{0, \dots, n\}$,

$$(\bar{v}_j^{S^{k_q}})^T \omega(S^{k_q}) - c_j^{S^{k_q}} \;= \sum_{i=0}^n \lambda_i^q \left((\bar{v}_j^{S^{k_q}})^T v_i^{k_q} - c_j^{S^{k_q}} \right) = -\lambda_j^q \;\le\; \bar{\rho}^{k(q)}\,.$$

The sequence $\{\bar{\rho}^k\}_{k \in \mathbb{N}}$ is bounded. Therefore, by passing to a subsequence, if necessary, we can assume that there holds

$$\lambda^q \;\to\; \bar{\lambda}\ (q \to \infty)$$

and, in particular, $\bar{\lambda} \in B_n$. Because of (4.3.3) each vertex sequence $\{v_i^{k_q}\}_{q \in \mathbb{N}}$ ($i = 0, \dots, n$) converges to s. It follows that

$$\omega(S^{k_q}) \;=\; \sum_{i=0}^n \lambda_i^q v_i^{k_q} \;\to\; \sum_{i=0}^n \bar{\lambda}_i s \;=\; s\ (q \to \infty)\,. \tag{4.3.6}$$

Now with respect to (4.3.4.b) it follows, for $j \in \{1, \dots, m\}$,

$$\begin{array}{ccccc}
a_j^T \omega(S^{k_q}) - b_j & & \le & \bar{\rho}^{k(q)} & \\
\downarrow & & (q \to \infty) & \downarrow & \\
a_j^T \;\; s \;\; - b_j & & \le & 0 & ,
\end{array}$$

and with (4.3.4.d) we obtain, for each $l \in \{1, \dots, p\}$,

$$\begin{array}{cccccc}
g^l(\omega(S^{k_q})) & + \sum\limits_{i=0}^n \lambda_i^q & f^l(v_i^{k_q}) & \le & \bar{\delta}^{k(q)} & \\
\downarrow & \downarrow & \downarrow & (q \to \infty) & \downarrow & \\
g^l(s) & + \sum\limits_{i=0}^n \bar{\lambda}_i & f^l(s) & \le & 0 & .
\end{array}$$

Note that f^l and g^l ($l \in \{0, \dots, p\}$) are continuous functions (see, e.g., [ROC70, Theorem 10.1]). Therefore, we know that s is a feasible point, i.e., $F \neq \emptyset$. From the construction of $\mu^{k_q} = \mu(S^{k_q})$ ($q \in \mathbb{N}$) it follows that $\{\mu^{k_q}\}_{q \in \mathbb{N}}$ is a nondecreasing sequence (see Remark 4.2.2(b)), which is bounded from above by the optimal value $(f^0 + g^0)^\star < \infty$ of (DCP) and hence convergent to a real value $\mu^\star$.

Using (4.3.4.e) we also obtain

$$\begin{array}{ccccccccc}
g^0(\omega(S^{k_q})) & + \sum\limits_{i=0}^{n} \lambda_i^q & f^0(v_i^{k_q}) & \leq & \mu(S^{k_q}) & + \bar{\epsilon}^{k(q)} & \leq & (f^0 + g^0)^\star & + \bar{\epsilon}^{k(q)} \\
\downarrow & \downarrow & \downarrow & (q \to \infty) & \downarrow & \downarrow & & & \downarrow \\
g^0(s) & + \sum\limits_{i=0}^{n} \bar{\lambda}_i & f^0(s) & \leq & \mu^\star & + \;\; 0 & \leq & (f^0 + g^0)^\star & + \;\; 0 \;\; ,
\end{array}$$

and because of the feasibility of s there holds

$$g^0(s) + f^0(s) \; = \; (f^0 + g^0)^\star \, ,$$

showing the optimality of s. Since the limit of a convergent sequence is unique, it follows from (4.3.6) that

$$s \; = \; \omega^\star \, ,$$

and we have proven the theorem. ■

The following corollary is a direct consequence of the proof of this theorem.

COROLLARY 4.3.2. *Assume that $\epsilon = \delta = \rho = 0$ and that an exhaustive subdivision rule is used, then Algorithm 4.1 stops after a finite number of iterations, if no feasible point exists, i.e., if $F = \emptyset$.*

REMARK 4.3.1. If an exhaustive subdivision rule is used, it is possible to avoid nonlinear subproblems (see, for example, [HT99]). Let $S = [v_0, \dots, v_n]$ be an n-simplex and $x_S \in S$ be an arbitrary point, e.g., $x_S = \frac{1}{n+1} \sum_{i=0}^{n} v_i$. Let further, for $l \in \{0, \dots, p\}$, ξ_S^l be a subgradient of g^l at the point x_S, i.e., $\xi_S^l \in \partial g^l(x_S)$, where $\partial g^l(x_S)$ denotes the subdifferential of g^l at the point x_S (see, e.g., [ROC70, SHO85] or Appendix B for the definition and the framework of subgradients and subdifferentials of convex functions). Then we know that the optimal solution $\bar{\mu}^\star(S)$ of the linear problem

$$\begin{aligned} \min\ (\xi_S^0)^T(x-x_S)+g^0(x_S)+\varphi_S^0(x) & \\ (\xi_S^l)^T(x-x_S)+g^l(x_S)+\varphi_S^l(x)\ \le\ 0 \qquad & l=1,\dots,p \\ x\in P\cap S & \end{aligned} \tag{LDCP^S}$$

is a lower bound for $\min_{x\in F\cap S}\left[g^0(x)+f^0(x)\right]$. If we replace in the formulation of Algorithm 4.1 the convex relaxation (DCP^S) of (DCP) with respect to the simplex S by the linear relaxation (LDCP^S), then the algorithm is still convergent (see again [HT99, Theorem 14]). Applying this concept to all-quadratic problems of type (QP) leads to the lower bounds, which we used in Algorithm 3.1. Note that (LP^S) and (LDCP^S) coincide, if x_S is chosen as v_0.

The choice between (DCP^S) and (LDCP^S) is a question of efficiency. The convex subproblems are in general harder to solve, but provide a better lower bound, since the objective function as well as the nonlinear constraints are better approximated. If an efficient solution method for (DCP^S) is available, the use of this method could lead to a faster algorithm than the use of a linear problem solver for (LDCP^S) (see the numerical results in Subsection 4.6.1). Whether an efficient solver for (DCP^S) exists, depends on the special structure of this convex subproblem. For convex quadratic subproblems, for example, interior point methods can be used if some additional assumptions are fulfilled (see, e.g., [Jar96]).

The previous convergence result for Algorithm 4.1 is not really surprising, since an exhaustive subdivision rule is assumed. In the next section we prove a similar convergence result for this method in the case that the ω-subdivision rule is employed and the approach is used for solving problems of type (DCP$_1$) and (DCP$_2$). Recognize that the ω-subdivision rule is not necessarily exhaustive.

4.4. Convergence with the ω-Subdivision Rule

In this section we assume that a CONVEXSOLVER$_{0,0,0}$ is known. If we apply Algorithm 4.1 for solving problems of type (DCP$_1$), we can use the Simplex-Algorithm as mentioned in Remark 4.2.1(a). For general problems of type (DCP$_2$) such a solution method does not exist to the author's knowledge. However, for example in the case that g^0 is a quadratic function and thus (DCP$_2^S$) is a convex quadratic optimization problem with linear constraints, a CONVEXSOLVER$_{0,0,0}$ is available (see Remark 4.2.1(b)). The existence of such a solution method for the convex subproblems implies that we choose 0-sequences for $\{\bar{\epsilon}^k\}_{k\in\mathbb{N}}$, $\{\bar{\delta}^k\}_{k\in\mathbb{N}}$

and $\{\bar{\rho}^k\}_{k\in\mathbb{N}}$ in the initialization of Algorithm 4.1, i.e., $\bar{\epsilon}^k = \bar{\delta}^k = \bar{\rho}^k = 0$ ($k \in \mathbb{N}$). Furthermore, in view of Remark 4.2.2(h) we know that, for each simplex $S^k = [v_0^k, \dots, v_n^k]$, the optimal solution $\omega(S^k)$ of the corresponding convex subproblem (DCP$^{S^k}$) is contained in the set $S^k \setminus \{v_0^k, \dots, v_n^k\}$. This implies that in the used generalized ω-subdivision rule (GWSR) we choose in each iteration $k \in \mathbb{N}$ the point w^k as $\omega(S^k)$.

As a consequence of the use of a CONVEXSOLVER$_{0,0,0}$ we set $\rho = 0$, since the linear constraints describing P are also involved in the description of the feasible set of the subproblems (DCP^S). A second consequence is that we set $\delta = 0$, if we apply Algorithm 4.1 for solving problems of type (DCP$_1$). Indeed, since the constraints of (DCP$_1$) are linear ($p = 0$), they are not relaxed in the formulation of (DCP$_1^S$). In this situation we know that each point $\omega(S)$ generated in Algorithm 4.1 for an arbitrary n-simplex S is feasible for the original Problem (DCP$_1$), i.e., $\omega(S) \in F$.

We assume further that in the formulation of Problem (DCP$_2$) the function f^0 is strictly concave, i.e., for any $x, y \in \mathbb{R}^n$ with $x \neq y$ and $\lambda \in (0, 1)$, there holds

$$f^0(\lambda x + (1-\lambda)y) \; > \; \lambda f^0(x) + (1-\lambda) f^0(y) \, . \tag{4.4.1}$$

This is not a restriction. The function $\bar{f}^0(x) := f^0(x) - \sigma\|x\|_2^2$ with a real value $\sigma > 0$ is strictly concave and $\bar{g}^0(x) + \sigma\|x\|_2^2$ is still convex. Therefore, we can solve

$$\min_{x\in F} \left[\bar{g}^0(x) + \bar{f}^0(x)\right]$$

with a strictly concave part of the objective function instead of $\min\limits_{x\in F} \left[g^0(x) + f^0(x)\right]$.

In the following we consider the version of Algorithm 4.1, which employs only ω-subdivisions. In order to obtain the desired convergence results for this approach, applied for solving problems of type (DCP$_1$) and (DCP$_2$), we show that this method is always finite, if the tolerance ϵ is chosen greater than 0, and in the case of (DCP$_2$), if δ is also greater than 0. This proof of finiteness is based on some lemmata and corollaries whose statements are presented in the sequel. Even though the pronounced results will be the same, some proofs are different depending on the problem class which we would like to solve with Algorithm 4.1. Each proof will be marked in a non-ambiguous way in order to clarify whether it is true for both types or only for one. The longer and more technical proofs of some results will be given in Appendix A.

The proof of finiteness will be done by contradiction. The results of the following lemmata will be proven using an infinite nested sequence $\{S^k\}_{k\in\mathbb{N}}$ of simplices with the properties that, for all $k \in \mathbb{N}$,

$$S^{k+1} \text{ is the } \textit{direct child} \text{ of } S^k = [v_0^k, \dots, v_n^k] \ , \tag{4.4.2.a}$$

$$\text{i. e., } S^{k+1} = [v_0^k, \dots, v_{i-1}^k, \omega(S^k), v_{i+1}^k, \dots, v_n^k] \quad (i \in \{0, \dots, n\}) \, ,$$

and

$$\mu^k = \mu(S^k) < \eta^k - \epsilon \, . \tag{4.4.2.b}$$

If the algorithm does not stop after a finite number of iterations, then – given the sequence $\{S^k\}_{k\in\mathbb{N}}$ of simplices which are selected in the simplex selection rule (SSR) of Algorithm 4.1 – there exists at least one subsequence with the additional attributes (4.4.2), as it will be shown in the proof of the Finiteness Theorem 4.4.9.

If we consider, for $k \in \mathbb{N}$, an n-simplex S^k and its direct child S^{k+1}, then it is a known fact (see, e.g., [HPT95, Theorem 1.23] or Remark 3.2.2(c)) that, for any $x \in S^{k+1}$, there holds

$$\varphi^l_{S^{k+1}}(x) \ = \ \varphi^l_{S^k}(x) + \tau^l(x) \qquad l = 0, \dots, p \, , \tag{4.4.3}$$

where $\tau^l : \mathbb{R}^n \to \mathbb{R}$ ($l \in \{0, \dots, p\}$) denotes a function with nonnegative values. The following lemma specifies, for each $x \in S^{k+1}$ and at least one $l \in \{0, \dots, p\}$, a lower bound for the function value $\tau^l(x)$. This lower bound depends on the barycentric coordinates of x with respect to S^{k+1} and on the tolerance ϵ, respectively δ.

Lemma 4.4.1. *Let S^k be the selected simplex in iteration $k \in \mathbb{N}$ of Algorithm 4.1 with $S^k = [v_0^k, \dots, v_{i-1}^k, v_i^k, v_{i+1}^k, \dots, v_n^k]$ ($i \in \{0, \dots, n\}$) and let $S^\star = [v_0^k, \dots, v_{i-1}^k, v^\star, v_{i+1}^k, \dots, v_n^k]$ be one of the simplices obtained by subdividing S^k with respect to $v^\star = \omega(S^k)$.*

Let x be an arbitrary element of $S^\star$ with the unique representation

$$x = \lambda_0 v_0^k + \dots + \lambda_i v^\star + \dots + \lambda_n v_n^k \, ,$$

with $\lambda \in B_n$. If S^k is not fathomed in the pruning rule (PR) of Algorithm 4.1, then there holds

$$\left.\begin{array}{r} \varphi^0_{S^\star}(x) \geq \varphi^0_{S^k}(x) + \epsilon\lambda_i \quad , \textit{if } v^\star \textit{ is } (\delta, 0)\textit{- feasible,} \\ \textit{or} \qquad\qquad \\ \exists l \in \{1, \dots, p\} : \ \varphi^l_{S^\star}(x) \geq \varphi^l_{S^k}(x) + \delta\lambda_i \quad , \textit{otherwise.} \qquad\quad \end{array}\right\} \tag{4.4.4}$$

PROOF FOR (DCP$_1$) AND (DCP$_2$): Using (4.2.1) and the linearity of the convex envelope $\varphi^l_{S^k}$ we know that, for $l \in \{0, \dots, p\}$, there holds

$$\varphi^l_{S^\star}(x) \;=\; \sum_{j=0, j\neq i}^{n} \lambda_j f^l(v^k_j) + \lambda_i f^l(v^\star)\,,$$

and

$$\varphi^l_{S^k}(x) \;=\; \sum_{j=0, j\neq i}^{n} \lambda_j f^l(v^k_j) + \lambda_i \varphi^l_{S^k}(v^\star)\,.$$

If $v^\star$ is (δ, 0)-feasible, this point was used for updating the current upper bound η^k in an earlier iteration. It follows that

$$g^0(v^\star) + f^0(v^\star) \;\geq\; \eta^k\,.$$

Since S^k is not fathomed in the pruning rule (PR) of Algorithm 4.1, there holds

$$\mu^k \;=\; g^0(v^\star) + \varphi^0_{S^k}(v^\star) \;<\; \eta^k - \epsilon\,,$$

and the first conclusion follows immediately.

If $v^\star$ is not (δ, 0)-feasible, then there must exist an index $l \in \{1, \dots, p\}$ with the property

$$g^l(v^\star) + f^l(v^\star) \;>\; \delta\,.$$

However, we know that $v^\star$ is feasible with respect to the constraints describing (DCP^S), i.e.,

$$g^l(v^\star) + \varphi^l_{S^k}(v^\star) \;\leq\; 0\,,$$

which proves the second part of (4.4.4). ■

With the result (4.4.4) we are now able to show that, given a nested sequence $\{S^k\}_{k\in\mathbb{N}}$ of simplices and a sufficiently large number $K \in \mathbb{N}$, at least one vertex of the residual simplices S^k ($k \geq K$) will be fixed. If we apply Algorithm 4.1 for solving problems of type (DCP$_1$), then we know that each generated optimal solution $\omega(S^k)$ ($k \in \mathbb{N}$) of the linear subproblem (DCP$^{S^k}$) is feasible. Recognize that a CONVEXSOLVER$_{0,0,0}$ is assumed to be used. Therefore, it is easy to see that in this situation at least one vertex of the simplices S^k ($k \in \mathbb{N}$) must be fixed. Indeed, if all vertices $v^k_0, \dots, v^k_n$ of S^k have been changed at least once, then they are all feasible. This means that the current upper bound η^k must be lower than or equal to the minimal value of f^0 with respect to $v^k_0, \dots, v^k_n$. Thus, the function values of the convex envelope $\varphi^0_{S^k}$ on the simplex S^k are higher than or equal to η^k

(compare with (4.2.1)), and the simplex S^k must be fathomed in the pruning rule (PR) of Algorithm 4.1 (see also the proof of Lemma 4.7.1).

In order to prove that one vertex must be fixed in the case of (DCP$_2$) we need more technical effort, since $\omega(S^k)$ is not necessarily (δ, 0)-feasible. However, applying this necessary effort, we obtain a stronger result. We are able to show that the nested simplex sequence shrinks to a lower-dimensional simplex S, where S is given by the fixed vertices of the residual simplices S^k ($k \geq K$). In Lemma 4.4.2 this stronger result is formulated and the corresponding proof is presented in Section A.1. In this proof we do not use the strict concavity of f^0. Thus, this proof, and consequently the following lemma, is also valid in the case of (DCP$_1$).

LEMMA 4.4.2. *Let $\{S^k\}_{k\in\mathbb{N}}$ be an infinite nested sequence of simplices generated by Algorithm 4.1 with Properties (4.4.2). Then there exist a number $K \in \mathbb{N}$ and an integer r with $0 \leq r < n$ such that, for each $k \geq K$, there holds*

$$S^k = [v_0, \ldots, v_r, v_{r+1}^k, \ldots, v_n^k] \ , \tag{4.4.5}$$

where $v_0, \ldots, v_r$ are fixed vectors, while $v_{r+1}^k, \ldots, v_n^k$ ($k \in \mathbb{N}, k \geq K$) change infinitely often. Moreover, there holds

$$\bigcap_{k\in\mathbb{N}} S^k \ = \ [v_0, \ldots, v_r] \ =: \ S\,. \tag{4.4.6}$$

In order to show that the number r of fixed vertices of the residual simplices S^k ($k \geq K$) must be greater than 1, i.e., $r \geq 1$, we first prove that each accumulation point of the sequence $\{\omega(S^k)\}_{k\in\mathbb{N}}$ is contained in the set $S \setminus \{v_0, \ldots, v_r\}$. This is the result of the following lemma. The proof of this lemma, which also does not depend on the considered problem class, is given in Section A.2.

LEMMA 4.4.3. *Let $\{S^k\}_{k\in\mathbb{N}}$ be an infinite nested sequence of simplices generated by Algorithm 4.1 with Properties (4.4.2). Let $K \in \mathbb{N}$ and $0 \leq r < n$ be given by Lemma 4.4.2. Denote by $S = [v_0, \ldots, v_r]$ the fixed face of the residual simplices*

$$S^k \ = \ [v_0, \ldots, v_r, v_{r+1}^k, \ldots, v_n^k] \quad (k \geq K)\,.$$

Then, for each accumulation point $\bar{\omega}$ of the sequence $\{\omega(S^k)\}_{k\in\mathbb{N}}$, there holds

$$\bar{\omega} \ \in \ S \setminus \{v_0, \ldots, v_r\}\,. \tag{4.4.7}$$

This result will also be helpful in the proof of the next Lemma 4.4.5. However, a direct consequence of the previous result is that at least two vertices of the residual simplices S^k ($k \geq K$) have to be fixed.

COROLLARY 4.4.4. *Let $\{S^k\}_{k\in\mathbb{N}}$ be an infinite nested sequence generated by Algorithm 4.1 with Properties (4.4.2). Let further $K \in \mathbb{N}$ and $0 \leq r < n$ be given by Lemma 4.4.2. Then there holds*

$$r \geq 1\,. \tag{4.4.8}$$

PROOF FOR (DCP$_1$) AND (DCP$_2$): Assume, by contradiction, that there holds $r = 0$, i.e.,

$$S \;=\; \{v_0\} \;=\; \bigcap_{k\in\mathbb{N}} S^k\,.$$

The sequence $\{\omega(S^k)\}_{k\in\mathbb{N}}$ is bounded. Therefore there exists an accumulation point $\bar{\omega}$ of this sequence. Since the simplex sequence $\{S^k\}_{k\in\mathbb{N}}$ consists of nested, compact and non-empty sets, it follows

$$\bar{\omega} \;\in\; S \;=\; \{v_0\}\,.$$

This is a contradiction to the result of Lemma 4.4.3, and hence we obtain $r \geq 1$. ■

REMARK 4.4.1. The result of the previous corollary follows also by the considerations regarding an exhaustive subdivision rule in Section 4.3. Indeed, if we are in the situation that only one vertex of the simplex sequence $\{S^k\}_{k\geq K}$ is fixed, then it follows by Lemma 4.4.2 that there holds

$$S \;=\; \{v_0\} \;=\; \bigcap_{k\in\mathbb{N}} S^k\,. \tag{4.4.9}$$

This relation is the essential part in the proof of Theorem 4.3.1, which guarantees the convergence of Algorithm 4.1 for an exhaustive subdivision rule. Therefore, by the same argumentation as in the corresponding proof (see Section 4.3) we would obtain finiteness of Algorithm 4.1 for ϵ, $\delta > 0$, if (4.4.9) is satisfied.

In order to prove finiteness of the variant of Algorithm 4.1, which employs only ω-subdivisions, it is not sufficient that each accumulation point $\bar{\omega}$ of the sequence $\{\omega(S^k)\}_{k\in\mathbb{N}}$ is contained in the set $[v_0,\ldots,v_r] \setminus \{v_0,\ldots,v_r\}$. Actually, we need a slightly stronger result. The next lemma signifies that the barycentric coordinates of $\omega(S^k)$ ($k \geq K$) with respect to the not-fixed vertices of S^k ($k \geq K$) converge to 0. The proofs of this lemma, which are different for the considered problem classes, are given in Section A.3.

LEMMA 4.4.5. *Let $\{S^k\}_{k\in\mathbb{N}}$ be an infinite nested sequence generated by Algorithm 4.1 with Properties (4.4.2). Let $K \in \mathbb{N}$ and $1 \leq r < n$ be chosen as in Lemma 4.4.2 and let*

$$\omega(S^k) = \sum_{i=0}^{r} \lambda_i^k v_i + \sum_{i=r+1}^{n} \lambda_i^k v_i^k, \tag{4.4.10}$$

with $\lambda^k \in B_n$ and $k \geq K$. Then there holds

$$\Lambda^k := \sum_{i=r+1}^{n} \lambda_i^k \longrightarrow 0 \ \ (k \to \infty)\,. \tag{4.4.11}$$

It follows immediately from the previous lemma that, for each $k \geq K$, the optimal solution $\omega(S^k)$ of the linear subproblem (DCP$_1^S$) can be represented as a combination of a point x^k, contained in the fixed face $S = [v_0, \ldots, v_r]$ of the simplices S^k, and a residual $\varsigma^k \in \mathbb{R}^n$. This representation is given in the following corollary. Furthermore, it is shown that the distance between the function values of $\varphi^0_{S^k}$ at the points $\omega(S^k)$ and x^k converges to 0.

COROLLARY 4.4.6. *Let $\{S^k\}_{k\in\mathbb{N}}$ be an infinite nested sequence generated by Algorithm 4.1 with Properties (4.4.2), and let $K \in \mathbb{N}$ and $1 \leq r < n$ be given by Lemma 4.4.2. Then there exists a number $\tilde{K} \in \mathbb{N}$ with $\tilde{K} \geq K$ such that, for each $k \geq \tilde{K}$, there exist a point $x^k \in [v_0, \ldots, v_r]$, a point $\varsigma^k \in \mathbb{R}^n$ with $\|\varsigma^k\|_2 \to 0$ $(k \to \infty)$ and a real value σ^k with $\sigma^k \to 0$ $(k \to \infty)$ satisfying*

$$\omega(S^k) = x^k + \varsigma^k \tag{4.4.12}$$

and

$$\varphi^0_{S^k}(\omega(S^k)) = \varphi^0_{S^k}(x^k) + \sigma^k\,. \tag{4.4.13}$$

PROOF FOR (DCP$_1$) AND (DCP$_2$): In view of Relation (4.4.11) we know that there exists an integer $\tilde{K} \geq K$ such that, for each $k \geq \tilde{K}$, there holds

$$\Lambda^k \ < \ 1\,.$$

Set, for $k \geq \tilde{K}$,

$$x^k := \sum_{i=0}^{r} \frac{\lambda_i^k}{1 - \Lambda^k} v_i$$

with $\lambda^k \in B_n$ and $\Lambda^k \in \mathbb{R}$ defined as in (4.4.10) and (4.4.11). Obviously there holds $x^k \in [v_0, \ldots, v_r]$, and we obtain the following representation of $\omega(S^k)$

$$\omega(S^k) = x^k + \underbrace{\sum_{i=r+1}^{n} \lambda_i^k v_i^k - \Lambda^k \sum_{i=0}^{r} \frac{\lambda_i^k}{1-\Lambda^k} v_i}_{=:\, \varsigma^k} .$$

For the function value of $\varphi^0_{S^k}$ at the point $\omega(S^k)$ we further get

$$\varphi^0_{S^k}(\omega(S^k)) = \varphi^0_{S^k}(x^k) + \underbrace{\sum_{i=r+1}^{n} \lambda_i^k f^0(v_i^k) - \Lambda^k \sum_{i=0}^{r} \frac{\lambda_i^k}{1-\Lambda^k} f^0(v_i)}_{=:\, \sigma^k} .$$

Because of the boundedness of S^0, there exist real values $C > 0$ and $D > 0$ such that, for each $x \in S^0$, there holds $\|x\|_2 \leq C$ and $|f^0(x)| \leq D$. With Lemma 4.4.5 it follows

$$\|\varsigma^k\|_2 = \| \sum_{i=r+1}^{n} \lambda_i^k v_i^k - \Lambda^k x^k \|_2 \leq 2\Lambda^k C \to 0 \ \ (k \to \infty) .$$

Furthermore, we obtain

$$\begin{aligned} |\sigma^k| \ &= \ | \sum_{i=r+1}^{n} \lambda_i^k f^0(v_i^k) - \Lambda^k \sum_{i=0}^{r} \frac{\lambda_i^k}{1-\Lambda^k} f^0(v_i)| \\ &\leq \ \Lambda^k D + \Lambda^k \underbrace{\sum_{i=0}^{r} \frac{\lambda_i^k}{1-\Lambda^k}}_{=1} D \to 0 \ \ (k \to \infty) . \end{aligned}$$

■

By using all previous results it is now possible to prove that there exists a number $\bar{K} \geq \tilde{K}$ such that, for each $k \geq \bar{K}$, we are able to replace the point $x^k \in S = [v_0, \ldots, v_r]$ in Relation (4.4.12) by a point $r^k \in S$ with the property that a lower bound for $\mu^k = \mu(S^k)$ depending on r^k can be given. This lower bound is the sum of the function value $g^0(r^k) + \varphi^0_{S^k}(r^k)$ of the objective function of Problem (DCP$^{S^k}$) and a residual part depending on Λ^k, which converges to 0 – with respect to Λ^k – slower than the Euclidean distance between the points $\omega(S^k)$ and r^k. This is the result of the following Lemma 4.4.7 and will be essential for the proof of finiteness of the version of Algorithm 4.1, which employs only

ω-subdivisions. The proofs of this lemma, which are again different for both problem classes, are given in Section A.4. The notation $o(x)$ is used for an arbitrary function $\tau : \mathbb{R} \to \mathbb{R}$ with the property

$$\frac{\tau(x)}{x} \;\to\; 0 \quad (x \to 0)\,.$$

LEMMA 4.4.7. *Let $\{S^k\}_{k\in\mathbb{N}}$ be an infinite nested simplex sequence generated by Algorithm 4.1 with Properties (4.4.2). Then there exist a number $\bar{K} \in \mathbb{N}$, a real value $\sigma > 0$ and, for each $k \geq \bar{K}$, a point $r^k \in [v_0, \ldots, v_r]$ satisfying*

$$g^0(\omega(S^k)) + \varphi^0_{S^k}(\omega(S^k)) \;\geq\; g^0(r^k) + \varphi^0_{S^k}(r^k) + \sigma\Lambda^k + o(\Lambda^k)\,, \quad \text{(4.4.14.a)}$$

$$\varphi^l_{S^k}(r^k) \;\leq\; o(\Lambda^k) \qquad l = 1, \ldots, p \quad \text{(4.4.14.b)}$$

and

$$\|\omega(S^k) - r^k\|_2 \;=\; o(\Lambda^k) \quad \text{(4.4.14.c)}$$

with Λ^k ($k \geq \bar{K}$) defined as in (4.4.11).

In order to obtain a contradiction in the proof of the final finiteness result it is still not sufficient that for each $k \geq \bar{K}$ we have got a point $r^k \in S = [v_0, \ldots, v_r]$ with Properties (4.4.14.a)-(4.4.14.c). Beyond it, we need a point $\bar{r}^k \in S$ with Properties (4.4.14.a) and (4.4.14.c), which is, additionally, feasible with respect to the convex subproblem (DCP$^{S^k}$). Thus, it is necessary to prove the existence of a point $\bar{r}^k$ satisfying (4.4.14.a), (4.4.14.c) and

$$\bar{r}^k \in P \quad , \quad \varphi^l_{S^k}(\bar{r}^k) \leq 0 \quad l = 1, \ldots, p\,. \quad \text{(4.4.15)}$$

Since the convex envelopes $\varphi^l_{S^k}$ ($l = 1, \ldots, p$, $k \geq K$) have, in view of the result of Lemma 4.4.2, the same function values on S independent of k, it follows, for each $x \in S$ and $k \geq K$,

$$\varphi^l_{S^k}(x) \;=\; \sum_{i=0}^{r} \lambda_i f^l(v_i)$$

with $\lambda \in B_r$, $x = \sum_{i=0}^r \lambda_i v_i$. Therefore, Condition (4.4.15) is fulfilled, if $\bar{r}^k$ is contained in the set

$$\bar{F} := \{x \in P \cap S : \textstyle\sum_{i=0}^r \lambda_i f^l(v_i) \leq 0\,,\; l = 1, \ldots, p$$
$$\qquad\qquad \text{with } \lambda \in B_r\,,\; x = \textstyle\sum_{i=0}^r \lambda_i v_i\}\,.$$

Note that in the case of problem class (DCP$_1$) there even holds $\bar{F} = P \cap S$. The proof of the next lemma presented in Section A.5 shows that the orthogonal projection of r^k on the set $\bar{F}$ satisfies Conditions (4.4.14.a), (4.4.14.c) and (4.4.15).

LEMMA 4.4.8. *Let $\{S^k\}_{k\in\mathbb{N}}$ be an infinite nested simplex sequence generated by Algorithm 4.1 with Properties (4.4.2). Let $\bar{K} \in \mathbb{N}$ and $\sigma > 0$ be given by Lemma 4.4.7. Then, for each $k \geq \bar{K}$, there exists a point $\bar{r}^k \in \bar{F}$ satisfying*

$$g^0(\omega(S^k)) + \varphi^0_{S^k}(\omega(S^k)) \;\geq\; g^0(\bar{r}^k) + \varphi^0_{S^k}(\bar{r}^k) + \sigma\Lambda^k + o(\Lambda^k) \tag{4.4.16.a}$$

and

$$\|\omega(S^k) - \bar{r}^k\|_2 \;=\; o(\Lambda^k) \tag{4.4.16.b}$$

with Λ^k ($k \geq \bar{K}$) defined as in (4.4.11).

With this last lemma we are now able to prove the postulated finiteness of Algorithm 4.1.

THEOREM 4.4.9. *The variant of Algorithm 4.1, which employs only ω-subdivisions, is finite, if a CONVEXSOLVER$_{0,0,0}$ is used and*

- *in the case of problem class (DCP$_1$), if $\epsilon > 0$ and δ, $\rho = 0$, or*
- *in the case of problem class (DCP$_2$), if ϵ, $\delta > 0$, $\rho = 0$ and f^0 is strictly concave.*

PROOF FOR (DCP$_1$) AND (DCP$_2$): Assume, by contradiction, that Algorithm 4.1 does not stop after a finite number of iterations, i.e., the algorithm generates an infinite sequence $\{S^k\}_{k\in\mathbb{N}}$ of simplices. Then there exists an infinite nested subsequence $\{S^{k_q}\}_{q\in\mathbb{N}} \subset \{S^k\}_{k\in\mathbb{N}}$ with Properties (4.4.2). We know – regarding Lemma 4.4.8 – that there exist a number $\bar{Q} \in \mathbb{N}$, an integer $1 \leq r < n$, a positive real value σ and a point sequence $\{\bar{r}^q\}_{q\geq\bar{Q}}$ such that, for each $q \geq \bar{Q}$, there holds

$$S^{k_q} \;=\; [v_0, \ldots, v_r, v^{k_q}_{r+1}, \ldots, v^{k_q}_n]\,, \tag{4.4.17.a}$$

$$\bar{r}^q \;\in\; \bar{F} \;\subset\; F_{S^{k_q}} = \{x \in S^{k_q} \cap P : \varphi^l_{S^{k_q}}(x) \leq 0\;,\; l = 1, \ldots, p\} \tag{4.4.17.b}$$

and

$$\begin{aligned} &g^0(\omega(S^{k_q})) + \varphi^0_{S^{k_q}}(\omega(S^{k_q})) \\ &\qquad\qquad \geq\; g^0(\bar{r}^q) + \varphi^0_{S^{k_q}}(\bar{r}^q) + \sigma\Lambda^{k_q} + o(\Lambda^{k_q}) \end{aligned} \tag{4.4.17.c}$$

with Λ^{k_q} defined as in (4.4.11).

The point $\omega(S^{k_q})$ is the optimal solution of the convex optimization problem

$$\min_{x \in F_{S^{k_q}}} \left[g^0(x) + \varphi^0_{S^{k_q}}(x) \right] .$$

Therefore, it follows from the feasibility of $\bar{r}^q$ with respect to the set $F_{S^{k_q}}$ (see (4.4.17.b)) that

$$g^0(\omega(S^{k_q})) + \varphi^0_{S^{k_q}}(\omega(S^{k_q})) \ \leq \ g^0(\bar{r}^q) + \varphi^0_{S^{k_q}}(\bar{r}^q) \, , \tag{4.4.18}$$

and from (4.4.17.c) we obtain, for $q \geq \bar{Q}$,

$$g^0(\bar{r}^q) + \varphi^0_{S^{k_q}}(\bar{r}^q) \ \geq \ g^0(\bar{r}^q) + \varphi^0_{S^{k_q}}(\bar{r}^q) + \sigma\Lambda^{k_q} + o(\Lambda^{k_q}) \, .$$

This relation is equivalent to

$$\underbrace{\sigma\Lambda^{k_q}}_{\geq 0} + o(\Lambda^{k_q}) \ \leq \ 0 \, .$$

Considering Lemma 4.4.5 we know that the sequence $\{\Lambda^{k_q}\}_{q\in\mathbb{N}}$ converges to 0, if q tends to infinity, and, furthermore, this sequence converges slower than $\{o(\Lambda^{k_q})\}_{q\in\mathbb{N}}$ to 0. Thus, there must exist a number $q' \geq \bar{Q}$ satisfying

$$\Lambda^{k_{q'}} \ = \ 0 \, . \tag{4.4.19}$$

Indeed, if Λ^{k_q} is always greater than 0, it follows by definition of $o(\Lambda)$ that

$$\begin{array}{rcl} \sigma + \frac{o(\Lambda^{k_q})}{\Lambda^{k_q}} & \leq & 0 \\ \downarrow & & (q \to \infty) \\ 0 \ < \ \sigma + \quad 0 \quad & \leq & 0 \ , \end{array}$$

which is a contradiction.

Relation (4.4.19) is only possible if there holds

$$\omega(S^{k_{q'}}) \ \in \ [v_0, \ldots, v_r] \, .$$

If $\omega(S^{k_{q'}})$ is contained in $[v_0, \ldots, v_r] \setminus \{v_0, \ldots, v_r\}$, $S^{k_{q'}+1}$ will be generated by replacing one of the vertices $v_0, \ldots, v_r$. However, this contradicts Property (4.4.17.a) of the simplex sequence $\{S^{k_q}\}_{q\in\mathbb{N}}$. Thus, there holds

$$\omega(S^{k_{q'}}) \ \in \ \{v_0, \ldots, v_r\} \tag{4.4.20.a}$$

and, additionally,

$$\varphi^l_{S^{k_{q'}}}(\omega(S^{k_{q'}})) \ = \ f^l(\omega(S^{k_{q'}})) \qquad l = 0, \ldots, p \, . \tag{4.4.20.b}$$

The point $\omega(S^{k_{q'}}) \in S^{k_{q'}} \cap P$ is feasible with respect to the convex optimization problem (DCP$^{S^{k_{q'}}}$). In view of (4.4.20.b) we obtain

$$f^l(\omega(S^{k_{q'}})) \;=\; \varphi^l_{S^{k_{q'}}}(\omega(S^{k_{q'}})) \;\leq\; 0 \qquad l = 1, \ldots, p\,,$$

and therefore we know that $\omega(S^{k_{q'}})$ is a (δ, 0)-feasible point of Problem (DCP$_i$) ($i = 1, 2$). Hence, the point $\omega(S^{k_{q'}})$ was used for updating the upper bound, and it follows

$$\begin{aligned}\mu^{k_{q'}} \;=\; \mu(S^{k_{q'}}) \;=\; g^0(\omega(S^{k_{q'}})) + \varphi^0_{S^{k_{q'}}}(\omega(S^{k_{q'}}))& \\ \leq\; \eta^{k_{q'}} \;\leq\; g^0(\omega(S^{k_{q'}})) + f^0(\omega(S^{k_{q'}}))\,.&\end{aligned}$$

In view of Relation (4.4.20.b) with $l = 0$ we obtain

$$\mu^{k_{q'}} \;=\; \eta^{k_{q'}}\,,$$

contradicting, for $\epsilon > 0$, Property (4.4.2.b) of the simplex sequence $\{S^{k_q}\}_{q \in \mathbb{N}}$ and completing the proof. ■

This finiteness result guarantees the convergence of Algorithm 4.1 only employing ω-subdivisions, applied for solving problems of type (DCP$_1$) and (DCP$_2$). However, the convergence of Algorithm 4.1 does not hold in the sense of Theorem 4.3.1, i.e., we do not know whether each accumulation point $\omega^\star$ of the point sequence $\{\omega(S^k)\}_{k \in \mathbb{N}}$ is optimal, if Algorithm 4.1 with $\epsilon = \delta = \rho = 0$ generates an infinite sequence $\{S^k\}_{k \in \mathbb{N}}$ of simplices. From Theorem 4.4.9 we obtain for problem class (DCP$_1$) that, if ϵ is also chosen as 0, then there holds

$$\mu^k \;=\; \mu(S^k) \;\rightarrow\; \min_{x \in P} f^0(x) \qquad (k \rightarrow \infty) \tag{4.4.21}$$

in the infinite case. Note that P is assumed to be non-empty and that $\{\mu(S^k)\}_{k \in \mathbb{N}}$ is by construction non-decreasing. For this problem class we obtain, furthermore, that each accumulation point $x_f^\star$ of the sequence $\{x_f^k\}_{k \in \mathbb{N}}$ is optimal for (DCP$_1$), where x_f^k denotes the best known point at iteration $k \in \mathbb{N}$, i.e., $\eta^k = f^0(x_f^k)$. Thus, in this case there holds a similar convergence result as in Theorem 4.3.1.

In the case of problems of type (DCP$_2$) we obtain from Theorem 4.4.9 convergence of Algorithm 4.1 only in the following sense. For arbitrary accuracies ϵ, $\delta > 0$ we know that this method detects in finite either the emptiness of F or an (ϵ, δ, 0)-solution of Problem (DCP$_2$). This implies that, if $F \neq \emptyset$, we are able to construct a sequence $\{x^k\}_{k \in \mathbb{N}}$ such that each accumulation point of this sequence is an optimal solution of (DCP$_2$). This can be done by successively applying Algorithm 4.1 with different positive accuracies ϵ^k, δ^k ($k \in \mathbb{N}$) belonging to sequences $\{\epsilon^k\}_{k\in\mathbb{N}}$, $\{\delta^k\}_{k\in\mathbb{N}}$ converging to 0. Such an iterative application of Algorithm 4.1 detects in particular the emptiness of F in finite time (compare with Corollary 4.3.2).

REMARK 4.4.2. At first glance the above convergence results for the variant of Algorithm 4.1, which employs only ω-subdivisions and is applied for the solution of problems of type (DCP$_1$) and (DCP$_2$), are weaker than the one obtained in the exhaustive case (Theorem 4.3.1) – especially the result for problem class (DCP$_2$). However – from a practical point of view – these different convergence concepts have the same quality. Note that the stronger convergence of algorithms in the sense of Theorem 4.3.1 is only needed in order to obtain finiteness of such approaches, when approximate solutions are sufficient. Hence, both results show that Algorithm 4.1 with an exhaustive subdivision rule as well as with ω-subivisions is finite, if we are satisfied with (ϵ, δ, 0)-solutions of Problem (DCP$_2$) (ϵ, $\delta > 0$). This is the essential result we need in order to apply this method in practice.

As the counterexample in the next section shows, it is not possible to extend the presented proof techniques in order to ensure the convergence of Algorithm 4.1 – in the above sense – in the general case. However, a careful checking of the proofs shows that all results until Lemma 4.4.7 are also provable in the case of the general problem class (DCP$_3$). Moreover, using a CONVEXSOLVER$_{\bar{\epsilon}^k,\bar{\delta}^k,0}$ ($k \in \mathbb{N}$) with non-increasing sequences $\{\bar{\epsilon}^k\}_{k\in\mathbb{N}}$ and $\{\bar{\delta}^k\}_{k\in\mathbb{N}}$ converging to 0 instead of a CONVEXSOLVER$_{0,0,0}$, it is also possible to verify all these results. However, in this situation it is necessary to change slightly the formulation of some of these lemmata, where these changes do not alter the essential content of the results. For example, the result of Lemma 4.4.1 does not hold for each $k \in \mathbb{N}$. It is only provable that there exists a $K \in \mathbb{N}$ such that Relation (4.4.4) with $\frac{\epsilon}{2}\lambda_i$ instead of $\epsilon\lambda_i$ and $\frac{\delta}{2}\lambda_i$ instead of $\delta\lambda_i$ is true, for each $k \geq K$.

Before presenting the counterexample we would like to give some ideas in order to understand, why the proofs fail in the general case. It is immediately clear, that the Finiteness Theorem 4.4.9 cannot be proven using a CONVEXSOLVER$_{\bar{\epsilon}^k,\bar{\delta}^k,0}$. Indeed, without an exact solution of the convex subproblem (DCP$^{S^k}$) the relation (4.4.18) is not fulfilled, and we are not able to derive the contradiction. This emphasizes that the assumption of a CONVEXSOLVER$_{0,0,0}$ is substantial for the proof of Theorem 4.4.9.

Apart from the problem that it is not reasonable to assume that a CONVEX-SOLVER$_{0,0,0}$ is available in the case of problems of type (DCP$_3$), the attempt of proving the presented finiteness result in this case even fails in the proof of Lemma 4.4.8. We show there (see Appendix A) that the orthogonal projection $\bar{r}^k$ of r^k on the set

$$\bar{F} := \{x \in P \cap S : \textstyle\sum_{i=0}^{r}\lambda_i f^l(v_i) \leq 0\ ,\ l = 1, \ldots, p \\ \text{with } \lambda \in B_r\ ,\ x = \textstyle\sum_{i=0}^{r}\lambda_i v_i\}$$

has the property $\|\bar{r}^k - r^k\|_2 = o(\Lambda^k)$ and, thus, we can derive that this point satisfies the required conditions of Lemma 4.4.8. We prove this property by using the Karush-Kuhn-Tucker(KKT)-conditions (see, e.g., [HOR79, FLE87, MAN94]) for the convex optimization problem $\min_{x\in\bar{F}} \|x - r^k\|_2^2$. Since this problem has only linear constraints we do not need a regularity condition for applying the KKT-theory. If we try to use the same argumentation in the case of problem class (DCP$_3$) with continuous differentiable functions g^l ($l \in \{1, \ldots, p\}$), then we need a regularity condition for the convex optimization problem $\min_{x\in\bar{F}_3} \|x - r^k\|_2^2$ with

$$\bar{F}_3 := \{x \in P \cap S : g^l(x) + \textstyle\sum_{i=0}^{r}\lambda_i f^l(v_i) \leq 0\ ,\ l = 1, \ldots, p \\ \text{with } \lambda \in B_r\ ,\ x = \textstyle\sum_{i=0}^{r}\lambda_i v_i\}\ ,$$

since this problem has also nonlinear convex constraints. In general, we are not able to assume that such a regularity condition is fulfilled for each possible simplex $S = [v_0, \ldots, v_r]$. The counterexample presented in the next section shows a situation where the KKT-conditions fail.

Even if we would be able to formulate a condition checkable in advance for problems of type (DCP$_3$) ensuring the applicability of the KKT-theory, another problem occurs. Apart from the application of the KKT-conditions a second essential part in the proof of Lemma 4.4.8 is the existence of a positive real value τ independent of the iteration counter k (see Relation (A.5.16) in Section A.5). The existence of this value depends on the fact that there is only a finite number of possible gradients in the formulation of the KKT-conditions for the problems $\min_{x\in\bar{F}} \|x - r^k\|_2^2$ ($k \in \mathbb{N}$). Formulating these conditions for $\min_{x\in\bar{F}_3} \|x - r^k\|_2^2$ with continuous differentiable functions g^l ($l = 0, \ldots, p$) we can obtain in each iteration gradients depending on $\nabla g^l(\bar{r}^k)$ ($l \in \{1, \ldots, p\}$). Therefore, the set of possible gradients is no longer finite, and the existence of the necessary value $\tau > 0$ is in general not provable, at least not provable with the used techniques.

4.5. A Counterexample

In this section we show that the variant of Algorithm 4.1, which employs only ω-subdivisions, can fail, if this approach is used for solving problems of type (DCP$_3$). Consider the following optimization problem

$$\begin{aligned} \min\ f\tbinom{x}{y} & \\ \tfrac{1}{4}x^2 + y^2 &\le 1 \\ g(x) + y^2 &\le 1 \\ \tbinom{x}{y} &\in P \subset \mathbb{R}^2 \end{aligned} \tag{CE}$$

with $f : \mathbb{R}^2 \to \mathbb{R}$, $f\tbinom{x}{y} = -\|\tbinom{x}{y}\|_2^2 = -x^2 - y^2$, $g : \mathbb{R} \to \mathbb{R}$

$$g(x) = \begin{cases} 0 & \text{, if } x \ge 0 \\ x^2 & \text{, otherwise} \end{cases}$$

and $P = \{\tbinom{x}{y} \in \mathbb{R}^2 : x \ge -1\,,\ -\tfrac{1}{2}x + y \ge -\tfrac{3}{2}\,,\ -x - y \ge -3\}$. The function f is concave, in particular strictly concave, the nonlinear constraint functions are obviously convex and P is a full-dimensional, non-empty polytope. Therefore, the Problem (CE) belongs to the class (DCP$_3$) and the feasible region of (CE) is given by

$$F = \{\tbinom{x}{y} \in P : \tfrac{1}{4}x^2 + y^2 \le 1\,,\ g(x) + y^2 \le 1\}$$

(see Figure 4.1). It is easy to see that the function f attains its unique minimum on F at the point $\tbinom{x^\star}{y^\star} = \tbinom{2}{0}$ with optimal value $f^\star = -4$.

In the following we apply the variant of Algorithm 4.1, which employs only ω-subdivisions, for solving Problem (CE). We will see that even with $\epsilon > 0$ and a CONVEXSOLVER$_{0,0,0}$ this approach generates an infinite sequence $\{S^k\}_{k\in\mathbb{N}}$ of simplices with the properties that, for each $k \in \mathbb{N}$, there holds

$$\mu(S^k) \ < \ -9\,, \tag{4.5.1.a}$$

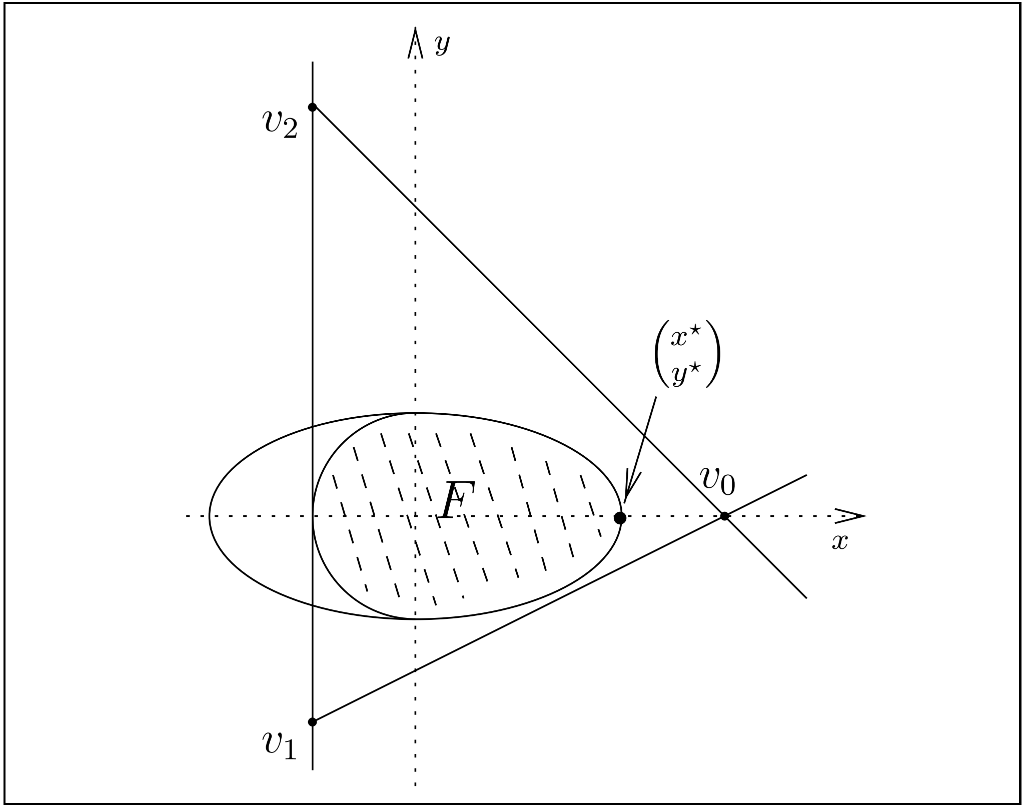
$$f(\omega(S^k)) \ = \ -1 \tag{4.5.1.b}$$

and

$$\tbinom{x^\star}{y^\star} \ \notin \ S^k\,. \tag{4.5.1.c}$$

Since the function f is continuous it follows that each accumulation point $\omega^\star$ of the sequence $\{\omega(S^k)\}_{k\in\mathbb{N}}$ has the function value $f(\omega^\star) = -1$. Thus, we know that

FIGURE 4.1. The feasible set F of Problem (CE)



$\omega^\star$ is not optimal. This shows that this variant of Algorithm 4.1 is not convergent in the sense of Theorem 4.3.1. Moreover, if the accuracies δ, $\rho > 0$ are chosen sufficiently small, we are able to guarantee that there holds

$$\min_{x \in F_{\delta,\rho}} f\binom{x}{y} \;\geq\; -5$$

with $F_{\delta,\rho} = \{\binom{x}{y} \in \mathbb{R}^2 : \frac{1}{4}x^2 + y^2 \leq 1 + \delta\,, g(x) + y^2 \leq 1 + \delta\,, -x \leq 1 + \rho\,, \frac{1}{2}x - y \leq \frac{3}{2} + \rho\,, x + y \leq 3 + \rho\}$. This implies that in Algorithm 4.1 the upper bound η^k ($k \in \mathbb{N}$) is always not smaller than -5, and from Property (4.5.1.a) of the lower bound sequence we know that for $\epsilon \in (0, 4)$ Algorithm 4.1 does not terminate after a finite number of iterations. Hence, this approach is also not convergent in the sense of Theorem 4.4.9.

The polytope P is a 2-simplex with the vertices $v_0 = \binom{3}{0}$, $v_1 = \binom{-1}{-2}$ and $v_2 = \binom{-1}{4}$ (see Figure 4.1). Thus, we can choose P as the start-simplex S^0, i.e., $S^0 = [v_0, v_1, v_2]$. In view of (4.2.2) we obtain for the convex envelope φ_{S^0} of f on the set S^0 the following relation

$$\varphi_{S^0}\binom{x}{y} \;=\; \underbrace{0\cdot(x-3)\;+\;(-2)\cdot(y-0)}_{=(\zeta_{S^0})^T\left(\binom{x}{y}-v_0\right)} \;+\; \underbrace{(-9)}_{=f(v_0)}\;.$$

It follows immediately that the optimal solution of $\min_{z\in F}\varphi_{S^0}(z)$ is the point $\omega(S^0)=\binom{0}{1}$ with optimal value $\varphi_{S^0}(\omega(S^0))=-11$. Thus we get $\mu^0=-11$ and $f(\omega(S^0))=-1$, and the simplex S^0 is subdivided – using the ω-subdivision rule – in the three simplices

$$S_1^0 \;=\; [v_0, v_1, \omega(S^0)]\,,$$

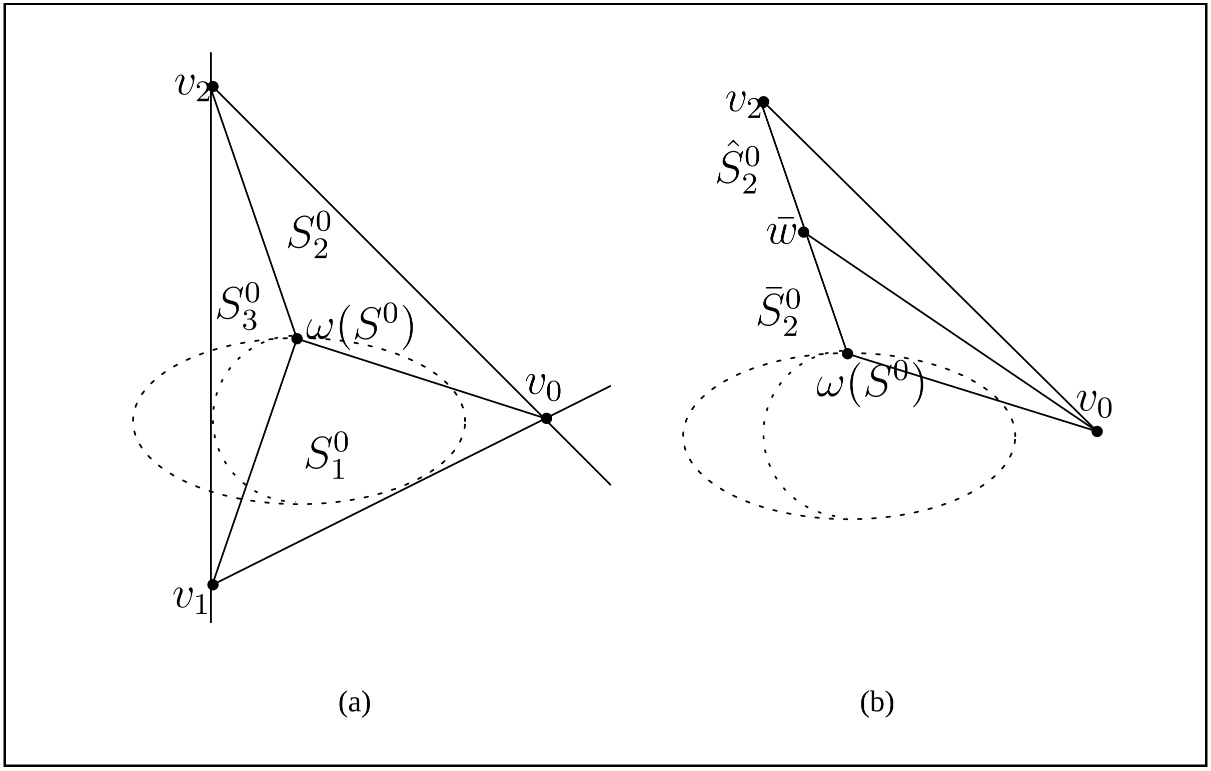
$$S_2^0 \;=\; [v_0, \omega(S^0), v_2]$$

and

$$S_3^0 \;=\; [\omega(S^0), v_1, v_2]$$

(see Figure 4.2(a)).

FIGURE 4.2. Subdivision of S^0 with respect to $\omega(S^0)$



We prove now that the following relations hold

$$\min\{\mu(S_1^0), \mu(S_2^0)\} \;\geq\; -9 \tag{4.5.2}$$

and

$$\mu(S_3^0) \;<\; -9\,, \tag{4.5.3}$$

which means that S_3^0 is chosen as the new simplex S^1 at the end of iteration 0. Because of $\min\{f(v_0), f(v_1), f(\omega(S^0))\} = \min\{-9, -5, -1\} = -9$ it is obvious that, for each $z \in S_1^0$, $\varphi_{S_1^0}(z) \geq -9$ and hence $\mu(S_1^0) \geq -9$. In order to prove this relation for S_2^0 we need more effort. Using again the representation (4.2.2) for convex envelopes we obtain

$$\varphi_{S_2^0}\tbinom{x}{y} \;=\; -5x - 7y + 6\,.$$

The point $\bar{w} = \binom{-0.5}{2.5} = 0.5\omega(S^0) + 0.5v_2$ belongs to the edge $[\omega(S^0), v_2]$ of the simplex S_2^0 and has the function value $\varphi_{S_2^0}(\bar{w}) = -9$. The simplex S_2^0 can be partitioned into two simplices

$$\bar{S}_2^0 \;=\; [v_0, \omega(S^0), \bar{w}] \quad \text{and} \quad \hat{S}_2^0 \;=\; [v_0, \bar{w}, v_2]$$

(see Figure 4.2(b)), with the properties

$$\varphi_{S_2^0}(z) \;\geq\; -9 \qquad \forall z \in \bar{S}_2^0$$

and

$$\varphi_{S_2^0}(z) \;\leq\; -9 \qquad \forall z \in \hat{S}_2^0\,.$$

Since the simplex $\hat{S}_2^0$ does not contain a feasible point of Problem (CE), i.e., $F \cap \hat{S}_2^0 = \emptyset$ (see again Figure 4.2(b)), we obtain $\mu(S_2^0) \geq -9$, which proves Relation (4.5.2).

Denote now for a point $w = \binom{w_x}{w_y} \in \{\binom{x}{y} \in \mathbb{R}^2 : x^2 + y^2 = 1\,,\ -1 < x \leq 0\,,\ 0 < y \leq 1\} \subset F$ by

$$F(w) \;:=\; \{\tbinom{x}{y} \in \mathbb{R}^2 : x^2 + y^2 = 1\,,\ -1 < x < w_x\,,\ 0 < y < w_y\}$$

the part of the boundary of the feasible region F of (CE) which lies between the points $\binom{-1}{0}$ and $\binom{w_x}{w_y}$. We verify Relation (4.5.3) by showing that the optimal solution of $\min\limits_{z \in F \cap S_3^0} \varphi_{S_3^0}(z)$ must be attained at a point $\hat{w} \in F(\omega(S^0))$ with the property

$$\varphi_{S_3^0}(\hat{w}) \;<\; -9\,. \tag{4.5.4}$$

This will be done by the next lemma. However, this lemma presents a more general result, which will also be helpful in the sequel.

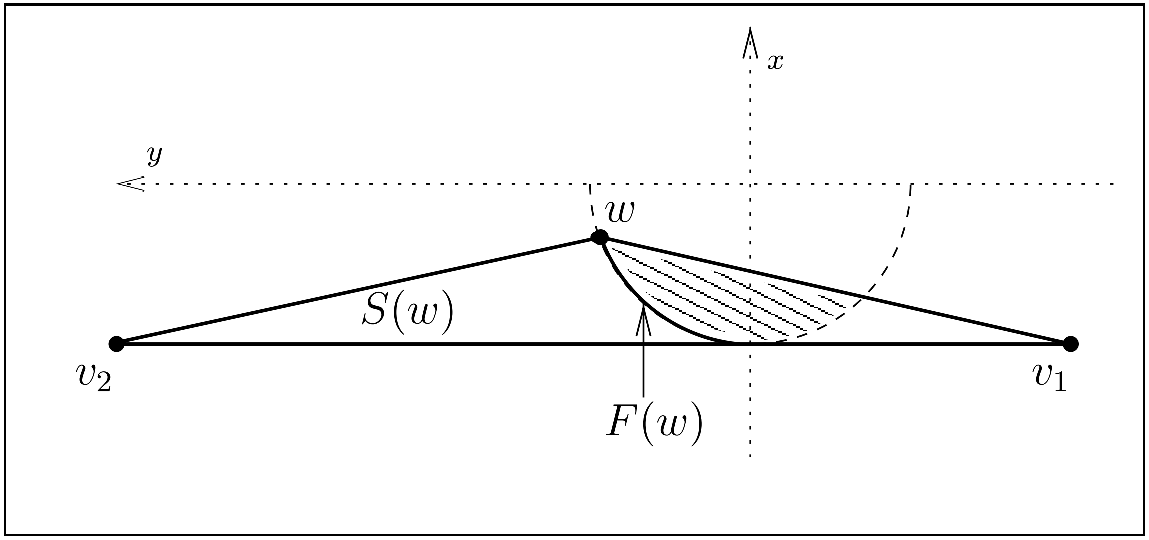
LEMMA 4.5.1. *Let w be a point on the part $\{\binom{x}{y} \in \mathrm{I\!R}^2 \,:\, x^2+y^2 = 1\,,$ $-1 < x \leq 0\,,\, 0 < y \leq 1\}$ of the boundary of the feasible region F of Problem (CE). Let $S(w)$ be the 2-simplex with the vertices w, $v_1 = \binom{-1}{-2}$ and $v_2 = \binom{-1}{4}$ (see Figure 4.3) and let $\varphi_{S(w)} : \mathrm{I\!R}^2 \to \mathrm{I\!R}$ be the convex envelope of f on the set $S(w) = [w, v_1, v_2]$. Let further $\hat{w}$ be the optimal solution of $\min\limits_{z \in F \cap S(w)} \varphi_{S(w)}(z)$. Then there holds*

$$\hat{w} \;\in\; F(w) \tag{4.5.5.a}$$

and

$$\varphi_{S(w)}(\hat{w}) \;<\; -9\,. \tag{4.5.5.b}$$

FIGURE 4.3. Situation in Lemma 4.5.1



PROOF: From (4.2.2) we know that $\varphi_{S(w)}\binom{x}{y}$ ($\binom{x}{y} \in \mathrm{I\!R}^2$) is given by

$$\varphi_{S(w)}\tbinom{x}{y} \;=\; 2\tfrac{b}{a}(x - w_x) + (-2)(y - w_y) - 1$$

with $w = \binom{w_x}{w_y}$, $b = 4 + w_y$ and $a = 1 + w_x$. Consider the set

$$\bar{F} \;:=\; F(w) \cap \left\{ \tbinom{\tau}{\sqrt{1-\tau^2}} \,:\, -1 < \tau \leq \tfrac{-b}{\sqrt{a^2+b^2}} \right\}\;.$$

Because of $w_x > -1$ we obtain $a > 0$ and, therefore, $\frac{-b}{\sqrt{a^2+b^2}} > -1$. It follows that the set $\bar{F}$ is not empty and, moreover, there holds $\bar{F} \subset S(w) \cap F$ (note that $F(w) \subset S(w)$, see Figure 4.3). Showing, for all $\bar{w} \in \bar{F}$, the relation

$$\varphi_{S(w)}(\bar{w}) \;<\; -9\,, \tag{4.5.6}$$

we will obtain that the minimal value of $\varphi_{S(w)}$ on the set $S(w) \cap F$ must be lower than -9. For this aim consider the one-dimensional function $\bar{\varphi}_{S(w)} : [-1, 0] \to \mathrm{I\!R}$,

$$\bar{\varphi}_{S(w)}(\tau) \;:=\; \varphi_{S(w)}\big(\tfrac{\tau}{\sqrt{1-\tau^2}}\big)\;.$$

There holds $\bar{\varphi}_{S(w)}(-1) = -9$. Therefore, in order to prove Relation (4.5.6) it is sufficient to show that the function $\bar{\varphi}_{S(w)}$ is monotonously decreasing along the line between -1 and $\frac{-b}{\sqrt{a^2+b^2}}$. The function $\bar{\varphi}_{S(w)}$ is obviously differentiable in each point $\tau \in (-1, 0)$, and there holds

$$\frac{\partial \bar{\varphi}_{S(w)}(\tau)}{\partial \tau} \;=\; 2\frac{b}{a} + \frac{2\tau}{\sqrt{1-\tau^2}}\;.$$

For $\tau \in (-1, \frac{-b}{\sqrt{a^2+b^2}})$ we know that $1 - \tau^2 \leq 1 - \frac{b^2}{a^2+b^2} = \frac{a^2}{a^2+b^2}$ and, thus, because of $\tau < 0$ we obtain

$$\frac{2\tau}{\sqrt{1-\tau^2}} \;\leq\; \frac{2\tau}{\frac{a}{\sqrt{a^2+b^2}}} \;<\; 2\frac{\frac{-b}{\sqrt{a^2+b^2}}}{\frac{a}{\sqrt{a^2+b^2}}} \;=\; -\frac{2b}{a}\;.$$

It follows that, for each $\tau \in (-1, \frac{-b}{\sqrt{a^2+b^2}})$, there holds

$$\frac{\partial \bar{\varphi}_{S(w)}(\tau)}{\partial \tau} \;<\; 0\;,$$

which shows that $\bar{\varphi}_{S(w)}$ is monotonously decreasing on $(-1, \frac{-b}{\sqrt{a^2+b^2}})$ and, thus, there holds $\varphi_{S(w)}(\hat{w}) < -9$, i.e., Relation (4.5.5.b) is fulfilled.

In order to prove Relation (4.5.5.a) assume, by contradiction, that there holds $\hat{w} \notin F(w)$. Because of the structure of the set $F \cap S(w)$ (see the shaded region in Figure 4.3) we know that, for each point $\tilde{w} \in (F \cap S(w)) \setminus (F(w) \cup \{w, \binom{-1}{0}\})$, the line between $\tilde{w}$ and v_2 must intersect the set $F(w)$. Because of $\varphi_{S(w)}(\hat{w}) < -9$ we have $\hat{w} \notin \{w, \binom{-1}{0}\}$. Let $\bar{w}$ be the intersection point of $[\hat{w}, v_2]$ and $F(w)$. It follows that there is a real value $\lambda \in (0, 1)$ with $\bar{w} = \hat{w} + \lambda(v_2 - \hat{w})$ and we obtain

$$\begin{aligned}\varphi_{S(w)}(\bar{w}) \;&=\; \varphi_{S(w)}(\hat{w}) + \lambda\left(\frac{2b}{a}\underbrace{(-1-\hat{w}_x)}_{<0,\hat{w}_x\in(-1,0]} -2\underbrace{(4-\hat{w}_y)}_{>0,\hat{w}_y\in[-1,1]}\right)\\ &<\; \varphi_{S(w)}(\hat{w})\end{aligned}$$

contradicting the optimality of $\hat{w}$. ■

With the notation used in the previous lemma there holds $S_3^0 = S(\omega(S^0))$, and therefore we obtain particularly the postulated result (4.5.4) for the solution $\omega(S_3^0)$ of the optimization problem $\min\limits_{z \in F \cap S_3^0} \varphi_{S_3^0}(z)$. This means that there holds

$$\mu(S_3^0) \;=\; \varphi_{S_3^0}(\omega(S_3^0)) \;<\; -9\,,$$

which implies $\mu^1 = \mu(S_3^0) < -9$.

In iteration 1 the simplex $S^1 = S_3^0$ is now subdivided with respect to the point $\omega(S^1) \in F(\omega(S^0))$ in the three subsimplices

$$S_1^1 \;=\; [\omega(S^0), v_1, \omega(S^1)]\,,$$

$$S_2^1 \;=\; [\omega(S^0), \omega(S^1), v_2]$$

and

$$S_3^1 \;=\; [\omega(S^1), v_1, v_2]\,.$$

Because of $\min\{f(\omega(S^0)), f(v_1), f(\omega(S^1))\} = \min\{-1, -5, -1\} > -9$ we obtain $\mu(S_1^1) > -9$ and regarding Lemma 4.5.1 we know that the minimal point of $\varphi_{S_3^1}$ on the set $F \cap S_3^1$ belongs to $F(\omega(S^1))$, and that there holds $\mu(S_3^1) < -9$. If we are able to show that the function value of $\varphi_{S_2^1}$ is greater than or equal to -9 for each feasible point $z \in F \cap S_2^1$, then we obtain

$$S^2 \;=\; S_3^1 \quad \text{and} \quad \mu^2 \;=\; \mu(S_3^1) \;<\; -9\,.$$

This means that we would be in the same situation as at the end of iteration 0. The next lemma shows that the relation

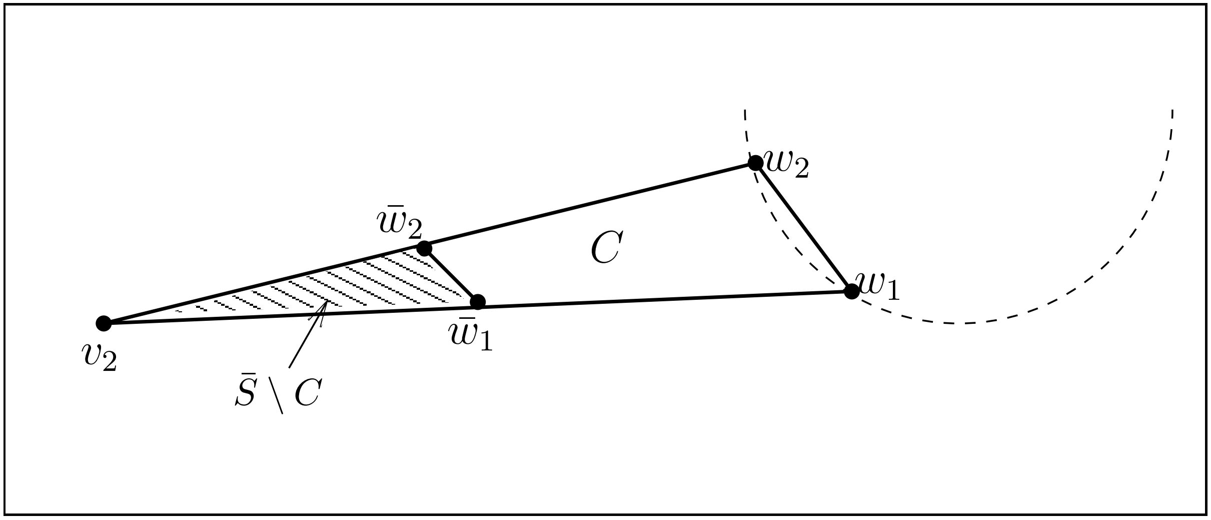
$$\min_{z \in F \cap \bar{S}} \varphi_{\bar{S}}(z) \;\geq\; -9$$

is true for each 2-simplex $\bar{S} = [w_1, w_2, v_2]$ with $w_1, w_2 \in F\binom{0}{1}$, and hence, in particular, for S_2^1.

LEMMA 4.5.2. *Let $\bar{S} = [w_1, w_2, v_2]$ be a 2-simplex with $w_1, w_2 \in F\binom{0}{1}$ and let $\varphi_{\bar{S}} : \mathrm{I\!R}^2 \to \mathrm{I\!R}$ be the convex envelope of f on $\bar{S}$. Then there holds*

$$\min_{z \in F \cap \bar{S}} \varphi_{\bar{S}}(z) \;\geq\; -9\,. \tag{4.5.7}$$

FIGURE 4.4. Situation in Lemma 4.5.2



PROOF: There holds $\varphi_{\bar{S}}(w_1) = \varphi_{\bar{S}}(w_2) = -1$ and $\varphi_{\bar{S}}(v_2) = -17$. With $\bar{w}_i := 0.5w_i + 0.5v_2$ ($i = 1, 2$) we obtain, for $i = 1, 2$,

$$\varphi_{\bar{S}}(\bar{w}_i) \ = \ -9 \, ,$$

and hence

$$\varphi_{\bar{S}}(z) \ \geq \ -9 \qquad \forall z \in [w_1, w_2, \bar{w}_1, \bar{w}_2] =: C \, .$$

In order to show result (4.5.7) it is sufficient to prove that each element of $\bar{S} \setminus C$ is infeasible with respect to Problem (CE) (see Figure 4.4).

Let w be an arbitrary element of $\bar{S} \setminus C$, i.e.,

$$w = \binom{w_x}{w_y} \ \in \ [v_2, \bar{w}_1, \bar{w}_2] \, .$$

By definition of $\bar{w}_i$ ($i = 1, 2$) we obtain $w_x \leq -0.5$ and $w_y \geq 2.0$. It follows immediately $\frac{1}{4}w_x^2 + w_y^2 > 1$, i.e., $w \notin F$. ■

Combining the results of Lemma 4.5.1 and Lemma 4.5.2 and regarding the considerations above we see that the variant of Algorithm 4.1, which uses only ω-subdivisions, generates an infinite sequence $\{S^k\}_{k\in\mathbb{N}}$of simplices with the properties that, for each $k \in \mathbb{N}$, there holds

$$S^k \ = \ [\omega(S^{k-1}), v_1, v_2] \, ,$$

$$\omega(S^k) \ \in \ F(\omega(S^{k-1}))$$

and

$$\mu^k \ = \ \mu(S^k) \ < \ -9 \, .$$

By definition of $F(\omega(S^{k-1}))$ ($k \in \mathbb{N}$) we obtain furthermore $f(\omega(S^k)) = -1$, and that the optimal point $\binom{x^\star}{y^\star} = \binom{2}{0}$ of Problem (CE) does not belong to the simplex S^k. Consequently, we have shown that Algorithm 4.1 applied for solving Problem (CE) generates an infinite simplex sequence with Properties (4.5.1.a)-(4.5.1.c). Note that Algorithm 4.1 generates this simplex sequence independent of the chosen accuracies. Too large values of ϵ, δ or ρ could only lead to a termination of Algorithm 4.1 after a finite number of iterations. However, it is obvious that these accuracies can be chosen – greater than 0 – such that Algorithm 4.1 makes infinitely many steps without fulfilling the stopping criterion, and hence does not solve Problem (CE).

REMARK 4.5.1. In this situation we see that the simplex S in the sense of Lemma 4.4.2 is the 1-simplex $[v_1, v_2]$. The unique point $\bar{z} \in F \cap S$ is $\binom{-1}{0}$, and the gradients of the constraints, which describe the set $F \cap S$ and which are active at $\bar{z}$, are linear dependent. Therefore, the KKT-theory is not applicable for the problem

$$\begin{aligned} \min \|\tilde{z} - z\|_2^2 \\ z \in F \cap S \end{aligned} \tag{CEOP}$$

for an arbitrary point $\tilde{z} \in S \setminus F$, i.e., the vector $(\tilde{z} - \bar{z})$ is not an element of the cone generated by the gradients of the active constraints of Problem (CEOP) in $\bar{z}$ (see the proof of Lemma 4.4.8 in Section A.5). As mentioned at the end of the previous section this is a situation, where the KKT-theory does not work.

In the next section we will see that it is nevertheless possible to make Algorithm 4.1 convergent for problems of each class, where convergence is meant in the sense that this approach detects in finite time either the emptiness of the feasible region or an approximate solution. For this aim we will change a little the generalized ω-subdivision rule (GWSR) by using the result of Lemma 4.4.2.

4.6. Numerical Comparisons

In this section we discuss the numerical performance of Algorithm 4.1. The proposed simplicial branch-and-bound Algorithm 4.1 was encoded in C++ with management of partition sets by AVL-trees. In fact, we used a modified version of the code mentioned in Chapter 3 (see, especially, Section 3.5). In order to test the computational performance of our algorithm we solved again the randomly generated set of all-quadratic problems described in Section 1.5.

We are interested in (ϵ, δ, ρ)-solutions of Problem (DCP) with ϵ, δ, $\rho > 0$. Note that the test examples were generated in a way, which ensured that the feasible set F is not empty. In view of the Convergence Theorem 4.3.1 we know that the variant of Algorithm 4.1, which employs only bisections, detects such a solution in finite time. Using the generalized ω-subdivision rule (GWSR) the finiteness of Algorithm 4.1 is no longer guaranteed, at least in the general case. Nevertheless, it is possible to make this variant of Algorithm 4.1 finite, as we will see later in this section. For this purpose we will modify (GWSR) by using the result of Lemma 4.4.2.

4.6.1. Comparison of Algorithm 4.1 Based on Bisection with Algorithm 3.1. First of all we would like to compare the computational performances of Algorithm 4.1 employing bisections and of Algorithm 3.1 (see Section 3.3). The used subproblems are the main difference between Algorithm 3.1 and Algorithm 4.1, if we apply these approaches for solving all-quadratic problems (QP). Note that we also use a (δ, ρ)-feasibility concept in order to obtain finiteness of Algorithm 3.1 (see the considerations at the end of Section 3.4). In Algorithm 3.1 we obtain lower bounds by linearizing the original Problem (QP) with respect to the current simplex (see Section 3.2) and in Algorithm 4.1 we use convex subproblems. Since the convex relaxation of an all-quadratic problem, presented in Section 4.2 for Algorithm 4.1, is of course a better approximation than the linear relaxation proposed in Section 3.2 for Algorithm 3.1 (see also Remark 3.2.1), we can expect that Algorithm 4.1 needs less iterations than Algorithm 3.1 in order to solve this problem. How much the running-times change is not predictable in advance. They can decrease, but also increase.

We solved all test problems with Algorithm 3.1 using the LP-subroutine *E04NFF* of the *NAG*-library. Since there is no sparse structure in our linear subproblems it is not reasonable to use *MINOS 5.4*, as we did in Section 3.5. Note that this tool is slower than E04NFF, if both are applied for solving non-sparse problems (compare with the computational results in Section 3.5 and the reason there to use *MINOS 5.4*). As a $\mathsf{CONVEXSOLVER}_{\bar\epsilon,\bar\delta,\bar\rho}$ in Algorithm 4.1 we used the *NAG*-subroutine *E04UCC* with the default value of $\bar\epsilon$ depending on the machine precision and $\bar\delta = \bar\rho = 10^{-6}$. This routine implements a *sequential quadratic programming (SQP)* method. In both algorithms we used the accuracies $\epsilon = \delta = 10^{-4}$ and $\rho = 10^{-6}$. As in the numerical tests in Section 3.5 we stopped branching in both algorithms, when the *relative* difference between η^k and μ^k ($k \in \mathbb{N}$) was smaller than the tolerance ϵ, i.e., if there held

$$\eta^k - \mu^k \ \leq \ \epsilon \max\{1.0\,,\, |\eta^k|\} \tag{$\overline{\overline{\text{SC}}}$}$$

(compare with page 103).

REMARK 4.6.1. We do not solve Problem (DCP^S) directly with the CONVEXSOLVER$_{\bar{\epsilon},\bar{\delta},\bar{\rho}}$. In order to avoid the calculation of the vectors $\bar{v}_i^S$ (see (4.2.3.a) and (4.2.3.b)) ($i = 0, \ldots, n$) it is cheaper to affinely transform Problem (DCP^S) by using $x = v_0 + W_S\lambda$, where W_S denotes the regular ($n \times n$)-matrix with columns $(v_i - v_0)$ ($i = 1, \ldots, n$) and λ is an element of $\{\lambda \in \mathbb{R}^n : \sum_{i=1}^n \lambda_i \leq 1\,,\, \lambda \geq 0\}$. By doing this we do not need the constraints $(\bar{v}_i^S)^T x \leq c_i^S$ in order to ensure that the feasible points of Problem (DCP^S) are contained in the current simplex $S = [v_0, \ldots, v_n]$. It is sufficient to require that there holds $\lambda \in [0,1]^n$ and $\sum_{i=1}^n \lambda_i \leq 1$ (see the derivation of the LP-relaxation in Section 3.2 and, especially, the Remarks 3.2.1 and 3.2.2(a)).

Tables 4.1 and 4.2 show some numerical results for the generated test problems run on a *SUN SPARCserver 1000* workstation. We use the abbreviations NuP Co<Li for the number of problems where Algorithm 4.1 with convex subproblems was faster with respect to the running-time than Algorithm 3.1 with linear subproblems. AvgNuSP is used for the average number of subproblems solved for each test problem with Algorithm 4.1 (Co) or Algorithm 3.1 (Li). StdSP is used for the standard deviation of the number of subproblems. AvgTime stands for the average computing time in seconds necessary for solving a problem and StdTime for the corresponding standard deviation values. Note that in the numerical tests of Algorithm 3.1 in Section 3.5 we used higher accuracies for checking the *"feasibility"* of generated solutions $\omega(S^k)$. Therefore, Algorithm 3.1 had in the numerical tests in the present chapter on average less linear subproblems to solve than it was the case in Section 3.5 (see Tables 3.1 and 3.2). The results for Algorithm 3.1 were obtained with the original variant of this approach, which did not apply any selection rule for the first vertex of a considered n-simplex (see Subsection 3.5.3).

The numerical results displayed in Table 4.1 show that for small dimensional problems ($n \leq 4$) the decrease of the number of subproblems, which had to be solved, did not lead to a decrease of the running-time. Algorithm 4.1 with convex subproblems needed on average the same or slightly more time in order to solve the test problems than the other one did. However, as we can see in Table 4.2, if more than twice as much linear subproblems had to be solved, Algorithm 4.1 showed a

TABLE 4.1. All test results for $n = 2, 3, 4$

p	NuP	AvgNuSP		StdSP		AvgTime		StdTime	
	Co<Li	Co	Li	Co	Li	Co	Li	Co	Li
$n = 2$									
1	21	29.6	42.0	12.9	17.1	0.12	0.11	0.05	0.04
2	20	23.6	38.2	12.4	18.8	0.12	0.13	0.06	0.08
3	12	33.9	55.5	14.1	30.4	0.19	0.16	0.09	0.07
4	3	34.4	50.2	11.1	23.4	0.21	0.14	0.08	0.06
$n = 3$									
1	16	78.4	122.5	52.2	83.0	0.43	0.42	0.29	0.30
2	14	80.2	133.0	47.1	131.7	0.47	0.44	0.32	0.38
3	9	101.4	173.0	67.6	145.1	0.68	0.55	0.45	0.43
4	6	82.0	132.0	43.4	74.7	0.57	0.42	0.28	0.22
5	8	88.3	156.4	48.4	106.8	0.74	0.53	0.44	0.33
6	10	88.8	159.5	44.5	86.2	0.70	0.56	0.33	0.33
$n = 4$									
1	14	172.6	304.4	145.8	316.4	1.27	1.12	1.07	1.10
2	14	179.2	324.4	158.4	323.9	1.39	1.35	1.26	1.38
3	10	155.3	310.2	105.9	372.1	1.28	1.26	0.86	1.43
4	18	234.1	536.2	214.7	741.2	2.14	2.24	2.17	3.13
5	10	178.4	332.6	110.7	257.7	1.67	1.41	1.12	1.03
6	10	228.5	671.6	202.5	1,661	2.32	3.47	2.12	10.10
7	10	207.2	382.3	141.6	292.9	2.22	1.82	1.53	1.34
8	4	204.6	372.9	154.1	378.2	2.43	1.92	1.85	1.98

better numerical performance with respect to the running-time than Algorithm 3.1. With growing dimensions and, in particular, with a growing number of quadratic constraints the relative difference between the average number of convex subproblems and the average number of linear subproblems increased. For dimensions higher than $n = 6$ Algorithm 4.1 solves more than 60% of the 50 test problems faster than Algorithm 3.1. On average Algorithm 4.1 was always faster for these test problems. Since the speedup, i.e., the quotient of the average running-time with linear subproblems and the average running-time with convex subproblems was mostly less than 1.5, we see that the use of convex subproblems was not a substantial acceleration of the considered solution process for all-quadratic problems. However, there was a small acceleration.

TABLE 4.2. Some test results for $n = 5, 6, 7, 8$

p	NuP	AvgNuSP		StdSP		AvgTime		StdTime	
	Co<Li	Co	Li	Co	Li	Co	Li	Co	Li
$n = 5$									
2	21	444.9	933.4	649.1	1432	4.56	4.71	5.90	6.18
4	21	479.0	1,033	615.7	1379	5.91	6.32	7.53	8.07
6	14	488.8	965.7	439.7	978.9	7.08	6.50	6.37	6.37
8	21	509.1	1,211	413.9	1214	8.45	9.42	7.05	9.59
10	15	375.9	800.8	280.7	742.9	7.28	6.61	5.43	5.90
$n = 6$									
2	30	1,058	2,546	1,019	2,731	13.99	18.25	13.00	19.32
4	29	1,632	5,315	2,735	14,899	26.87	41.30	47.13	105.8
6	28	1,918	5,184	3,768	12,043	38.29	46.03	73.76	105.0
8	21	1,534	4,191	1,885	5,984	34.75	41.10	42.36	56.15
10	20	907	2,345	927.1	2,509	24.49	25.41	24.77	26.59
12	31	1,228	3,467	1,349	3,885	37.97	43.82	41.82	45.73
$n = 7$									
2	32	3,601	11,319	8,416	25,815	63.91	100.2	150.2	230.1
4	40	3,246	12,510	7,069	38,315	72.10	147.7	161.2	375.3
6	34	2,246	7,015	3,001	8,855	56.49	82.07	73.09	104.5
8	31	2,928	9,236	3,568	11,230	93.75	119.1	115.1	138.5
10	31	2,885	9,038	3,526	12,054	101.2	127.5	114.5	163.8
12	35	2,768	8,631	3,576	10,830	111.8	137.5	149.6	175.4
14	23	3,091	9,969	3,667	13,217	137.2	168.7	163.8	230.8
$n = 8$									
2	34	6,126	18,002	14,354	25,313	128.2	211.3	280.0	285.6
4	33	5,585	17,282	14,449	33,490	154.3	226.7	379.5	438.5
6	30	5,808	19,474	13,115	43,043	189.3	288.6	396.4	670.1
8	35	9,398	32,306	16,262	57,404	416.6	522.1	747.2	921.8
10	30	4,285	21,935	4,989	43,103	214.8	378.4	255.1	728.8
12	33	5,102	19,783	5,470	23,321	284.4	380.9	304.5	432.1
14	25	5,419	23,616	8,654	49,231	368.3	493.3	554.7	1,014
16	32	5,749	20,977	6,934	27,565	393.7	503.5	465.6	656.5

Remember that it is possible to improve the performance of Algorithm 3.1 by introducing a selection rule for the first vertex of a considered simplex, as we did in Subsection 3.5.3. The convex relaxation used in Algorithm 4.1 is unique and does particularly not depend on the first vertex, as it is the case for the LP-relaxation applied in Algorithm 3.1. Hence such a selection rule does not alter the numerical performance of Algorithm 4.1 and, in view of the results in Table 3.4, it is likely that the use of a selection rule in Algorithm 3.1 reduces the running-time advantage of Algorithm 4.1. On the other hand, in Algorithm 4.1 we applied a CONVEXSOLVER$_{\bar{\epsilon},\bar{\delta},\bar{\rho}}$, which only uses the differentiability of the convex functions. Maybe another solution method, which exploits the quadratic structure of (DCP^S), can solve the occurring convex subproblems faster (see, e.g., [JAR96]). Thus, we can expect that Algorithm 4.1 with convex subproblems is – with respect to the running-time – a better solution method for all-quadratic optimization problems than Algorithm 3.1 with linear subproblems, at least for dimensions higher than $n = 4$.

Another interesting numerical effect of the use of convex subproblems instead of linear subproblems is that the standard deviation values are in some cases significantly smaller. Note that, especially in Table 4.2, for the numbers of solved subproblems as well as for the running-times the values of the standard deviation are very high, when they are compared with the average values. The reason is that, in particular for growing dimensions, the number of test problems, which needed substantially more time to be solved than the average, increased, and that the difference between the effort for solving such numerical outliers and the effort for solving average problems also grew. These effects were stronger if linear subproblems were used. From this point of view we see that Algorithm 4.1 shows, at least

TABLE 4.3. Comparison of the medians of the running-times of Algorithm 4.1 based on bisection and of Algorithm 3.1

	$p = 2$	$p = 4$	$p = 6$	$p = 8$	$p = 10$	$p = 12$	$p = 14$	$p = 16$
$n = 5$								
Co	2.64	3.53	5.14	6.06	6.06			
Li	2.93	3.83	4.75	4.99	4.96			
$n = 6$								
Co	10.21	14.39	15.97	21.04	16.51	23.84		
Li	14.18	15.12	16.01	19.70	17.04	30.03		
$n = 7$								
Co	27.08	35.69	29.15	55.18	70.46	71.66	78.11	
Li	34.20	47.23	35.97	73.75	64.33	88.12	72.13	
$n = 8$								
Co	34.61	59.79	68.14	136.4	114.4	179.8	192.0	191.1
Li	52.63	98.18	92.30	204.9	133.5	215.9	174.3	234.7

for the examined test problems, a more stable behavior in the sense that less numerical outliers occur. In view of the existence of numerical outliers we can expect that a large number of test problems could be solved with less effort than the average values imply. In Table 4.3 we display the medians of the running-times in seconds, corresponding to the numerical results of Table 4.2. The presented values show that at least 50% of the test problems could be solved significantly faster than the average. We also see that the medians of the running-times for Algorithm 4.1 (Co) are not always smaller than the corresponding values for Algorithm 3.1 (Li), as it is the case for the average values, at least for dimensions higher than $n = 5$ (see Table 4.2). This corroborates the effect mentioned above that the use of convex subproblems reduces the occurrence of numerical outliers and the worst case effort, respectively.

4.6.2. A Convergent Subdivision Rule Based on (GWSR). In the following we would like to use Algorithm 4.1 with the (GWSR) strategy in order to solve the same set of test problems. Since – in view of Section 4.5 – we cannot be sure that the variant of Algorithm 4.1, which employs only the generalized ω-subdivision rule, detects in finite time an (ϵ, δ, ρ)-solution, we introduce a modification of (GWSR) ensuring finiteness of Algorithm 4.1.

For this aim we will exploit the result of Lemma 4.4.2. At the end of Section 4.4 we pointed out that also in the general case (DCP$_3$) all results until Lemma 4.4.7 are provable for Algorithm 4.1 using a $\mathsf{CONVEXSOLVER}_{\bar{\epsilon}^k,\bar{\delta}^k,0}$ ($k \in \mathbb{N}$). As long as a solution method for the convex subproblems is used, which generates a point $\omega(S^k) \in S^k$, we know that w^k is chosen as $\omega(S^k)$ in the (GWSR)-rule (if $\bar{\delta}^k \le \delta$ and $\bar{\epsilon}^k \le \epsilon$, see Remark 4.2.2(h)). In the numerical tests we use a $\mathsf{CONVEXSOLVER}_{\bar{\epsilon},\bar{\delta},\bar{\rho}}$ with arbitrary accuracies $\bar{\epsilon}$, $\bar{\delta}$, $\bar{\rho} > 0$. Therefore, there does not necessarily hold $\omega(S^k) \in S^k$, and it is not immediately clear that at least the results of Lemma 4.4.2 still hold. Nevertheless, if the accuracies $\bar{\epsilon}$, $\bar{\delta}$ and $\bar{\rho}$ are chosen as in the following lemma, then we are able to prove all results until Lemma 4.4.2 for the version of Algorithm 4.1 with (GWSR) also in the general case.

LEMMA 4.6.1. *Let ϵ, δ, $\rho \ge 0$ be given. Let L^l be a Lipschitz constant of g^l ($l \in \{0, \dots, p\}$) on the n-simplex*

$$\bar{S}^0 \;=\; \{x \in \mathbb{R}^n : (\bar{v}_i^{S^0})^T x \;\le\; c_i^{S^0} + \rho\,,\, i = 0, \dots, n\}$$

(see (4.2.2) for the definition of $\bar{v}_i^{S^0}$ and $c_i^{S^0}$), let $D \in \mathbb{R}^n$ be an upper bound for $\|\cdot\|_2$ on $S^0 = [v_0^0, \dots, v_n^0]$, e.g., $D = \max_{i=0,\dots,n} \|v_i^0\|_2$, and let C^l be an upper

bound for f^l ($l \in \{0, \ldots, p\}$) on the same set. For $k \in \mathbb{N}$ denote by

$$\bar{I}^k \ := \ \{i \in \{0, \ldots, n\} : \bar{\lambda}_i^k > 0\}$$

with $\bar{\lambda}^k \in \{\lambda \in \mathbb{R}^{n+1} : \sum_{i=0}^n \lambda_i = 1\}$, $\omega(S^k) = \sum_{i=0}^n \bar{\lambda}_i^k v_i^k$ and set

$$w^k \ := \ \sum_{i \in \bar{I}^k} \frac{\bar{\lambda}_i^k}{\gamma^k} v_i^k$$

with $\gamma^k = \sum_{i \in \bar{I}^k} \bar{\lambda}_i^k$. If $\omega(S^k)$ is an ($\bar{\epsilon}$, $\bar{\delta}$, $\bar{\rho}$)-solution of (DCP$^{S^k}$) with

$$\bar{\epsilon} \ \leq \ \tfrac{1}{4}\epsilon \quad , \quad \bar{\delta} \ \leq \ \tfrac{1}{4}\delta$$

and

$$\bar{\rho} \ \leq \ \min\left\{ \frac{\rho}{4} \, , \, \frac{\rho}{4nD\|a_i\|_2} \, , \, i = 1, \ldots, m \, , \, \frac{\delta}{4n(C^l + DL^l)} \, , \, l = 1, \ldots, p \, , \right.$$
$$\left. \frac{\epsilon}{4n(C^0 + DL^0)} \right\} ,$$

then there holds

$$w^k \textit{ is a } (\tfrac{3}{4}\delta, \, \tfrac{3}{4}\rho)\textit{-feasible point for } (\mathrm{DCP}^{S^k}) \tag{4.6.1.a}$$

and

$$g^0(w^k) + \varphi^0_{S^k}(w^k) \ \leq \ \mu(S^k) + \tfrac{3}{4}\epsilon \, . \tag{4.6.1.b}$$

PROOF: In the proof of Theorem 4.3.1 we showed that, if $\omega(S^k)$ is an ($\bar{\epsilon}$, $\bar{\delta}$, $\bar{\rho}$)-solution of (DCP$^{S^k}$), then there holds, for each $k \in \mathbb{N}$ and $i \in \{0, \ldots, n\}$,

$$\bar{\lambda}_i^k \ \geq \ -\bar{\rho} \, .$$

Therefore, using the relation

$$\sum_{i \in \bar{I}^k} \left(\bar{\lambda}_i^k - \frac{\bar{\lambda}_i^k}{\gamma^k} \right) \ = \ \gamma^k - 1 \ = \ \sum_{i \notin \bar{I}^k} |\bar{\lambda}_i^k|$$

we obtain

$$\sum_{i \in \bar{I}^k} \left(\bar{\lambda}_i^k - \frac{\bar{\lambda}_i^k}{\gamma^k} \right) + \sum_{i \notin \bar{I}^k} |\bar{\lambda}_i^k| \ = \ 2 \sum_{i \notin \bar{I}^k} \underbrace{|\bar{\lambda}_i^k|}_{\leq \bar{\rho}} \ \underset{|\bar{I}^k| \geq 1}{\leq} \ 2n\bar{\rho} \, . \tag{4.6.2}$$

Because of the definition of w^k it follows

$$\|\omega(S^k) - w^k\|_2 \ = \ \| \sum_{i \in \bar{I}^k} \left(\bar{\lambda}_i^k - \frac{\bar{\lambda}_i^k}{\gamma^k} \right) v_i^k + \sum_{i \notin \bar{I}^k} \bar{\lambda}_i^k v_i^k \|_2$$

$$\leq\ D\left(\sum_{i\in\bar{I}^k}\left(\bar{\lambda}_i^k-\frac{\bar{\lambda}_i^k}{\gamma^k}\right)+\sum_{i\notin\bar{I}^k}|\bar{\lambda}_i^k|\right)\ \underset{(4.6.2)}{\leq}\ 2nD\bar{\rho} \tag{4.6.3}$$

and, for $l\in\{0,\dots,p\}$,

$$\begin{aligned}|\varphi^l_{S^k}(\omega(S^k))-\varphi^l_{S^k}(w^k)| &= |\sum_{i\in\bar{I}^k}\left(\bar{\lambda}_i^k-\frac{\bar{\lambda}_i^k}{\gamma^k}\right)f^l(v_i^k)+\sum_{i\notin\bar{I}^k}\bar{\lambda}_i^k f^l(v_i^k)|\\ &\leq\ C^l\left(\sum_{i\in\bar{I}^k}\left(\bar{\lambda}_i^k-\frac{\bar{\lambda}_i^k}{\gamma^k}\right)+\sum_{i\notin\bar{I}^k}|\bar{\lambda}_i^k|\right)\ \underset{(4.6.2)}{\leq}\ 2nC^l\bar{\rho}\,. \end{aligned}\tag{4.6.4}$$

From (4.6.3) and (4.6.4) we conclude, for $i\in\{1,\dots,m\}$,

$$\begin{aligned}a_i^Tw^k\ &\leq\ \underbrace{a_i^T\omega(S^k)}_{\leq b_i+\bar{\rho}}+\underbrace{|a_i^Tw^k-a_i^T\omega(S^k)|}_{\leq\|a_i\|_2\|w^k-\omega(S^k)\|_2}\ \leq\ b_i+\underbrace{\bar{\rho}}_{\leq\frac{\rho}{4}}+\underbrace{\|a_i\|_2 2nD\bar{\rho}}_{\leq\frac{\rho}{2}}\\ &\leq\ b_i+\tfrac{3}{4}\rho \end{aligned}\tag{4.6.5}$$

and, for $l\in\{1,\dots,p\}$,

$$\begin{aligned}g^l(w^k)+\varphi^l_{S^k}(w^k)\ &\leq\ g^l(\omega(S^k))+\varphi^l_{S^k}(\omega(S^k))+\underbrace{|g^l(\omega(S^k))-g^l(w^k)|}_{\leq L^l\|\omega(S^k)-w^k\|_2}\\ &\qquad+\underbrace{|\varphi^l_{S^k}(\omega(S^k))-\varphi^l_{S^k}(w^k)|}_{\leq 2nC^l\bar{\rho}}\\ &\leq\ \bar{\delta}+2nDL^l\bar{\rho}+2nC^l\bar{\rho}\\ &\leq\ \underbrace{\bar{\delta}}_{\leq\frac{\delta}{4}}+\underbrace{2n(DL^l+C^l)\bar{\rho}}_{\leq\frac{\delta}{2}}\ \leq\ \tfrac{3}{4}\delta\,, \end{aligned}\tag{4.6.6}$$

which shows the $(\frac{3}{4}\delta,\ \frac{3}{4}\rho)$-feasibility of w^k with respect to the feasibility set of (DCP$^{S^k}$). In order to show (4.6.1.b) we obtain by the same argumentation as in (4.6.6)

$$\begin{aligned}g^0(w^k)+\varphi^0_{S^k}(w^k)\ &\leq\ \underbrace{g^0(\omega(S^k))+\varphi^0_{S^k}(\omega(S^k))}_{\leq\mu(S^k)+\bar{\epsilon}}+\underbrace{2n(DL^0+C^0)\bar{\rho}}_{\leq\frac{\epsilon}{2}}\\ &\leq\ \mu(S^k)+\tfrac{3}{4}\epsilon\,. \end{aligned}$$

■

REMARK 4.6.2.

(a) It follows immediately that the situation $|\bar{I}^k| = 1$ (see (GWSR)) cannot occur, if $\bar{\epsilon}$, $\bar{\delta}$ and $\bar{\rho}$ are chosen as in the previous lemma. Indeed, $|\bar{I}^k| = 1$ means that

$$w^k \;=\; \sum_{i\in\bar{I}^k} \frac{\bar{\lambda}_i^k}{\gamma^k} v_i^k \;\in\; \{v_0^k, \ldots, v_n^k\}\,.$$

Regarding (4.6.1.a) and because of $\varphi^l_{S^k}(v_i^k) = f^l(v_i^k)$ ($i \in \{0,\ldots,n\}$), we know that there is a (δ, ρ)-feasible vertex $v_{i'}^k \in \{v_0^k,\ldots,v_n^k\}$. Since each vertex of S^k ($k \in \mathbb{N}$) is used for updating the upper bound in previous iterations of Algorithm 4.1 and in view of (4.6.1.b), there holds

$$\begin{aligned}\eta^k \;\le\; g^0(v_{i'}^k) + f^0(v_{i'}^k) \;&=\; g^0(w^k) + \varphi^0_{S^k}(w^k)\\ &\le\; \underbrace{\mu(S^k)}_{<\eta^k-\epsilon} + \tfrac{3}{4}\epsilon \;<\; \eta^k - \tfrac{1}{4}\epsilon\,,\end{aligned}$$

which is a contradiction.

(b) In the quadratic case the Lipschitz constants L^l ($l \in \{0,\ldots,p\}$) can be calculated in the following way

$$L^l \;=\; \max_{x\in V(\bar{S}^0)} \|\nabla f^l(x)\|_2\,,$$

where $V(\bar{S}^0)$ denotes the vertex set of $\bar{S}^0$. For the calculation of L^l in the general case we refer to Section B.2.

(c) Since we have a $\mathsf{CONVEXSOLVER}_{\bar{\epsilon},\bar{\delta},\bar{\rho}}$ for arbitrary accuracies $\bar{\epsilon}$, $\bar{\delta}$, $\bar{\rho} > 0$ the necessary upper bounds C^l for the concave functions f^l ($l \in \{0,\ldots,p\}$) on the set S^0 can be determined using this solution method.

With the result of Lemma 4.6.1 and using an analogous argumentation as in the proof of Lemma 4.4.1 the following corollary is easy to verify.

COROLLARY 4.6.2. *Assume that ϵ, δ, $\rho > 0$. Let S^k be the selected simplex in iteration $k \in \mathbb{N}$ of Algorithm 4.1 employing the generalized ω-subdivision rule (GWSR), let $\omega(S^k)$ be an ($\bar{\epsilon}$, $\bar{\delta}$, $\bar{\rho}$)-solution of (DCP$^{S^k}$) with $\bar{\epsilon}$, $\bar{\delta}$ and $\bar{\rho}$ chosen as in Lemma 4.6.1, and let $S^\star$ be one of the simplices obtained by subdividing S^k with respect to w^k. If S^k is not fathomed in the pruning rule (PR) of Algorithm 4.1, then there holds, for each $x \in S^\star$ with $x = \sum_{j=0,j\neq i}^{n} \lambda_j v_j^k + \lambda_i w^k$, $\lambda \in B_n$,*

$i \in \{0, \dots, n\}$,

$$\left.\begin{array}{rll} & \varphi^0_{S^\star}(x) \geq \varphi^0_{S^k}(x) + \frac{1}{4}\epsilon\lambda_i & \textit{, if } w^k \textit{ is } (\delta, \rho)\textit{- feasible,} \\ & \textit{or} & \\ \exists l \in \{1, \dots, p\} : & \varphi^l_{S^\star}(x) \geq \varphi^l_{S^k}(x) + \frac{1}{4}\delta\lambda_i & \textit{, otherwise.} \end{array}\right\} \quad (4.6.7)$$

PROOF: As mentioned above, this proof is analogous to the proof of Lemma 4.4.1. Therefore, we would not like to expatiate this proof. However, note that – regarding Remark 4.6.2(a) – there holds

$$w^k \ = \ \sum_{i \in \bar{I}^k} \frac{\bar{\lambda}^k_i}{\gamma^k} v^k_i \ \notin \ \{v^k_0, \dots, v^k_n\} \ ,$$

i.e, $|\bar{I}^k| > 1$ in (GWSR). Note, furthermore, that w^k is, in view of Lemma 4.6.1, always ρ-feasible with respect to the linear constraints of (DCP). ■

The result of this corollary coincides with Lemma 4.4.1, where this lemma was the essential part in the proof of Lemma 4.4.2. A careful checking of the proofs for Lemma 4.4.2 (see Appendix A, especially the proof of Lemma A.1) shows that this result is also true in the general case of problems of type (DCP$_3$), if the assumptions of Corollary 4.6.2 are fulfilled.

COROLLARY 4.6.3. *Assume that ϵ, δ and ρ are chosen greater than 0 in the initialization of Algorithm 4.1, and assume that a CONVEXSOLVER$_{\bar{\epsilon},\bar{\delta},\bar{\rho}}$ is used with $\bar{\epsilon}$, $\bar{\delta}$ and $\bar{\rho}$ chosen as in Lemma 4.6.1. Let $\{S^k\}_{k \in \mathbb{N}}$ be an infinite nested sequence of simplices generated by the variant of Algorithm 4.1, which employs only (GWSR). Assume further that this sequence has the properties that, for each $k \in \mathbb{N}$, there holds*

$$S^{k+1} \ = \ [v^k_0, \dots, v^k_{i-1}, w^k, v^k_{i+1}, \dots, v^k_n]$$

and

$$\mu(S^k) \ < \ \eta^k - \epsilon \, .$$

Then there exist a number $K \in \mathbb{N}$ and an integer r with $0 \leq r < n$, such that

$$S^k = [v_0, \dots, v_r, v^k_{r+1}, \dots, v^k_n] \ \ \forall k \geq K \ ,$$

where $v_0, \dots, v_r$ are fixed vectors, while $v^k_{r+1}, \dots, v^k_n$ ($k \in \mathbb{N}, k \geq K$) change infinitely often. Moreover, there holds

$$\bigcap_{k \in S} S^k \ = \ [v_0, \dots, v_r] \ =: \ S \, .$$

If we modify the generalized ω-subdivision rule in such a way, that in the result of Corollary 4.6.3 there must hold $r = 0$, then we can obtain an algorithm, which delivers in finite time either an (ϵ, δ, ρ)-solution of Problem (DCP) with ϵ, δ, $\rho > 0$ or the emptiness of F. The following **modified generalized ω-subdivision rule (MGWSR)** yields this intention. For this rule we need an additional counter $N(i, S^k)$, which shows how long the vertex v_i^k ($i \in \{0, \dots, n\}$) of S^k did not change.

In the initialization phase of Algorithm 4.1 we set, for each $i \in \{0, \dots, n\}$, $N(i, S^0) = 0$, and in each iteration this counter is adjusted, for each simplex S_j^k ($j \in I^k$), in the following way

$$\begin{aligned} N(i, S_j^k) &= N(i, S^k) + 1 \text{ , if } i \in \{0, \dots, n\} \setminus \{j\} \\ N(j, S_j^k) &= 0 \,. \end{aligned}$$

In order to formulate the modified rule we need, additionally, an arbitrary, but fixed number $\bar{N} \in \mathbb{N}$, which has to be chosen in the initialization phase of Algorithm 4.1. The (MGWSR) is now as follows.

```
Choose $\bar\lambda^k \in \{\lambda \in \mathbb{R}^{n+1} : \sum_{i=0}^n \lambda_i = 1\}$ with $\omega(S^k) = \sum_{i=0}^n \bar\lambda_i^k v_i^k$.
$\bar I^k \leftarrow \{i \in \{0, \dots, n\} : \bar\lambda_i^k > 0\}$
If $|\bar I^k| = 1$ Then
   $w^k \leftarrow \frac{1}{2}\left(v_{i_0}^k + v_{i_1}^k\right)$                                  (SR1)
   (i.e., choose a classical bisection, see (4.2.12) for the definition of $i_0$ and $i_1$)
Else
   Determine $N_1$ and $i_2$ with $N_1 = N(i_2, S^k) = \max_{i \in \{0,\dots,n\}} N(i, S^k)$
   and $N_2 = \max_{i \in \{0,\dots,n\} \setminus \{i_2\}} N(i, S^k)$.
   If $N_2 > \bar N$ Then
      $w^k \leftarrow \frac{1}{2}\left(v_{i_0}^k + v_{i_1}^k\right)$                               (SR2)
   Else
      Determine $\gamma^k := \sum_{i \in \bar I^k} \bar\lambda_i^k$
      $w^k \leftarrow \sum_{i \in \bar I^k} \frac{\bar\lambda_i^k}{\gamma^k} v_i^k$                                (SR3)
   EndIf
EndIf
```

THEOREM 4.6.4. *Assume that ϵ, δ and ρ are chosen greater than 0 in the initialization of Algorithm 4.1, and assume that a CONVEXSOLVER$_{\bar{\epsilon},\bar{\delta},\bar{\rho}}$ is used with $\bar{\epsilon}$, $\bar{\delta}$ and $\bar{\rho}$ chosen as in Lemma 4.6.1. Then the variant of Algorithm 4.1, which employs only (MGWSR), detects for Problem (DCP), in particular for (DCP$_3$), in finite time either the emptiness of the feasible region F or an (ϵ, δ, ρ)-solution.*

PROOF: Assume, by contradiction, that Algorithm 4.1 is not finite and let, without loss of generality, $\{S^k\}_{k\in\mathbb{N}}$ be an infinite nested sequence of simplices generated by this approach with the property $\mu(S^k) < \eta^k - \epsilon$ ($k \in \mathbb{N}$). In view of Remark 4.6.2(a) we know that w^k must be chosen by the selection rules (SR2) or (SR3). If there holds

$$|\{k \in \mathbb{N} : \ w^k \text{ is chosen by (SR2)}\}| \ < \ \infty \,, \tag{4.6.8}$$

then we can assume that w^k ($k \in \mathbb{N}$) is always chosen by (SR3). With respect to Corollary 4.6.3 we obtain the existence of a number $K \in \mathbb{N}$ and an integer $0 \leq r < n$ with

$$S^k \ = \ [v_0, \ldots , v_r, v_{r+1}^k, \ldots , v_n^k] \quad \forall k \geq K \,.$$

Since each w^k is chosen by (SR3) there holds $r = 0$. Otherwise, we would obtain

$$\min \{N(0, S^k) \ , \ N(1, S^k)\} \ > \ \bar{N}$$

for $k \in \mathbb{N}$ big enough, which would force the selection of w^k by (SR2). The fact that r is equal to 0 together with the second result of Corollary 4.6.3 implies the exhaustiveness of the sequence $\{S^k\}_{k\in\mathbb{N}}$, and, because of ϵ, δ, $\rho > 0$, Theorem 4.3.1 yields a contradiction.

Therefore, (4.6.8) cannot be true, i.e., an infinite number of elements of $\{S^k\}_{k\in\mathbb{N}}$ must be generated by the classical bisection rule. Denote by $d(S^k) := \max_{i,j=0,\ldots,n} \|v_i^k - v_j^k\|_2$ the diameter of S^k and the maximal distance of w^k to any vertex of S^k by $d(w^k, S^k) := \max_{i=0,\ldots,n} \|w^k - v_i^k\|_2$. If there exists a constant value $\tau \in (0, 1)$ with the property that, for each $k \in \mathbb{N}$, there holds

$$d(w^k, S^k) \ \leq \ \tau d(S^k) \,, \tag{4.6.9}$$

then it is a known fact [HT96B, Proposition VII.4] that infinitely many bisections guarantee the exhaustiveness of $\{S^k\}_{k\in\mathbb{N}}$.

As mentioned before we know, in view of Theorem 4.3.1 and because of ϵ, δ, $\rho > 0$, that $\{S^k\}_{k\in\mathbb{N}}$ cannot shrink to a singleton. Thus, there does not exist a value

$\tau \in (0,1)$ with property (4.6.9). I.e., there is at least a subsequence $\{S^{k_q}\}_{q\in\mathbb{N}}$ of $\{S^k\}_{k\in\mathbb{N}}$ with

$$\frac{d(w^{k_q}, S^{k_q})}{d(S^{k_q})} \;\to\; 1 \qquad (q \to \infty) \tag{4.6.10}$$

and, for each $q \in \mathbb{N}$,

$$w^{k_q} \text{ is chosen by (SR3)} .$$

Note that $\{d(S^k)\}_{k\in\mathbb{N}}$ is a non-increasing sequence, which is, in view of the non-exhaustiveness of $\{S^k\}_{k\in\mathbb{N}}$, convergent to a real value $\bar{d} > 0$, and note, furthermore, that there holds

$$d(w^k, S^k) \;\leq\; \tfrac{\sqrt{3}}{2} d(S^k) ,$$

if w^k is chosen by bisection (see, e.g., the proof of Proposition 3.14 in [HPT95]).

Since each vertex sequence $\{v_i^{k_q}\}_{q\in\mathbb{N}}$ ($i \in \{0, \ldots , n\}$) is bounded, we can assume, without loss of generality, that they are convergent to points $\bar{v}_i$ ($i = 0, \ldots , n$). Let $\lambda^q \in B_n$ ($q \in \mathbb{N}$) be chosen such that

$$w^{k_q} \;=\; \sum_{i=0}^{n} \lambda_i^q v_i^{k_q} .$$

Because of the boundedness of $\{\lambda^q\}_{q\in\mathbb{N}}$ we can further assume, without loss of generality, that this sequence is also convergent to a vector $\bar{\lambda} \in B_n$ and we obtain

$$w^{k_q} \;\to\; \sum_{i=0}^{n} \bar{\lambda}_i \bar{v}_i \;=:\; \bar{w} . \tag{4.6.11}$$

We prove now that, taking (4.6.10) into account, there holds

$$\bar{w} \;\in\; \{\bar{v}_0, \ldots , \bar{v}_n\} . \tag{4.6.12}$$

PROOF OF (4.6.12): Obviously we know that

$$d(S^{k_q}) \;\to\; \max_{i,j=0,\ldots,n} \|\bar{v}_i - \bar{v}_j\|_2 = \bar{d} \quad (q \to \infty) . \tag{4.6.13}$$

For each $q \in \mathbb{N}$, there is an index $i(q) \in \{0, \ldots , n\}$ with

$$d(w^{k_q}, S^{k_q}) \;=\; \|w^{k_q} - v_{i(q)}^{k_q}\|_2 .$$

Assume, without loss of generality, that $i(q)$ is always the same index, i.e.,

$$d(w^{k_q}, S^{k_q}) \;=\; \|w^{k_q} - v_{i'}^{k_q}\|_2 \tag{4.6.14}$$

for each $q \in \mathbb{N}$ and a fixed $i' \in \{0, \ldots , n\}$.

Combining (4.6.10), (4.6.11), (4.6.13) and (4.6.14) we obtain

$$\begin{aligned} 0 \ < \ \bar{d} \ = \ \|\bar{w} - \bar{v}_{i'}\|_2 \ &= \ \|\sum_{j=0}^{n} \bar{\lambda}_j \bar{v}_j - \bar{v}_{i'}\|_2 \\ &= \ \|\sum_{j=0}^{n} \bar{\lambda}_j \left(\bar{v}_j - \bar{v}_{i'}\right)\|_2 \ \le \ \sum_{j=0}^{n} \bar{\lambda}_j \underbrace{\|\bar{v}_j - \bar{v}_{i'}\|_2}_{\le \bar{d}} \ \le \ \bar{d}\,. \end{aligned}$$

Therefore, we see that,

$$\|\sum_{j=0}^{n} \bar{\lambda}_j \left(\bar{v}_j - \bar{v}_{i'}\right)\|_2 \ = \ \sum_{j=0}^{n} \bar{\lambda}_j \| \left(\bar{v}_j - \bar{v}_{i'}\right)\|_2 \ = \ \bar{d}\,. \tag{4.6.15}$$

Because of $\bar{d} > 0$ we know that the set

$$L \ := \ \{j \in \{0, \dots, n\} : \ (\bar{v}_j - \bar{v}_{i'}) \ne 0 \text{ and } \bar{\lambda}_j > 0\}$$

is not empty, and, in view of the right-hand side of (4.6.15), for each $j \notin L$, there holds $\bar{\lambda}_j = 0$. Indeed, if L is empty, then we obtain $\sum_{j=0}^{n} \bar{\lambda}_j \|\bar{v}_j - \bar{v}_{i'}\|_2 = 0$ contradicting $\bar{d} > 0$. Moreover, if there is an index $j' \notin L$ with $\bar{\lambda}_{j'} > 0$, then it follows that $\sum_{j=0}^{n} \bar{\lambda}_j \|\bar{v}_j - \bar{v}_{i'}\|_2 \le \sum_{j=0, j\ne j'}^{n} \bar{\lambda}_j \bar{d} < \bar{d}$ contradicting (4.6.15). The left-hand equality in Relation (4.6.15) is only possible if there exists, for each pair $i, j \in L$, a scalar $\gamma_{i,j} \in \mathrm{I\!R} \setminus \{0\}$ with

$$\bar{v}_i - \bar{v}_{i'} \ = \ \gamma_{i,j} \frac{\bar{\lambda}_j}{\bar{\lambda}_i} (\bar{v}_j - \bar{v}_{i'})\,.$$

Furthermore, again in view of the right-hand equation of (4.6.15), we obtain

$$\|\bar{v}_i - \bar{v}_{i'}\|_2 \ = \ \|\bar{v}_j - \bar{v}_{i'}\|_2 \ = \ \bar{d}\,, \tag{4.6.16}$$

and, therefore,

$$\gamma_{i,j} \frac{\bar{\lambda}_j}{\bar{\lambda}_i} \ \in \ \{-1, 1\}\,.$$

Assume that $\gamma_{i,j} \frac{\bar{\lambda}_j}{\bar{\lambda}_i} = -1$. Then there holds $(\bar{v}_i - \bar{v}_{i'}) = (\bar{v}_{i'} - \bar{v}_j)$. This implies

$$2\bar{d} \ = \ \|2(\bar{v}_i - \bar{v}_{i'})\|_2 \ = \ \|\bar{v}_i - \bar{v}_{i'} + \bar{v}_{i'} - \bar{v}_j\|_2 \ = \ \|\bar{v}_i - \bar{v}_j\|_2 \ \le \ \bar{d}\,,$$

which contradicts $\bar{d} > 0$. It follows, that there is an index $i'' \in L$ such that, for each $j \in L$, there holds

$$\bar{v}_{i''} \ = \ \bar{v}_j \ ,$$

and we conclude

$$\bar{w} \ = \ \sum_{j \in L} \bar{\lambda}_j \bar{v}_j + \sum_{j \notin L} \underbrace{\bar{\lambda}_j}_{=0} \bar{v}_j \ = \ \bar{v}_{i''} \underbrace{\sum_{j \in L} \bar{\lambda}_j}_{=1} \ = \ \bar{v}_{i''}\ ,$$

i.e., (4.6.12) is true. □

Assume, without loss of generality, that there holds $\bar{w} = \bar{v}_0$. Since the points w^{k_q} ($q \in \mathbb{N}$) are chosen by (SR3) we know, regarding Lemma 4.6.1, that, for each $q \in \mathbb{N}$, w^{k_q} is ($\frac{3}{4}\delta$, $\frac{3}{4}\rho$)-feasible for the convex subproblem (DCP$^{S^{k_q}}$). The point sequences $\{w^{k_q}\}_{q\in\mathbb{N}}$ and $\{v_0^{k_q}\}_{q\in\mathbb{N}}$ converge to the same limit point $\bar{v}_0$. Therefore, we obtain by continuity of f^l and g^l ($l \in \{0, \ldots, p\}$) (see, e.g., again [ROC70, Theorem 10.1]) and by using the same representation of w^{k_q} ($q \in \mathbb{N}$) as in the proof of (4.6.12), for each $l \in \{0, \ldots, p\}$,

$$g^l(w^{k_q}) - g^l(v_0^{k_q}) \;\to\; 0 \quad (q \to \infty)$$

and

$$\varphi^l_{S^{k_q}}(w^{k_q}) - f^l(v_0^{k_q}) = \varphi^l_{S^{k_q}}(w^{k_q}) - \varphi^l_{S^{k_q}}(v_0^{k_q}) \;\to\; 0 \quad (q \to \infty)\,.$$

It follows, that there exists a $Q \in \mathbb{N}$ such that, for each $q \geq Q$,

$$v_0^{k_q} \text{ is } (\delta,\, \rho)\text{-feasible}$$

and

$$g^0(v_0^{k_q}) + f^0(v_0^{k_q}) \;\leq\; g^0(w^{k_q}) + \varphi^0_{S^{k_q}}(w^{k_q}) + \tfrac{1}{8}\epsilon\,. \tag{4.6.17}$$

Since each (δ, ρ)-feasible vertex of an iteration simplex S^k ($k \in \mathbb{N}$) was used for updating the upper bound, it follows from Relation (4.6.17) and Relation (4.6.1.b) of Lemma 4.6.1 that

$$g^0(v_0^{k_q}) + f^0(v_0^{k_q}) \;\leq\; \mu(S^{k_q}) + \tfrac{7}{8}\epsilon \;<\; \eta^{k_q} - \tfrac{1}{8}\epsilon \;\leq\; g^0(v_0^{k_q}) + f^0(v_0^{k_q}) - \tfrac{1}{8}\epsilon\,,$$

which is also a contradiction and completes the proof. ■

4.6.3. Comparison of Different Subdivision Strategies Based on (MGWSR).

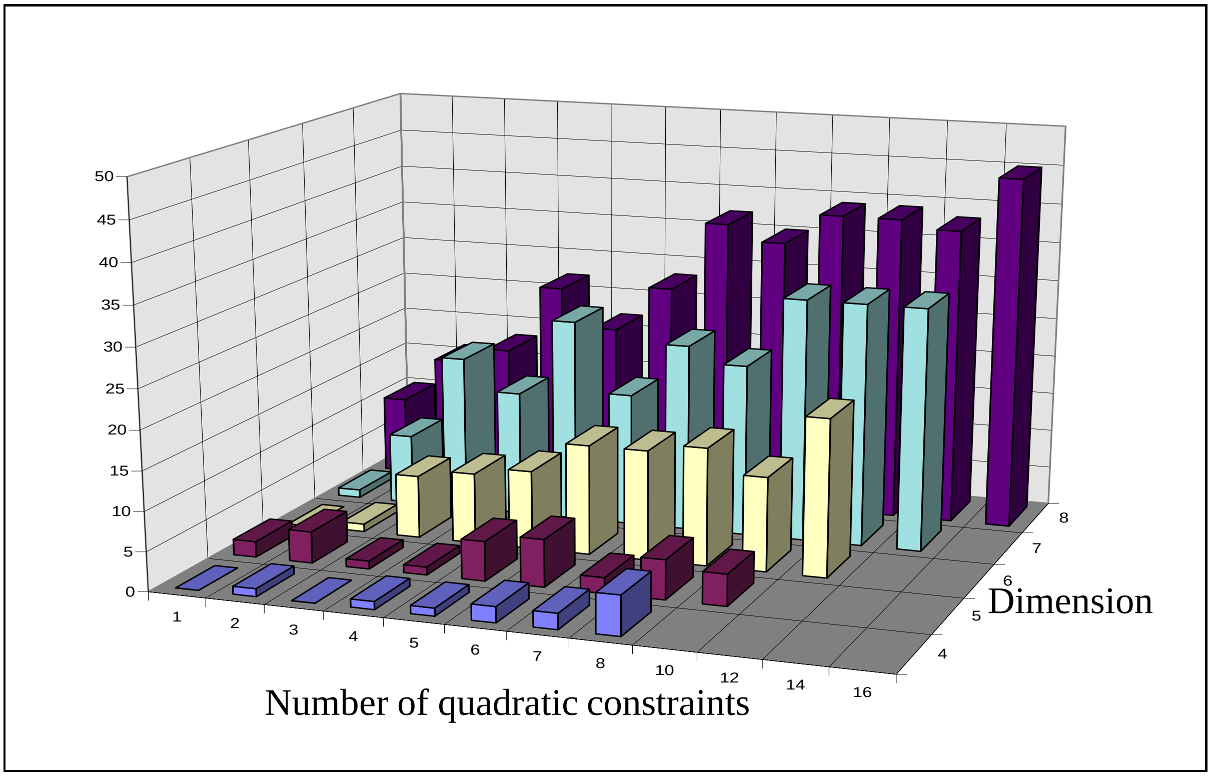
Theorem 4.6.4 guarantees that Algorithm 4.1 with (MGWSR) delivers in finite time an (ϵ, δ, ρ)-solution of our test problems, if the accuracies for the used CONVEX-SOLVER$_{\bar{\epsilon},\bar{\delta},\bar{\rho}}$ are chosen sufficiently small. We implemented the modified generalized ω-subdivision rule and used again the *NAG*-routine *E04UCC*, where the accuracies $\bar{\epsilon}$, $\bar{\delta}$ and $\bar{\rho}$ were calculated as required in Lemma 4.6.1. With this implementation we solved all test problems. However, in order to avoid excessive running-time we stopped the calculations, if more than $200,000$ convex subproblems were solved. The variant of Algorithm 4.1, which uses only bisections, needed less than this maximal number of convex subproblems for solving any test problem. The tolerances ϵ, δ and ρ were the same as in the numerical experiments using bisection and for $\bar{N}$ we chose $2n$.

In Table 4.4 we compare the numerical performance of Algorithm 4.1 using (MGWSR) with the performance of Algorithm 4.1 only employing bisections. The displayed test results for the dimensions $n = 2$ and $n = 3$ were run on *SUN SPARC 10* workstations. The used abbreviations are the same as in Table 4.1 and 4.2. **M**ω stands for Algorithm 4.1 with (MGWSR) and **Bi** for Algorithm 4.1 using bisections. Even though Algorithm 4.1 using (MGWSR) was mostly in more than 50% of the test examples faster than Algorithm 4.1 using bisections, the average running-time was only in a few cases lower. Furthermore, the standard deviation of the number of solved convex subproblems as well as of the running-time was higher in the case of (MGWSR) and was growing faster. The reason is that Algorithm 4.1 using (MGWSR) was in many test examples slightly faster than the version of Algorithm 4.1 with bisections, at least for small dimensions, but, in particular for growing dimensions and a growing number of quadratic constraints, this approach was in more and more test examples significantly slower than Algorithm 4.1 using bisections. For dimensions higher than $n = 3$ Algorithm 4.1 only employing (MGWSR) did not solve all 50 test problems for each couple (n, p) of the dimension n and the number of quadratic constraints p with less than $200,000$ convex subproblems. As it can be seen in Figure 4.5, the number of not-solved test problems increased with growing dimensions and a growing number of quadratic constraints. For example, for dimension $n = 8$ and $p = 16$ quadratic constraints

TABLE 4.4. Comparison of (MGWSR) and bisection

p	NuP	AvgNuSP		StdSP		AvgTime		StdTime	
	Mω<Bi	Mω	Bi	Mω	Bi	Mω	Bi	Mω	Bi
$n = 2$									
1	46	16.50	29.64	33.16	12.96	0.13	0.23	0.16	0.10
2	40	14.94	23.60	26.08	12.40	0.14	0.21	0.17	0.10
3	43	28.54	33.92	43.50	14.12	0.30	0.34	0.36	0.17
4	35	86.10	34.40	366.9	10.93	1.31	0.36	3.58	0.13
$n = 3$									
1	43	105.6	78.00	386.3	51.47	1.04	0.91	2.97	0.60
2	37	78.06	80.16	97.05	47.03	1.05	1.03	1.43	0.66
3	37	181.7	101.3	517.9	67.40	2.76	1.43	6.35	0.95
4	25	113.1	82.00	131.4	43.38	1.64	1.20	1.64	0.59
5	30	294.1	88.20	966.1	48.50	3.94	1.40	10.98	0.72
6	22	205.4	88.80	308.5	44.31	4.52	1.54	7.17	0.72

FIGURE 4.5. Number of test problems where Algorithm 4.1 using (MGWSR) needed more than $200,000$ convex subproblems



Algorithm 4.1 using (MGWSR) needed less than $200,000$ convex subproblems for only 5 test problems. Remember that the same approach, which used only bisections, solved all test problems with less than this maximal number of calls of the subroutine *E04UCC*.

Due to this reason we obtain that Algorithm 4.1 with bisections led to a more robust solution process, where *more robust* is meant in the following sense. The number of necessary convex subproblems and, thus, the running-time did not vary so keenly, as it was the case, when (MGWSR) was applied. The effort for detecting an (ϵ, δ, ρ)-solution of the quadratic test problems was rather predictable. Even though the variant of Algorithm 4.1, which employs only bisections, was also for higher dimensions not always the fastest approach, it was a better approach than the same algorithm using (MGWSR), since it did not show numerical outliers.

Our numerical experiments further showed a regularization effect of the bisection with respect to the effort for solving the convex subproblems. If the generated simplices tend to degenerate, i.e., they become too *flat*, as it is possible by using (MGWSR), then numerical problems, e.g., ill-conditioned constraint matrices, can occur and can lead to a substantially growing effort for solving the convex subproblems. If the classical bisection rule is used, then the risk of the occurrence of such numerical problems is much lower. In our numerical tests such problems did not appear in connection with the use of the bisection.

In order to make (MGWSR) more robust in the sense mentioned above, we could reduce the number $\bar{N}$. With $\bar{N} = 2n$ Algorithm 4.1 using (MGWSR) chose w^k by (SR3) on average in 91.5% of the performed subdivisions (see also Table 4.8 and Table 4.9). If we reduce $\bar{N}$, the number of bisections will increase and the numerical performance of Algorithm 4.1 using (MGWSR) will approach to the numerical performance of the variant of this algorithm, which employs only bisections. Note that, for each nested sequence $\{S^k\}_{k \in \mathbb{N}}$, all simplices S^k $(k > \bar{N})$ must be generated by bisection, if $\bar{N}$ is chosen smaller than $n - 1$.

As mentioned before, there were also for higher dimensions test examples where the (MGWSR) strategy was the best one. If we simply reduce $\bar{N}$, then we will obtain a more robust algorithm, but, at the end we have nothing else than an algorithm using bisections, and it is likely that we loose the not frequent, but really good results of (MGWSR) for some test examples. Maybe it is possible to develop a strategy, which is a mixture of the classical bisection rule and (MGWSR), and which shows a good performance in all cases, i.e., which use (MGWSR), if this strategy is the fastest one, and bisection, if (MGWSR) does not work.

In the case of problems of type (DCP$_1$) we know that $\omega(S^k)$ $(k \in \mathbb{N})$ is always a feasible point, at least if a CONVEXSOLVER$_{0,0,0}$ is used. Therefore, it is reasonable to hope that this point is a better choice than the point obtained by bisection, since $\omega(S^k)$ is connected to the information returned by the algorithm inside the selected simplex S^k. This is the main reason for the suggestion of the ω-subdivision, e.g., in [HT96B]. In the general case of problems of type (DCP$_3$) the point $\omega(S^k)$ is the approximate solution of a subproblem, where the objective function as well as the constraints are relaxed. Therefore, we cannot hope that $\omega(S^k)$ has something to do with the solution of the original problem. If $\omega(S^k)$ is at least (δ, ρ)-feasible, then this point is used for updating the upper bound, and we have more hope that $\omega(S^k)$ bears some information about the original problem. Thus the first mixed strategy we used was the following **(MGWSR1)**

```
If $\bar{w}^k$ is $(\delta, \rho)$-feasible Then
    Choose $w^k$ by (MGWSR).
Else
    Choose $w^k$ by bisection.
EndIf
```

where $\bar{w}^k$ is defined as $\sum_{i \in \bar{I}^k} \frac{\bar{\lambda}_i^k}{\gamma^k} v_i^k$ with $\bar{\lambda}^k \in \mathrm{I\!R}^{n+1}$, $\gamma^k \in \mathrm{I\!R}$ and $\bar{I}^k \subset \{0, \ldots, n\}$ given as in Lemma 4.6.1.

This strategy was much more robust than (MGWSR) alone, but the average running-times for higher dimensions were still slower than by using only bisections. In this case 2 test problems with $n = 7$ and 17 problems with $n = 8$ were not solved. Therefore, we developed further strategies. In these strategies we try to use more information about the Problem (DCP).

Denote by m^k the middle point of the longest edge of S^k ($k \in \mathrm{I\!N}$), i.e.,

$$m^k \;=\; \tfrac{1}{2}(v_{i_0}^k + v_{i_1}^k)\,,$$

with i_0 and i_1 defined as in (4.2.12). Denote, further, for $l \in \{0, \ldots, p\}$ and $x \in \mathrm{I\!R}^n$, by

$$\tau_{S^k}^l(x) \;:=\; f^l(x) - \varphi_{S^k}^l(x)$$

the difference between the function values of the concave function f^l and its convex envelope at the point x.

In the second mixed strategy **(MGWSR2)** we require now, additionally, that $\tau_{S^k}^0(\bar{w}^k)$ is greater than $\tau_{S^k}^0(m^k)$. This means that we choose just the point $\bar{w} \in \{\bar{w}^k, m^k\}$ which pushes most the convex envelope. Note that the convex envelopes $\varphi_{S_j^k}^0$ ($j \in I^k$) coincide at $\bar{w}$ with f^0, where S_j^k is the simplex resulting from the subdivision of S^k with respect to the point $\bar{w}$ (see the formulation of Algorithm 4.1).

This strategy showed on average a better running-time performance than (MGWSR1). However, 5 test problems with $n = 7$ and 8 problem with $n = 8$ were still not solved with less than $200,000$ convex subproblems.

Therefore, in the third mixed strategy we further intensified the decision criterion for the use of (MGWSR) by considering also the constraints. The point w^k is chosen only by (MGWSR), if there holds

$$\bar{w}^k \;\text{ is } (\delta,\, \rho)\text{-feasible,} \tag{C.1}$$

$$\tau^0_{S^k}(\bar{w}^k) \; > \; \tau^0_{S^k}(m^k) \tag{C.2}$$

and

$$\max_{l \in \{1,\dots,p\}} \tau^l_{S^k}(\bar{w}^k) \; > \; \max_{l \in \{1,\dots,p\}} \tau^l_{S^k}(m^k)\,. \tag{C.3}$$

This strategy **(MGWSR3)** is the most robust strategy with respect to all strategies using (MGWSR) we tested. Only one test problem with dimension $n = 8$ was not solved. On the other hand, as we will see later, applying this strategy only in a few iterations of Algorithm 4.1 the new simplices were generated by using $\bar{w}^k$ as subdivision point (see Tables 4.8 and 4.9). Thus, this strategy did not show in all relevant examples, i.e., in just the examples where (MGWSR) is the best approach, the good performance of (MGWSR) mentioned before. Therefore, in the last examined mixed strategy **(MGWSR4)** we relaxed again the criteria which had to be fulfilled in order to choose (MGWSR) instead of the bisection rule. The $(\delta,\, \rho)$-feasibility of $\bar{w}^k$ is no longer required. The number of iterations, where (MGWSR) is applied, increased again (see Tables 4.8 and 4.9), but, this strategy was less robust than (MGWSR3), as we can see in Figure 4.6.

FIGURE 4.6. Number of test problems where Algorithm 4.1 using (MGWSR4) needed more than $200,000$ convex subproblems

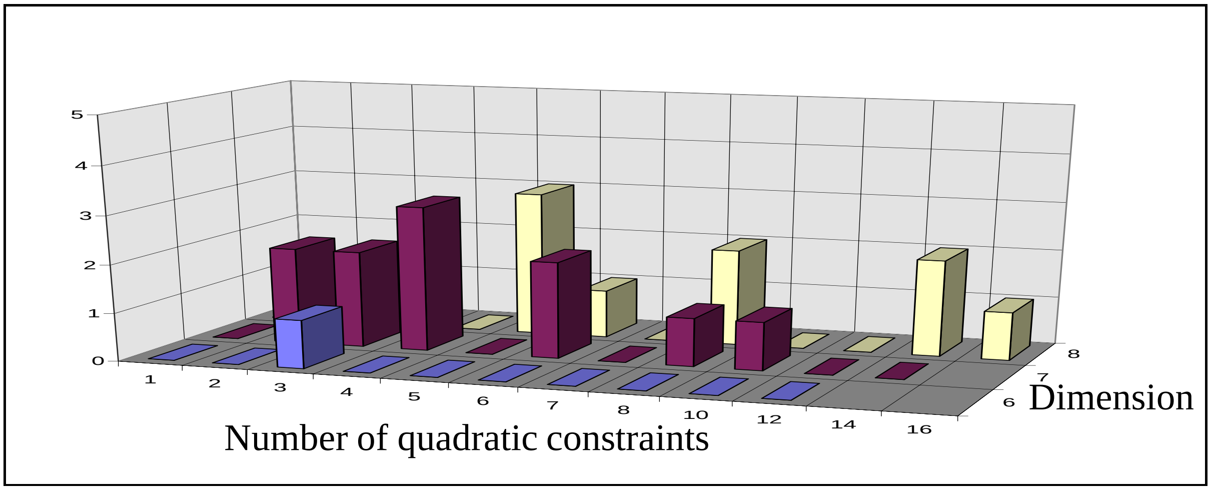


Table 4.5 gives an overview of the different subdivision strategies we tested and of the criteria which have to be fulfilled such that either (MGWSR) or the classical bisection is used.

TABLE 4.5. Different strategies and the used subdivision rules

STRATEGY	**USED SUBDIVISION RULE** (MGWSR)	Bisection
Bisection	*never*	*always*
(MGWSR)	*always*	*never*
(MGWSR1)	*if (C.1) is satisfied*	*otherwise*
(MGWSR2)	*if (C.1) and (C.2) are satisfied*	*otherwise*
(MGWSR3)	*if (C.1), (C.2) and (C.3) are satisfied*	*otherwise*
(MGWSR4)	*if (C.2) and (C.3) are satisfied*	*otherwise*

We also tested strategies requiring the (δ, ρ)-feasibility of m^k before analyzing the criterions (C.2) and (C.3). Our numerical tests showed that the feasibility of m^k was nearly never fulfilled and, thus, by requiring this feasibility we obtained a strategy which almost coincided with (MGWSR) or (MGWSR1).

With each of the presented six strategies we tried to solve all test problems. Because of the high number of test examples ($3,000$ for each strategy) we used several workstations, as it can be seen in Table 4.7. In order to make the running-times comparable all problems with the same dimension and the same number of quadratic constraints were calculated on the same machine.

In Tables 4.6 and 4.7 the average running-times in seconds are displayed for some of the solved test problems. Note that in the calculation of the average running-times we considered only the problems, which were solved by considering less than $200,000$ convex subproblems. Therefore, the corresponding number of solved problems is given in brackets next to the average running-time. The columns with respect to the bisection strategy are not comparable with the corresponding columns of Table 4.1 or Table 4.2, respectively, since other workstations were used for the calculations.

The third mixed strategy (MGWSR3) shows with respect to the running-time the best numerical performance among all strategies involving (MGWSR). In some cases this strategy was even faster than bisection. However, none of the strategies using ω-subdivision beats the numerical performance of the bisection strategy. Thus, and with respect to the possible numerical problems by using a strategy with (MGWSR) mentioned before, the bisection seems to be the best choice, at least for the tested set of all-quadratic optimization problems.

Table 4.6. Comparison of the average running-time in seconds for all strategies and $n = 2, 3, 4$

p	Bisection	(MGWSR)	(MGWSR1)	(MGWSR2)	(MGWSR3)	(MGWSR4)
$n = 2^a$						
1	0.22 (50)	0.13 (50)	0.13 (50)	0.16 (50)	0.18 (50)	0.18 (50)
2	0.21 (50)	0.14 (50)	0.14 (50)	0.15 (50)	0.18 (50)	0.18 (50)
3	0.34 (50)	0.30 (50)	0.30 (50)	0.34 (50)	0.32 (50)	0.31 (50)
4	0.36 (50)	1.31 (50)	0.60 (50)	0.55 (50)	0.34 (50)	0.34 (50)
$n = 3^a$						
1	0.91 (50)	1.04 (50)	0.62 (50)	0.65 (50)	0.71 (50)	0.72 (50)
2	1.03 (50)	1.05 (50)	1.00 (50)	0.86 (50)	0.93 (50)	0.92 (50)
3	1.43 (50)	2.76 (50)	1.51 (50)	1.39 (50)	1.25 (50)	1.36 (50)
4	1.20 (50)	1.64 (50)	1.54 (50)	1.37 (50)	1.10 (50)	1.10 (50)
5	1.40 (50)	3.94 (50)	1.74 (50)	1.55 (50)	1.31 (50)	1.41 (50)
6	1.54 (50)	4.52 (50)	2.47 (50)	1.86 (50)	1.46 (50)	1.54 (50)
$n = 4^a$						
1	1.94 (50)	2.53 (50)	1.53 (50)	1.51 (50)	1.59 (50)	1.69 (50)
2	2.26 (50)	2.89 (49)	2.10 (50)	1.79 (50)	1.86 (50)	2.51 (50)
3	2.10 (50)	13.5 (50)	2.12 (50)	1.98 (50)	1.96 (50)	2.76 (50)
4	3.62 (50)	19.8 (49)	3.90 (50)	3.75 (50)	3.42 (50)	3.70 (50)
5	2.78 (50)	15.6 (50)	3.16 (50)	2.99 (50)	2.65 (50)	2.83 (50)
6	3.97 (50)	36.7 (48)	4.85 (50)	4.48 (50)	3.80 (50)	4.06 (50)
7	3.79 (50)	28.3 (48)	4.92 (50)	4.54 (50)	3.70 (50)	3.99 (50)
8	3.95 (50)	13.3 (45)	5.39 (50)	4.83 (50)	3.84 (50)	4.03 (50)

[a] run on *SUN SPARC 10* workstations

We conclude the numerical comparisons by a consideration of the average number of ω-subdivisions used by the different strategies (MGWSR), (MGWSR1)-(MGWSR4), i.e., we consider the number of subdivisions, where w^k was chosen by (SR3) and not by bisection. Tables 4.8 and 4.9 show the average number of ω-subdivisions for some test results in percent. Note that also for the calculation of these average numbers we considered only the problems, which were solved with less than $200,000$ convex subproblems. For that reason, the corresponding numbers of solved test problems are given again in brackets (compare with Tables 4.6 and 4.7). As it was to be expected, the number of ω-subdivisions was reduced, if a stronger criterion for the choice of (MGWSR) was applied. Furthermore, it is not surprising that for strategies using at least criterion (C.1), i.e., for (MGWSR1), (MGWSR2) and (MGWSR3), the average proportional part of ω-subdivisions decreased, if the number of nonlinear constraints increased. The

TABLE 4.7. Comparison of the average running-times in seconds for all strategies and $n = 5, 6, 7, 8$

p	Bisection	(MGWSR)	(MGWSR1)	(MGWSR2)	(MGWSR3)	(MGWSR4)
$n = 5$[b]						
2	6.14 (50)	23.98 (46)	7.80 (50)	6.29 (50)	5.42 (50)	8.56 (50)
4	8.11 (50)	148.2 (49)	11.17 (50)	9.56 (50)	7.86 (50)	18.36 (50)
6	9.66 (50)	152.9 (44)	11.19 (50)	11.06 (50)	9.18 (50)	12.27 (50)
8	11.91 (50)	275.5 (45)	15.06 (50)	14.42 (50)	11.45 (50)	12.66 (50)
10	9.61 (50)	385.3 (46)	12.64 (50)	11.84 (50)	9.19 (50)	10.19 (50)
$n = 6$						
2[c]	14.27 (50)	175.4 (49)	99.29 (50)	30.42 (50)	11.70 (50)	21.92 (50)
4[c]	27.21 (50)	293.1 (41)	34.47 (50)	28.37 (50)	25.84 (50)	35.60 (50)
6[c]	37.10 (50)	403.8 (36)	46.37 (50)	43.33 (50)	36.96 (50)	83.81 (50)
8[a]	59.42 (50)	1530 (35)	73.45 (50)	72.13 (50)	57.96 (50)	68.60 (50)
10[b]	32.40 (50)	824.9 (38)	50.89 (50)	44.79 (50)	32.30 (50)	41.74 (50)
12[b]	51.83 (50)	2391 (30)	73.73 (50)	71.18 (50)	49.67 (50)	66.19 (50)
$n = 7$[d]						
2	24.97 (50)	109.8 (41)	55.20 (49)	21.19 (48)	31.17 (50)	48.23 (49)
4	29.86 (50)	251.8 (34)	58.50 (49)	32.11 (49)	30.98 (50)	41.02 (47)
6	23.05 (50)	382.1 (33)	31.55 (50)	29.47 (50)	22.59 (50)	29.51 (48)
8	36.92 (50)	396.4 (28)	43.86 (50)	41.27 (50)	35.95 (50)	66.27 (49)
10	41.16 (50)	973.1 (19)	48.07 (50)	46.09 (50)	40.04 (50)	60.06 (49)
12	44.21 (50)	784.3 (19)	52.07 (50)	51.23 (50)	43.06 (50)	53.53 (50)
14	52.46 (50)	688.2 (19)	60.58 (50)	60.13 (50)	51.46 (50)	75.98 (50)
$n = 8$						
2[c]	127.7 (50)	506.3 (34)	339.4 (47)	158.0 (49)	124.9 (50)	263.5 (49)
4[c]	149.5 (50)	756.2 (23)	303.2 (50)	201.6 (49)	153.2 (50)	309.9 (50)
6[c]	184.4 (50)	2115 (22)	313.3 (50)	234.5 (50)	199.2 (50)	239.0 (49)
8[c]	391.7 (50)	750.9 (15)	627.7 (50)	620.3 (50)	388.3 (50)	523.3 (48)
10[c]	226.8 (50)	3169 (11)	265.3 (50)	357.0 (50)	219.4 (50)	326.5 (50)
12[d]	118.2 (50)	1080 (10)	138.7 (50)	134.2 (50)	114.5 (50)	299.9 (50)
14[d]	134.7 (50)	1641 (12)	166.1 (50)	164.0 (50)	131.4 (50)	141.3 (48)
16[d]	164.4 (50)	2106 (5)	217.2 (50)	215.2 (50)	167.2 (50)	207.8 (49)

[a] run on *SUN SPARC 10* workstations

[b] run on *SUN SPARC 20* workstations

[c] run on *SUN SPARCserver 1000* workstations

[d] run on *SUN ULTRA 60* workstations

same is also true for strategy (MGWSR4), as we can see in the last columns of Table 4.8 and Table 4.9. However, using this strategy the variation of the proportional part with respect to a fixed dimension was not so high as for the other mixed

TABLE 4.8. Comparison of the average proportional part of subdivisions, where w^k is chosen by (SR3), for $n = 2, 3, 4$

p	(MGWSR)	(MGWSR1)	(MGWSR2)	(MGWSR3)	(MGWSR4)
$n = 2$					
1	97.46 (50)	27.04 (50)	7.88 (50)	3.98 (50)	5.19 (50)
2	93.44 (50)	18.27 (50)	7.36 (50)	2.91 (50)	5.04 (50)
3	95.75 (50)	14.99 (50)	4.20 (50)	1.21 (50)	2.75 (50)
4	98.49 (50)	11.60 (50)	4.49 (50)	0.590 (50)	2.57 (50)
$n = 3$					
1	95.24 (50)	24.53 (50)	7.70 (50)	3.10 (50)	5.83 (50)
2	93.87 (50)	10.06 (50)	4.01 (50)	1.14 (50)	5.68 (50)
3	95.57 (50)	9.08 (50)	4.53 (50)	1.41 (50)	5.51 (50)
4	91.71 (50)	4.63 (50)	2.01 (50)	0.298 (50)	2.45 (50)
5	93.66 (50)	4.26 (50)	1.40 (50)	0.227 (50)	2.89 (50)
6	92.56 (50)	4.29 (50)	1.92 (50)	0.088 (50)	2.60 (50)
$n = 4$					
1	95.44 (50)	23.58 (50)	5.76 (50)	5.47 (50)	9.85 (50)
2	95.09 (49)	19.85 (50)	5.18 (50)	2.25 (50)	5.33 (50)
3	94.44 (50)	4.62 (50)	2.77 (50)	2.40 (50)	6.92 (50)
4	94.67 (49)	3.60 (50)	1.85 (50)	0.303 (50)	4.79 (50)
5	93.14 (49)	1.76 (50)	0.831 (50)	0.155 (50)	3.97 (50)
6	94.65 (48)	3.44 (50)	1.62 (50)	0.170 (50)	4.39 (50)
7	92.83 (48)	2.55 (50)	0.746 (50)	0.186 (50)	3.13 (50)
8	94.68 (45)	1.95 (50)	0.532 (50)	— (50)	3.65 (50)

strategies. This shows that the feasibility criterion (C.1) is a strong one, at least for a high number of quadratic constraints.

Our numerical tests showed further that there are, in particular for higher number of nonlinear constraints ($p \geq n$), a lot of test examples where the three criterions are rarely satisfied together. Using (MGWSR3) there are many examples where only bisections were used for subdivision. This is demonstrated by the small numbers in the corresponding columns of Tables 4.8 and 4.9. For the pairs $(n, p) \in \{(4, 8), (5, 10)\}$ we even had a situation where for solving all 50 test problems with Algorithm 4.1 using (MGWSR3) a simplex was never generated by choosing the point w^k according to the rule (SR3), i.e., in this situation the strategy (MGWSR3) led to the same iterations as the bisection strategy.

TABLE 4.9. Comparison of the average proportional part of subdivisions, where w^k is chosen by (SR3), for $n = 5, 6, 7, 8$

p	(MGWSR)	(MGWSR1)	(MGWSR2)	(MGWSR3)	(MGWSR4)
$n = 5$					
2	90.50 (46)	20.84 (50)	5.70 (50)	3.57 (50)	8.02 (50)
4	91.71 (49)	7.86 (50)	3.05 (50)	1.02 (50)	4.91 (50)
6	92.62 (44)	1.61 (50)	0.583 (50)	0.138 (50)	5.08 (50)
8	88.95 (45)	0.868 (50)	0.423 (50)	0.0844 (50)	4.02 (50)
10	89.60 (46)	1.21 (50)	0.253 (50)	— (50)	3.71 (50)
$n = 6$					
2	94.05 (49)	69.38 (50)	4.48 (50)	3.21 (50)	18.21 (50)
4	91.29 (41)	9.75 (50)	2.95 (50)	0.583 (50)	5.52 (50)
6	78.99 (36)	4.46 (50)	1.58 (50)	0.302 (50)	5.46 (50)
8	87.23 (35)	1.37 (50)	2.03 (50)	0.229 (50)	4.78 (50)
10	80.53 (38)	3.61 (50)	0.413 (50)	0.0388 (50)	3.63 (50)
12	83.40 (30)	0.864 (50)	0.402 (50)	0.042 (50)	3.99 (50)
$n = 7$					
2	99.17 (41)	49.50 (49)	9.03 (48)	7.32 (50)	12.53 (49)
4	75.30 (34)	2.29 (49)	2.47 (49)	1.35 (50)	10.36 (47)
6	89.65 (33)	56.19 (50)	8.89 (50)	1.12 (50)	7.02 (48)
8	93.56 (28)	1.07 (50)	0.421 (50)	0.0283 (50)	7.08 (49)
10	91.96 (19)	1.24 (50)	0.185 (50)	0.0717 (50)	7.02 (49)
12	87.96 (19)	0.175 (50)	0.107 (50)	0.0028 (50)	2.71 (50)
14	94.06 (19)	0.345 (50)	0.108 (50)	0.0076 (50)	3.72 (50)
$n = 8$					
2	88.22 (34)	31.94 (47)	8.02 (49)	4.03 (50)	11.66 (49)
4	71.34 (23)	18.42 (50)	6.49 (49)	2.20 (50)	8.68 (50)
6	85.04 (22)	7.36 (50)	2.39 (50)	0.537 (50)	6.92 (49)
8	99.77 (15)	3.40 (50)	2.69 (50)	0.759 (50)	6.07 (48)
10	92.81 (11)	0.977 (50)	0.111 (50)	0.0536 (50)	5.77 (50)
12	86.51 (10)	0.383 (50)	0.253 (50)	0.0367 (50)	2.35 (50)
14	94.62 (12)	0.185 (50)	0.065 (50)	0.0022 (50)	6.20 (48)
16	86.25 (5)	0.0533 (50)	0.0198 (50)	0.0026 (50)	5.65 (49)

Finally, we have to note that all these suggested strategies did not manage our aim to develop a mixed strategy, which is the best one in all cases. Maybe it is possible to develop such a strategy by using other problem information than the information we used. The existence of such a strategy is still an open question.

4.7. A Finiteness Result

We conclude the chapter about the convergence of simplicial branch-and-bound algorithms based on ω-subdivisions with a partial answer to the theoretical problem of the finiteness of Algorithm 4.1, even with $\epsilon = \delta = \rho = 0$. In this section we consider problems of type (DCP$_1$), i.e., concave minimization problems over polytopes. In Section 4.4 we showed that the variant of Algorithm 4.1, which employs only ω-subdivisions, applied for problems of this type is convergent, if a CONVEXSOLVER$_{0,0,0}$ is available. Remember that in this situation there holds in the generalized ω-subdivision rule (GWSR) always $w^k = \omega(S^k)$. As we will see in this section it is even possible to prove the finiteness of this approach, if two additional assumptions are fulfilled.

Some finite simplicial branch-and-bound algorithms for solving problems of type (DCP$_1$) were proposed in the literature (see, e.g., [BEN85, TB85, BS94, NAS96]). In some cases finiteness was obtained by combining the simplicial approach with some other tools. For instance, in [BEN85] finiteness is yielded by the introduction of a neighbor generation mechanism. In some other cases finiteness was obtained by using different subdivision rules: instead of subdividing the selected simplex with respect to one of its points, as we did in Algorithm 4.1 (see, especially, the point selection rule (PSR) and the following lines in the formulation of this approach in Section 4.2), the selected simplex is subdivided by other techniques. One of such different subdivision techniques is described in [NAS96] and the corresponding algorithm is proven to be finite. On the other hand, the number of new simplices generated at each iteration by this technique may be extremely high, while by subdividing with respect to a point of the simplex, i.e., by applying a radial subdivision, this number is bounded from above by $n + 1$.

To the author's knowledge, no proof of finiteness for the basic simplicial branch-and-bound Algorithm 4.1 applied for solving problems of type (DCP$_1$) has been given apart from [LR97A]. Some simplicial branch-and-bound algorithms can even be proven to be infinite. For instance, for Algorithm 4.1 with bisections it is possible to construct counterexamples showing that the algorithm, though convergent, is not finite. In some other cases finiteness is still an open question.

As mentioned before, we need two additional assumptions in order to prove the finiteness of the version of Algorithm 4.1, which uses ω-subdivisions (with $\epsilon = \delta = \rho = 0$) and is applied for solving problems of type (DCP$_1$). The first of these assumptions is a mild one and can always be enforced, as it will be shown in the sequel. The second assumption is a strong one and cannot be guaranteed all the time. However, as we will see, this assumption is easy to check and holds in some special cases.

The assumptions are as follows.

(A.1) The function value of the concave objective function f^0 in a vertex of the start-simplex $S^0 = [v_0^0, \dots, v_n^0]$, which does not belong to the polytope P, is smaller than the optimal value $f^\star$ of f^0 over P, i.e., for each $i \in \{0, \dots, n\}$ with $v_i^0 \notin P$, there holds

$$f^0(v_i^0) \; < \; f^\star \, . \tag{A.1}$$

(A.2) Each vertex of the start-simplex S^0, which does not belong to the polytope P, violates one and only one of the constraints describing P, i.e., for each $i \in \{0, \dots, n\}$ with $v_i^0 \notin P$, there is exactly one index $j(i) \in \{1, \dots, m\}$ satisfying

$$a_{j(i)}^T v_i^0 \; > \; b_{j(i)} \, . \tag{A.2}$$

REMARK 4.7.1.

(a) The assumption (A.1) can be enforced for any start-simplex $S^0 \supset P$ by considering the concave function

$$\tilde{f}^0(x) \; := \; f^0(x) + M \min \{0, \min_{j=1,\dots,m} (b_j - a_j^T x)\} \, , \; x \in \mathbb{R}^n$$

with

$$M \; := \; \max_{\substack{i \, \in \, \{0, \dots, n\} \\ v_i^0 \, \notin \, P}} \left(\frac{f^0(v_i^0) - \mu(S^0) + 1}{|\min_{j=1,\dots,m} (b_j - a_j^T v_i^0)|} \right) ,$$

which has the same optimal value and optimal solutions as f^0 over the polytope P. Therefore, this assumption is not a substantial restriction. Remember that $\mu(S^0)$ denotes the optimal value of the initial linear problem (DCP$_1^{S^0}$).

(b) The assumption (A.2) depends only on the start-simplex S^0 and it is a strong one. Given an n-simplex S^0 it is easy to check whether S^0 satisfies (A.2). However, we cannot assume that for an arbitrary polytope P there exists an n-simplex $S^0 \supset P$ which fulfills this assumption. Nevertheless, there are some well-known instances, where it is possible to construct such simplices. If P is a hypercube, for example, then a start-simplex $S^0 \supset P$ satisfying (A.2) exists (compare, e.g., the construction of start polytopes in [HPT95, pp. 145f]).

Apart from these two assumptions we assume again, as in Section 4.4, that a CONVEXSOLVER$_{0,0,0}$ like the Simplex-Method is used for solving the linear subproblems (DCP$_1^S$). If the start-simplex S^0 and the concave objective function f^0 satisfy (A.1) and (A.2), then it can be shown that Algorithm 4.1 only employing ω-subdivisions, applied for solving problems of type (DCP$_1$), will stop after a finite number of iterations, even with $\epsilon = \delta = \rho = 0$. This is the result of the final Finiteness Theorem 4.7.3. At first, however, two additional lemmata are needed to establish this theorem.

The following lemma is equivalent to the first part of Lemma 4.4.2. Since there we used ϵ, $\delta > 0$ in order to prove Lemma 4.4.2 for problems of type (DCP$_1$) and (DCP$_2$), it is necessary to give another proof, which is valid in the case of (DCP$_1$) also for $\epsilon = \delta = 0$.

LEMMA 4.7.1. *Let $\{S^k\}_{k\in\mathbb{N}}$ be an infinite nested sequence of simplices generated by Algorithm 4.1 with the properties that, for each $k \in \mathbb{N}$, there holds*

$$S^{k+1} \;=\; [v_0^k, \ldots , v_{i-1}^k, \omega(S^k), v_{i+1}^k, \ldots , v_n^k] \tag{4.7.1.a}$$

and

$$\varphi^0_{S^k}(\omega(S^k)) \;=\; \mu(S^k) \;<\; \eta^k \tag{4.7.1.b}$$

(compare with Properties (4.4.2.a) and (4.4.2.b)). Then there exist a number $K \in \mathbb{N}$ and an integer r with $0 \le r < n$ such that, for each $k \ge K$, there holds

$$S^k \;=\; [v_0, \ldots , v_r, v_{r+1}^k, \ldots , v_n^k]\,, \tag{4.7.2}$$

where $v_0, \ldots , v_r$ are fixed vectors, while $v_{r+1}^k, \ldots , v_n^k$ ($k \ge K$) change infinitely often. Moreover, for each $i \in \{0, \ldots , r\}$, there holds

$$v_i \notin P \quad \Rightarrow \quad v_i \;=\; v_i^0\,. \tag{4.7.3}$$

PROOF: As we pointed out before Lemma 4.4.2, result (4.7.2) is a direct consequence of the feasibility of each generated point $\omega(S^k)$. Note that in the considered case (DCP$_1$) the feasible region of each subproblem is a subset of F and that we use a CONVEXSOLVER$_{0,0,0}$.

Indeed, assume, by contradiction, that in the infinite nested sequence $\{S^k\}_{k\in\mathbb{N}}$ all the vertices of the simplices $S^k = [v_0^k, \ldots , v_n^k]$ change infinitely often. Choose a number $\bar{K} \in \mathbb{N}$ such that each vertex of $S^{\bar{K}}$ has changed at least once, i.e.,

$$v_i^{\bar{K}} \;\ne\; v_i^0 \qquad \forall i \in \{0, \ldots , n\}\,.$$

Because of (4.7.1.a) it follows that, for each $i \in \{0, \ldots, n\}$, there is an index $k(i) \in \mathbb{N}$, $k(i) < \bar{K}$ satisfying

$$v_i^{\bar{K}} = \omega(S^{k(i)}) \in P = F . \tag{4.7.4}$$

Each feasible point $\omega(S^k)$ ($k \in \mathbb{N}$) is used for updating the upper bound. Therefore, we know that, for each $i \in \{0, \ldots, n\}$, there holds

$$f^0(v_i^{\bar{K}}) \geq \eta^{\bar{K}} ,$$

and, thus, we obtain, for each $x \in S^{\bar{K}}$,

$$\varphi^0_{S^{\bar{K}}}(x) = \sum_{i=0}^{n} \lambda_i f^0(v_i^{\bar{K}}) \geq \sum_{i=0}^{n} \lambda_i \eta^{\bar{K}} = \eta^{\bar{K}} \tag{4.7.5}$$

with $\lambda \in B_n$ and $x = \sum_{i=0}^{n} \lambda_i v_i^{\bar{K}}$. Combining (4.7.1.b) and (4.7.5) it follows for $\omega(S^{\bar{K}}) \in S^{\bar{K}}$

$$\eta^{\bar{K}} \leq \varphi^0_{S^{\bar{K}}}(\omega(S^{\bar{K}})) < \eta^{\bar{K}} ,$$

which is a contradiction and proves result (4.7.2).

Since each new vertex belongs to P (see (4.7.4)) Relation (4.7.3) is follows readily. ■

In order to obtain a contradiction in the proof of the final finiteness result we prove in the next lemma that, given an infinite nested sequence $\{S^k\}_{k \in \mathbb{N}}$ with Properties (4.7.1), there is a number $\hat{K} \in \mathbb{N}$ such that the infinitely changing vertices v of the residual simplices $\{S^k\}_{k \geq \hat{K}}$ have a special property, if (A.1) and (A.2) are satisfied. We are able to show that such a vertex v must be contained in the intersection of all hyperplanes, which are described by just the constraints of P, which are violated by at least one of the fixed vertices of $\{S^k\}_{k \geq \hat{K}}$.

LEMMA 4.7.2. *Let $\{S^k\}_{k \in \mathbb{N}}$ be an infinite sequence of simplices with Properties (4.7.1). Let further S^0 be a start-simplex in Algorithm 4.1 and $f^0 : \mathbb{R}^n \to \mathbb{R}$ be a concave function satisfying Assumptions (A.1) and (A.2), and let $K \in \mathbb{N}$ and $0 \leq r < n$ be given by Lemma 4.7.1, i.e., $S^k = [v_0, \ldots, v_r, v_{r+1}^k, \ldots, v_n^k]$ ($k \geq K$). Then there exists a number $\hat{K} \in \mathbb{N}$ such that, for each $i \in \{0, \ldots, r\}$ with $v_i \notin P$, there holds, for all $k \geq \hat{K}$ and $l \in \{r+1, \ldots, n\}$,*

$$a_{j(i)}^T v_l^k = b_{j(i)} , \tag{4.7.6}$$

where $j(i) \in \{1, \dots , m\}$ denotes – with respect to (A.2) – the unique constraint of P violated by v_i.

Proof: Choose an arbitrary, but fixed index $i' \in \{0, \dots , r\}$ with $v_{i'} \notin P$. From (4.7.3) we know that $v_{i'}$ is a vertex of S^0, and regarding (A.1) and the structure of the convex envelope we obtain, for $k \geq K$,

$$\varphi^0_{S^k}(v_{i'}) \ = \ f^0(v_{i'}) \ < \ f^\star \ . \tag{4.7.7}$$

Theorem 4.4.9 implies that the variant of Algorithm 4.1, which employs only ω-subdivisions and is applied for solving problems of type (DCP$_1$), is convergent in the sense that for the sequence $\{\mu(S^k)\}_{k\in\mathbb{N}}$ we obtain

$$\mu(S^k) \ \to \ f^\star \qquad (k \to \infty)$$

(compare with (4.4.21)). Moreover, we know that, for each $k \in \mathbb{N}$,

$$\mu(S^k) \ \leq \ f^\star \ .$$

Since (4.7.7) is fulfilled for each index $i \in \{0, \dots , r\}$ with $v_i \notin P$ we hence know that there is a number $\bar{K} \in \mathbb{N}$, $\bar{K} \geq K$ such that, for each $k \geq \bar{K}$,

$$\mu(S^k) \ > \max_{i\in\{0,\dots,r\}, v_i \notin P} f^0(v_i) \ . \tag{4.7.8}$$

Choose a number $\hat{K} \in \mathbb{N}$, $\hat{K} \geq \bar{K}$ such that, for each $k \geq \hat{K}$ and $l \in \{r + 1, \dots , n\}$, there exists an index $k(l) \in \mathbb{N}$, $\bar{K} \leq k(l) < k$ with

$$v^k_l \ = \ \omega(S^{k(l)}) \ .$$

Assume now, by contradiction, that Relation (4.7.6) is not true for i', i.e., there is an index $l' \in \{r + 1, \dots , n\}$ and a number $k' \geq \hat{K}$ satisfying

$$a^T_{j(i')} v^{k'}_{l'} \ < \ b_{j(i')} \ . \tag{4.7.9}$$

This means that $v^{k'}_{l'}$ is not active in the unique constraint of P violated by $v_{i'}$. Note that $v^{k'}_{l'}$ must be feasible.

Let $\bar{v}$ be the intersection point of the line segment between the two points $\omega(S^{k'(l')}) = v^{k'}_{l'}$ and $v_{i'}$ with the hyperplane $H := \{x \in \mathbb{R}^n : a^T_{j(i')} x = b_{j(i')}\}$, i.e.,

$$\{\bar{v}\} \ = \ [\omega(S^{k'(l')}), v_{i'}] \cap H \ \subset \ S^{k'(l')} \ .$$

Since $v_{i'}$ violates only the constraint $a_{j(i')}^T x \leq b_{j(i')}$ and $\omega(S^{k'(l')})$ does not violate any constraint, it follows that $\bar{v}$ is contained in $S^{k'(l')} \cap P$, and, furthermore, by using (4.7.9), that there exists a $\lambda \in (0,1)$ satisfying

$$\bar{v} \;=\; \lambda v_{i'} + (1-\lambda)\omega(S^{k'(l')}) \,.$$

Therefore, we obtain

$$\varphi^0_{S^{k'(l')}}(\bar{v}) \;=\; \lambda \underbrace{\varphi^0_{S^{k'(l')}}(v_{i'})}_{=f^0(v_{i'})} + (1-\lambda) \underbrace{\varphi^0_{S^{k'(l')}}(\omega(S^{k'(l')}))}_{=\mu(S^{k'(l')})}$$

and from (4.7.8) it follows that

$$\varphi^0_{S^{k'(l')}}(\bar{v}) \;<\; \varphi^0_{S^{k'(l')}}(\omega(S^{k'(l')})) \,,$$

which contradicts the minimality of $\omega(S^{k'(l')})$ with respect to $S^{k'(l')} \cap P$. ■

With this result the postulated finiteness of Algorithm 4.1 can now be shown.

THEOREM 4.7.3. *Assume that $\epsilon = \delta = \rho = 0$ and that a CONVEXSOLVER$_{0,0,0}$ is used. Assume further that the start-simplex S^0 is chosen and the concave objective function $f^0 : \mathbb{R}^n \rightarrow \mathbb{R}$ is given in a way such that Assumptions (A.1) and (A.2) are satisfied. Then the version of Algorithm 4.1, which employs only ω-subdivisions, will stop after a finite number of iterations, if it is applied for solving a problem of type (DCP$_1$).*

PROOF: Assume, by contradiction, that this version of Algorithm 4.1 does not stop after a finite number of iterations, i.e., the algorithm generates an infinite sequence $\{S^k\}_{k\in\mathbb{N}}$ of simplices. Then there exists an infinite nested subsequence $\{S^{k_q}\}_{q\in\mathbb{N}} \subset \{S^k\}_{k\in\mathbb{N}}$ with Properties (4.7.1). Choose $\hat{Q} \in \mathbb{N}$ as in Lemma 4.7.2 and $0 \leq r < n$ as in Lemma 4.7.1 and let, for each $q \geq \hat{Q}$, a vector $\lambda^q \in B_n$ be given with

$$\omega(S^{k_q}) \;=\; \sum_{i=0}^{r} \lambda_i^q v_i + \sum_{i=r+1}^{n} \lambda_i^q v_i^{k_q} \,.$$

Set, for $q \geq \hat{Q}$,

$$\beta^q \;:=\; \sum_{i=0}^{r} \lambda_i^q \,.$$

We prove first that, for each $q \geq \hat{Q}$, there holds

$$\beta^q \; \in (0,1) \, . \tag{4.7.10}$$

It is obvious that β^q belongs to $[0,1]$. If β^q is not different from 0 for an index $q' \geq \hat{Q}$, we obtain $\omega(S^{k_{q'}}) = \sum_{i=r+1}^{n} \lambda_i^{q'} v_i^{k_{q'}}$ and $\sum_{i=r+1}^{n} \lambda_i^{q'} = 1$. This implies because of the feasibility of $v_i^{k_{q'}}$ $(i = r+1, \dots, n)$ (see the choice of $\hat{Q}$ in the proof of Lemma 4.7.2) that

$$\mu(S^{k_{q'}}) \; = \; \varphi^0_{S^{k_{q'}}}(\omega(S^{k_{q'}})) \; = \; \sum_{i=r+1}^{n} \lambda_i^{q'} f^0(\underbrace{v_i^{k_{q'}}}_{\in P}) \; \geq \; \eta^{k_{q'}} \; , \tag{4.7.11}$$

contradicting (4.7.1.b). If there holds $\beta^{q'} = 1$, then we have $\omega(S^{k_{q'}}) \in [v_0, \dots, v_r]$. By the same argumentation as in the proof of Theorem 4.4.9 we obtain that this is either a contradiction to the property $S^{k_q} = [v_0, \dots, v_r, v_{r+1}^{k_q}, \dots, v_n^{k_q}]$ $(q \geq \hat{Q})$ of the simplex sequence $\{S^{k_q}\}_{q \in \mathbb{N}}$ or to Property (4.7.1.b). Therefore, we know that (4.7.10) is fulfilled for any $q \geq \hat{Q}$.

Now choose an arbitrary, but fixed $q \geq \hat{Q}$. By using (4.7.10) we are able to represent $\omega(S^{k_q})$ in the following way

$$\omega(S^{k_q}) \; = \; \beta^q \; \underbrace{\frac{1}{\beta^q}\left(\sum_{i=0}^{r} \lambda_i^q v_i\right)}_{:=w_1^q} + (1-\beta^q) \; \underbrace{\frac{1}{1-\beta^q}\left(\sum_{i=r+1}^{n} \lambda_i^q v_i^{k_q}\right)}_{:=w_2^q} \; , \tag{4.7.12}$$

such that we obtain

$$\mu(S^{k_q}) \; = \; \varphi^0_{S^{k_q}}(\omega(S^{k_q})) \; = \; \beta^q \varphi^0_{S^{k_q}}(w_1^q) + (1-\beta^q)\varphi^0_{S^{k_q}}(w_2^q) \, .$$

By the same argumentation as in (4.7.11) it follows that $\varphi^0_{S^{k_q}}(w_2^q) \; \geq \; \eta^{k_q}$ and, therefore, with Property (4.7.1.b) we know that

$$\varphi^0_{S^{k_q}}(w_1^q) \; < \; \eta^{k_q} \; \leq \; \varphi^0_{S^{k_q}}(w_2^q) \, .$$

This implies that $\varphi^0_{S^{k_q}}$ is strictly monotonously decreasing on the line between w_2^q and w_1^q. In view of (4.7.10) there holds $w_1^q \neq \omega(S^{k_q})$, and it follows

$$\varphi^0_{S^{k_q}}(w_1^q) \; < \; \varphi^0_{S^{k_q}}(\omega(S^{k_q})) \, .$$

Since $\omega(S^{k_q})$ is the optimal solution of $\min_{x \in S^{k_q} \cap F} \varphi^0_{S^{k_q}}(x)$, it follows

$$w_1^q \; \notin \; P = F \, .$$

Let $j' \in \{1, \ldots, m\}$ be the index of a constraint defining P violated by w_1^q. Then this constraint must be violated by one of the infeasible vertices of the fixed face $[v_0, \ldots, v_r]$ of S^{k_q} and it follows by Lemma 4.7.2 that, for each $l \in \{r+1, \ldots, n\}$, there holds

$$a_{j'}^T v_l^{k_q} \ = \ b_{j'}$$

and, thus,

$$a_{j'}^T w_2^q \ = \ b_{j'} \quad \text{and} \quad a_{j'}^T w_1^q \ > \ b_{j'} \ . \tag{4.7.13}$$

In view of (4.7.12) we know that $\omega(S^{k_q})$ is an element of the open line segment $LS := \{x \in \mathrm{I\!R}^n : \exists\, \lambda \in (0,1)$ with $x = \lambda w_1^q + (1-\lambda) w_2^q\}$ between the points w_1^q and w_2^q. However, using (4.7.13) we obtain, for each $x \in LS$,

$$x \ \notin \ P$$

contradicting the feasibility of $\omega(S^{k_q})$ and completing the proof. ■

The proofs for obtaining this finiteness result use substantially the feasibility of each generated solution $\omega(S^k)$ ($k \in \mathrm{I\!N}$), i.e.,

$$\omega(S^k) \ \in \ P \ = \ F \ .$$

This feasibility property of $\omega(S^k)$ is only given for the problem class (DCP$_1$) as long as a CONVEXSOLVER$_{0,0,0}$ is used. In the case of problems of type (DCP$_2$) we do not have this property of $\omega(S^k)$ ($k \in \mathrm{I\!N}$), and, therefore, the proof techniques suggested here cannot be extended to more general problem classes. Note that it is not necessary that Algorithm 4.1 applied for solving problems of type (DCP$_2$) or (DCP$_3$) detects in finite time a feasible point. In order to overcome this difficulty we just introduced the (δ, ρ)-feasibility concept in Section 4.2 (see Definition 4.2.1).

CHAPTER 5

Packing Equal Circles in a Square

In the previous three chapters we developed some approaches applicable for the solution of arbitrary nonconvex all-quadratic problems of type (QP). As mentioned in Section 1.1, the problem of packing equal circles with maximum radius into a square, which we would like to call in the following the *packing problem*, is an application of this class of global optimization problems. In the first section of the present chapter we will see that there is a one-to-one relation between solutions of the packing problem and solutions of Problem (PP) given on page 5. Problem (PP) is hence an equivalent formulation of the packing problem as an all-quadratic program and could be solved – at least theoretically – with one of the methods developed so far.

However, Problem (PP) is a $(2n + 1)$-dimensional program with $\binom{n}{2}$ concave quadratic constraints, where n is the number of circles which we would like to pack into the square. For reasons becoming evident in Section 5.1 we are interested in solutions of the packing problem for more than 20 circles. Therefore, we have to solve Problem (PP) with $n > 20$. In the numerical tests done for the approaches for solving (QP) discussed so far (see particularly Subsection 3.5.2) we recognized that these general methods are not able to solve all-quadratic problems with dimensions higher than 10, at least that they are not able to solve such problems with acceptable computational effort. Consequently, it is not surprising that these methods developed for general problems of type (QP) fail to solve Problem (PP). They are not able to determine approximate solutions of (PP) – for the required sizes of the dimension and the number of quadratic constraints – with acceptable effort.

Exploiting the structure of the packing problem, respectively of Problem (PP), we are non the less able to derive a new global solution method based on a rectangular branch-and-bound scheme, which can determine approximate solutions of the packing problem for more than 20 circles. The description of this new solution method is the main content of the present chapter.

5.1. Introduction

The packing problem is a widely explored problem in the field of optimization. One tries to find the maximum radius r of n equal and non-overlapping circles located within the unit square. This problem can be formulated as

$$\begin{aligned} &\max\ r \\ S(x_i, r) &\subset U && i = 1, \ldots, n \\ S(x_i, r) \cap S(x_j, r) &= \emptyset && 1 \leq i < j \leq n \end{aligned} \tag{CPP}$$

where, for each $i \in \{1, \ldots, n\}$, $S(x_i, r) := \{x \in \mathbb{R}^2 : \|x - x_i\|_2 < r\}$ denotes the open sphere with center $x_i \in \mathbb{R}^2$ and radius r, and $U := [0, 1]^2$ denotes the unit square.

As we will see below, the circle packing problem (CPP) is equivalent to the problem of scattering n points into the unit square such that the minimum pairwise distance d becomes as large as possible. This point scattering problem is given by

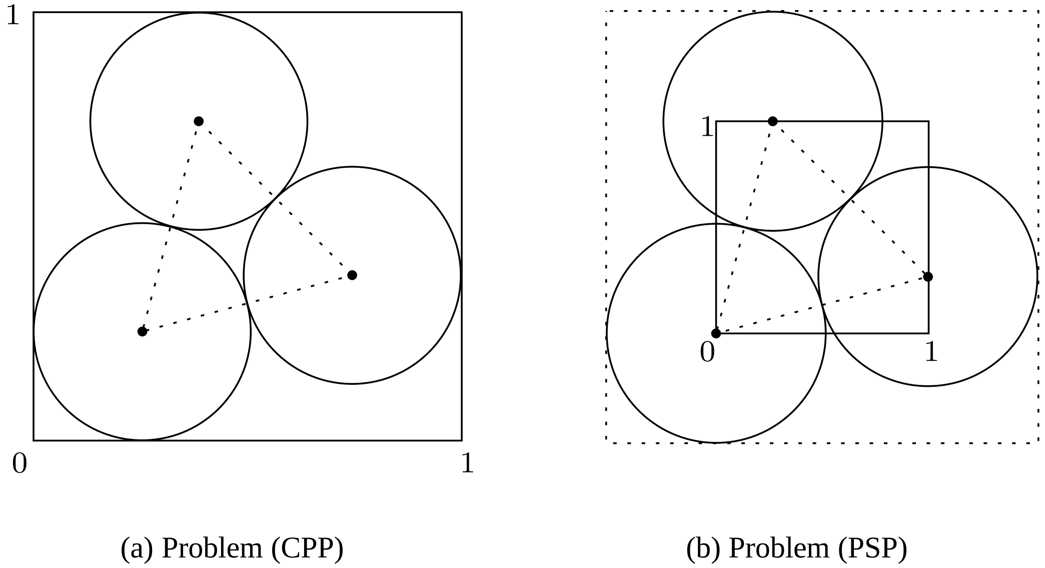
$$\begin{aligned} &\max\ d \\ d &\leq \|x_i - x_j\|_2 && 1 \leq i < j \leq n \\ x_i &\in U && i = 1, \ldots, n\,. \end{aligned} \tag{PSP}$$

In Problem (PSP) one considers only the centers of the circles. In contrast to Problem (CPP) it is allowed that a center-point x_i ($i \in \{1, \ldots, n\}$) belongs to the boundary of U, i.e., the constraints $S(x_i, r) \subset U$ ($i = 1, \ldots, n$) are neglected. In (PSP) we only require that x_i ($i \in \{1, \ldots, n\}$) is contained in U. The second group of constraints in (CPP) is obviously equivalent to the constraints $d \leq \|x_i - x_j\|_2^2$ ($1 \leq i < j \leq n$) in the formulation of (PSP). Even though Problem (CPP) and Problem (PSP) are not equivalent at first glance, there is still a one-to-one relation between the optimal solutions of both problems. It can be seen that there holds

$$r^\star(n) = \frac{d^\star(n)}{2(d^\star(n) + 1)} \tag{5.1.1}$$

(see, for example, [DGPWM91]), where $r^\star(n)$ is the optimal radius of the packing problem (CPP) with n circles and $d^\star(n)$ is the optimal minimal distance for the scattering of n points. Solving (PSP) one obtains the centers of n circles, which form an optimal solution of (CPP) on a slightly larger square. Indeed, one solves (CPP) on a square with edge-length $1 + d^\star(n)$ (see Figure 5.1). Note that a variation of the edge-length of the square in the packing problem does not alter the packing of an optimal combination of circles. Such a variation leads only to a scaling of a solution of (CPP).

FIGURE 5.1. Solutions for $n = 3$



(a) Problem (CPP)

(b) Problem (PSP)

Problem (PSP) is obviously equivalent to

$$\begin{array}{lll} & \max\ t & \\ t - \|x_i - x_j\|_2^2 \le 0 & & 1 \le i < j \le n \\ x_i \in U & & i = 1, \dots, n\,, \end{array} \qquad \text{(PP)}$$

which is just the formulation of the packing problem as an all-quadratic problem mentioned in Section 1.1. The optimal value $t^\star(n)$ of (PP) is equal to the squared optimal distance $d^\star(n)$ of (PSP).

According to the intention of this thesis we will consider the all-quadratic formulation (PP) of the circle packing problem (CPP) throughout this chapter. We say that $x^\star = (x_1^\star, \dots, x_n^\star)^T$ with $x_i^\star = (x_{i_1}^\star, x_{i_2}^\star)^T \in U$ is an optimal solution of Problem (PP) with optimal value $t^\star$, if there holds

$$t^\star(n) \ = \ t^\star \ = \ \min_{1 \le i < j \le n} \|x_i^\star - x_j^\star\|_2^2\,.$$

Any point $\bar{x} = (\bar{x}_1, \dots, \bar{x}_n)^T \in U^n$ satisfying

$$t^\star(n) - \min_{1 \le i < j \le n} \|\bar{x}_i - \bar{x}_j\|_2^2 \ \le \ \epsilon$$

will be called an ϵ-optimal solution, where $\epsilon > 0$ is some prespecified tolerance possibly depending on the number n of points.

As an interpretation for this problem, we can think of $x_1, \dots , x_n$ as the positions of "objects" which interfere with each other. The interference is inversely proportional to the minimum distance between the objects. Therefore, the solution of (PP) is an arrangement of the objects such that the interference is reduced to a minimum. For instance, $x_1, \dots , x_n$ may be positions of radio stations, which we want to place in such a way that the interferences between them are reduced to a minimum.

Problem (PP) has received a great deal of attention in the last years. In spite of its apparent simplicity, it turns out to be a quite difficult one. Papers about it can be divided into two categories. The first category contains papers in which proofs of optimality of packings for some values of n are given. Optimal solutions for $n \leq 9$ were already found in the sixties by geometric arguments. The cases $n = 2, \dots , 5$ are easy; the solution for $n = 6$ was given by Graham; the cases $n = 7, 8$ were solved in [SM65] and the case $n = 9$ in [SCH65]. Optimal solutions were geometrically derived also for bigger values of n. Optimal solutions for $n = 14, 16, 25, 36$ are proposed in [WEN83, WEN87A, WK87, WEN87B]. In [DGPW90] a computer proof for the cases $n = 10 - 13$ is suggested, while in [DGPWM91] the computer proof of optimality is extended to the cases $n = 14 - 20$. To the author's knowledge no proof of optimality for $n > 20$ has ever been given in the literature.

The second category includes papers in which improvements with respect to the best known solutions for $n > 20$ are presented – without giving any proof of optimality. Good packings for n up to 27 and for a few values greater than 27 are given in [GOL70]. In [MFP95] the formulation (PP) for the circle packing problem is employed and good packings for $n \leq 30$ are calculated using a stochastical approach.

It seems to be obvious that the following implication

$$n = k^2 \quad \Longrightarrow \quad d^\star(n) = \frac{1}{k-1}\,, \tag{5.1.2}$$

is true. However, Relation (5.1.2) is only fulfilled for $2 \leq k \leq 6$. Indeed, in [NO97], where good packings for $n \leq 50$ are given, a packing for 49 is presented with a bigger minimum pairwise distance than $\frac{1}{6}$. In [GL96] good packings for $n \leq 52$ and for a few other values greater than 52 are proposed. In particular, for $n = 21, 28, 34, 40, 43, 45, 47$ the presented results are better than those in [NO97].

For a short overview about the circle packing problem with respect to squares and to other related objects like circles, triangles or hemispheres we refer also to [STE98].

Since the optimal solutions of Problem (PP) for up to 20 points are reported to be known, we are interested in solutions of (PP) with $n > 20$. As mentioned before, the optimization approaches for all-quadratic problems discussed so far in this thesis are not able to solve Problem (PP) with a dimension higher than 42 and with more than $\binom{20}{2}$ concave quadratic constraints. With the rectangular branch-and-bound method by Al-Khayyal et al. [AKLV95] as well as with our simplicial approach (see Chapter 3) we were only able to solve Problem (PP) with less than 10 points. Therefore, we developed a new rectangular branch-and-bound approach, which is theoretically able to solve Problem (PP) in each dimension and which showed a good performance with up to 27 points.

Before formulating the algorithm in Section 5.3 we study in Section 5.2 some theoretical properties of optimal solutions of Problem (PP). We state the intuitive fact that there exists at least one optimal solution such that as many points as possible belong to the boundary of the unit square U. In Subsection 5.2.1 it is shown that there exists an optimal solution of Problem (PP) with a special behavior at each vertex of the unit square U. Another result describing the behavior of at least one optimal solution along each edge of U is discussed in Subsection 5.2.2. In Section 5.3 a rectangular branch-and-bound algorithm for solving (PP) is proposed. Even though we do not expatiate the details of our algorithm in this section, we are able to prove the convergence of this approach under some restrictions. In the following three Sections 5.4 - 5.6 we describe the details of our method. The calculation of the critical upper bounds is developed in Section 5.4. Exploiting the special structure of Problem (PP) we are able to derive in Section 5.5 a special splitting strategy for the relevant hyperrectangles $R \subset U^n$, which shows a better performance for this problem than the well-known bisection (for the definition of the bisection see, e.g., page 101). Using the theoretical results derived in Section 5.2 and an idea mentioned in [DGPWM91] we develop strategies for reducing the size of the relevant hyperrectangles in Section 5.6. These strategies enabled us to further improve the performance of our approach. In Section 5.7 we present preliminary computational results. In particular, we give approximately optimal solutions for $n = 21 - 24, 26, 27$. We finish this chapter with a discussion of some further improvements of the introduced method, which enable us to solve even larger problems. In particular, we present in Section 5.8 for $n = 32$ a solution, which constitutes an improvement of the best solution known so far. Apart from these improvements of our method and the numerical results for more than 27 points the presented results are also given in [LR98A, LR98B].

5.2. Theoretical Results

We start the treatment of (PP) by a theoretical examination of this problem. One would intuitively expect that as many as possible members $x_i^\star$ ($i \in \{1, \ldots, n\}$) of an optimal solution $x^\star \in U^n$ of Problem (PP) lie on or near to the boundary of U, since we try to maximize the minimum squared distance t between any two points. In the following two subsections we derive some properties, which have to be fulfilled by at least one optimal solution of Problem (PP). These properties corroborates the intuitive fact mentioned above and will later be useful in order to improve the numerical performance of our new rectangular branch-and-bound method introduced in Section 5.3.

5.2.1. Properties of an Optimal Solution at each Vertex. As the known optimal solution for the case $n = 6$ shows (see [SM65] or Figure 5.5), we cannot expect that each vertex of U belongs to an optimal solution of (PP). However we are able to derive the existence of an optimal solution with a special property at each vertex v of U. Either this vertex v is a member of the solution $x^\star = (x_1^\star, \ldots, x_n^\star)^T$ itself or there exist two points $x_i^\star$, $x_j^\star$ ($i, j \in \{1, \ldots, n\}$), which lie on the two boundary lines of U forming the vertex v and which have exactly the optimal distance $d^\star(n)$, i.e., $d^\star(n) = \|x_i^\star - x_j^\star\|_2$. This will be the result of Theorem 5.2.3 and the subsequent corollary.

In order to prove this theorem we need to show the existence of an optimal solution of (PP) with another special property. We need an optimal solution $x^\star = (x_1^\star, \ldots, x_n^\star)^T \in U^n$ such that each member $x_i^\star$ of $x^\star$ ($i \in \{1, \ldots, n\}$) belonging to the convex hull $[x_1^\star, \ldots, x_n^\star]$ of the points $x_1^\star, \ldots, x_n^\star$ belongs even to the boundary of U. The existence of such a solution is ensured by the next lemma. In the following corollary we prove, moreover, that there is an optimal solution of Problem (PP) such that the set $[x_1^\star, \ldots, x_n^\star]$ touches each boundary line of U.

LEMMA 5.2.1. *There exists an optimal solution* $(x_1^\star, \ldots, x_n^\star)^T \in \mathbb{R}^{2n}$ *of Problem (PP) with the property*

$$x_i^\star \in \partial([x_1^\star, \ldots, x_n^\star]) \implies x_i^\star \in \partial U = \partial([0,1]^2)\,, \tag{P1}$$

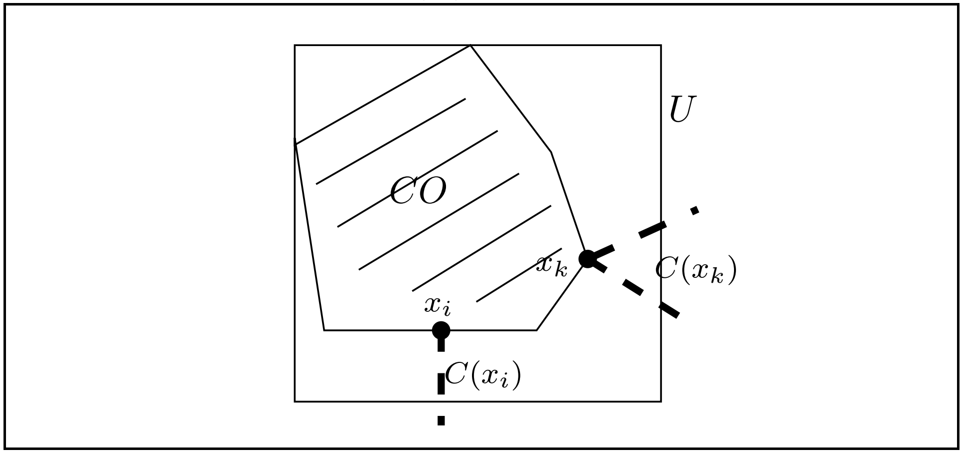
i.e., each member $x_i^\star$ of the optimal solution $x^\star$ belonging to the boundary of the convex hull of the points $x_1^\star, \dots, x_n^\star$ belongs to the boundary of the unit square U.

PROOF: Let $(x_1, \dots, x_n)^T \in \mathrm{I\!R}^{2n}$ be an optimal solution of (PP) with optimal value $t^\star(n)$. Denote by CO the convex hull of the points $x_1, \dots, x_n$, i.e., $CO = [x_1, \dots, x_n]$. If solution x does not have Property (P1), then there exists a member $x_i \in \partial CO$ satisfying $x_i \notin \partial U$. Consider the normal cone $C(x_i)$ of set CO at point x_i (see [ROC70] for the general definition of a normal cone), i.e.,

$$C(x_i) = \{y \in \mathrm{I\!R}^2 : y = x_i + d\,,\, d^T(x_j - x_i) \leq 0\,,\, j \in \{1, \dots, n\} \setminus \{i\}\}$$

(compare with Figure 5.2).

FIGURE 5.2. Normal cones



The set $C(x_i)$ has the following properties. These properties can be found in [ROC70]. For completeness we present a detailed proof.

(A). $C(x_i) \setminus \{x_i\} \neq \emptyset$

The convex hull CO is a polytope, which can be described by a finite set of facets. Let, for each $j \in I^i$, the set $F_j^i := \{y \in \mathrm{I\!R}^2 : (f_j^i)^T y = (f_j^i)^T x_i\}$ denote a facet of CO through the point x_i with the property $CO \subset \{y \in \mathrm{I\!R}^2 : (f_j^i)^T y \leq (f_j^i)^T x_i\}$, where I^i is an index set satisfying $|I^i| \leq 2$. Choose a vector $d \in \mathrm{I\!R}^2$, which is a convex combination of the normals f_j^i ($j \in I^i$), i.e., choose $\lambda \in \mathrm{I\!R}_+^{|I^i|}$ with $\sum_{i \in I^i} \lambda_i = 1$ and set

$$d := \sum_{j \in I^i} \lambda_j f_j^i \neq 0$$

Then, for each $j \in \{1,\dots,n\} \setminus \{i\}$, there holds

$$d^T(x_j - x_i) = \sum_{j \in I^i} \lambda_j \underbrace{((f_j^i)^T(\underbrace{x_j}_{\in CO} - x_i))}_{\leq 0} \leq 0\,.$$

Hence we obtain $x_i + d \in C(x_i) \setminus \{x_i\}$.

(B). $CO \cap C(x_i) \;=\; \{x_i\}$
Choose an arbitrary, but fixed $x \in CO \cap C(x_i)$. There exists a vector $\lambda \in \mathbb{R}_+^n$ with $\sum_{j=1}^n \lambda_j = 1$ and $x = \sum_{j=1}^n \lambda_j x_j$. Because of $x \in C(x_i)$ we obtain, for each $j \in \{1,\dots,n\} \setminus \{i\}$,

$$\begin{aligned} 0 \;&\geq\; (\underbrace{x - x_i}_{=d})^T(x_j - x_i) \\ &=\; \left(\sum_{j=1}^n \lambda_j (x_j - x_i)\right)^T (x_j - x_i) \;=\; \sum_{j=1}^n \lambda_j \underbrace{\|x_j - x_i\|_2^2}_{>0,\text{ if } i \neq j}\,. \end{aligned}$$

It follows that $\lambda_i = 1$ and thus $x = x_i$.

(C). $y \in C(x_i) \implies \|y - x_j\|_2^2 \geq t^\star(n)\;,\; j \in \{1,\dots,n\} \setminus \{i\}$
Choose an arbitrary, but fixed $y \in C(x_i)$. For each $j \in \{1,\dots,n\} \setminus \{i\}$, there holds

$$\begin{aligned} \|y - x_j\|_2^2 \;&=\; \|x_i - x_j + d\|_2^2 \\ &=\; \|x_i - x_j\|_2^2 + \underbrace{2d^T(x_i - x_j)}_{\geq 0} + \underbrace{\|d\|_2^2}_{\geq 0} \\ &\geq\; \|x_i - x_j\|_2^2 \;\geq\; t^\star(n)\,. \end{aligned}$$

Choosing a point $d \in \mathbb{R}^2$ with $0 \neq d \in C(x_i) - \{x_i\}$, which exists because of (A), we obtain that, for each $\lambda \in \mathbb{R}_+$, the point $x_i + \lambda d$ is an element of $C(x_i)$. Since by assumption x_i does not belong to the boundary of U, there holds

$$\bar{\lambda} \;:=\; \max\{\lambda \geq 0 : x_i + \lambda d \in U\} \;>\; 0\,.$$

Setting $x_i^\star := x_i + \bar{\lambda} d$ we obtain an element of the boundary of U. Taking Properties (B) and (C) of the normal cone $C(x_i)$ into account it follows that $(x_1,\dots,x_{i-1},x_i^\star,x_{i+1},\dots,x_n)^T$ is also an optimal solution of Problem (PP) and, moreover, $x_i^\star \in \partial[x_1,\dots,x_{i-1},x_i^\star,x_{i+1},\dots,x_n] \cap \partial U$. Repeating the argumentation presented in this proof we obtain a solution $(x_1^\star,\dots,x_n^\star)^T$ of (PP) satisfying Property (P1). ■

As mentioned before, in order to prove the existence of an optimal solution of Problem (PP) with the claimed special behavior at each vertex of U, we need a solution $x^\star$ of (PP) with Property (P1) and, additionally, satisfying that the convex hull of the members $x_1^\star, \dots, x_n^\star$ of $x^\star$ touches each boundary line of U. If an optimal solution of (PP) is given, it is easy to see that an altering of this solution – using the same ideas as in the previous proof – leads to an optimal solution of (PP) with the required attributes.

COROLLARY 5.2.2. *There exists an optimal solution $(x_1^\star, \dots, x_n^\star)^T \in \mathbb{R}^{2n}$ of Problem (PP) with Property (P1) and, additionally, with the property that, for each $i \in \{1, 2\}$ and $j \in \{0, 1\}$, there holds*

$$[x_1^\star, \dots, x_n^\star] \cap e_i^j \neq \emptyset \,, \tag{P2}$$

where $e_i^j = \{x \in U : x_i = j\}$ is a boundary line of the unit square U. This means that the convex hull $[x_1^\star, \dots, x_n^\star]$ of the set $\{x_1^\star, \dots, x_n^\star\}$ touches each edge of the unit square U.

PROOF: Let $(x_1, \dots, x_n)^T \in U^n$ be an optimal solution of (PP) with optimal value $t^\star(n)$ fulfilling Property (P1). If the convex hull of the set $\{x_1, \dots, x_n\}$ does not touch an edge e of U, then we choose one of the members x_i ($i \in \{1, \dots, n\}$), which are closest to e. Moving x_i towards this edge in the direction perpendicular to e we obtain a point $x_i^\star \in e$. This direction belongs to the normal cone of the set $[x_1, \dots, x_n]$ at point x_i. Hence, it follows by an analogous argumentation as in the proof of Lemma 5.2.1 that the minimum pairwise squared distance of $\bar{x} = (x_1, \dots, x_{i-1}, x_i^\star, x_{i+1}, \dots x_n)^T$ is still equal to $t^\star(n)$. Consequently, $\bar{x}$ is also an optimal solution of (PP) satisfying Property (P1) and, additionally, the convex hull of the members of $\bar{x}$ touches e. ■

Using optimal solutions of Problem (PP) satisfying Properties (P1) and (P2) we are now able to derive the existence of optimal solutions with a special behavior at each vertex of the unit square, as the following theorem shows.

THEOREM 5.2.3. *There exists an optimal solution $(x_1^\star, \dots, x_n^\star)^T \in \mathbb{R}^{2n}$ of Problem (PP) with optimal value $t^\star(n)$ such that, for each vertex v of the unit square U, i.e., $v = \binom{v_1}{v_2} \in \{\binom{0}{0}, \binom{0}{1}, \binom{1}{0}, \binom{1}{1}\}$, one and only one of the following statements is true*

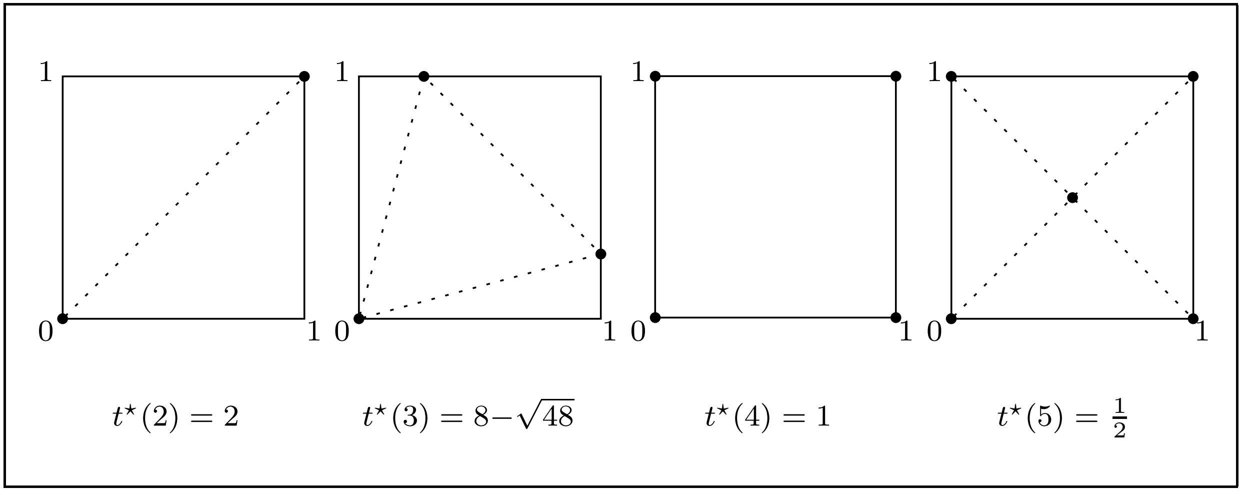
$$(i) \qquad \exists i \in \{1,\ldots,n\} \textit{ with } v = x_i^\star\ , \tag{P3a}$$

$$\begin{aligned} (ii) \qquad & \exists i,j \in \{1,\ldots,n\} \textit{ with } x_{i_1}^\star = v_1\ ,\ x_{j_2}^\star = v_2 \\ & \textit{and, for } l \in \{i{,}j\}\ ,\ \|v - x_l^\star\|_2^2 < t^\star(n)\ . \end{aligned} \tag{P3b}$$

This means that either the vertex v itself belongs to the optimal solution or there exist two members $x_i^\star$, $x_j^\star$ of this solution lying on the boundary lines of U forming the vertex v, which have a squared distance to v smaller than the optimal one.

PROOF: For $n \leq 5$ the known optimal solutions (see Figure 5.3) have Property (P3). Therefore, we can assume that there holds $n > 5$ and, in particular, $t^\star(n) < 1.0$.

FIGURE 5.3. Known solutions for $n = 2, \ldots, 5$



Let $(x_1, \ldots, x_n)^T \in U^n$ be an optimal solution of (PP) with optimal value $t^\star(n)$ satisfying Properties (P1) and (P2). Choose an arbitrary, but fixed vertex v of U and define (using $t := t^\star(n)$)

$$S^2(v,t) \ := \ \{y \in \mathbb{R}^2 : \|v - y\|_2^2 < t\}$$

and

$$\bar{S}^2(v,t) \ := \ S^2(v,t) \cap \{x_1, \ldots, x_n\}\ .$$

The set $\bar{S}^2(v,t)$ contains all members of $(x_1, \ldots, x_n)^T$, which have a squared distance smaller than t to the vertex v. Depending on the cardinality of the set $\bar{S}^2(v,t)$ we distinguish four cases.

Case 1: $|\bar{S}^2(v,t)| = 0$

For each $i \in \{1, \ldots, n\}$, there holds $\|v - x_i\|_2^2 \geq t$. Setting

$$x_1^\star \ := \ v$$

and, for $i \in \{2, \dots, n\}$,

$$x_i^\star \ := \ x_i$$

we obtain an optimal solution of Problem (PP), which fulfills (P3a) at vertex v, and not (P3b).

<u>Case 2</u>: $|\bar{S}^2(v,t)| = 1$
Without loss of generality assume that x_1 is the only element of $\bar{S}^2(v,t)$, i.e., $\bar{S}^2(v,t) = \{x_1\}$. For each $i \in \{2, \dots, n\}$, there holds $\|v - x_i\|_2^2 \geq t$. Using the same definition for $x^\star \in U^n$ as in the previous case we obtain again an optimal solution with the same properties as before.

<u>Case 3</u>: $|\bar{S}^2(v,t)| = 2$
We will show in this case that only (P3b) is true – not (P3a). We prove this for the vertex $v = \binom{0}{0}$. The argumentation for the other vertices is analogous. Let $\bar{S}^2(v,t)$ be given by $\{x_i, x_j\}$ with $i, j \in \{1, \dots, n\}$, $i \neq j$. Regarding the definition of $\bar{S}^2(v,t)$ it follows that there holds

$$v \notin \{x_i, x_j\} \,. \tag{5.2.1}$$

Hence Property (P3a) is not fulfilled. If x_i and x_j belong to the boundary of U, it is easy to see that (P3b) is satisfied. Indeed, since there holds $t < 1.0$ we know that x_i and x_j must lie on the boundary lines forming the vertex v. Moreover, with respect to the definition of $\bar{S}^2(v,t)$ and because of $\|x_i - x_j\|_2^2 \geq t$ they cannot both belong to the same edge of U.

We prove now that x_i and x_j must always belong to the boundary of U. Assume, by contradiction, that x_i does not belong to an edge of the unit square, i.e., $x_i \notin \partial U$. The optimal solution $(x_1, \dots, x_n)^T$ has Property (P1) such that there also holds

$$x_i \notin \partial CO$$

with $CO = [x_1, \dots, x_n]$. The set CO is a polytope and we know that v does not belong to this set. It follows that there exists a facet of CO which separates v and x_i. Each facet of CO is a line connecting two elements of $\{x_1, \dots, x_n\}$ belonging to the boundary of CO and hence – taking (P1) into account – belonging to the boundary of U. Since the point $(x_1, \dots, x_n)^T \in \mathbb{R}^{2n}$ fulfills by assumption Property (P2), there must exist two elements $x_k, x_l \in \{x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n\}$

satisfying

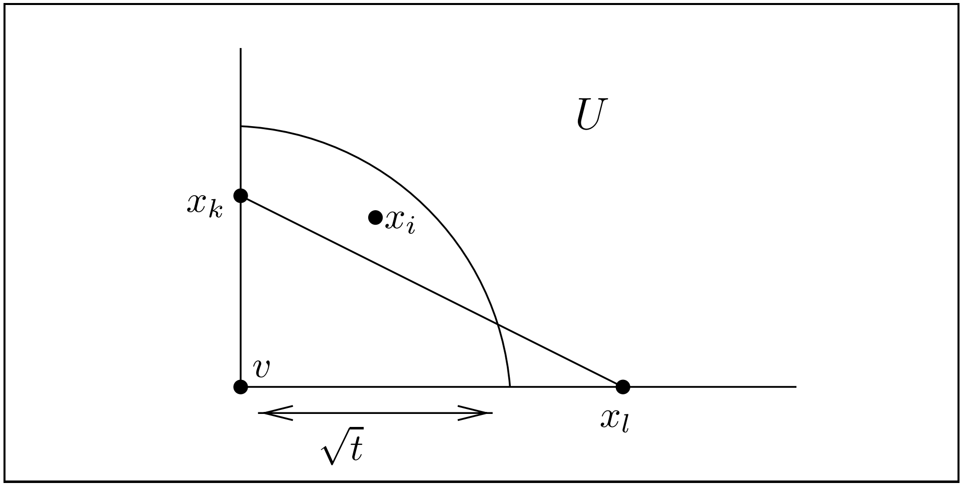
$$x_{k_1} = 0 \quad , \quad x_{k_2} > 0$$
$$x_{l_1} > 0 \quad , \quad x_{l_2} = 0$$

and

$$\frac{x_{i_1}}{x_{l_1}} + \frac{x_{i_2}}{x_{k_2}} \; > \; 1 \,, \tag{5.2.2}$$

i.e., x_i lies on the right-hand side of the facet of CO formed by the points x_k and x_l (compare with Figure 5.4).

FIGURE 5.4. Vertex situation



Because of $\|x_k - x_i\|_2^2 \geq t$ and $\|x_l - x_i\|_2^2 \geq t$ we obtain

$$x_{i_1}^2 + (x_{k_2} - x_{i_2})^2 \geq t \quad \text{and} \quad (x_{l_1} - x_{i_1})^2 + x_{i_2}^2 \geq t \,. \tag{5.2.3}$$

Moreover, in view of $x_i \in S^2(v,t)$ we know that

$$x_{i_1}^2 + x_{i_2}^2 \leq t \,. \tag{5.2.4}$$

Combining (5.2.3) and (5.2.4) it follows that

$$(x_{k_2} - x_{i_2})^2 \geq x_{i_2}^2 \qquad , \qquad (x_{l_1} - x_{i_1})^2 \geq x_{i_1}^2 \tag{5.2.5}$$

and hence

$$\tfrac{1}{2} x_{k_2} \geq x_{i_2} \qquad , \qquad \tfrac{1}{2} x_{l_1} \geq x_{i_1} \,. \tag{5.2.6}$$

From this relation we obtain

$$\frac{x_{i_1}}{x_{l_1}} + \frac{x_{i_2}}{x_{k_2}} \leq \frac{1}{2} + \frac{1}{2} = 1 \,,$$

contradicting (5.2.2) and, therefore, contradicting the assumption $x_i \notin \partial U$. Analogously, it can be proven that x_j must belong to ∂U.

Hence we have seen that in Case 3 the solution $(x_1, \ldots, x_n)^T$ must fulfill Property (P3b) itself and cannot satisfy (P3a).

Case 4: $|\bar{S}^2(v,t)| \geq 3$
It follows from the argumentation in the previous case that any point of the set $\bar{S}^2(v,t)$ is contained in the boundary of the unit square. Therefore, at least two points must belong to the same edge. However, this is not possible since they would have a squared distance smaller than t. Thus Case 4 cannot occur. ■

In the introduction of this subsection we claimed that there is an optimal solution of Problem (PP) satisfying (P3) and the additional property that there holds $\|x_i^\star - x_j^\star\|_2^2 = t^\star(n)$ in the case of (P3b). In order to strengthen Property (P3b) in this sense we will need some technical effort.

COROLLARY 5.2.4. *There exists an optimal solution* $(x_1^\star, \ldots, x_n^\star)^T$ *of Problem (PP) with optimal value* $t^\star(n)$ *satisfying Properties (P1)-(P3) and, additionally, fulfilling*

$$\|x_i^\star - x_j^\star\|_2^2 = t^\star(n) \tag{*}$$

in the case of (P3b).

PROOF: Let $(x_1, \ldots, x_n)^T \in \mathbb{R}^{2n}$ be an optimal solution of (PP) with Properties (P1)-(P3). Let further v be a vertex of the unit square U such that (P3b) is fulfilled, i.e.,

$$\bar{S}^2(v,t) \;=\; \{x_i, x_j\} \;\subset\; \partial U \;.$$

As in Case 3 of the proof of the previous Theorem 5.2.3 we assume that v is the origin. Furthermore, without loss of generality, we can assume that

$$x_{j_1} \;=\; x_{i_2} \;=\; 0\;.$$

If the squared distance between x_i and x_j is equal to $t^\star(n)$, then we know that Property (*) is fulfilled at vertex v. Otherwise there must hold

$$\|x_i - x_j\|_2^2 \;>\; t^\star(n) \;=:\; t\;.$$

In this case it is possible to move one of the points x_i or x_j towards the origin such that (*) is fulfilled and the distance to all other points x_l ($l \in \{1, \ldots, n\} \setminus \{i, j\}$) is still big enough. This will be proven in the sequel.

In order to derive this result we need at first a more technical statement. For any $l \in \{1, \dots, n\} \setminus \{i, j\}$, there holds

$$x_{l_1} \geq x_{i_1} \text{ or } x_{l_2} \geq x_{j_2} \ . \tag{5.2.7}$$

PROOF OF (5.2.7): Assume that (5.2.7) is not true, i.e.,

$$\exists l \in \{1, \dots, n\} \setminus \{i, j\} \text{ with } x_{l_1} < x_{i_1} \text{ and } x_{l_2} < x_{j_2} \ .$$

From $\|x_i - x_l\|_2^2 \geq t$ and $\|x_j - x_l\|_2^2 \geq t$ we obtain

$$(x_{i_1} - x_{l_1})^2 + x_{l_2}^2 \ \geq \ t \tag{5.2.8.a}$$

$$x_{l_1}^2 + (x_{j_2} - x_{l_2})^2 \ \geq \ t \tag{5.2.8.b}$$

and hence

$$t \ \leq \ x_{i_1}^2 \underbrace{-2x_{i_1}x_{l_1}}_{\leq -2x_{l_1}^2} + x_{l_1}^2 + x_{l_2}^2 \ \leq \ x_{i_1}^2 - x_{l_1}^2 + x_{l_2}^2 \tag{5.2.9.a}$$

$$t \ \leq \ x_{l_1}^2 + x_{j_2}^2 \underbrace{-2x_{j_2}x_{l_2}}_{\leq -2x_{l_2}^2} + x_{l_2}^2 \ \leq \ x_{l_1}^2 + x_{j_2}^2 - x_{l_2}^2 \ . \tag{5.2.9.b}$$

Adding (5.2.9.a) to (5.2.9.b) and using the fact that $\{x_i, x_j\} \subset S^2(\binom{0}{0}, t)$ it follows

$$2t \ \leq \ x_{i_1}^2 + x_{j_2}^2 \ < \ 2t \, ,$$

which is a contradiction. Therefore, (5.2.7) must be true. □

If there holds $x_{i_1} \leq x_{j_2}$, it is possible to move x_i towards the origin in order to satisfy (*). Indeed, set $x_i(\epsilon) := \binom{x_{i_1} - \epsilon}{0}$ for $\epsilon \geq 0$. It is provable that, for any $\epsilon \in (0, x_{i_1})$ and $l \in \{1, \dots, n\} \setminus \{i, j\}$, there holds

$$t \ \leq \ \|x_l - x_i(\epsilon)\|_2^2 = (x_{l_1} - x_{i_1} + \epsilon)^2 + x_{l_2}^2 \ . \tag{5.2.10}$$

This relation shows that it is possible to move x_i towards the origin – altering the first coordinate x_{i_1} – without decreasing too much the squared distance between the moved point $x_i(\epsilon)$ and x_l ($l \in \{1, \dots, n\} \setminus \{i, j\}$).

PROOF OF (5.2.10): Choose an arbitrary, but fixed index $l \in \{1, \dots, n\} \setminus \{i, j\}$. In order to show (5.2.10) we have to distinguish two cases.

Case 1: $x_{i_1} \leq x_{l_1}$

It follows immediately that

$$x_{l_1} - x_{i_1} + \epsilon \geq x_{l_1} - x_{i_1} \ \geq \ 0 \ .$$

This implies

$$(x_{l_1} - x_{i_1} + \epsilon)^2 + x_{l_2}^2 \geq (x_{l_1} - x_{i_1})^2 + x_{l_2}^2 = \|x_i - x_l\|_2^2 \geq t\,.$$

Case 2: $x_{i_1} > x_{l_1}$
From (5.2.7) we obtain that $x_{l_2} \geq x_{j_2}$. Therefore, we can conclude

$$\begin{aligned} t \;\leq\; \|x_l - x_j\|_2^2 \;=\; & x_{l_1}^2 + x_{j_2}^2 \underbrace{-2x_{l_2}x_{j_2}}_{\leq -2x_{j_2}^2} + x_{l_2}^2 \\ \leq\; & x_{l_1}^2 - \underbrace{x_{j_2}^2}_{\geq x_{i_1}^2} + x_{l_2}^2 \\ \leq\; & \underbrace{x_{l_1}^2 - x_{i_1}^2}_{\leq 0} + x_{l_2}^2 \;\leq\; x_{l_2}^2\,. \end{aligned}$$

It follows

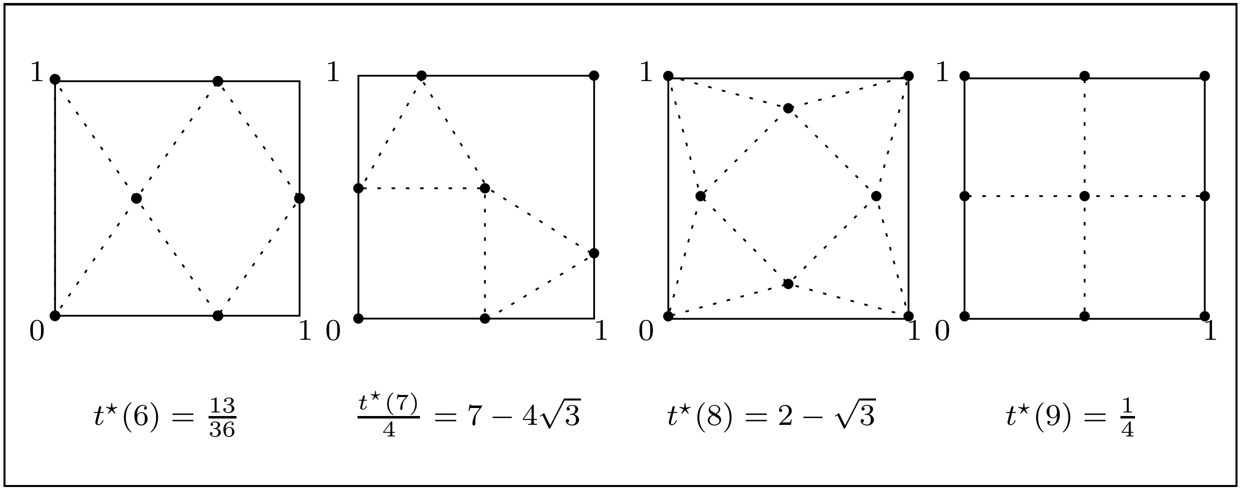
$$(x_{l_1} - x_{i_1} + \epsilon)^2 + x_{l_2}^2 \;\geq\; x_{l_2}^2 \;\geq\; t\,. \qquad \square$$

If we choose $\bar{\epsilon} \in (0, x_{i_1})$ satisfying $\|x_i(\bar{\epsilon}) - x_j\|_2^2 = t$, we obtain from (5.2.10) that $(x_1, \dots, x_{i-1}, x_i(\bar{\epsilon}), x_{i+1}, \dots, x_n)^T$ is also an optimal solution of Problem (PP) fulfilling (P1)-(P3) and, additionally, fulfilling (*) at vertex v.

In order to move x_i towards the origin we assumed that there holds $x_{i_1} \leq x_{j_2}$. If this is not true, it is possible to move x_j – altering the second coordinate x_{j_2} – towards the origin in an analogous way, such that in each case we obtain an optimal solution of (PP) with all required attributes at vertex v.

For $n \leq 9$ it follows from the known solutions (see Figure 5.3 and Figure 5.5) that solutions of Problem (PP) exist, which fulfill the stronger Property (*) at each

FIGURE 5.5. Known solutions for $n = 6, \dots, 9$



$t^\star(6) = \frac{13}{36}$ $\quad \frac{t^\star(7)}{4} = 7 - 4\sqrt{3}$ $\quad t^\star(8) = 2 - \sqrt{3}$ $\quad t^\star(9) = \frac{1}{4}$

vertex v of U with Property (P3b). Hence, we have to verify the existence of such points only for $n \geq 10$. This implies that there holds $t = t^\star(n) \leq 0.25$ and thus

$$\bar{S}^2(v,t) \cap \bar{S}^2(w,t) \; = \; \emptyset \tag{5.2.11}$$

for two different vertices v and w of the unit square. Therefore, we can apply the argumentation used above in order to guarantee that (*) is fulfilled at each vertex of U, which has Property (P3b). Property (5.2.11) implies that the points we would like to move in order to enforce (*) must be different for different vertices of U.

■

In the following we denote by Property (P3) this stronger version.

5.2.2. Properties of an Optimal Solution along each Edge. If we consider the behavior of an optimal solution on the boundary lines of the unit square U, we cannot expect that two consecutive points have exactly the optimal distance (compare, e.g., the optimal solutions for $n = 6$ or $n = 7$, see [SM65] or Figure 5.5). We are only able to verify the existence of an optimal solution with the property that this distance is smaller than two times the optimal one. This is the result of the following theorem. At first, however, one additional lemma is needed in order to establish this statement.

LEMMA 5.2.5. *Let* $(x_1, \ldots, x_n)^T \in \mathrm{I\!R}^{2n}$ *be an optimal solution of Problem (PP) with optimal value* $t^\star(n)$. *Assume further that there exist indices* $i, j \in \{1, \ldots, n\}$ *and* $l \in \{1, 2\}$ *satisfying*

$$\|x_i - x_j\|_2^2 \; \geq \; 4t^\star(n) \quad \textit{and} \quad x_{i_l} \; = \; x_{j_l} \in \{0, 1\}\,, \tag{5.2.12}$$

and, moreover, that there does not exist an index $k \in \{1, \ldots, n\} \setminus \{i, j\}$ *with* $x_k \in [x_i, x_j]$. *Then there holds*

$$\mathit{int}U \cap \{x_1, \ldots, x_n\} \; \neq \; \emptyset\,. \tag{5.2.13}$$

This means, if two consecutive points x_i *and* x_j *lying on the same boundary line of the unit square* U *have a distance not smaller than two times the optimal distance* $\sqrt{t^\star(n)}$, *then there exists a member* x_m *of this optimal solution belonging to the interior of* U.

PROOF: The points x_i and x_j belong by Assumption (5.2.12) to the same edge of U. Hence there holds

$$\|x_i - x_j\|_2^2 \leq 1$$

and consequently regarding the left part of (5.2.12) we know $t^\star(n) \leq 0.25$.

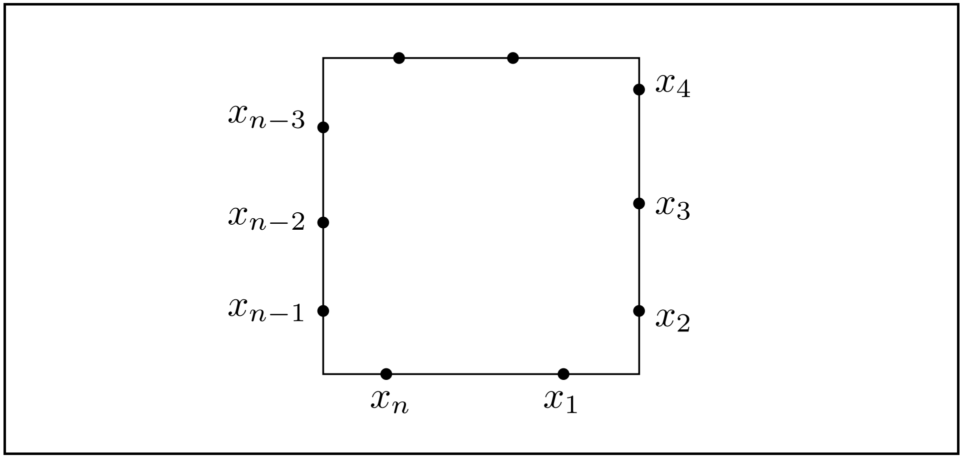
We prove Relation (5.2.13) by contradiction. Assume that there does not exist a member of the optimal solution belonging to the interior of U. With respect to the value of $t^\star(n)$ we distinguish again two cases.

Case 1: $t^\star(n) = 0.25$
From the first part of (5.2.12) it follows that $\|x_i - x_j\|_2^2 = 1$ and thus these points are vertices of U. It can be seen that in this situation n cannot be greater than 7. Indeed, it is not possible to place more points on the boundary lines of U such that the squared distance is not smaller than 0.25. However, for $n \leq 7$ solutions with larger minimum distances are known [SM65] (see also Figures 5.3 and 5.5). Consequently, the feasible point $(x_1, \dots, x_n)^T$ is not optimal for (PP), contradicting the assumption.

Case 2: $t := t^\star(n) < 0.25$
In this case it is possible to explicitly construct a point $\bar{x} = (\bar{x}_1, \dots, \bar{x}_n)^T$ with a bigger minimum squared distance than t.

Assume, without loss of generality, that there holds $i = n$ and $j = 1$ and that the members x_i $(i = 1, \dots, n)$ are numbered in such a way that x_{i+1} is a direct neighbor of x_i (compare with Figure 5.6). Denote, for $i \in \{1, \dots, n-1\}$, by $d_{i,i+1}$ the 1-norm distance between the two consecutive points x_i and x_{i+1}, i.e.,

FIGURE 5.6. Numbering



$$d_{i,i+1} \;:=\; \|x_i - x_{i+1}\|_1 \;\geq\; \|x_i - x_{i+1}\|_2 \;\geq\; \sqrt{t}\,. \tag{5.2.14.a}$$

Denote, furthermore, by $d_{n,1}$ the according distance between x_n and x_1, i.e.,

$$d_{n,1} \;:=\; \|x_n - x_1\|_1 \;\geq\; \|x_n - x_1\|_2 \;\geq\; 2\sqrt{t}\,. \tag{5.2.14.b}$$

Since the total length of the boundary lines of U is 4 it follows that there holds

$$\sum_{i=1}^{n-1} d_{i,i+1} + d_{n,1} \; = \; 4 \, .$$

Therefore, from (5.2.14.a) and (5.2.14.b) we obtain

$$(n+1)\sqrt{t} \; \leq \; 4 \, . \tag{5.2.15}$$

Set

$$\delta \; := \; \frac{4}{n+0.5} - \sqrt{t} \, .$$

From Relation (5.2.15) we see that

$$0 \; < \; \delta$$

and because of $n \geq 8$ (compare with Case 1) we know that

$$\delta \; < 0.5 - \sqrt{t} \, .$$

In the sequel we construct now a solution $\bar{x} = (\bar{x}_1, \ldots, \bar{x}_n)^T$ of Problem (PP) with a minimum pairwise distance of $\sqrt{t} + \delta$. In order to do this we will place one point at the center of U and the remaining $(n-1)$ points will be placed at the boundary of the unit square. Let us first interpret the edges of U as one connected line, i.e., as the interval $[0,4]$, where each integer in this interval coincides with a vertex of U. 0 and 4 coincide with the same vertex, namely the origin. We construct now a sequence of $(n-1)$ real numbers lying inside this interval in such a way that two successive numbers have a distance equal to $\sqrt{t}+\delta$, if no integer lies between them, and a distance equal to $\sqrt{2}(\sqrt{t}+\delta)$ otherwise. This is sufficient in order to obtain a solution of Problem (PP) with a minimum distance in the Euclidean norm not smaller than $\sqrt{t}+\delta$, as required.

The needed sequence of real numbers is defined as follows

$$\tilde{x}_1 \; := \; 0.0$$

and, for $i \in \{2, \ldots, n-1\}$,

$$\tilde{x}_i \; := \; \begin{cases} \tilde{x}_{i-1} + (\sqrt{t}+\delta) & \text{, if } \lfloor \tilde{x}_{i-1} \rfloor = \lfloor \tilde{x}_{i-1} + (\sqrt{t}+\delta) \rfloor \\ \tilde{x}_{i-1} + \sqrt{2}(\sqrt{t}+\delta) & \text{, otherwise.} \end{cases}$$

In order to verify that this sequence has the claimed properties it is sufficient to show that $\tilde{x}_{n-1}$ lies in the interval $[0,4]$ and, additionally, that $\tilde{x}_{n-1}$ has a distance not smaller than $\sqrt{t}+\delta$ to 4 (and hence to the origin). If we look for the biggest quantity $k \in \mathbb{N}$ of numbers constructed by the foregoing prescription, which belong to the interval $[0, 4 - (\sqrt{t} + \delta)]$, then we have to solve the problem

$$\begin{array}{c} \max \quad k \\ (k-4)(\sqrt{t}+\delta) + 3\sqrt{2}(\sqrt{t}+\delta) \ \leq \ 4 - (\sqrt{t}+\delta) \\ k \in \mathbb{N} \qquad\qquad\qquad\qquad\qquad . \end{array}$$

Note that there are three integers inside the interval $[0, 4]$. Hence there are at most three pairs $(\tilde{x}_i, \tilde{x}_{i+1})$ ($i \in \{1, \ldots, k-1\}$) with a distance of $\sqrt{2}(\sqrt{t} + \delta)$. The remaining $(k - 4)$ distances are by construction equal to $(\sqrt{t} + \delta)$. From the properties of δ it is easy to verify that we obtain $k^\star = n - 1$, where $k^\star$ is the optimal solution of the previous problem.

With this sequence of real numbers we are now able to construct the required feasible point $\bar{x} \in U^n$ with the claimed minimal pairwise distance. Set, for $i = 1, \ldots, n - 1$,

$$\bar{x}_i := \begin{cases} (\tilde{x}_i, 0.0)^T & \text{, if } 0.0 \leq \tilde{x}_i \leq 1.0 \\ (1.0, \tilde{x}_i - 1.0)^T & \text{, if } 1.0 < \tilde{x}_i \leq 2.0 \\ (1.0 - (\tilde{x}_i - 2.0), 1.0)^T & \text{, if } 2.0 < \tilde{x}_i \leq 3.0 \\ (0.0, 1.0 - (\tilde{x}_i - 3.0))^T & \text{, otherwise.} \end{cases}$$

Adding $\bar{x}_n := (0.5, 0.5)^T$ we obtain obviously a feasible point for (PP). Straightforward calculation shows that this point has a larger minimum squared distance than $(x_1, \ldots, x_n)^T$. This contradicts the optimality of $(x_1, \ldots, x_n)^T$ and completes the proof. ■

If two consecutive points of an optimal solution of Problem (PP) lying on the same edge of U have a distance bigger than or equal to two times the optimal one, the previous lemma guarantees that there always exists a member of this optimal solution belonging to the interior of the unit square U. In the proof of the next theorem we show that it is possible to move one of these interior points to the boundary of U without decreasing the minimum distance. This leads to the existence of an optimal solution of Problem (PP) with the claimed property at each boundary line of the unit square U.

THEOREM 5.2.6. *There exists an optimal solution $(x_1^\star, \dots, x_n^\star)^T$ of Problem (PP) with the following property:*

If there exist indices $i, j \in \{1, \dots, n\}$ with $i \neq j$ and an index $l \in \{1, 2\}$ with $x_{i_l}^\star = x_{j_l}^\star \in \{0, 1\}$ and, furthermore, there does not exist an index $k \in \{1, \dots, n\} \setminus \{i, j\}$ with $x_k^\star \in [x_i^\star, x_j^\star]$, then there holds

$$\|x_i^\star - x_j^\star\|_2^2 \; < \; 4t^\star(n)\,. \tag{P4}$$

I.e., two consecutive members of $(x_1^\star, \dots, x_n^\star)^T$ belonging to the same edge of the unit square have a distance smaller than two times the optimal one.

PROOF: Let $(x_1, \dots, x_n)^T \in \mathbb{R}^{2n}$ be an optimal solution of Problem (PP) with optimal value $t := t^\star(n)$. Assume that there exist two consecutive members of this optimal solution with a distance not smaller than two times the optimal one. This means that there exist indices $i, j \in \{1, \dots, n\}$, $i \neq j$ and an index $l \in \{1, 2\}$ satisfying

$$\|x_i - x_j\|_2^2 \; \geq \; 4t \tag{5.2.16.a}$$

and

$$x_{i_l} \; = \; x_{j_l} \in \{0, 1\}\,, \tag{5.2.16.b}$$

and there does not exist an index $k \in \{1, \dots, n\} \setminus \{i, j\}$ such that x_k belongs to $[x_i, x_j]$. As long as there are two consecutive points with Properties (5.2.16.a) and (5.2.16.b), Lemma 5.2.5 yields that there exists a member of $(x_1, \dots, x_n)^T$ located within the interior of U. In order to prove the existence of an optimal solution with Property (P4) it is hence sufficient to show that we are able to move one of these interior points to the boundary of U without decreasing the minimum distance.

Without loss of generality, we assume $i = 1$, $j = 2$, $l = 2$, $x_{1_2} = x_{2_2} = 0.0$ and $x_{1_1} < x_{2_1}$, i.e., x_1 and x_2 lie on the edge $e = \{y \in \mathbb{R}^2 : 0.0 \leq y_1 \leq 1.0,$ $y_2 = 0.0\}$ and there holds $x_{2_1} \geq x_{1_1} + 2\sqrt{t}$.

Denote by

$$A \; := \; \{x_i : i \in \{3, \dots, n\}, x_{i_1} \leq x_{1_1} \text{ or } x_{i_1} \geq x_{2_1}\}$$

the set of all members of $(x_1, \dots, x_n)^T$ different from x_1 and x_2 with the property that the first coordinate does not belong to the open interval (x_{1_1}, x_{2_1}). It is easy to verify that, for each $\lambda \in [x_{1_1} + \sqrt{t}, x_{2_1} - \sqrt{t}]$ and $x \in A$, there holds

$$\|x - \tbinom{\lambda}{0}\|_2^2 \; \geq \; t\,. \tag{5.2.17}$$

Depending on the structure of set A we distinguish two cases.

Case 1: $\{x_3, \dots, x_n\} \setminus A = \emptyset$
Set

$$x^\star_{3_1} \; := \; \frac{x_{1_1} + x_{2_1}}{2} \quad \text{and} \quad x_{3_2} \; := \; 0.0 \; .$$

From (5.2.17) we obtain, for each $k \in \{4, \dots, n\}$,

$$\|x^\star_3 - x_k\|_2^2 \; \geq \; t \; .$$

Moreover, for $i \in \{1, 2\}$, there holds also $\|x^\star_3 - x_i\|_2^2 \geq t$. Therefore, the point $(x_1, x_2, x^\star_3, x_4, \dots, x_n)^T$ is another optimal solution of (PP) with the property that the number of members belonging to the boundary of U is increased by one – in comparison with $(x_1, \dots, x_n)^T$.

Case 2: $\{x_3, \dots, x_n\} \setminus A \neq \emptyset$
Choose an index $l \in \{3, \dots, n\}$ such that there holds

$$x_{l_2} \; = \; \min\{y_2 \,|\, y \in \{x_3, \dots, x_n\} \setminus A\} \; . \tag{5.2.18}$$

Construct a new point $x^\star_l \in \mathbb{R}^2$ belonging to $[x_1, x_2]$ according to the following rule

$$x^\star_{l_1} := \begin{cases} x_{l_1} & \text{, if } x_{1_1} + \sqrt{t} \leq x_{l_1} \leq x_{2_1} - \sqrt{t} \\ x_{1_1} + \sqrt{t} & \text{, if } x_{l_1} < x_{1_1} + \sqrt{t} \\ x_{2_1} - \sqrt{t} & \text{, otherwise} \end{cases} \qquad \text{and} \;\; x^\star_{l_2} := 0.0 \; .$$

For $i \in \{1, 2\}$ there holds obviously $\|x^\star_l - x_i\|_2^2 \geq t$. In order to finish the proof we have to show that the point $x^\star_l$ has a squared distance not smaller than t to any member of $(x_1, \dots, x_n)^T$ belonging to the set

$$\bar{A} \; := \; \{x_3, \dots, x_n\} \setminus (A \cup \{x_l\}) \; .$$

Choose an arbitrary, but fixed element x_k of $\bar{A}$. Depending on the definition of $x^\star_{l_1}$ it is necessary to distinguish three subcases.

Case 2.1: $x^\star_{l_1} = x_{l_1}$

From (5.2.18) we know that $x_{k_2} \geq x_{l_2}$. It follows

$$\begin{aligned} \|x_k - x^\star_l\|_2^2 \; &= \; (x_{k_1} - x_{l_1})^2 + x_{k_2}^2 \\ &\geq \; (x_{k_1} - x_{l_1})^2 + (x_{k_2} - x_{l_2})^2 \\ &= \; \|x_k - x_l\|_2^2 \; \geq \; t \; . \end{aligned}$$

<u>Case 2.2</u>: $x^\star_{l_1} = x_{1_1} + \sqrt{t}$

The following assertions are true

$$x_{l_2} \leq x_{k_2} \text{ (compare with Relation (5.2.18))} \tag{5.2.19.a}$$

$$x_{l_1} - x_{1_1} < \sqrt{t} \text{ (definition of } x^\star_{l_1}\text{)} \tag{5.2.19.b}$$

$$x_{1_1} \leq x_{l_1} \text{ (since } x_l \notin A\text{)} \tag{5.2.19.c}$$

$$t \leq \|x_k - x_l\|_2^2 = (x_{k_1} - x_{l_1})^2 + (x_{k_2} - x_{l_2})^2 \tag{5.2.19.d}$$

$$t \leq \|x_l - x_1\|_2^2 = (x_{1_1} - x_{l_1})^2 + x_{l_2}^2\ . \tag{5.2.19.e}$$

Using these statements we can conclude for the squared distance between $x^\star_l$ and x_k

$$\begin{aligned}
\|x_k - x^\star_l\|_2^2 &= (x_{k_1} - x_{1_1} - \sqrt{t})^2 + x_{k_2}^2\\
&= [(x_{k_1} - x_{l_1}) + x_{l_1} - x_{1_1} - \sqrt{t}]^2 + [(x_{k_2} - x_{l_2}) + x_{l_2}]^2\\
&= (x_{k_1} - x_{l_1})^2 + 2(x_{k_1} - x_{l_1})(x_{l_1} - x_{1_1} - \sqrt{t})\\
&\qquad + (x_{l_1} - x_{1_1})^2 - 2\sqrt{t}(x_{l_1} - x_{1_1}) + t\\
&\qquad + (x_{k_2} - x_{l_2})^2 + 2x_{l_2}\underbrace{(x_{k_2} - x_{l_2})}_{\geq 0,\ (5.2.19.a)} + x_{l_2}^2\\
&\geq (x_{k_1} - x_{l_1})^2 + (x_{k_2} - x_{l_2})^2 + \underbrace{\|x_l - x_1\|_2^2}_{\geq t,\ (5.2.19.e)} + t\\
&\quad + 2(x_{k_1} - x_{l_1})(x_{l_1} - x_{1_1} - \sqrt{t}) + 2(x_{1_1} - x_{l_1})\sqrt{t}\\
&\geq 2t + (x_{k_1} - x_{l_1})^2 + (x_{k_2} - x_{l_2})^2\\
&\qquad + 2(x_{k_1} - x_{l_1})(x_{l_1} - x_{1_1} - \sqrt{t}) + 2(x_{1_1} - x_{l_1})\sqrt{t}\\
&=: C
\end{aligned}$$

We need that C is not smaller than t. In order to prove this we have to distinguish two further subcases.

<u>Case 2.2.1</u>: $x_{k_1} - x_{l_1} \leq \sqrt{t}$

In this situation we obtain

$$\begin{aligned}
C &= 2t + \|x_k - x_l\|_2^2 + 2[\underbrace{(x_{k_1} - x_{l_1})}_{\leq\sqrt{t}}\underbrace{(x_{l_1} - x_{1_1} - \sqrt{t})}_{\leq 0,\ (5.2.19.b)} + (x_{1_1} - x_{l_1})\sqrt{t}]\\
&\geq 2t + \|x_k - x_l\|_2^2 + 2\sqrt{t}\underbrace{[x_{l_1} - x_{1_1} - \sqrt{t} + x_{1_1} - x_{l_1}]}_{=-\sqrt{t}}\\
&= \|x_k - x_l\|_2^2 \ \geq\ t \quad \text{(see (5.2.19.d)).}
\end{aligned}$$

Case 2.2.2: $x_{k_1} - x_{l_1} > \sqrt{t}$

It follows

$$\begin{aligned} C &= 2t + (x_{k_1} - x_{l_1})^2 + \underbrace{(x_{k_2} - x_{l_2})^2}_{\geq 0} \\ &\qquad + 2[(x_{k_1} - x_{l_1})(x_{l_1} - x_{1_1} - \sqrt{t}) + \underbrace{(x_{1_1} - x_{l_1})}_{\leq 0,\ (5.2.19.c)} \quad \underbrace{\sqrt{t}}_{<(x_{k_1} - x_{l_1})} \quad] \\ &\geq 2t + (x_{k_1} - x_{l_1})^2 - 2\sqrt{t}(x_{k_1} - x_{l_1}) \\ &= \underbrace{(x_{k_1} - x_{l_1} - \sqrt{t})^2}_{\geq 0} + t \ \geq \ t\,. \end{aligned}$$

Hence we obtain in Case 2.2 that $\|x_k - x_l^\star\|_2^2 \geq t$.

Case 2.3: $x_{l_1}^\star = x_{2_1} - \sqrt{t}$

By analogous calculations as in Case 2.2 it is possible to conclude

$$\|x_k - x_l^\star\|_2^2 \geq t\,.$$

We showed that, for each index $k \in \{1, \ldots, n\} \setminus \{l\}$, there holds

$$\|x_l^\star - x_k\|_2^2 \ \geq \ t\,.$$

Therefore, the point $(x_1, \ldots, x_{l-1}, x_l^\star, x_{l+1}, \ldots, x_n)^T$ is also an optimal solution of Problem (PP) with the same additional property as the new solution constructed in Case 1. ■

The results of Theorem 5.2.3 in connection with Corollary 5.2.4 and the result of Theorem 5.2.6 are independent from each other. Combining both we know that there exists an optimal solution $(x_1^\star, \ldots, x_n^\star)^T$ of Problem (PP) fulfilling Properties (P1)-(P4). In particular, as we will see in Section 5.5, Property (P4) and the strong version of Property (P3) give us a powerful tool in order to reduce the dimension of subproblems in a rectangular branch-and-bound algorithm.

5.3. The Algorithm

After the derivation of the theoretical results in the previous section giving us more insight into the structure of possible solutions of Problem (PP) we present now an algorithm for solving (PP). As in the solution approaches for (QP) developed in Chapters 3 and 4 we use a branch-and-bound scheme (see also Subsection 1.2.2). In Chapter 3 we saw that the use of simplices as subdivision sets can lead to a faster solution approach than the application of hyperrectangles, in particular, if we try to solve all-quadratic problems with a large number of constraints. Even though Problem (PP) has – in comparison with the dimension – a large number of constraints we prefer in the following algorithm hyperrectangles as subdivision sets. Using this type of sets the required initial hyperrectangle (see again Subsection 1.2.2) is immediately given by U^n. Moreover, the strategies, which we will develop in subsequent sections in order to improve the numerical performance of our solution scheme for (PP), need hyperrectangles.

In the present section we describe the basic algorithm without expatiating the details. These are discussed in the following sections. The presented algorithm guarantees to detect for a prespecified tolerance $\epsilon > 0$ an ϵ-optimal solution for the point scattering problem in finite time. Some preliminary notes about the convergence of our approach are given at the end of this section.

Denote by $f : \mathbb{R}^{2n} \to \mathbb{R}$

$$f(x) \ := \ \min_{1 \leq i < j \leq n} \|x_i - x_j\|_2^2$$

the minimum pairwise squared distance of the members $x_i = (x_{i_1}, x_{i_2})^T$ ($i \in \{1, \ldots, n\}$) of a $2n$-dimensional point $x = (x_1, \ldots, x_n)^T \in \mathbb{R}^{2n}$. Using this notation Problem (PP) can be written as

$$\begin{array}{c} \max f(x) \\ x \in U^n \; . \end{array} \tag{PP}$$

Assume that a point $\bar{x} \in U^n$ is known with $f(\bar{x}) > 0$. We can generate such a point $\bar{x}$ by using a local optimizer alone or in combination with a stochastical approach like a multi-start algorithm (see, e.g., [BR95] for an introduction to stochastic methods for global optimization). However, it is not necessary that $\bar{x}$ is a local optimal point for Problem (PP). Therefore, it is also sufficient to simply construct $\bar{x} \in U^n$ geometrically.

Assume further that an upper bound $\bar{\mu}$ for the optimal value $t^\star(n)$ of Problem (PP) is given. If the optimal value $t^\star(n-1)$ for Problem (PP) with $n-1$ points or an upper bound for $t^\star(n-1)$ is known, we can choose this value for $\bar{\mu}$. Otherwise it is possible to set $\bar{\mu} := 2.0$ since, for any $x \in U^n$, there holds $0 \leq f(x) \leq 2$.

Similar to Algorithm 3.1 and Algorithm 4.1 the formulation of the algorithm for Problem (PP) follows nearly the guidelines of a basic branch-and-bound scheme given in [HPT95, Algorithm 3.5]. Note that the following algorithm has special adaptations to Problem (PP). In particular, we do not insist that the union of all partition sets forms the full set (see (5.3.1) below). Since the details of the following approach will be expatiated later in this chapter, we use another type of description than before.

ALGORITHM 5.1 (***Rectangular Branch-and-Bound Algorithm for (PP)***).

Initialization

Choose a real number $\epsilon \geq 0$.
Set, for $i \in \{1, \dots, n\}$, $R_i^0 \leftarrow [0,1] \times [0,1] =: [l_{i_1}^0, L_{i_1}^0] \times [l_{i_2}^0, L_{i_2}^0]$,
$R^0 \leftarrow R_1^0 \times \ldots \times R_n^0 \subset \mathbb{R}^{2n}$, $\mathcal{R}^0 \leftarrow \{R^0\}$,
$x^0 \leftarrow \bar{x}$, $\eta^0 \leftarrow f(x^0)$, $Q \leftarrow \{x^0\}$, $\mu^0 \leftarrow \bar{\mu}$, $\mu_{R^0} \leftarrow \mu^0$, $k \leftarrow 0$.

Loop

Step I: (*Stopping criterion*)
If there holds $\mu^k - \eta^k \leq \epsilon$, then **STOP**.
$x^\star := x^k$ is an ϵ-optimal solution of Problem (PP), i.e.,
$t^\star(n) - \eta^k = t^\star(n) - f(x^k) \leq \epsilon$.

Step II:
Choose the smallest index $j \in \{1, \dots, n\}$ satisfying

$$\max\{L_{j_1}^k - l_{j_1}^k, L_{j_2}^k - l_{j_2}^k\} = \max\left\{\max\{L_{i_1}^k - l_{i_1}^k, L_{i_2}^k - l_{i_2}^k\}, i = 1, \dots, n\right\} .$$

Step III: (**Subdivision strategies**)
Construct $l \in \mathbb{N}$ two-dimensional rectangles $R_j^{k_1}, \dots, R_j^{k_l}$ with equal size fulfilling, for each $i = 1, \dots, l$,

$$R_j^{k_i} \subset R_j^k$$

and, for each $1 \leq i < p \leq l$,

$$\mathrm{int} R_j^{k_i} \cap \mathrm{int} R_j^{k_p} = \emptyset .$$

Step IV: (**Size reduction strategies**)
If possible, reduce, for each $i \in \{1, \dots, l\}$, the size of the hyperrectangles

$$R^{k_i} = R_1^k \times \ldots \times R_{j-1}^k \times R_j^{k_i} \times R_{j+1}^k \times \ldots \times R_n^k ,$$

i.e., construct hyperrectangles $\bar{R}^{k_i} \subset R^{k_i}$.

```
Step V: (Upper bounds)
    For $p = 1$ To $l$ Do
        If $\bar{R}^{k_p} = \emptyset$ Then
            $\mu_{\bar{R}^{k_p}} \leftarrow -\infty$
        Else
            Construct an upper bound $\mu_{\bar{R}^{k_p}}$ for the optimization problem

                $\max\ t$
                $t - \|x_i - x_j\|_2^2 \leq 0 \qquad 1 \leq i < j \leq n$
                $(x_1, \ldots, x_n)^T \in \bar{R}^{k_p}$                          (SP)
                $\eta^k \leq t \leq \mu^k$ .

            Use each point $y \in \bar{R}^{k_p}$ found during the calculation of $\mu_{\bar{R}^{k_p}}$ in order
            to update the lower bound, i.e.,

                $\eta^k \leftarrow \max\{\eta^k, f(y)\}$
                $Q \leftarrow Q \cup \{y\}$ .

        EndIf
    EndFor
Step VI:
    Adjust the set $\mathcal{R}^k$ of relevant subdivision sets by setting

        $\mathcal{R}^k \leftarrow (\mathcal{R}^k \setminus R^k) \cup \{\bar{R}^{k_p} : p \in \{1, \ldots, l\}$ with $\mu_{\bar{R}^{k_p}} \geq \eta^k\}$ .

Step VII: (Pruning rule)
    $\mathcal{R}^{k+1} \leftarrow \{R \in \mathcal{R}^k : \mu_R \geq \eta^k\}$
Step VIII:
    Update the lower and the upper bound by setting

        $\eta^{k+1} \leftarrow \eta^k$
        $\mu^{k+1} \leftarrow \begin{cases} \max\{\mu_R : R \in \mathcal{R}^{k+1}\} & \text{, if } \mathcal{R}^{k+1} \neq \emptyset \\ \eta^{k+1} & \text{, otherwise} \end{cases}$ .

    Choose a new node $R^{k+1}$ of the partition tree satisfying $\mu_{R^{k+1}} = \mu^{k+1}$.
    Select a point $x^{k+1} \in Q$ with $f(x^{k+1}) = \eta^{k+1}$. $k \leftarrow k + 1$.
    Go to Step I.
```

REMARK 5.3.1. Problem (PP) is a maximization problem. Therefore, we have changed in contrast to the previous algorithms the meaning of the Greek symbols η and μ. In Algorithm 5.1 η denotes a lower bound and μ is an upper bound.

As mentioned before, the formulation of Algorithm 5.1 follows nearly the guidelines of a general branch-and-bound scheme given in [HPT95]. There are two main ways in order to adapt this general algorithm to a special problem or problem class. First of all it is necessary to decide how the bounds should be constructed. For the lower bounds η^k ($k \in \mathbb{N}$) we use the most common and simple idea, which we also used in Algorithm 3.1 and Algorithm 4.1 for the bounds η^k. The lower bound is updated each time the algorithm generates a new point $y \in \mathbb{R}^{2n}$ belonging to the feasible region of (PP) with a function value $f(y)$ bigger than the current bound η^k.

In the construction of the upper bound μ_R with respect to a given hyperrectangle R we invest more effort. Similar to the solution approaches for (QP) we calculate this bound by solving an LP-relaxation of Problem (PP) with the additional constraint $(t, x) \in [\eta^k, \mu^k] \times R$ (see Subproblem (SP) in Step V). By interpreting Problem (SP) as an all-quadratic problem or as a polynomial problem we could choose the LP-relaxations proposed in [AKLV95] or [ST92] (see also Section 1.3). However, doing this we do not stay abreast of the special structure of (SP). Note that each quadratic constraint depends only on the four variables x_{i_1}, x_{i_2}, x_{j_1} and x_{j_2}. Exploiting this structure we are able to construct a better linear approximation of the feasible region of Problem (SP) than by using one of these general approaches. In Section 5.4 our method for calculating upper bounds for this special problem is discussed.

The second step of adapting a general branch-and-bound scheme is to determine in which way we subdivide the current subdivision set. In Algorithm 3.1 and Algorithm 4.1 we used radial subdivisions of the applied n-simplices (see Definition 1.2.2), which result in a partition (Definition 1.2.1) of the subdivided set. The hyperrectangle R^k used in Algorithm 5.1 is the Cartesian product of n two-dimensional rectangles R_i^k ($i \in \{1, \ldots, n\}$). We partition R^k by splitting one of these rectangles R_i^k. In Step II we decide which rectangle R_i^k ($i \in \{1, \ldots, n\}$) we would like to subdivide and in Step III we use a strategy to generate $l \in \mathbb{N}$ rectangular subsets of R_i^k. How we do this is described in Section 5.5. Our strategy is similar to the well-known bisection approach (see, e.g., page 101). However, in contrast to this strategy we divide R^k in each iteration with respect to two dimensions and not only regarding one dimension. Furthermore, exploiting the structure of Problem (PP) we are able to eliminate a lot of possible partition sets in advance even without computing upper bounds. Therefore – at the end of Step III – the sets $R^{k_1}, \ldots, R^{k_l}$ do not form a partition of R^k. The property

$$\bigcup_{i=1}^{l} R^{k_i} = R^k \quad \Leftrightarrow \quad \bigcup_{i=1}^{l} R_j^{k_i} = R_j^k \tag{5.3.1}$$

is not necessarily satisfied (compare with the required properties in Step III).

In branch-and-bound algorithms derived for general problem classes it is usually not possible to manipulate the current subdivision set R^k apart from its splitting. In Algorithm 3.1 and Algorithm 4.1 we do not know how to manipulate further the n-simplices S_j^k resulting from the partition of the current set S^k, since in general no additional information about the structure of Problem (QP) is available. If a branch-and-bound scheme is developed for a special problem instance, as it is the case for Algorithm 5.1, then exploiting the structure of this instance could enable us to derive manipulation strategies for the sets resulting after the subdivision step. In fact, in the case of Problem (PP) we can reduce under some circumstances the size of the relevant hyperrectangles R^{k_i} using the theoretical results derived in Section 5.2 and the knowledge of the current best known value η^k, as mentioned in Step IV of the algorithm. The resulting ***size reduction strategies*** are presented in Section 5.6.

Before expatiating the details of the suggested algorithm in the following sections let us first give some notes on the convergence of this approach. We would like to formulate three conditions, which have to be satisfied by the upper bounds (Step V) and by the subdivision set manipulation strategies, i.e., by the subdivision strategies and the size reduction strategies (Step III and Step IV). Using these conditions we are able to prove the convergence of our method.

The conditions are as follows.

(C1) The subdivision strategy is exhaustive, i.e., for each infinite sequence of hyperrectangles $\{R^k\}_{k\in\mathbb{N}}$ satisfying $R^{k+1} \subset R^k$ for each $k \in \mathbb{N}$, there exists a point $s \in U^n$ with

$$\lim_{k\to\infty} R^k = \bigcap_{k\in\mathbb{N}} R^k = \{s\}$$

(compare with Definition 4.3.1).

(C2) If an infinite nested sequence of hyperrectangles $\{R^k\}_{k\in\mathbb{N}}$ with the property $\lim_{k\to\infty} R^k = \{s\} \subset U^n$ is given, then there holds

$$\lim_{k\to\infty} \mu_{R^k} \;=\; f(s)\,.$$

(C3) The subdivision strategies and the size reduction strategies are *consistent* in the following sense. Let $P^k \subset U^n$ be the union of the relevant subdivision sets in Step I in iteration $k \in \mathbb{N}$, i.e., $P^k = \bigcup_{R\in\mathcal{R}^k} R$. Then there holds that $P^k \cup Q$ contains an optimal solution of Problem (PP), i.e.,

$$\big(P^k \cup Q\big) \cap SOL(n) \;\neq\; \emptyset$$

where $SOL(n)$ denotes the set of optimal solution of Problem (PP) with n scattering points.

Condition (C1) for the subdivision strategy and Condition (C2) for the upper bounds are often used in order to prove the convergence of branch-and-bound schemes for general problem classes (see, e.g., [HPT95, Section 3.7]). Note that in the convergence proof for Algorithm 3.1 (see Section 3.4) and in the convergence proof for Algorithm 4.1 in the exhaustive case (see Section 4.3) we have just verified these conditions. In both algorithms we had the property that the sets resulting from the subdivision of the current n-simplex form a partition of S^k and that these sets are not further manipulated. Therefore, these two conditions were sufficient for the convergence of these approaches.

As mentioned before, the subdivision strategy used in Step III of Algorithm 5.1 does not lead to a partition, since Relation (5.3.1) is not necessarily satisfied. Moreover, we are able to reduce the size of the hyperrectangles $R^{k_1}, \ldots, R^{k_l}$ with our size reduction strategies in Step IV. Therefore, in order to prove the correctness of Algorithm 5.1 in the sense that this method detects an ϵ-optimal solution of Problem (PP), it is not sufficient that Condition (C1) and Condition (C2) are fulfilled. In Section 5.5 and in Section 5.6 we will see that using our subdivision set manipulation strategies we may lose optimal solutions, i.e., we cut away parts of the feasible region of Problem (PP) containing optimal solutions without detecting them. However, as long as the strategies applied in our method guarantee that there still exist at least one optimal solution in the part of the feasible area of (PP), which is not eliminated by the set manipulation strategies, we are able to prove the correctness of Algorithm 5.1. If Algorithm 5.1 fulfills Condition (C3), it is ensured that not all optimal solutions are eliminated without detecting them.

Under the assumption that the required conditions are satisfied by the strategies used in Algorithm 5.1 we are able to show that our method detects in finite time an ϵ-optimal solution of Problem (PP), if ϵ is chosen greater than 0. This will be a direct consequence of the following convergence theorem, which proves the correctness of Algorithm 5.1 for the case $\epsilon = 0$.

THEOREM 5.3.1. *Assume that $\epsilon = 0$ and that Algorithm 5.1 fulfills Conditions (C1), (C2) and (C3). Then the following assertions are true:*

(i) *If Algorithm 5.1 stops after a finite number of iterations with $\eta^k = \mu^k$, then it follows that x^k is an optimal solution of Problem (PP) with optimal value η^k.*

(ii) *If Algorithm 5.1 generates an infinite point sequence $\{x^k\}_{k\in\mathbb{N}}$, then there holds that each accumulation point $x^\star$ of this sequence is an optimal solution of Problem (PP) with optimal value $f(x^\star)$.*

PROOF: Denote, for $k \in \mathbb{N}$, by P^k the part of U^n still to be analyzed in Step I of iteration k, i.e.,

$$P^k \;=\; \bigcup\nolimits_{R\in\mathcal{R}^k} R\,.$$

From the description of the algorithm it follows immediately that, for any $k \in \mathbb{N}$,

$$f(x^k) = \eta^k \le \max_{x\in P^k\cup Q} f(x) = \max\{ \max_{(t,x)\in F^k} t\;,\; \underbrace{\max_{x\in Q} f(x)}_{=\eta^k} \} \tag{5.3.2}$$

with $F^k = \{(t,x) \in \mathbb{R} \times P^k \text{ with } t - \|x_i - x_j\|_2^2 \le 0\,,\, 1 \le i < j \le n\}$, and

$$\max_{x\in P^k\cup Q} f(x) \;\le\; \mu^k\,. \tag{5.3.3}$$

Since Condition (C3) is satisfied for each $k \in \mathbb{N}$, there holds

$$\max_{x\in P^k\cup Q} f(x) \;=\; t^\star(n)\,. \tag{5.3.4}$$

Combining (5.3.2), (5.3.3) and (5.3.4) we obtain the first result (i).

In order to prove (ii) we can use the general convergence theory proposed in [HPT95, Section 3.7]. Because of Property (C3) our algorithm has the same essential properties as the general branch-and-bound scheme used in [HPT95, Algorithm 3.5]. Note that we are interested in detecting one global solution of Problem (PP). Therefore, assertion (ii) follows immediately from [HPT95, Theorem

3.8], if for each infinite nested subsequence $\{R^{k_q}\}_{q\in\mathbb{N}}$ of the generated sequence $\{R^k\}_{k\in\mathbb{N}}$ of $2n$-dimensional hyperrectangles, there holds

$$\lim_{q\to\infty}[\mu_{R^{k_q}} - \eta^{k_q}] \;=\; 0\,. \tag{5.3.5}$$

Let $\{R^{k_q}\}_{q\in\mathbb{N}}$ be a subsequence of $\{R^k\}_{k\in\mathbb{N}}$ satisfying $R^{k_{q+1}} \subset R^{k_q}$ for each $q \in \mathbb{N}$. From Condition (C1) we know that in this situation there exists a point $s \in U^n$ with

$$\lim_{q\to\infty} R^{k_q} \;=\; \{s\}\,. \tag{5.3.6}$$

Algorithm 5.1 generates, for each $q \in \mathbb{N}$, a point $y^{k_q} \in R^{k_q}$ satisfying

$$f(y^{k_q}) \;\leq\; \eta^{k_q}\,.$$

Note that in Step V at least one point belonging to R^{k_q} ($q \in \mathbb{N}$) is used for updating the lower bound η^{k_q}. From (5.3.6) we obtain hence

$$\lim_{q\to\infty} f(y^{k_q}) = f(s) \;\leq\; \lim_{q\to\infty} \eta^{k_q}\,. \tag{5.3.7}$$

Condition (C2) implies that

$$\lim_{q\to\infty} \mu_{R^{k_q}} \;=\; f(s)\,. \tag{5.3.8}$$

Using (5.3.7) and (5.3.8) Property (5.3.5) follows readily. ■

REMARK 5.3.2. In the convergence proofs for Algorithm 3.1 and Algorithm 4.1 in the exhaustive case it was not possible to use the general convergence theory given in [HPT95, Section 3.7] (see Remark 3.4.1(b)). In both approaches we do not necessarily generate feasible points for each considered subdivision set. Therefore, we do not know how the sequence $\{\eta^k\}_{k\in\mathbb{N}}$ behaves. Since in Algorithm 5.1 we consider the formulation of Problem (PP) given on page 210 we can use each point belonging to the current hyperrectangle in order to update the lower bound η^k ($k \in \mathbb{N}$). This guarantees that the sequence $\{\eta^k\}_{k\in\mathbb{N}}$ converges to the optimal value $t^\star(n)$ in the infinite case.

As a consequence of the convergence result presented above it is immediately clear that, for each $\epsilon > 0$, Algorithm 5.1 generates an ϵ-optimal solution x^k in finite time. In the following sections we describe the details of the calculation of the upper bounds and the details of the diverse subdivision set manipulation strategies, which we used in our implementation of Algorithm 5.1. In order to guarantee a correct functioning of the suggested approach we will have to show at the respective places that the postulated conditions are fulfilled.

5.4. Upper Bounds

In this section we describe the calculation of an upper bound for the optimization problem

$$\begin{aligned} &\max\ t \\ &t - \|x_i - x_j\|_2^2 \le 0 \qquad 1 \le i < j \le n \\ &x_i \in R_i \qquad\qquad\qquad i = 1, \dots , n \\ &\eta \le t \le \mu\,, \end{aligned} \tag{SP'}$$

where, for $i \in \{1, \dots , n\}$, $R_i = [l_{i_1}, L_{i_1}] \times [l_{i_2}, L_{i_2}]$ ($0 \le l_{i_j} \le L_{i_j} \le 1$, $j \in \{1, 2\}$) is a two-dimensional rectangle, and $\eta > 0$ and $\mu \le 4$ are real numbers. This problem coincides with Subproblem (SP) in the description of Algorithm 5.1 in the previous section.

We calculate an upper bound of (SP') by solving an LP-relaxation of this problem. In order to obtain such a relaxation we need for any concave quadratic constraint

$$t - \|x_i - x_j\|_2^2 \le 0$$

($1 \le i < j \le n$) a piecewise affine convex function $h_{ij} : \mathbb{R} \times \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ with the property

$$\begin{aligned} &\{(t, x_i, x_j)^T \in [\eta, \mu] \times R_i \times R_j :\ t - \|x_i - x_j\|_2^2 \le 0\} \\ &\qquad \subset \{(t, x_i, x_j)^T \in [\eta, \mu] \times R_i \times R_j :\ h_{ij}(t, x_i, x_j) \le 0\}\,. \end{aligned} \tag{5.4.1}$$

In order to simplify the presentation we ignore at first the indices i, j and consider one concave quadratic constraint

$$g(t, x, y)\ :=\ t - \|x - y\|_2^2\ \le\ 0\,,$$

where $(t, x, y)^T$ is restricted to the set $[\eta, \mu] \times R_x \times R_y$ with

$$R_x\ :=\ [l_{x_1}, L_{x_1}] \times [l_{x_2}, L_{x_2}]$$

and

$$R_y\ :=\ [l_{y_1}, L_{y_1}] \times [l_{y_2}, L_{y_2}]\,.$$

In order to construct a piecewise affine convex function h such that

$$\begin{aligned} F\ :=\ &\{(t, x, y)^T \in [\eta, \mu] \times R_x \times R_y :\ t - \|x - y\|_2^2 \le 0\} \\ &\qquad \subset \{(t, x, y)^T \in [\eta, \mu] \times R_x \times R_y :\ h(t, x, y) \le 0\}\,, \end{aligned} \tag{5.4.2}$$

let us examine the quadratic part of g, i.e., consider the function $\bar{g} : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ given by

$$\bar{g}(x, y) \ := \ \|x - y\|_2^2 \ = \ (x_1 - y_1)^2 + (x_2 - y_2)^2$$

over the 4-dimensional rectangle $\bar{R} := R_x \times R_y$. Substituting v for the term $(x_1 - y_1)$ and w for $(x_2 - y_2)$ we can interpret $\bar{g}$ as a two-dimensional function $\hat{g} : \mathbb{R}^2 \to \mathbb{R}$ with

$$\hat{g}(v, w) \ := \ v^2 + w^2 \ .$$

According to the feasible region of x and y the new variables are restricted as follows

$$\begin{aligned} -1 \ \leq \ l_v := \ l_{x_1} - L_{y_1} \leq v \leq L_{x_1} - l_{y_1} =: \ L_v \ \leq \ 1 \\ -1 \ \leq \ l_w := \ l_{x_2} - L_{y_2} \leq w \leq L_{x_2} - l_{y_2} =: \ L_w \ \leq \ 1 \ . \end{aligned}$$

If a piecewise affine function $\hat{h} : \mathbb{R}^2 \to \mathbb{R}$ with the properties

- $\hat{h}$ is concave, i.e., the minimum of a finite number of affine functions,
- there holds, for any $\binom{v}{w} \in \hat{R} := [l_v, L_v] \times [l_w, L_w]$,

$$\hat{h}(v, w) \begin{cases} \geq \mu & \text{, if } \hat{g}(v, w) \geq \mu \\ \geq \hat{g}(v, w) & \text{, if } \eta \leq \hat{g}(v, w) < \mu \end{cases} , \qquad (5.4.3)$$

is given, we obtain by setting

$$h(t, x, y) := t - \hat{h}(x_1 - y_1, x_2 - y_2)$$

a function h fulfilling (5.4.2). Note that if there is a point $(t, x, y)^T \in [\eta, \mu] \times R_x \times R_y$ with $\bar{g}(x, y) \geq \mu$ and $g(t, x, y) \leq 0$, then – regarding (5.4.3) – there holds $\hat{h}(x_1 - y_1, x_2 - y_2) \geq \mu$ and hence $h(t, x, y) \leq 0$. Note further that there does not exist a point $(t, x, y)^T \in [\eta, \mu] \times R_x \times R_y$ with $\bar{g}(x, y) < \eta$ and $g(t, x, y) \leq 0$.

In the next part of this section we describe the construction of a piecewise affine concave function $\hat{h}$ with Property (5.4.3) in detail. Denote by

$$V(\hat{R}) \ = \ \{\begin{pmatrix} l_v \\ l_w \end{pmatrix}, \begin{pmatrix} L_v \\ l_w \end{pmatrix}, \begin{pmatrix} L_v \\ L_w \end{pmatrix}, \begin{pmatrix} l_v \\ L_w \end{pmatrix}\} \ = \ \{v_1, v_2, v_3, v_4\}$$

the set of vertices of the two-dimensional rectangle $\hat{R}$ and assume, without loss of generality, that there holds

$$\hat{g}(v_1) \ = \ \max_{z \in \hat{R}} \hat{g}(z) \ . \qquad (5.4.4)$$

Note that a convex function always attains its maximum over a polytope P in a vertex of P [HPT95, Theorem 1.19].

If an affine function $\ell : \mathrm{I\!R}^2 \to \mathrm{I\!R}$, $\ell(z) = a^T z + b$ coinciding in three vertices of $\hat{R}$ with the function values of $\hat{g}$ is given, it follows by straightforward calculation that ℓ also coincides with $\hat{g}$ in the fourth vertex, i.e., for each $i \in \{1, \dots, 4\}$, there holds

$$\ell(v_i) \;=\; \hat{g}(v_i)\,. \tag{5.4.5}$$

PROOF OF (5.4.5): If, for $i \in \{1, 2, 3\}$, there holds $\ell(v_i) = \hat{g}(v_i)$, we obtain

$$\begin{aligned} \ell(v_1) \;&=\; a_1 l_v + a_2 l_w + b = l_v^2 + l_w^2 \;=\; \hat{g}(v_1) \\ \ell(v_2) \;&=\; a_1 L_v + a_2 l_w + b = L_v^2 + l_w^2 \;=\; \hat{g}(v_2) \\ \ell(v_3) \;&=\; a_1 L_v + a_2 L_w + b = L_v^2 + L_w^2 \;=\; \hat{g}(v_3) \quad , \end{aligned}$$

and therefore

$$\begin{aligned} \hat{g}(v_4) \;=\; l_v^2 + L_w^2 \;&=\; \ell(v_1) - \ell(v_2) + \ell(v_3) \\ &=\; a_1 l_v + a_2 L_w + b = \ell(v_4) \end{aligned}$$

□

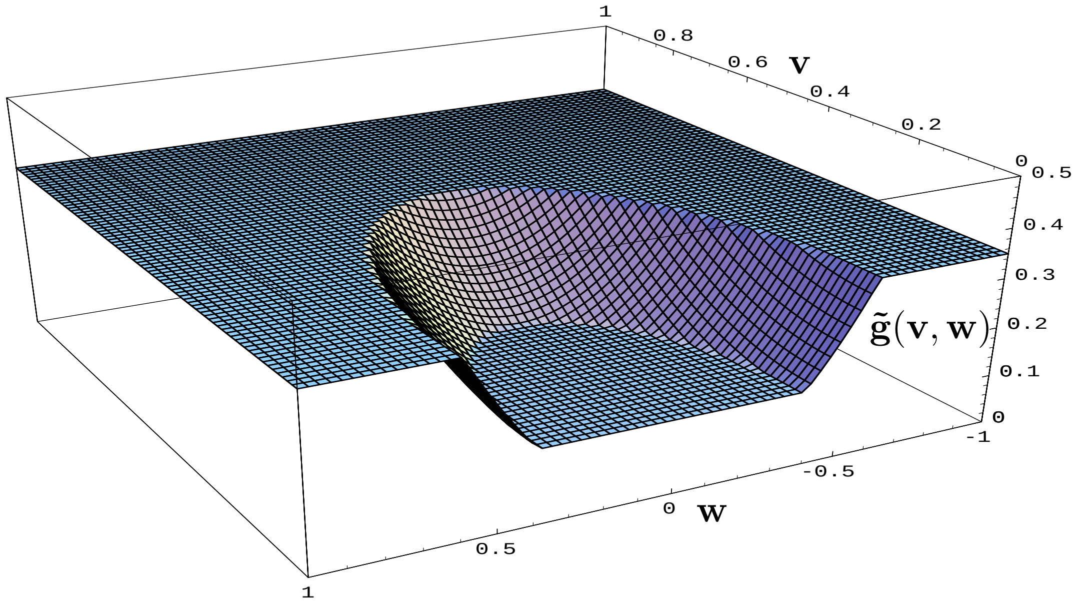
REMARK 5.4.1. The function ℓ is the concave envelope of $\hat{g}$ on the rectangle $[v_1, v_2, v_3, v_4]$ (see Subsection 1.2.4). Note that the concave envelope of a separable function $f : \mathrm{I\!R}^n \to \mathrm{I\!R}$, $f(x) = \sum_{i=1}^n f_i(x_i)$ on a rectangle $R = \{x \in \mathrm{I\!R}^n : l_i \le x_i \le L_i\,,\, i = 1, \dots, n\}$ is the sum of the concave envelopes of each part $f_i : \mathrm{I\!R} \to \mathrm{I\!R}$ of f on the interval $[l_i, L_i]$ ($i = 1, \dots, n$) [HT96B, Theorem IV.8].

Because of Relation (5.4.5) it is obvious that the affine function ℓ is an overestimating function for $\hat{g}$ on the whole rectangle $\hat{R}$, and especially that ℓ fulfills (5.4.3). Hence the simplest way in order to obtain the required function $\hat{h}$ is to take the function ℓ itself. However, in some circumstances it is possible to "improve" this overestimator, where "improve" is meant in the sense of a concave approximation of $\hat{g}$ with respect to the feasible region F, which has function values smaller than or equal to ℓ. Considering the structure of F we recognize that it is not necessary to overestimate $\hat{g}$ on the whole rectangle $\hat{R}$, as we do by choosing ℓ. We only need a concave overestimator for the function

$$\min\{\, \mu\,,\, \hat{g}(v, w)\,\} \tag{5.4.6}$$

on the set $\hat{R} \cap \{\binom{v}{w} \in \mathrm{I\!R}^2 : \hat{g}(v, w) \ge \eta\}$ (see Figure 5.7 and compare with Property (5.4.3)). Note that all points $\binom{v}{w} \in \hat{R}$ with a function value $\hat{g}(v, w)$ lower than η (see the bottom of the function $\tilde{g}$ in Figure 5.7) are infeasible, i.e., $F \cap \{\binom{v}{w} \in \mathrm{I\!R}^2 : \hat{g}(v, w) < \eta\} = \emptyset$.

FIGURE 5.7. The relevant function



$$\tilde{g}(v,w) = \min\{\mu, \max\{\hat{g}(v,w),\eta\}\,\}$$
with $(v,w)^T \in [0,1] \times [-1,1]$

Depending on the function values of $\hat{g}$ in the vertices of $\hat{R} \subset [-1,1] \times [-1,1]$ we distinguish four main cases.

Case 1: $\hat{g}(v_i) < \eta$, $i \in \{1,\dots,4\}$ (see Figure 5.8)

Since $\hat{g}$ is a convex function it follows immediately that, for each $(v,w)^T \in \hat{R}$, there holds $\hat{g}(v,w) < \eta$. This implies $F = \emptyset$. In this case it is not necessary to construct a function $\hat{h}$ because we do not need an upper bound for Problem (SP').

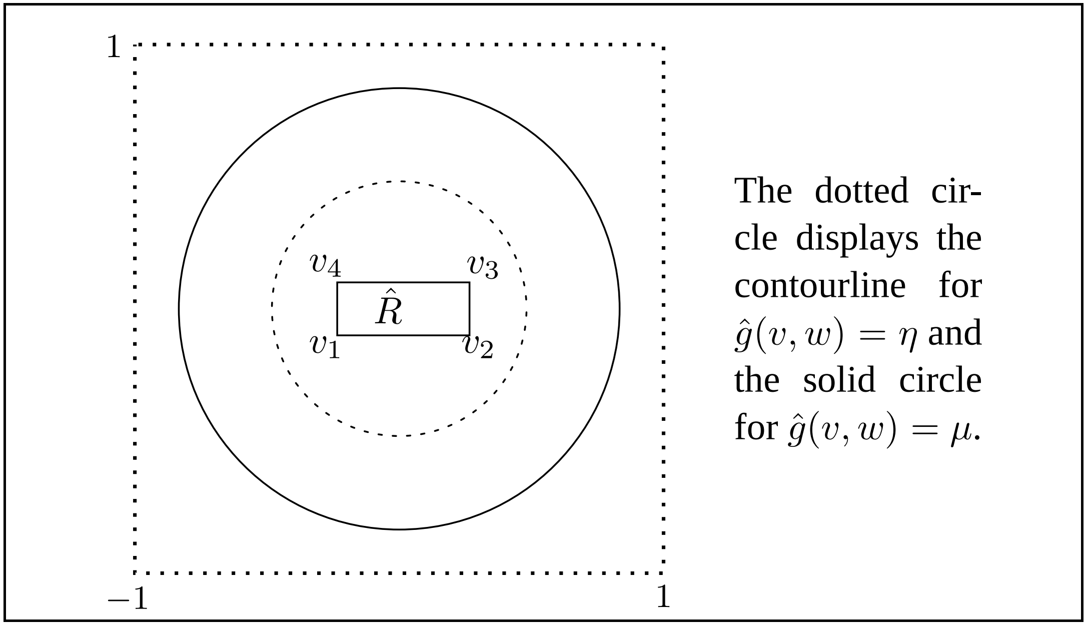
Case 2: $\hat{g}(v_i) \geq \mu$, $i \in \{1,\dots,4\}$ (see Figure 5.9(a))

Since, for any $(v,w)^T \in \hat{R}$, there holds

$$\min\{\,\mu\,,\,\max\{\hat{g}(v,w),\eta\}\,\} \;\leq\; \mu\,,$$

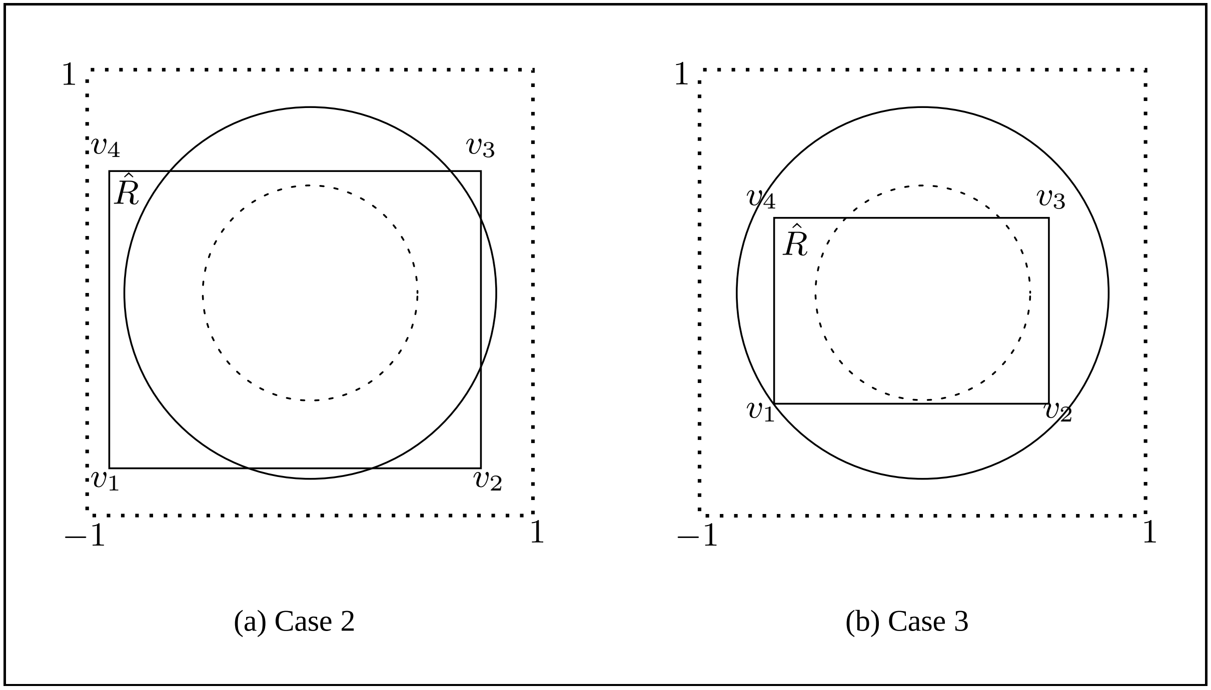
we obtain by setting $\hat{h} \equiv \mu$ a constant function with Property (5.4.3). This function is a better approximation of $\hat{g}$ with respect to the feasible region F than the affine function ℓ. Moreover, there is no possibility to further improve $\hat{h}$ without losing the

FIGURE 5.8. Case 1



The dotted circle displays the contourline for $\hat{g}(v,w) = \eta$ and the solid circle for $\hat{g}(v,w) = \mu$.

FIGURE 5.9. Case 2 and 3



(a) Case 2

(b) Case 3

concavity.

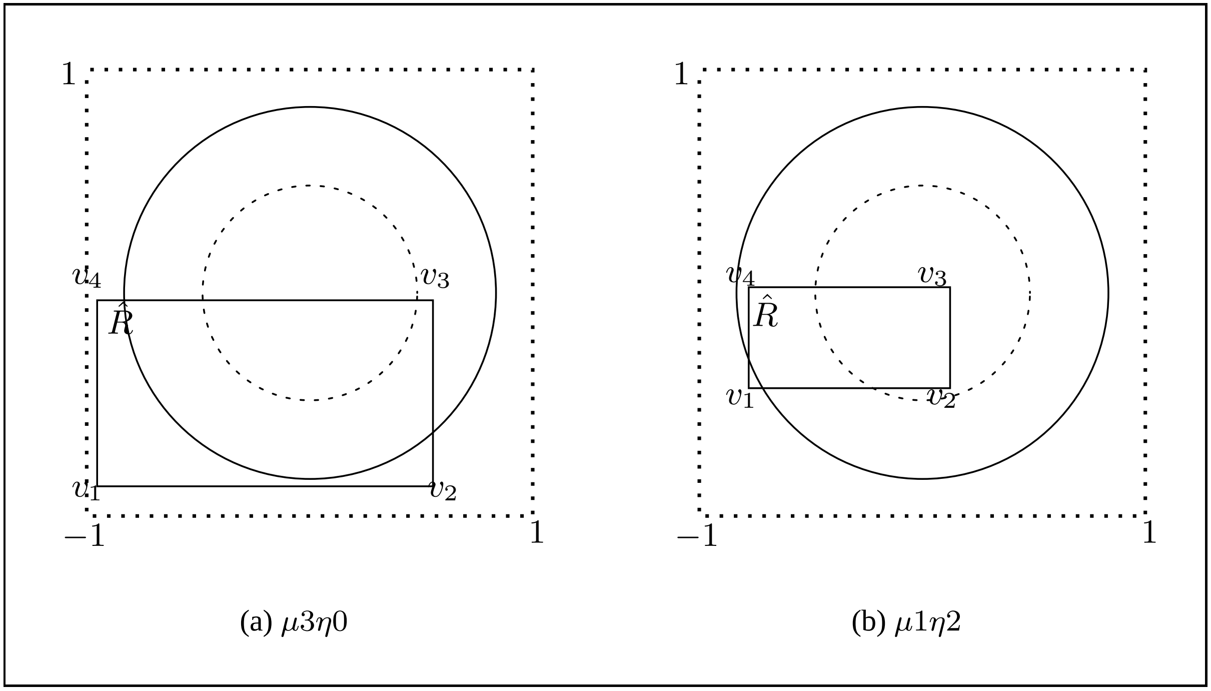
Case 3: $\eta \leq \hat{g}(v_i) \leq \mu$, $i \in \{1, \dots, 4\}$ (see Figure 5.9(b))

In this situation the previously defined affine function ℓ is the best approximation of $\hat{g}$ fulfilling (5.4.3). As in Case 2 it is not possible to improve $\hat{h} \equiv \ell$ and preserve simultaneously the concavity of $\hat{h}$.

<u>Case 4</u>: $\exists i, j \in \{1, \ldots, 4\}$ with $(\hat{g}(v_i) > \mu$ and $\hat{g}(v_j) < \mu)$ (see Figure 5.10(a))
or $(\hat{g}(v_i) < \eta$ and $\hat{g}(v_j) > \eta)$ (see Figure 5.10(b))

In this situation the affine function ℓ is not the best concave approximation of $\hat{g}$ with

FIGURE 5.10. Case 4



(a) $\mu 3 \eta 0$

(b) $\mu 1 \eta 2$

Property (5.4.3). Using a piecewise affine function we are able to improve ℓ. It is possible that up to three vertices v_i of $\hat{R}$ have a function value $\hat{g}(v_i)$ bigger than μ or that up to three vertices have a function value smaller than η. Depending on the number of vertices of $\hat{R}$ with a function value bigger or smaller than μ or η, respectively, there are hence 12 possible subcases (see Table 5.1).

To depict the rather simple ideas in order to obtain a better approximation of $\hat{g}$ we describe the construction of the function $\hat{h}$ for Subcase $\mu 0 \eta 1$ in detail (compare with Figure 5.11). Let v_5 and v_6 be the intersection points of the level curve $\{(v, w) \in [-1, 1] \times [-1, 1] : \hat{g}(v, w) = \eta\}$ with the boundary of the rectangle $\hat{R}$ and assume at first that there holds

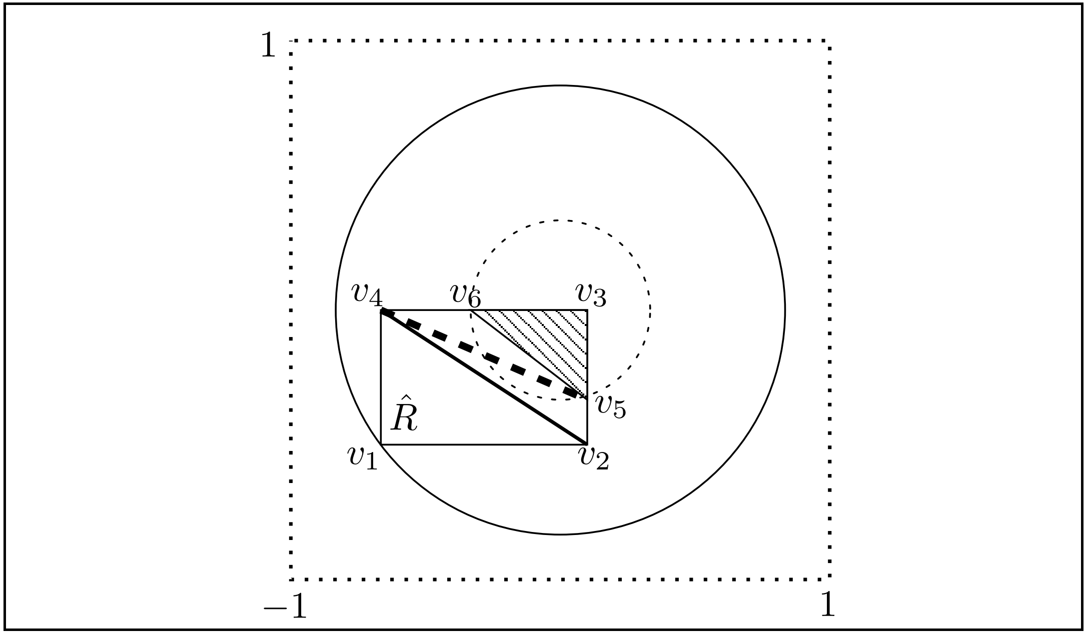
$$\begin{aligned} v_5 &\in \operatorname{relint}([v_2, v_3]) \\ v_6 &\in \operatorname{relint}([v_3, v_4]) \end{aligned} \tag{5.4.7}$$

TABLE 5.1. Possible subcases

Subcase	$\lvert\{i \in \{1,\ldots,4\} : \hat{g}(v_i) > \mu\}\rvert$	$\lvert\{i \in \{1,\ldots,4\} : \hat{g}(v_i) < \eta\}\rvert$
$\mu 0 \eta 1$	0	1
$\mu 0 \eta 2$	0	2
$\mu 0 \eta 3$	0	3
$\mu 1 \eta 0$	1	0
$\mu 1 \eta 1$	1	1
$\mu 1 \eta 2$	1	2
$\mu 1 \eta 3$	1	3
$\mu 2 \eta 0$	2	0
$\mu 2 \eta 1$	2	1
$\mu 2 \eta 2$	2	2
$\mu 3 \eta 0$	3	0
$\mu 3 \eta 1$	3	1

(for the definition of the relative interior (relint) of a set we refer to [ROC70]). The points belonging to the triangle formed by v_3, v_5 and v_6 – except the points v_5 and v_6 themselves – (see the shaded region in Figure 5.11) are not feasible. Therefore, it is not necessary to overestimate $\hat{g}$ on this region. If we kink the affine function ℓ along the line between v_2 and v_4 and pull down the part lying over $[v_2, v_4, v_5, v_6]$

FIGURE 5.11. Case $\mu 0 \eta 1$



as much as possible, we improve the approximation of $\hat{g}$ over the feasible region F.

Let us explain this strategy now in a more technical way. Let ℓ_1 be the affine function, which coincides with the function $\hat{g}$ at the vertices v_1, v_2 and v_4, i.e., $\ell_1 \equiv \ell$. Let further ℓ_{21} be the affine function, which coincides at the points v_2, v_4 and v_5 with $\hat{g}$ and, analogously, ℓ_{22} be the affine function coinciding with $\hat{g}$ at the points v_2, v_4 and v_6. The affine functions ℓ_{21} and ℓ_{22} are by construction equal along the line joining the points v_2 and v_4. This line splits the two-dimensional real space $\mathbb{R}^2$ into two halfspaces, where the points v_5 and v_6 belong to the same of these halfspaces. Therefore, one of the following relations have to be satisfied

$$\ell_{21}(v_6) \ \geq \ \ell_{22}(v_6) \ = \ \hat{g}(v_6) \tag{5.4.8.a}$$

or

$$\ell_{22}(v_5) \ \geq \ \ell_{21}(v_5) \ = \ \hat{g}(v_5)\,. \tag{5.4.8.b}$$

If (5.4.8.a) is fulfilled, we set $\ell_2 := \ell_{21}$. Otherwise we choose $\ell_2 := \ell_{22}$. It is immediately clear that $\hat{h}_1 : \hat{R} \to \mathbb{R}$ given by

$$\begin{aligned}\hat{h}_1(v) &:= \min\{\ell_1(v), \ell_2(v)\}\\ &= \begin{cases} \ell_1(v) & \text{, if } v \in [v_1, v_2, v_4] \\ \ell_2(v) & \text{, if } v \in [v_2, v_4, v_3] \end{cases}\end{aligned}$$

is a piecewise affine concave function satisfying Property (5.4.3).

Assume now, without loss of generality, that (5.4.8.a) is true, i.e., we choose $\ell_2 \equiv \ell_{21}$. We are able to improve $\hat{h}_1$ further by kinking ℓ_2 along the line between v_4 and v_5 and now pulling down the part over $[v_4, v_5, v_6]$. Let ℓ_3 be the affine function coinciding with $\hat{g}$ at the points v_4, v_5 and v_6. It follows by the same arguments as before that $\hat{h} : \hat{R} \to \mathbb{R}$

$$\begin{aligned}\hat{h}(v) &:= \min\{\hat{h}_1(v), \ell_3(v)\}\\ &= \begin{cases} \ell_1(v) & \text{, if } v \in [v_1, v_2, v_4] \\ \ell_2(v) & \text{, if } v \in [v_2, v_4, v_5] \\ \ell_3(v) & \text{, if } v \in [v_3, v_4, v_5] \end{cases}\end{aligned}$$

is also a piecewise affine function fulfilling (5.4.3). This function has the additional property that, for each $v \in [v_5, v_6, v_3]$, there holds

$$\hat{h}(v) \ \leq \ \eta\,. \tag{5.4.9}$$

If Assumption (5.4.7) is not true, we can simplify the definition of $\hat{h}$ as described in the following cases:

Case 1: $v_5 = v_2$ and $v_6 \in \operatorname{relint}([v_3, v_4])$

$$\hat{h}(v) := \min\{\ell_1(v), \ell_{22}(v)\}\,,\, v \in \hat{R}$$

Case 2: $v_6 = v_4$ and $v_5 \in \operatorname{relint}([v_2, v_3])$

$$\hat{h}(v) := \min\{\ell_1(v), \ell_{21}(v)\}\,,\, v \in \hat{R}$$

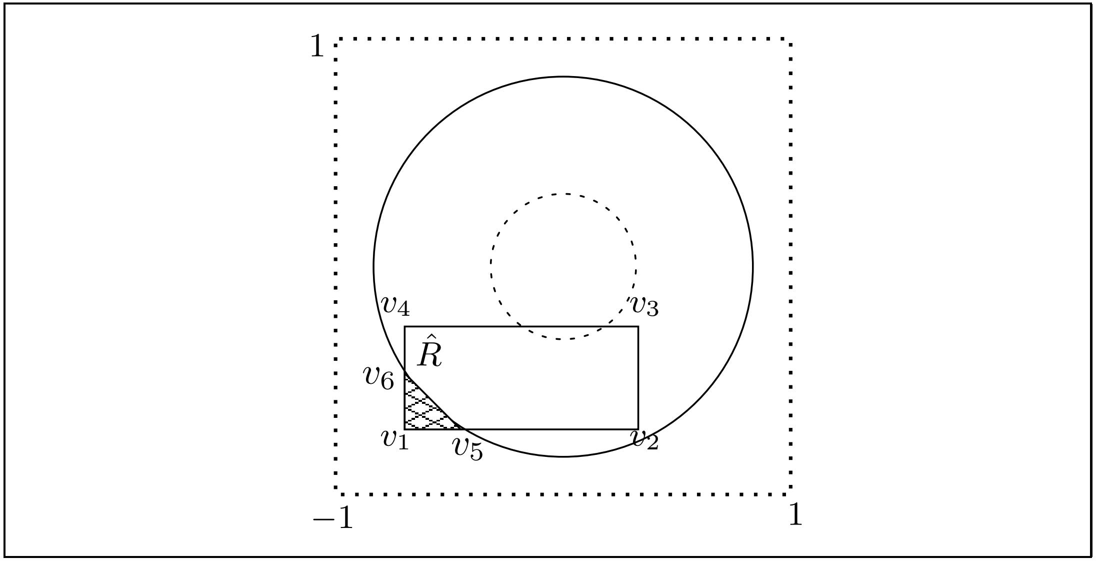
Case 3: $v_5 = v_2$ and $v_6 = v_4$

$$\hat{h}(v) := \ell_1(v)\,,\, v \in \hat{R}$$

In the described manner we are able to construct the required function $\hat{h}$ for Case $\mu 0 \eta 1$ by a minimum of up to three affine functions. The construction of $\hat{h}$ in the remaining eleven cases (see again Table 5.1) follows the same ideas.

REMARK 5.4.2. Note that there is one difference in the argumentation, if we construct the function $\hat{h}$ for the cases where at least one vertex of $\hat{R}$ has a function value bigger than μ, i.e., if there holds $g(v_1) > \mu$ (see (5.4.4)). Let us consider Case $\mu 1 \eta 0$ in order to explain this difference. Denote by v_5 and v_6 the intersection points of the level curve $\{(v, w) \in [-1, 1] \times [-1, 1] : \hat{g}(v, w) = \mu\}$ with the boundary of the rectangle $\hat{R}$ (see Figure 5.12) and let Δ be the triangle formed by the points v_1, v_5 and v_6. According to the described ideas we pull down twice the affine function ℓ over the triangle Δ. In Case $\mu 0 \eta 1$ this operation was allowed since

FIGURE 5.12. $\mu 1 \eta 0$



all elements of the shaded triangle in Figure 5.11 were infeasible. In contrast to this in the present case the points belonging to Δ are feasible (see the shaded region in Figure 5.12). However, the elements of Δ have function values with respect to $\hat{g}$ bigger than or equal to μ. Regarding the structure of the feasible region F it is hence not necessary to overestimate $\hat{g}$ on the triangle Δ (compare with (5.4.6)). It is sufficient if the function $\hat{h}$ is bigger than or equal to μ on this set. The application of the described ideas leads obviously to a function $\hat{h}$ – minimum of up to three affine functions – fulfilling, for each $v \in \Delta$,

$$\hat{h}(v) \;\geq\; \mu$$

(compare with Relation (5.4.9)). Therefore, in the cases with $\hat{g}(v_1) > \mu$ the concave overestimating function $\hat{h}$ for the function $\hat{g}$ on the set F can be constructed using the same ideas as described in Case $\mu 0 \eta 1$.

Table 5.2 shows the maximum number of affine functions needed for the construction of $\hat{h}$ in each subcase. These maximum numbers coincide with the number of triangles we use in order to partition the region of $\hat{R}$, where $\hat{g}$ has function values not smaller than η and not greater than μ. The choice of this triangle partition depends on the function values of $\hat{g}$ in the relevant corner points, as we have described in detail for the construction of $\hat{h}$ in Case $\mu 0 \eta 1$.

TABLE 5.2. Maximum number of affine functions

Subcase	maximum number	**Subcase**	maximum number
$\mu 0 \eta 1$	3	$\mu 1 \eta 3$	2
$\mu 0 \eta 2$	2	$\mu 2 \eta 0$	2
$\mu 0 \eta 3$	1	$\mu 2 \eta 1$	3
$\mu 1 \eta 0$	3	$\mu 2 \eta 2$	2
$\mu 1 \eta 1$	4	$\mu 3 \eta 0$	1
$\mu 1 \eta 2$	3	$\mu 3 \eta 1$	2

Consider now again the optimization problem (SP’). In the described way we are able to build for each pair (i, j) ($1 \leq i < j \leq n$) the required piecewise affine function $h_{ij} : \mathbb{R} \times \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ satisfying Condition (5.4.1), which is needed in order to obtain an upper bound for Problem (SP’). It follows immediately that the solution μ_R of the optimization problem

$$\begin{aligned} &\max\ t \\ &h_{ij}(t,x_i,x_j) \le 0 \qquad 1 \le i < j \le n \\ &x_i \in R_i \qquad\qquad\quad i = 1,\dots,n \\ &\eta \le t \le \mu\,, \end{aligned} \tag{LSP'}$$

delivers such an upper bound for the optimal value of (SP'). Note that (LSP') can be formulated as a linear program, since h_{ij} is a maximum of a finite number of affine functions.

In Section 5.3 we pointed out that it is necessary for a correct functioning of Algorithm 5.1 that the described upper bounds μ_R satisfy Condition (C2). This is ensured by the following lemma.

LEMMA 5.4.1. *Let $\{R^k = R_1^k \times \ldots \times R_n^k\}_{k\in\mathbb{N}}$ be an infinite sequence of $2n$-dimensional hyperrectangles with $R^1 \subset U^n$ and $R^k \supset R^{k+1}$ for each $k \in \mathbb{N}$. Assume further that there exists a point $s = (s_1,\dots,s_n)^T \in U^n$ satisfying*

$$\lim_{k\to\infty} R^k \ = \ \{s\} \tag{5.4.10}$$

Then there holds

$$\lim_{k\to\infty} \mu_{R^k} \ = \ f(s) \ = \ \min_{1\le i<j\le n} \|s_i - s_j\|_2^2\,. \tag{5.4.11}$$

PROOF: Let, for $1 \le i < j \le n$, $\ell_{ij}^{R^k} : \mathbb{R}^2 \to \mathbb{R}$ be the affine function ℓ with respect to the rectangles R_i^k and R_j^k, i.e., for each $l \in \{1,\dots,4\}$ there holds

$$\ell_{ij}^{R^k}(v_l^{ijR^k}) \ = \ (v_{l_1}^{ijR^k})^2 + (v_{l_2}^{ijR^k})^2 \ = \ \|v_l^{ijR^k}\|_2^2\,, \tag{5.4.12}$$

where $v_l^{ijR^k}$ is a vertex of the rectangle $R_i^k - R_j^k \ = \ [l_{i_1}^k - L_{j_1}^k, L_{i_1}^k - l_{j_1}^k] \times [l_{i_2}^k - L_{j_2}^k, L_{i_2}^k - l_{j_2}^k]$. According to the construction of $\hat{h}$ we know that, for each $1 \le i < j \le n$, $x_i \in R_i^k$, $x_j \in R_j^k$ and $t \in [\eta^k, \mu^k]$, there holds

$$\begin{aligned} h_{ij}^{R^k}(t,x_i,x_j) \ &\ge t - \ell_{ij}^{R^k}(x_{i_1} - x_{j_1}, x_{i_2} - x_{j_2}) \\ &= t - \ell_{ij}^{R^k}(x_i - x_j)\,. \end{aligned} \tag{5.4.13}$$

From (5.4.10) and (5.4.12) it follows immediately, for each $1 \le i < j \le n$ and $l \in \{1,\dots,4\}$, that

$$\ell_{ij}^{R^k}(v_l^{ijR^k}) \quad \to \quad \|s_i - s_j\|_2^2 \qquad (k \to \infty)\,. \tag{5.4.14}$$

Since, for each $1 \le i < j \le n$ and $k \in \mathbb{N}$, the affine function $\ell_{ij}^{R^k}$ attains its maximum over the rectangle $R_i^k - R_j^k$ in a vertex of this set, i.e.,

$$\max_{w \in R_i^k - R_j^k} \ell_{ij}^{R^k}(w) \;=\; \max_{l=1,\dots,4} \ell_{ij}^{R^k}(v_l^{ijR^k}) \, , \tag{5.4.15}$$

we obtain – taking (5.4.10) and (5.4.14) into account – that the following relation is satisfied, for each $1 \le i < j \le n$,

$$\max_{\substack{x_i \in R_i^k \\ x_j \in R_j^k}} \ell_{ij}^{R^k}(x_i - x_j) \quad \to \quad \|s_i - s_j\|_2^2 \qquad (k \to \infty) \, . \tag{5.4.16}$$

The point s is an element of each hyperrectangle R^k ($k \in \mathbb{N}$). Hence, we know that $f(s)$ is bounded from above by μ_{R^k} ($k \in \mathbb{N}$). Regarding (5.4.13) and (5.4.16) we can conclude

$$\begin{aligned}
f(s) \;\le\; \mu_{R^k} = & \max_{\substack{h_{ij}^{R^k}(t, x_i, x_j) \le 0 \\ x \in R^k \\ t \in [\eta^k, \mu^k]}} t \quad \le \max_{\substack{h_{ij}^{R^k}(t, x_i, x_j) \le 0 \\ x \in R^k}} t \\
\le & \max_{\substack{t - \ell_{ij}^{R^k}(x_i - x_j) \le 0 \\ x \in R^k}} t \quad = \max_{x \in R^k} \min_{1 \le i < j \le n} \ell_{ij}^{R^k}(x_i - x_j) \\
\le & \underbrace{\min_{1 \le i<j \le n} \underbrace{\max_{\substack{x_i \in R_i^k \\ x_j \in R_j^k}} \ell_{ij}^{R^k}(x_i - x_j)}_{\substack{\downarrow \ (k \to \infty) \\ \|s_i - s_j\|_2^2}}}_{\substack{\downarrow \quad (k \to \infty) \\ f(s)}} \; ,
\end{aligned}$$

which proves Relation (5.4.11). ■

Considering the structure of Problem (PP), respectively of Subproblem (SP), we were able to construct upper bounds, which can expect to be better than those obtained by a general approach for all-quadratic programs (see, e.g., [AKLV95, ST92]). In the subsequent two sections we will see that the examination of the structure of (PP) can also be exploited for the subdivision of the current hyperrectangle R^k. Doing this we will be able to substantially reduce the effort for solving Problem (PP) with Algorithm 5.1. However, we have to keep in mind that the subdivision set manipulation strategies introduced in Sections 5.5 and 5.6 fulfill Condition (C1) and Condition (C3).

5.5. Subdivision Strategies

Let $R^k = R_1^k \times \ldots \times R_n^k$ be the current hyperrectangle considered in iteration $k \in \mathbb{N}$ of Algorithm 5.1. As pointed out in the description of our approach (see Section 5.3) we choose in Step II an index $j \in \{1, \ldots, n\}$ such that the rectangle $R_j^k = [l_{j_1}^k, L_{j_1}^k] \times [l_{j_2}^k, L_{j_2}^k]$ has the longest edge-length among all rectangles forming R^k. In Step III of Algorithm 5.1 we subdivide this rectangle in order to obtain a subdivision of the hyperrectangle R^k. However, until now we did not say how we do that.

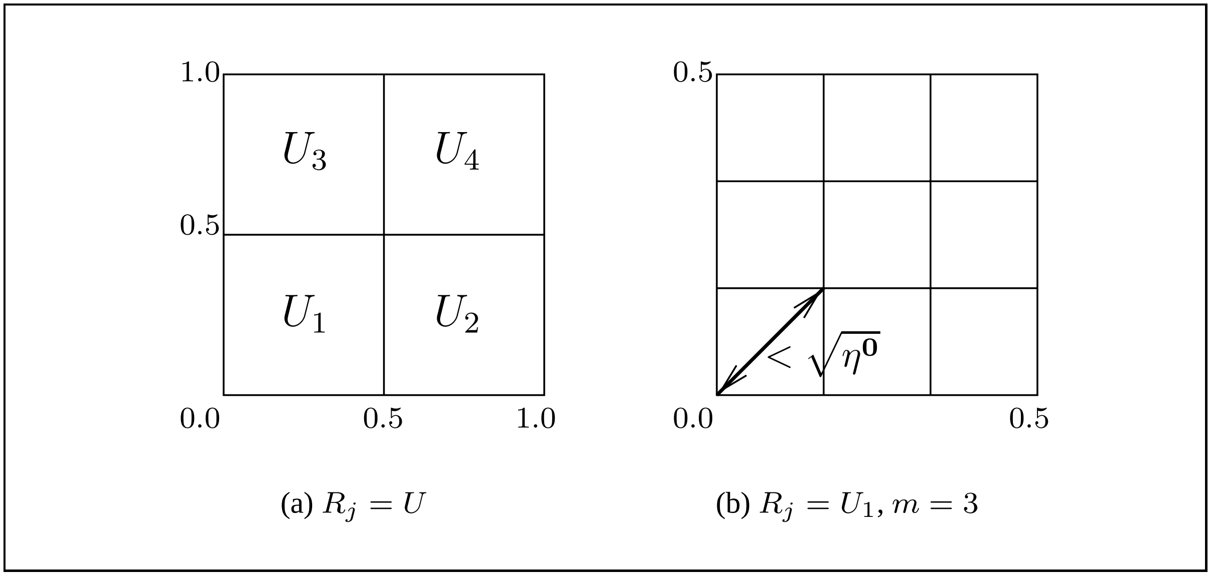
In the present section we describe the strategy applied for splitting the chosen rectangle R_j^k into a finite number of subrectangles. In order to simplify the presentation we ignore the iteration counter k in the following. We start this section with a description of the basic strategy leading to a partition of R_j. After this we discuss some special features of our subdivision strategy. Exploiting the structure of Problem (PP) they enable us to avoid redundant computations.

5.5.1. Basic Strategy. If the two-dimensional rectangle R_j coincides with the unit square U, we construct a partition of R_j (see Definition 1.2.1) consisting of the four squares

$$\begin{aligned} U_1 &= [0.0, 0.5] \times [0.0, 0.5] \\ U_2 &= [0.5, 1.0] \times [0.0, 0.5] \\ U_3 &= [0.0, 0.5] \times [0.5, 1.0] \\ U_4 &= [0.5, 1.0] \times [0.5, 1.0]\,. \end{aligned} \tag{5.5.1}$$

(see Figure 5.13(a)). Note that in Algorithm 5.1 the rectangle R_j is initialized with $U = [0,1]^2$. In the next level, i.e., if R_j is equal to one of the squares U_1, U_2, U_3 or U_4, we obtain a refined partition by constructing m^2 squares $\bar{U}_l$ ($l \in \{1, \ldots, m^2\}$) with equal size and edge-length $\frac{0.5}{m}$. The choice of the integer m with $m \geq 2$ depends on the value of the first lower bound $\eta^0 > 0$ determined by the first best known solution for (PP) (see the initialization phase of Algorithm 5.1). This choice shall assure that the squared diameter of $\bar{U}_l$ ($l \in \{1, \ldots, m^2\}$) is

FIGURE 5.13. Basic subdivision strategy



(a) $R_j = U$ (b) $R_j = U_1, m = 3$

smaller than η^0 (see Figure 5.13(b)). For that reason we choose m as the solution of the optimization problem

$$\begin{array}{c} \min\ m \\ \left(\frac{L_{j_1}-l_{j_1}}{m}\right)^2 + \left(\frac{L_{j_2}-l_{j_2}}{m}\right)^2 < \eta^0 \\ m \in \mathbb{N}\,,\, m \geq 2 \end{array} \quad .$$

Selecting m in this way we know that at most one member $x_k^\star$ of an optimal solution $(x_1^\star, \ldots, x_n^\star)^T \in U^n$ of Problem (PP) can belong to one of these squares $\bar{U}_l$ ($l \in \{1, \ldots, m^2\}$).

REMARK 5.5.1. In our numerical tests it was sufficient to choose $m = 2$ for $n \leq 13$ and $m = 3$ for $n \leq 27$.

In deeper levels, i.e., if R_j has a maximal edge-length smaller than or equal to $\frac{0.5}{m}$, we subdivide R_j again into four rectangles with equal size by bisecting the edges of this rectangle.

REMARK 5.5.2. As we will see in Section 5.6, it is possible that R_j shrinks to an interval, i.e., to a one-dimensional rectangle. In these cases we simply split R_j by halving this interval.

The reason for choosing a partition consisting of more than four squares in the second level has a heuristical nature. Our numerical tests showed that this strategy – in connection with the following special features and the possible reductions of the size of relevant rectangles discussed in Section 5.6 – has a much better running-time performance than the simpler strategy, where R_j is always divided into four squares.

For the implementation of the following special features it is essential that at least in the first level we use squares as partition sets instead of two-dimensional rectangles with different edge-length, as it is done in [DGPWM91].

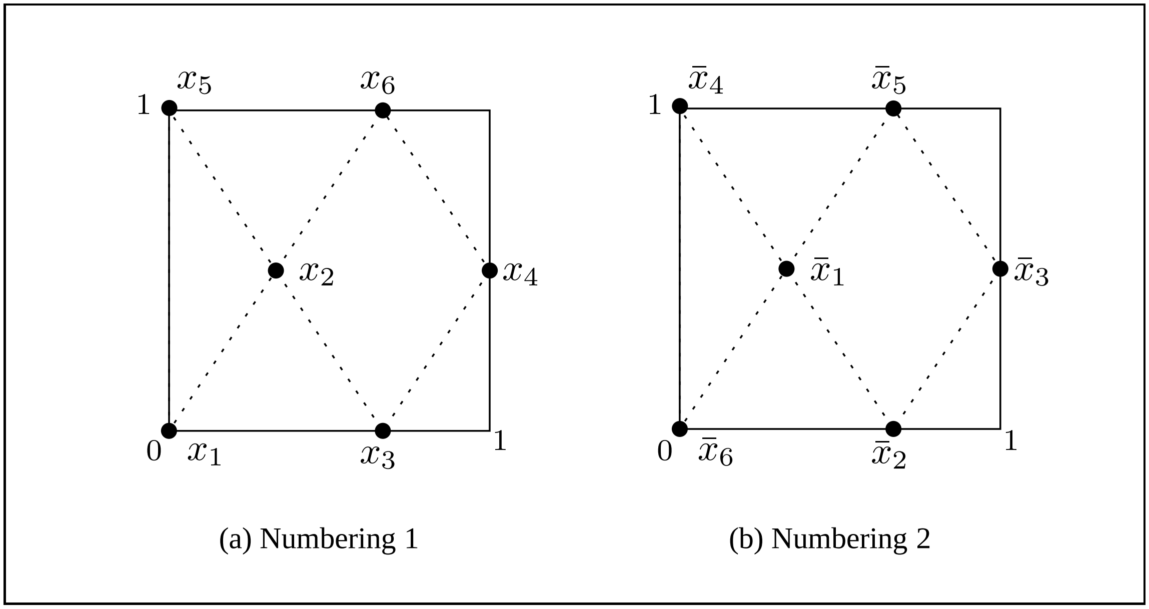
5.5.2. Special Features. Because of the special structure of Problem (PP) there are many optimal solutions differing only by the numbering of their members or differing by a rotation or a reflection.

Consider the case $n = 6$. An optimal solution of Problem (PP) is given by

$$\begin{gathered} x_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, x_2 = \begin{pmatrix} \frac{1}{3} \\ \frac{1}{2} \end{pmatrix}, x_3 = \begin{pmatrix} \frac{2}{3} \\ 0 \end{pmatrix}, x_4 = \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix}, \\ x_5 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, x_6 = \begin{pmatrix} \frac{2}{3} \\ 1 \end{pmatrix}, t = \frac{13}{36} \end{gathered} \tag{5.5.2}$$

(see Figure 5.14(a)). Setting, for each $i \in \{1, \ldots, 5\}$, $\bar{x}_i := x_{i+1}$ and $\bar{x}_6 := x_1$

FIGURE 5.14. Same solutions with different numbering



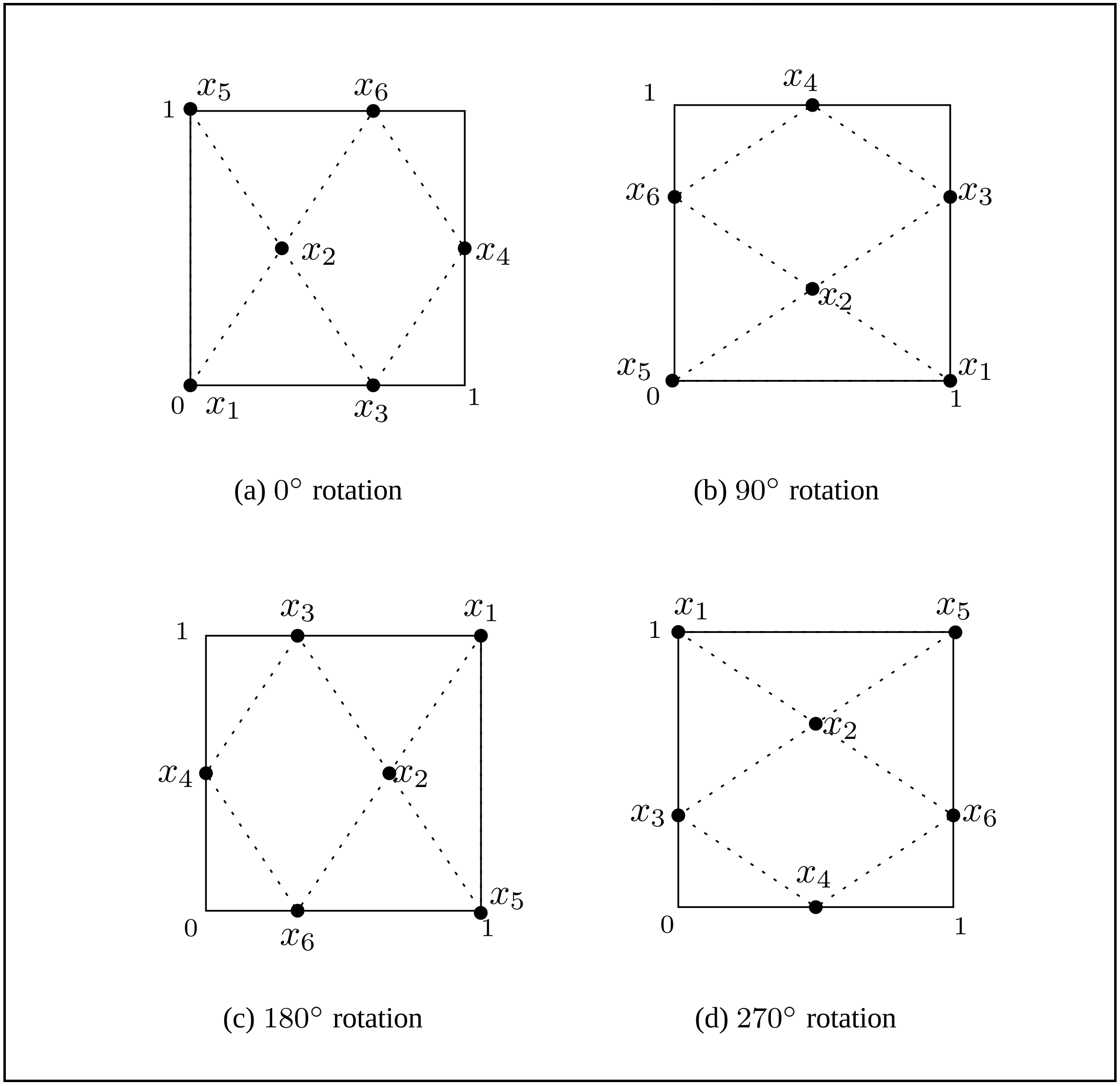
(a) Numbering 1

(b) Numbering 2

we obtain the "same" optimal solution (see Figure 5.14(b)). The members x_i of an optimal solution are permutable. However, for the solution of the point scattering problem it is sufficient if we detect one of these $n!$ optimal solutions. Hence, we need a subdivision strategy, which guarantees a ***unique numbering***.

In order to illustrate the problem of possible rotations and reflections consider again the case $n = 6$. There are 8 possible symmetric arrangements of an optimal solution $(x_1, \ldots, x_n)^T \in U^n$ in the unit square U. For the optimal solution given in (5.5.2) (see Figure 5.15(a)), there are three possible rotations (Figures 5.15(b)-5.15(d)) and four reflections (Figures 5.16(a)-5.16(d)). For Algorithm 5.1 it would be sufficient, if only one of these possibilities is considered. Therefore, we need also a strategy, which avoids that Algorithm 5.1 looks for more than one of these symmetric solutions. Note that these reflections and rotations are the results of orthogonal transformations, which do not change the Euclidean distances between the members of a point $x \in U^n$.

FIGURE 5.15. Solutions differing by rotation



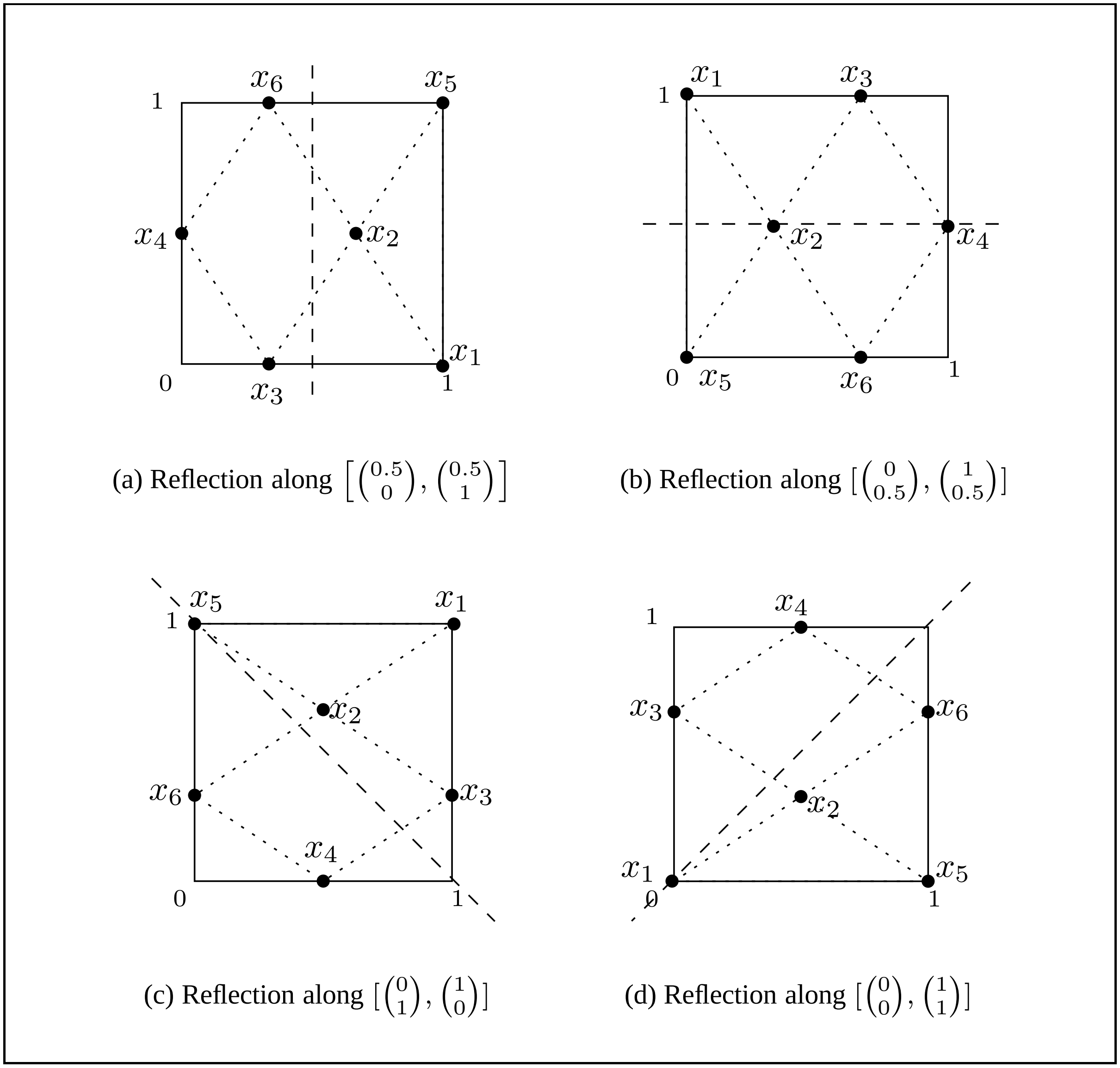
(a) 0° rotation

(b) 90° rotation

(c) 180° rotation

(d) 270° rotation

FIGURE 5.16. Solutions differing by reflection



(a) Reflection along $\left[\binom{0.5}{0}, \binom{0.5}{1}\right]$

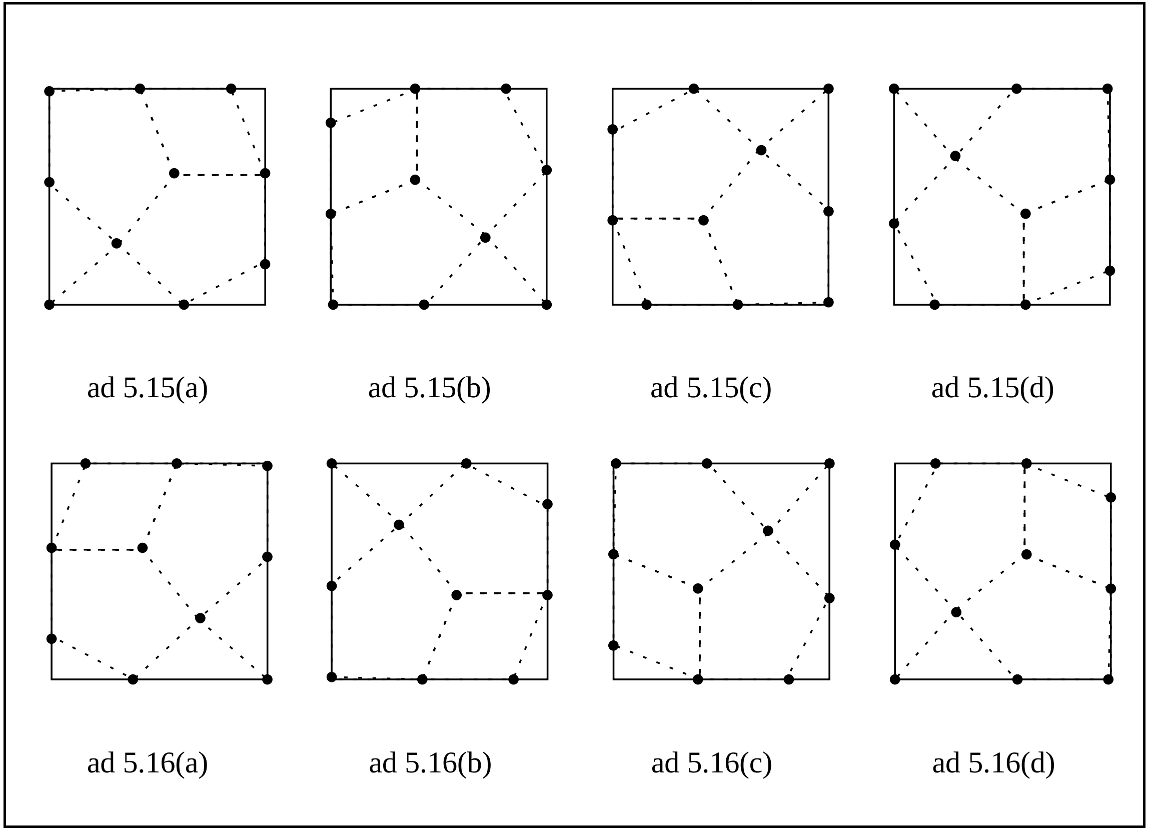
(b) Reflection along $[\binom{0}{0.5}, \binom{1}{0.5}]$

(c) Reflection along $[\binom{0}{1}, \binom{1}{0}]$

(d) Reflection along $[\binom{0}{0}, \binom{1}{1}]$

REMARK 5.5.3. The arrangements displayed in Figure 5.15 and in Figure 5.16 differ only by the numbering of the members of the solution $(x_1, \dots, x_n)^T$ (compare, e.g., Figures 5.15(a) and 5.16(b)). Consequently, one could assume that a *unique numbering* strategy also reduces the number of possible symmetric arrangements, and it would be thus not necessary to consider all displayed cases in a ***symmetry avoiding strategy***. However, the fact that the reflections lead only to a different numbering in comparison with the rotations depends on the special structure of the considered solution for $n = 6$. This solution is symmetric itself. If an optimal solution for Problem (PP) is not symmetric, as it is for example the case for $n = 10$ (see Figure 5.17), then there exist 8 completely different arrangements.

FIGURE 5.17. A solution of Problem (PP) for $n = 10$



In the following we would like to sketch how we obtain the required unique numbering and how we try to avoid the appearance of symmetric solutions. We stress that the elimination of redundant solutions is crucial for the efficiency of Algorithm 5.1 (see also Remark 5.6.1 in the next section). The algorithm necessarily refines the branch-and-bound tree near all optimal solutions that have not been identified as "simple modifications" of each other. Thus the amount of time saved by eliminating $k-1$ of k solutions is nearly a factor of k.

Unique Numbering. In order to describe the simple idea of this special strategy let us assume that a hyperrectangle

$$R = R_1 \times \cdots \times R_n$$

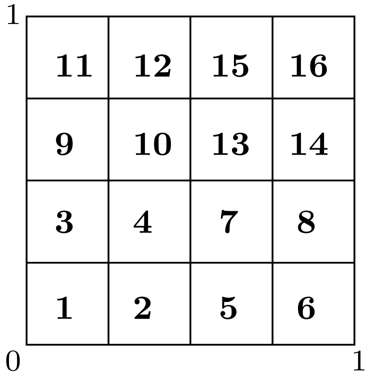
is given such that, for each $i \in \{1, \ldots, n\}$, the rectangle R_i is subdivided twice, i.e., R_i is the result of a subdivision of one of the four squares U_i ($i = 1, \ldots, 4$) (see (5.5.1)). Taking the basic strategy into account it follows that, for each rectangle R_i ($i \in \{1, \ldots, n\}$), there exists a unique square $\bar{U}_i \subset U$ with edge length $\frac{0.5}{m}$ satisfying $R_i \subset \bar{U}_i$. If we identify each of the $4m^2$ possible squares $\bar{U}$ with a unique number $no(\bar{U})$ and require that, for any $i = 1, \ldots , n-1$, there holds

$$no(\bar{U}_i) \; < \; no(\bar{U}_{i+1}) \, , \tag{5.5.3}$$

then we are able to guarantee a unique numbering of the rectangles forming the hyperrectangle R.

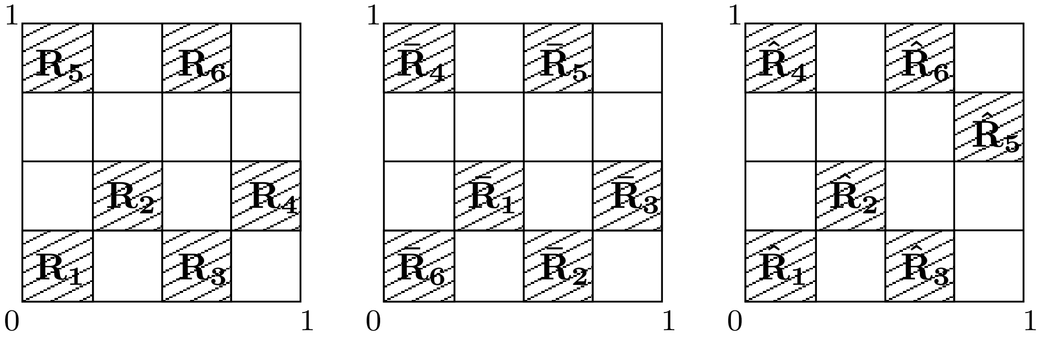
Let us illustrate this approach with an example. Consider again the case $n = 6$ and assume that m is chosen as 2. In this situation there are 16 possible squares $\bar{U}$. Numbering these possibilities as shown in Figure 5.18 and requiring that (5.5.3) is true, we see that solution $\bar{x}$ displayed in Figure 5.14(b) is not possible. Indeed, $\bar{x}$ is a member of the hyperrectangle $\bar{R} = \bar{R}_1 \times \cdots \times \bar{R}_6$ given in Figure 5.19(b) with the property $no(\bar{U}_6) < no(\bar{U}_1)$. This violates Condition (5.5.3). On the other hand, a numbering of solution x as in Figure 5.14(a) is allowed, since this point is contained in the hyperrectangle $R = R_1 \times \cdots \times R_6$ shown in Figure 5.19(a).

FIGURE 5.18. Unique numbering



11	**12**	**15**	**16**
9	**10**	**13**	**14**
3	**4**	**7**	**8**
1	**2**	**5**	**6**

FIGURE 5.19. Different numberings



(a) Possible

(b) Impossible

(c) Possible

As mentioned before, this strategy guarantees a unique numbering of the rectangles R_i forming the considered hyperrectangle R. Unfortunately, this strategy is not able to ensure a unique numbering of an optimal solution, i.e., Algorithm 5.1 using this strategy can still look for several solutions of Problem (PP) differing only by the numbering of their members. Indeed, with respect to the basic strategy we know that each square $\bar{U}$ resulting from the second partition of U contains at most one member of an optimal solution. However, if a member of an optimal solution belongs to the boundary of such a square $\bar{U}$, this set is not unique, as it is the case for $x_i = \binom{0.3\bar{3}}{0.5}$ or $x_j = \binom{1.0}{0.5}$ in our present example. In such a situation our strategy does not guarantee a unique numbering of the optimal solution. The numbering of the hyperrectangle $\hat{R} = \hat{R}_1 \times \cdots \times \hat{R}_6$ given in Figure 5.19(c) fulfills also Property (5.5.3). Hence, Algorithm 5.1 – even using this unique numbering strategy – has to detect an optimal solution in hyperrectangle R (Figure 5.19(a)) as well as in hyperrectangle $\hat{R}$ (Figure 5.19(c)). Nevertheless, this special feature strongly reduces the necessary effort for solving Problem (PP). Note that we eliminate till the second partitioning level $n! - 1$ of $n!$ possible hyperrectangles.

Symmetry Avoiding Strategy. Let

$$\begin{aligned}\mathcal{R} \;=\; \{R = R_1 \times \cdots \times R_n \subset U^{2n},& \\ & R_i = [l_{i_1}, L_{i_1}] \times [l_{i_2}, L_{i_2}] \subset U,\, i = 1, \ldots, n\}\end{aligned}$$

be the set of all possible $2n$-dimensional hyperrectangles. The symmetries resulting from the relevant rotations and reflections (see again Figure 5.15 and Figure 5.16) can be interpreted as an equivalence relation $\sim$ on the set $\mathcal{R}$.

$$\boxed{R, Q \in \mathcal{R} : R \sim Q \iff \begin{array}{l}\text{R is the result of one of the three possible}\\ \text{rotations of Q or the result of one of the}\\ \text{four possible reflections of Q}\end{array}}$$

The equivalence relation $\sim$ divides $\mathcal{R}$ into equivalence classes $\mathcal{R}_\iota$ ($\iota \in I$, I index set), i.e.,

- $\mathcal{R}_\iota \subset \mathcal{R}, \forall \iota \in I$
- $R \sim Q, \forall R, Q \in \mathcal{R}_\iota, \iota \in I$
- $\mathcal{R}_\iota \cap \mathcal{R}_\kappa = \emptyset, \forall \iota, \kappa \in I, \iota \neq \kappa$
- $\bigcup_{\iota \in I} \mathcal{R}_\iota = \mathcal{R}$.

Obviously, it is sufficient for a correct functioning of Algorithm 5.1 if this method considers only one member of the equivalence classes $\mathcal{R}_\iota$, which are relevant during the execution of our approach. We developed a strategy able to decide whether a given hyperrectangle $R = R_1 \times \cdots \times R_n$ in a node of the branch-and-bound tree is a *special* representative of an equivalence class or not. Let us shortly sketch the basic ideas of this strategy.

Assume at first that we are in a situation such that the current hyperrectangle R is given as the Cartesian product of the squares U_i ($i \in \{1, \ldots, 4\}$) (see (5.5.1)), i.e., each member R_i of R has been subdivided once. If the previously described unique numbering strategy is applied, then we know that there are three integers i_1, i_2, $i_3 \in \{1, \ldots, n+1\}$ ($i_1 \le i_2 \le i_3$) satisfying

$$\begin{aligned} R_i &= U_1 \qquad i = 1, \ldots, i_1 - 1\,, \\ R_i &= U_2 \qquad i = i_1, \ldots, i_2 - 1\,, \\ R_i &= U_3 \qquad i = i_2, \ldots, i_3 - 1 \end{aligned}$$

and

$$R_i = U_4 \qquad i = i_3, \ldots, n\,.$$

Note that the unique numbering strategy can also be applied for hyperrectangles with this structure, even though we described the ideas of this method under the assumption that each member of the considered hyperrectangle is subdivided twice. Denote by

$$C1 := i_1 - 1\ ,\ C2 := i_2 - i_1\ ,\ C3 := i_3 - i_2\ ,\ C4 := n + 1 - i_3$$

the number of members R_j ($j \in \{1, \ldots, n\}$) of R, which are equal to U_i ($i \in \{1, \ldots, 4\}$). In order to avoid that Algorithm 5.1 considers more than one representative of the equivalence class of the set $\mathcal{R}$ containing the hyperrectangle R we require that these counters $C1$, $C2$, $C3$ and $C4$ fulfill special ordering conditions.

Ordering conditions for the first level

$$\boxed{\begin{array}{ll} C1 \ \ge\ \max\{C1, C2, C3\} & \text{(OC1)} \\ C2 \ \ge\ C3 & \text{(OC2)} \\ \textbf{If } C1 \ =\ C2 \textbf{ Then } C3 \ \ge\ C4 & \text{(OC3)} \\ \textbf{If } C1 \ =\ C3 \textbf{ Then } C2 \ \ge\ C4 & \text{(OC4)} \end{array}}$$

If a hyperrectangle $R = R_1 \times \ldots \times R_n$ does not fulfill (OC1), it is possible to rotate the rectangles forming this set such that (OC1) is satisfied. Condition (OC2) can be reached by a reflection along the line $\left[\binom{0}{0}, \binom{1}{1}\right]$ and Condition (OC3) is yielded by a reflection along $\left[\binom{0.5}{0}, \binom{0.5}{1}\right]$. If Condition (OC4) is not satisfied we can reflect the members of R along the line $\left[\binom{0}{0.5}, \binom{1}{0.5}\right]$ and obtain an element of the same equivalence class fulfilling this condition. Note that all Conditions (OC1)-(OC4) are satisfiable simultaneously. The examination of the reflection along the line $\left[\binom{1}{0}, \binom{0}{1}\right]$ in the first level does not lead to another ordering condition. We could only require $C1 \geq C4$. However, this is fulfilled regarding Condition (OC1) and thus unnecessary.

Let us illustrate these conditions for the case $n = 6$. As mentioned before, we would like to avoid that Algorithm 5.1 tries to determine the solutions displayed in Figures 5.15(b)-5.15(d). In order to detect these solutions Algorithm 5.1 has to generate the hyperrectangles

$$\begin{aligned} R^1 &= U_1 \times U_1 \times U_2 \times U_2 \times U_3 \times U_4 \,, \\ R^2 &= U_1 \times U_2 \times U_2 \times U_3 \times U_4 \times U_4 \,, \\ R^3 &= U_1 \times U_2 \times U_3 \times U_3 \times U_4 \times U_4 \,, \\ R^4 &= U_1 \times U_1 \times U_2 \times U_3 \times U_3 \times U_4 \,, \\ R^5 &= U_1 \times U_2 \times U_2 \times U_3 \times U_3 \times U_4 \end{aligned}$$

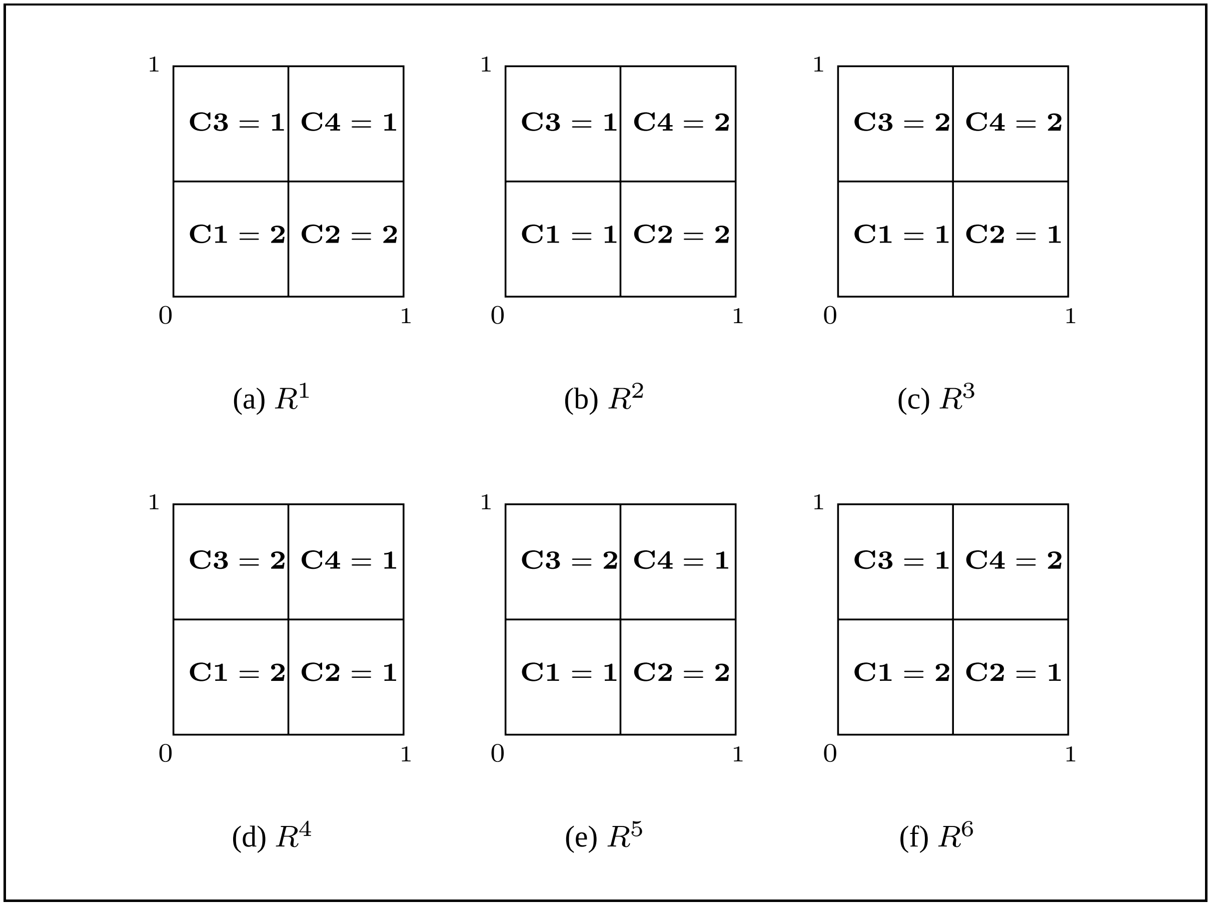
and

$$R^6 = U_1 \times U_1 \times U_2 \times U_3 \times U_4 \times U_4 \,.$$

(see Figure 5.20). Only the hyperrectangles R^1 and R^6 satisfy (OC1)-(OC4) simultaneously. Indeed, R^2, R^3 and R^5 violate (OC1) and R^4 does not fulfill (OC2). Hence, Algorithm 5.1 eliminates these four sets from further considerations. This means that only one-third of the hyperrectangles containing optimal solutions have to be analyzed further. However, because of the special structure of the solution for $n = 6$ – x_2 and x_4 does not belong to a unique square U_i $(i \in \{1, \ldots, 4\})$ – the remaining hyperrectangles R^1 and R^6 still contain the four possible symmetric arrangements of x. The solutions given in Figures 5.15(a) and 5.15(c) lie in R^1 and all solutions are located within R^6. For that reason we have to examine the symmetry structure of the relevant hyperrectangles also in the second level.

Assume now that the current hyperrectangle R is given by the Cartesian product of n rectangles with edge-length $\frac{0.5}{m}$, i.e., each member of R is subdivided

FIGURE 5.20. Hyperrectangles containing an optimal solution



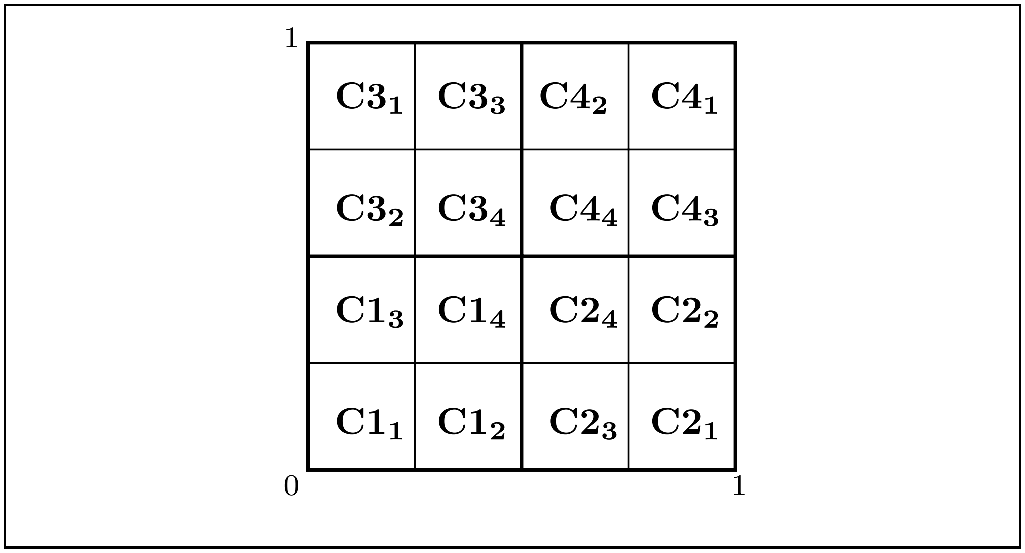
(a) R^1 (b) R^2 (c) R^3

(d) R^4 (e) R^5 (f) R^6

twice. According to the choice of m we know that each rectangle R_i ($i \in \{1, \ldots, n\}$) forming R contains at most one member of an optimal solution of Problem (PP) and hence the rectangles R_i ($i \in \{1, \ldots, n\}$ must be different from each other. Moreover, we know that R is a child of a hyperrectangle $\hat{R}$, which has satisfied all conditions in the first level and we know which type of symmetry are still possible, i.e., with respect to which type of symmetry is $\hat{R}$ invariant. For example, in the case $n = 6$ there holds that for R^1 and hence for all its children only the reflection along the line $\left[\binom{0.5}{0}, \binom{0.5}{1}\right]$ can be considered. The other possible hyperrectangle R^6 is invariant with respect to $180°$ rotations as well as with respect to reflections along $\left[\binom{0}{0}, \binom{1}{1}\right]$ and along $\left[\binom{0}{1}, \binom{1}{0}\right]$.

In order to avoid that Algorithm 5.1 examines more than one representative of the equivalence classes containing hyperrectangles with the described structure we assign to each square U_i ($i \in \{1, \ldots, 4\}$) a number, as we did in the first level. However, this number is not a pure integer anymore. We use binary representations of integer values with a length of m^2. An element of the m^2-dimensional binary vector corresponds to one of the m^2 possible subsquares obtained by partitioning U_i ($i \in \{1, \dots, 4\}$) according to the basic strategy. Such an element is set to 1, if the corresponding square is a member of R, and 0 otherwise. Consider the case $m = 2$. We have 4 possible subsquares of edge-length 0.25 for each square U_i ($i \in \{1, \dots, 4\}$). In order to examine rotation symmetries we define the vectors $C1, C2, C3, C4 \in \{0, 1\}^4$ as shown in Figure 5.21. As in the first level we require that the integers given by the binary vectors $C1$, $C2$, $C3$ and $C4$ fulfill special ordering conditions – like $C1 \geq \max\{C2, C3, C4\}$ (compare with Condition (OC1) for the first level). Note that the numbering of the vectors is chosen such that they are invariant with respect to the rotations. The same has to be done, if we examine possible reflections. In these cases we choose for $m = 2$ the numberings given in Figures 5.22(a)-5.22(d). The resulting integers are also checked, whether they fulfill special ordering conditions.

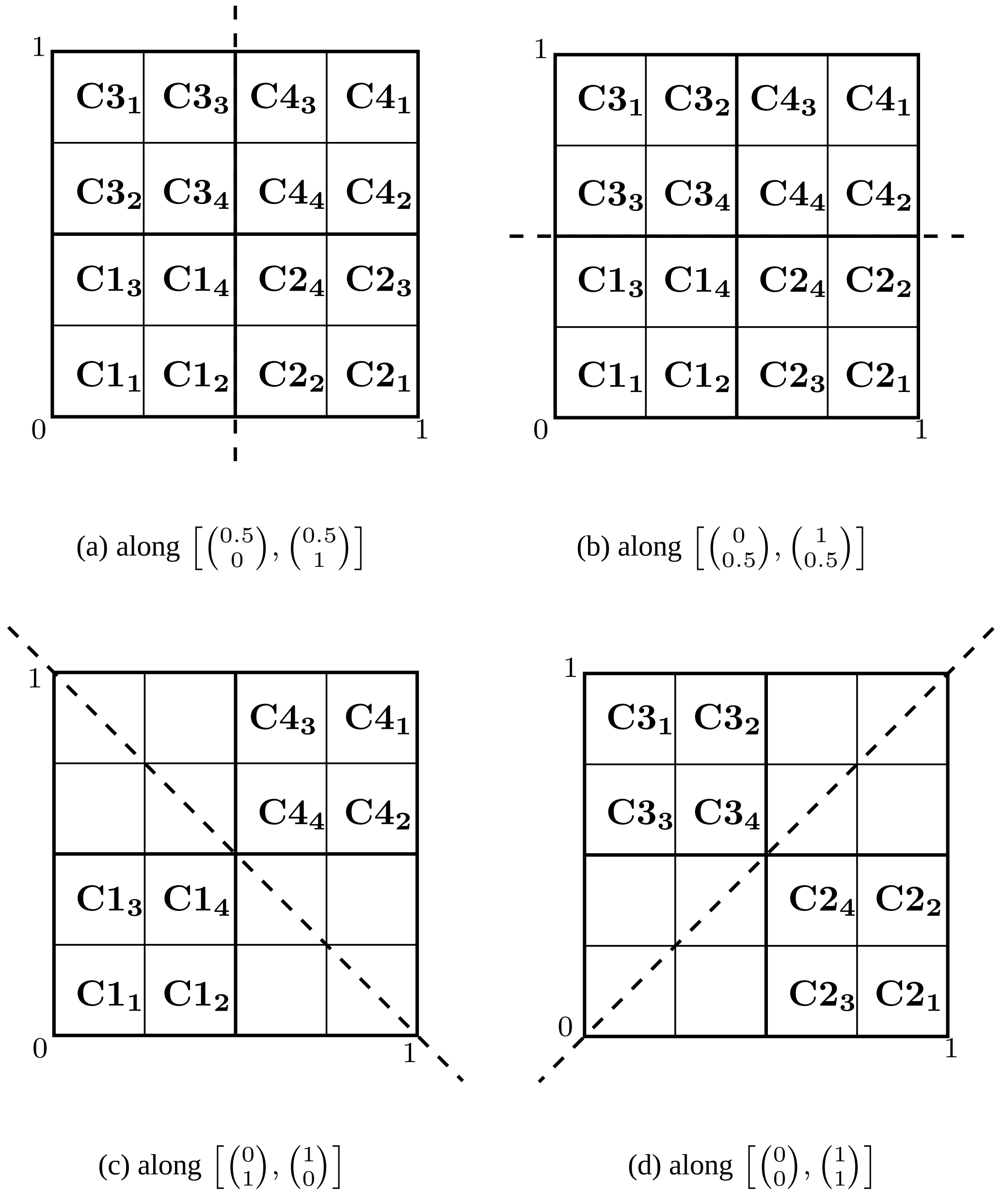
FIGURE 5.21. Numbering of $C1, C2, C3, C4$ for rotation symmetries and $m = 2$



$\mathbf{C3_1}$	$\mathbf{C3_3}$	$\mathbf{C4_2}$	$\mathbf{C4_1}$
$\mathbf{C3_2}$	$\mathbf{C3_4}$	$\mathbf{C4_4}$	$\mathbf{C4_3}$
$\mathbf{C1_3}$	$\mathbf{C1_4}$	$\mathbf{C2_4}$	$\mathbf{C2_2}$
$\mathbf{C1_1}$	$\mathbf{C1_2}$	$\mathbf{C2_3}$	$\mathbf{C2_1}$

It is essential to note that in the cases where more than one type of symmetry can be examined we have to pay attention that the applied conditions are consistent. Recognize that in contrast to the first level the integers used here for checking the ordering conditions can vary for different symmetries. This means that we have to guarantee that all conditions, which we require, can be fulfilled simultaneously by at least one element of each relevant equivalence class of $\mathcal{R}$. This problem leads to a distinction of many cases in order to formulate these conditions. Therefore, we abandon an explicit formulation of the used conditions in the present work.

FIGURE 5.22. Numbering of $C1, C2, C3, C4$ for reflection symmetries and $m = 2$


1
$\mathbf{C3_1}$ $\mathbf{C3_3}$ $\mathbf{C4_3}$ $\mathbf{C4_1}$
$\mathbf{C3_2}$ $\mathbf{C3_4}$ $\mathbf{C4_4}$ $\mathbf{C4_2}$
$\mathbf{C1_3}$ $\mathbf{C1_4}$ $\mathbf{C2_4}$ $\mathbf{C2_3}$
$\mathbf{C1_1}$ $\mathbf{C1_2}$ $\mathbf{C2_2}$ $\mathbf{C2_1}$
0
1
1
$\mathbf{C3_1}$ $\mathbf{C3_2}$ $\mathbf{C4_3}$ $\mathbf{C4_1}$
$\mathbf{C3_3}$ $\mathbf{C3_4}$ $\mathbf{C4_4}$ $\mathbf{C4_2}$
$\mathbf{C1_3}$ $\mathbf{C1_4}$ $\mathbf{C2_4}$ $\mathbf{C2_2}$
$\mathbf{C1_1}$ $\mathbf{C1_2}$ $\mathbf{C2_3}$ $\mathbf{C2_1}$
0
1
1
$\mathbf{C4_3}$ $\mathbf{C4_1}$
$\mathbf{C4_4}$ $\mathbf{C4_2}$
$\mathbf{C1_3}$ $\mathbf{C1_4}$
$\mathbf{C1_1}$ $\mathbf{C1_2}$
0
1
1
$\mathbf{C3_1}$ $\mathbf{C3_2}$
$\mathbf{C3_3}$ $\mathbf{C3_4}$
$\mathbf{C2_4}$ $\mathbf{C2_2}$
$\mathbf{C2_3}$ $\mathbf{C2_1}$
0
1

(a) along $\left[\binom{0.5}{0}, \binom{0.5}{1}\right]$

(b) along $\left[\binom{0}{0.5}, \binom{1}{0.5}\right]$

(c) along $\left[\binom{0}{1}, \binom{1}{0}\right]$

(d) along $\left[\binom{0}{0}, \binom{1}{1}\right]$

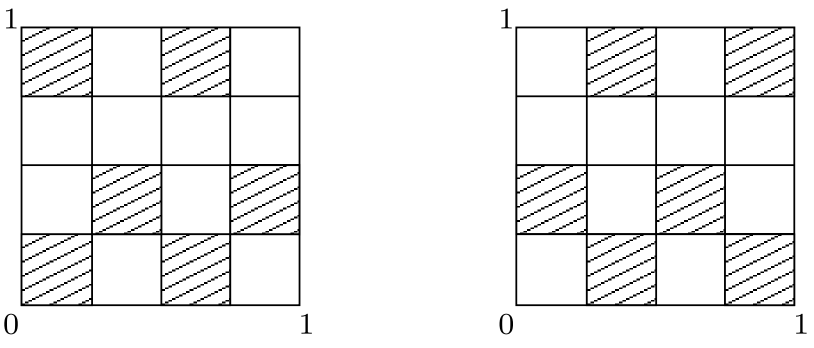
REMARK 5.5.4.

(a) If m is different from 2, as it is the case for numerically interesting numbers of points (see Remark 5.5.1), we have to adjust the numbering of the m^2-dimensional binary vectors $C1$, $C2$, $C3$ and $C4$ according to the previous ideas. Moreover, the *consistent* ordering conditions, which we require to be fulfilled, have also to be adapted in these cases.

(b) If the considered hyperrectangle is a child of a set invariant with respect to reflections along the line $\left[\binom{1}{0}, \binom{0}{1}\right]$ or along $\left[\binom{0}{0}, \binom{1}{1}\right]$, it is also possible to formulate additional conditions considering the subsquares of U_2 and U_3 or of U_1 and U_4, respectively, which are not crossed by the reflection line (see the not-numbered subsquares in Figure 5.22(c) or in Figure 5.22(d)).

Let us finally illustrate the described symmetry avoiding strategy in the second level for our example $n = 6$. The hyperrectangles R^{11} and R^{12} given in Figure 5.23 are children of R^1 and hence possible during the execution of Algorithm 5.1.

FIGURE 5.23. Possible children of R^1 for $n = 6$



(a) R^{11} (b) R^{12}

As we have pointed out before, the hyperrectangle R^1 is only invariant with respect to reflections along the line $\left[\binom{0.5}{0}, \binom{0.5}{1}\right]$. According to the numbering of the 4-dimensional binary vectors given in Figure 5.22(a) we obtain:

$$R^{11}: \quad \begin{aligned} C1 &= (1,0,0,1)^T \\ C2 &= (0,1,1,0)^T \\ C3 &= (1,0,0,0)^T \\ C4 &= (0,0,1,0)^T \end{aligned} \qquad R^{12}: \quad \begin{aligned} C1 &= (0,1,1,0)^T \\ C2 &= (1,0,0,1)^T \\ C3 &= (0,0,1,0)^T \\ C4 &= (1,0,0,0)^T \end{aligned}$$

If we use the condition

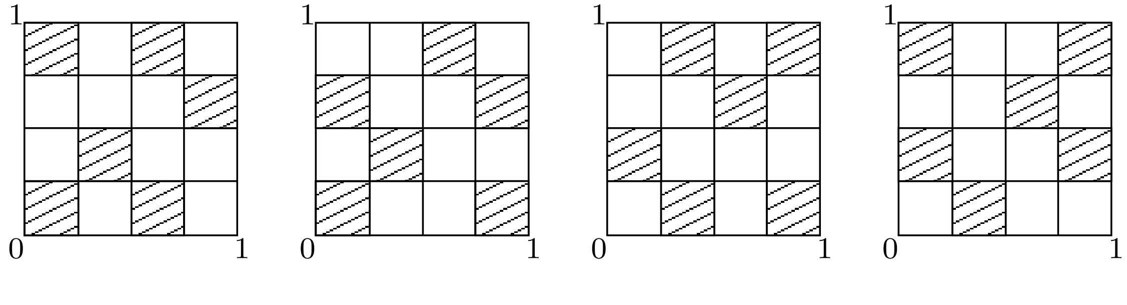
$$C1 \geq C2 \,,$$

we see that R^{11} satisfies this condition and R^{12} violates it. Hence R^{12} is eliminated from further considerations and we obtain that the optimal solution given in Figure

5.15(c), which was still contained in R^1, is with respect to the children of this hyperrectangle no longer possible.

The hyperrectangle R^6 , which fulfills also the ordering conditions in the first level, can lead to the four hyperrectangles given in Figure 5.24 containing the four possible arrangements of the optimal solutions shown in Figure 5.15. We know that R^6 is invariant with respect to $180°$ rotations. Using the numbering of the binary vectors $C1$, $C2$, $C3$ and $C4$ given in Figure 5.21 and requiring $C1 \geq C4$ – as we did in the first level – we obtain that R^{63} and R^{64} violates this condition. If we consider the reflection along the line $\left[\binom{0}{0}, \binom{1}{1}\right]$ and require again $C2 \geq C3$ (compare with (OC2)), where $C2$ and $C3$ are now numbered as in Figure 5.22(d), it follows that R^{61} does not fulfill this condition. Thus using our ordering conditions, only the hyperrectangle R^{62} remains.

FIGURE 5.24. Possible children of R^6 for $n = 6$



(a) R^{61} (b) R^{62} (c) R^{63} (d) R^{64}

Applying the described symmetry avoiding strategy in the second level it follows again that two-third of the hyperrectangles containing symmetric solutions are eliminated from further considerations. Unfortunately, there are still two symmetric solutions (see Figure 5.15(a) and Figure 5.15(b)), which have to be detected by Algorithm 5.1, since R^{11} and R^{62} fulfill all ordering conditions and contain both arrangements of x. The reason for this fact is again the special structure of the solution x for $n = 6$ given in (5.5.2). As in the first level, there are two different equivalence classes of $\mathcal{R}$, whose members contain symmetric arrangements of x.

Hence our symmetry avoiding strategy is not able to fully avoid that Algorithm 5.1 has to detect different symmetric solutions. We can only guarantee that Algorithm 5.1 does not look for optimal solutions in more than one representative of an equivalence class of $\mathcal{R}$. This is the best we can obtain and – as it was the case for the unique numbering strategy – we have seen in our example that the use of the suggested symmetry avoiding strategy reduces significantly the effort for solving Problem (PP) with Algorithm 5.1.

If a hyperrectangle R fulfilling all required conditions in the second level is still invariant with respect to some types of symmetry, we could examine these symmetries also in deeper levels. However, our numerical experience showed that this effort does not lead to an improvement of the numerical performance of Algorithm 5.1 – at least as long as the size reduction strategies introduced in the next section are used.

Besides these two special features, which do not depend on the current upper and lower bounds, we also use a third idea in order to reduce the effort for solving Problem (PP). This idea exploits explicitly the knowledge of the current best known value η.

Using the Current Lower Bound $\boldsymbol{\eta}$. If we assume that for the cases $2 \leq l < n$ upper bounds $\mu(l)$ for the optimal solution value $t^\star(l)$ are known, then it is possible to further reduce the number of subdivision sets $R = R_1 \times \cdots \times R_n$, which are relevant during the execution of Algorithm 5.1. Note that the presented approach can deliver the necessary upper bounds $\mu(l)$.

For a given hyperrectangle $R = R_1 \times \cdots \times R_n \subset U^{2n}$ and for $2 \leq l < n$, let

$$\mathcal{R}_l \;=\; \{\{R_i : i \in I\} \text{ with } I \subset \{1, \ldots, n\}, |I| = l\}$$

be the set of all subsets of $\{R_i, i = 1, \ldots, n\}$ with cardinality l. Choose, for $l < n$, a set $Q \in \mathcal{R}_l$ and let

$$\bar{Q} \;=\; [l_1, L_1] \times [l_2, L_2]$$

be the smallest rectangle containing all elements of Q. Since there holds $\mu(l) \geq t^\star(l)$ it is obvious that the maximal minimum pairwise squared distance of l points lying inside a square with edge length d is not greater than $\mu(l)d^2$.

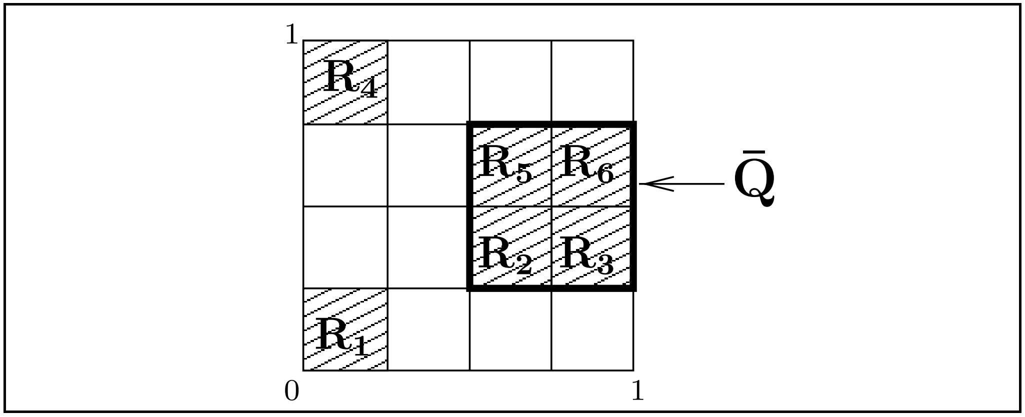
If there holds

$$\mu(l)(\max\{L_1 - l_1, L_2 - l_2\})^2 \;<\; \eta\,, \tag{5.5.4}$$

it is not possible that l points lie inside $\bar{Q}$ with a minimum squared distance bigger than or equal to η. Thus it is not necessary to consider R further, since R cannot contain a point $x \in \mathbb{R}^{2n}$ with a better distance behavior than the current best known point.

Let us illustrate this method with an example. Consider again the case $n = 6$ and assume that the current rectangle $R = R_1 \times \cdots \times R_6$ has the structure given in Figure 5.25. We know that $\mu(4) = 1 = t^\star(4)$. Hence we can derive that the maximal minimum pairwise squared distance of four points lying inside the square

FIGURE 5.25. Eliminable case



$$\bar{Q} = [0.5, 1.0] \times [0.25, 0.75] \;=\; R_2 \cup R_3 \cup R_5 \cup R_6$$

is equal to $0.25 = \mu(4)0.5^2$. If a current best known value η greater than 0.25 is given, we are able to eliminate R from the set of relevant hyperrectangles without losing an optimal solution of Problem (PP).

REMARK 5.5.5.

(a) In Algorithm 5.1 (see Section 5.3) we use the subdivision strategies described so far in the following way. At first we partition the current rectangle R_j in 4 or m^2 rectangles $R_j^1, \cdots, R_j^{\bar{l}}$ ($\bar{l} \in \{4, m^2\}$) with equal size following the basic strategy. After this we use the special features in order to test whether it is possible to eliminate some of the resulting hyperrectangles

$$R^i \;=\; R_1 \times \cdots \times R_{j-1} \times R_j^i \times R_{j+1} \times \cdots \times R_n \;, i = 1, \ldots, \bar{l} \,.$$

In this way we obtain l ($0 \leq l \leq \bar{l}$) hyperrectangles, which have to be analyzed further.

(b) Note that in the execution of Algorithm 5.1 it is not necessary that all two-dimensional rectangles forming the hyperrectangle R have equal size, since we split in each iteration only one part R_j. For simplicity of presentation we have assumed in the description of the subdivision strategies in the present section that all rectangles $R_1, \ldots, R_n$ forming the current hyperrectangle

> have equal size (compare with the previous figures). In the implementation of Algorithm 5.1 we took the possibility of different sizes into account. It is possible to adapt all subdivision strategies mentioned before to the examination of hyperrectangles consisting of members without equal size. However, doing this we have to pay attention to the fact that the strategies used in our algorithm can interfere with each other. For instance, the conditions for the symmetry avoiding strategies have to recognize the unique numbering strategy. Therefore, we have to ensure that all conditions, which we require to be satisfied, work simultaneously.

In the description of Algorithm 5.1 in Section 5.3 we postulate that the subdivision strategies fulfill Conditions (C1) and (C3). Since our basic strategy generates more and smaller subsets of R than the bisection strategy would do, it follows immediately that our subdivision strategy is exhaustive, i.e., satisfies (C1). Note that the bisection of hyperrectangles is exhaustive.

In the discussion of the special features we have seen that we lose optimal solutions applying our strategies. Nevertheless, it is possible to implement these strategies such that we never lose all solutions. Therefore, our subdivision strategies are also consistent in the sense of Condition (C3).

The described special features reduce the effort for solving Problem (PP) by avoiding possible, but redundant partition sets in advance. Only the last idea takes advantage of the information generated by the algorithm itself. Exploiting these information in a stronger way it is possible to reduce the size of the hyperrectangles, which are not eliminated from consideration by these subdivision strategies. How we realize this is the content of the next section.

5.6. Size Reduction Strategies

Let $R = R_1 \times \ldots \times R_n \subset \mathbb{R}^{2n}$ be a hyperrectangle in an iteration of Algorithm 5.1 with $R_i = [l_{i_1}, L_{i_1}] \times [l_{i_2}, L_{i_2}] \subset U$ ($i \in \{1, \ldots, n\}$), and let $\eta > 0$ and $\mu \leq 2$ respectively be the current lower and upper bound. Assume that this hyperrectangle belongs to the sets remaining after the execution of the subdivision strategies in Step III. Note that we ignore the iteration index k_i ($i \in \{1, \ldots, l\}$, $k \in \mathbb{N}$) in order to reduce the number of necessary indices, as we did in the previous section.

In the formulation of Algorithm 5.1 (see, especially, Step IV) we claimed that is can be possible to diminish the size of the hyperrectangle R. In the present section we describe the strategy, which can lead to such a reduction of the size of the set R.

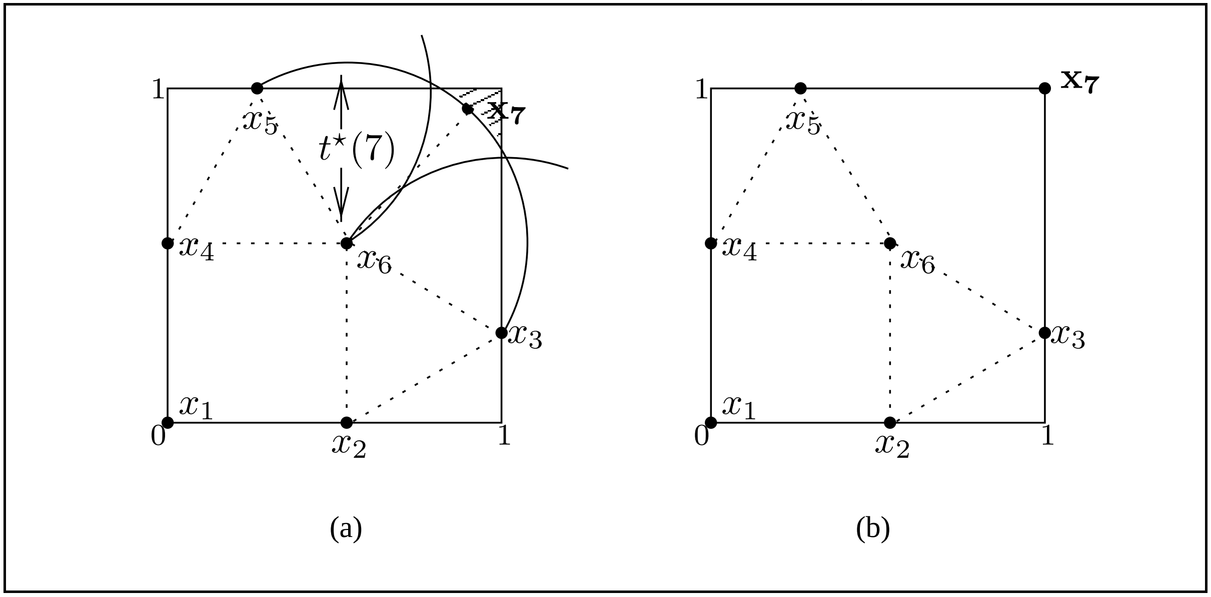
In the derivation of the theoretical results in Section 5.2 we saw that for Problem (PP) there always exist optimal solutions satisfying special properties. In particular, we know that there is an optimal solution $x^\star = (x_1^\star, \ldots, x_n^\star)^T \in U^n$ with optimal value $t^\star(n)$ fulfilling the properties

(P3): either a vertex v of the unit square U is a member of $x^\star$ itself, or there exist two members of $x^\star$ lying on the edge-lines of U forming the vertex v, which have exactly the optimal distance (Theorem 5.2.3 and Corollary 5.2.4), and

(P4): two consecutive members of $x^\star$ belonging to the same edge of U have a distance smaller than two times the optimal one (Theorem 5.2.6).

It is sufficient, if Algorithm 5.1 looks only for optimal solutions satisfying these properties. This means that we can interpret each point $x \in U^n$, which does not have these attributes, as an infeasible point for (PP). Doing this we can further reduce the number of possible optimal solutions of Problem (PP), as it was the case by applying the special features of the subdivision strategy developed in the previous section. Indeed, consider the case $n = 7$. One optimal solution is displayed in Figure 5.26(a). The point x_7 is not unique. We can choose each point in the shaded region without changing the minimum pairwise distance. However, only the solution shown in Figure 5.26(b) fulfills Property (P3).

FIGURE 5.26. Solutions for $n = 7$



(a) (b)

REMARK 5.6.1. Global optimization approaches, in particular branch-and-bound methods, generally have problems if many global optimal solutions exist. For example, branch-and-bound methods often have to strongly refine the subdivision sets in a neighborhood of an optimal solution in order to reduce the distance between the upper and lower bounds until the required tolerance is reached. Therefore, it is not surprising that the effort for solving a problem increases significantly if the number of global optimal solutions grows. On the other hand, we are satisfied if the solution method detects one global solution. Consequently, each strategy reducing the number of possible solutions of a problem can improve the numerical performance of a global optimization approach. However, such strategies cannot be derived in general. Nevertheless, exploiting the structure of special problem instances we can expect to obtain such *solution elimination strategies*. Note that the unique numbering strategy as well as the symmetry avoiding strategy introduced in the previous section and the strategies enforcing the satisfaction of Property (P3) and of Property (P4) discussed in the sequel can be interpreted as such solution elimination strategies.

In the subsequent two subsections we will see, how it is possible to enforce that a solution of Problem (PP) detected by Algorithm 5.1 has the required attributes and how this enforcement leads to a reduction of the size of R. Using Property (P3) we derive the so-called *corner rules*, which can result in a shrinkage of some rectangles R_i ($i \in \{1, \ldots, n\}$) – forming the hyperrectangle R – to an interval or even to a single point. Exploiting Property (P4) we obtain the so-called *edge rules*. In Subsection 5.6.2 we will see that the application of these rules can also reduce some rectangles to intervals. Hence the enforcement of Properties (P3) and (P4) lead to a reduction of the size of R through a reduction of the dimension of this set.

We complete the size reduction strategies with a third strategy reducing the volume of the hyperrectangle R. This strategy does not base on the properties of an optimal solution mentioned above. As the last special feature of our subdivision strategies, it uses the knowledge of the current best known value η. We will see that the use of this knowledge enables us to eliminate parts of R, which cannot contain a point $x \in U^n$ with a larger minimum pairwise distance than the best known so far.

5.6.1. Corner Rules. If we analyze the behavior of the given subdivision set $R = R_1 \times \ldots \times R_n$ in the neighborhood

$$S(v, \mu) \;=\; \{x \in U : \|x - v\|_2^2 \leq \mu\}$$

of the vertex v of the unit square U, then we recognize that there exist some situations allowing us to reduce the dimension of selected rectangles R_i ($i \in \{1, \ldots, n\}$). Remember that μ is an upper bound for the optimal value $t^\star(n)$ of Problem (PP).

Indeed, let $v \in \mathrm{I\!R}^2$ be an arbitrary vertex of the unit square U and denote by e_1 and e_2 the edge-lines of U forming this vertex, i.e., for $i \in \{1, 2\}$, there holds $e_i = \{x \in U : x_i = v_i\}$. Denote now by

$$\begin{aligned}\tilde{S}(v,\mu) \;=\; \{R_i : i \in \{1,\ldots,n\} \text{ with } R_i \cap S(v,\mu) \neq \emptyset \\ \text{and } \exists l \in \{1,2\} \text{ satisfying } R_i \cap e_l \neq \emptyset\}\end{aligned}$$

the set of all two-dimensional rectangles R_i ($i \in \{1, \ldots, n\}$) forming the hyperrectangle R, which have a non-empty intersection with $S(v, \mu)$ and which, additionally, touch the edge-line e_1 or the edge-line e_2 or both. Depending on the cardinality of $\tilde{S}(v, \mu)$ we distinguish four cases (compare $\tilde{S}(v, \mu)$ with $\bar{S}(v, t)$ used in the proof of Theorem 5.2.3 in Section 5.2).

Case 1: $|\tilde{S}(v,\mu)| = 0$

Since μ is an upper bound for the optimal solution $t^\star(n)$ of Problem (PP), it follows immediately that there does not exist a point $x \in R$ fulfilling Property (P3) at vertex v. Hence, it is not necessary to analyze R further, i.e., R can be pruned. Note that we interpret each point $x \in U^n$ without Properties (P3) and (P4) as infeasible, and with respect to this interpretation we know that in this case R contains only infeasible points.

Case 2: $|\tilde{S}(v,\mu)| = 1$

Assume, without loss of generality, that there holds $\tilde{S}(v, \mu) = \{R_1\}$. If $v \notin R_1$ (see Figure 5.27(a)), it follows by the same argumentation as in Case 1 that R is eliminable. Otherwise (see Figure 5.27(b)) we know that only points $x = (x_1, \ldots, x_n)^T \in R$ with $x_1 = v$ fulfill Property (P3). Note that (P3b) is not satisfiable in this case. Thus, we do not lose all optimal solutions of Problem (PP), if we set

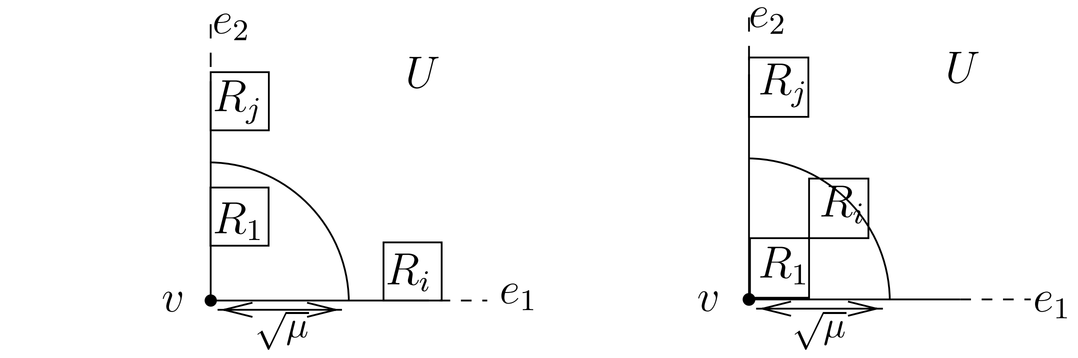
$$R = \bar{R}_1 \times R_2 \times \ldots \times R_n$$

with

$$\bar{R}_1 = \{v\} = [v_1, v_1] \times [v_2, v_2] \,,$$

i.e., if we shrink R_1 to a single point.

FIGURE 5.27. Corner rules (Case 2)



(a) Eliminable case (b) Adjustable case

Case 3: $|\tilde{S}(v,\mu)| = 2$

Assume again, without loss of generality, that $\tilde{S}(v,\mu)$ is equal to $\{R_1, R_2\}$. Depending on the location of v with respect to R_1 and R_2 we have to distinguish two further subcases.

Case 3.1: $v \notin R_1 \cup R_2$

It is clear that at vertex v Property (P3a) cannot be satisfied by an element x of R. If R_1 and R_2 touch the same edge-line e_i ($i \in \{1,2\}$) (see Figure 5.28(a)), we are able to eliminate R, since in this situation (P3b) is also not possible. Otherwise (see Figure 5.28(b)) we are able to replace R_1 and R_2 by intervals. Setting

$$\bar{R}_1 := R_1 \cap (e_1 \cup e_2)$$

and

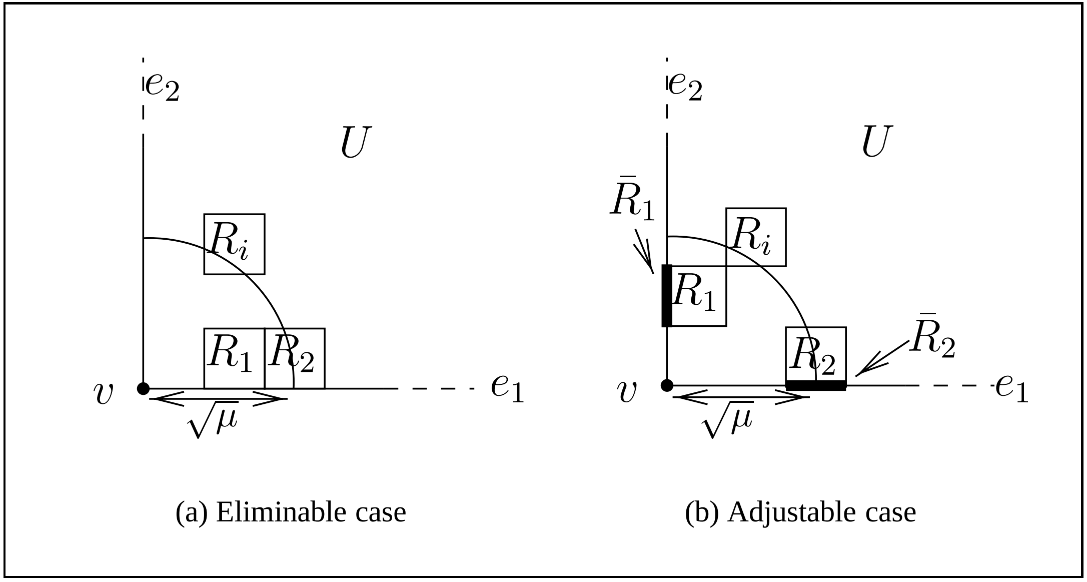
$$\bar{R}_2 := R_2 \cap (e_1 \cup e_2)$$

we do not lose any point $x \in R$ fulfilling Property (P3b). Note that R_1 and R_2 touches either e_1 or e_2, but not both.

Case 3.2: $v \in R_1 \cup R_2$

In this case it is possible that there exist points $x \in R$ fulfilling (P3a) and points $x \in R$ such that (P3b) is true at vertex v. In general there is no way to reduce the dimension of both rectangles R_1 and R_2, as we did in the

FIGURE 5.28. Corner rules (Case 3.1)



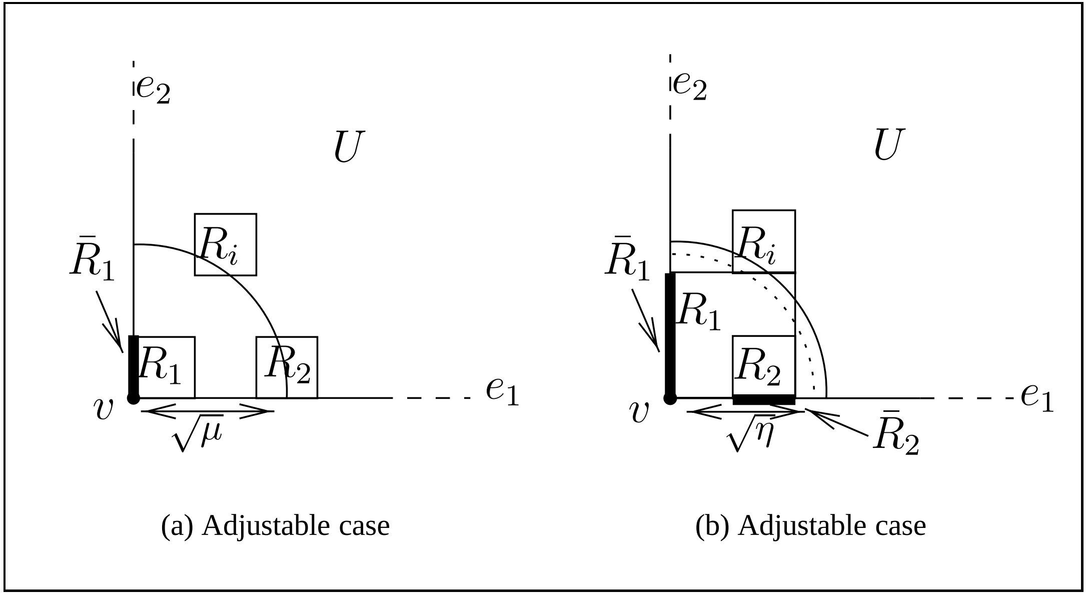
(a) Eliminable case (b) Adjustable case

previous subcase. However, if v belongs to one and only one of the rectangles R_1 or R_2 (see Figure 5.29), we can shrink the rectangle containing v to an interval. Assume that there holds $v \in R_1 \setminus R_2$ (see Figure 5.29(a)). The hyperrectangle $\bar{R}_1 \times R_2 \times \ldots \times R_n$ with

$$\bar{R}_1 \ := \ \begin{cases} R_1 \cap e_1 & \text{, if } R_2 \cap e_1 = \emptyset \\ R_1 \cap e_2 & \text{, if } R_2 \cap e_2 = \emptyset \end{cases}$$

contains any point $x \in R$ satisfying (P3) at vertex v.

FIGURE 5.29. Corner rules (Case 3.2)



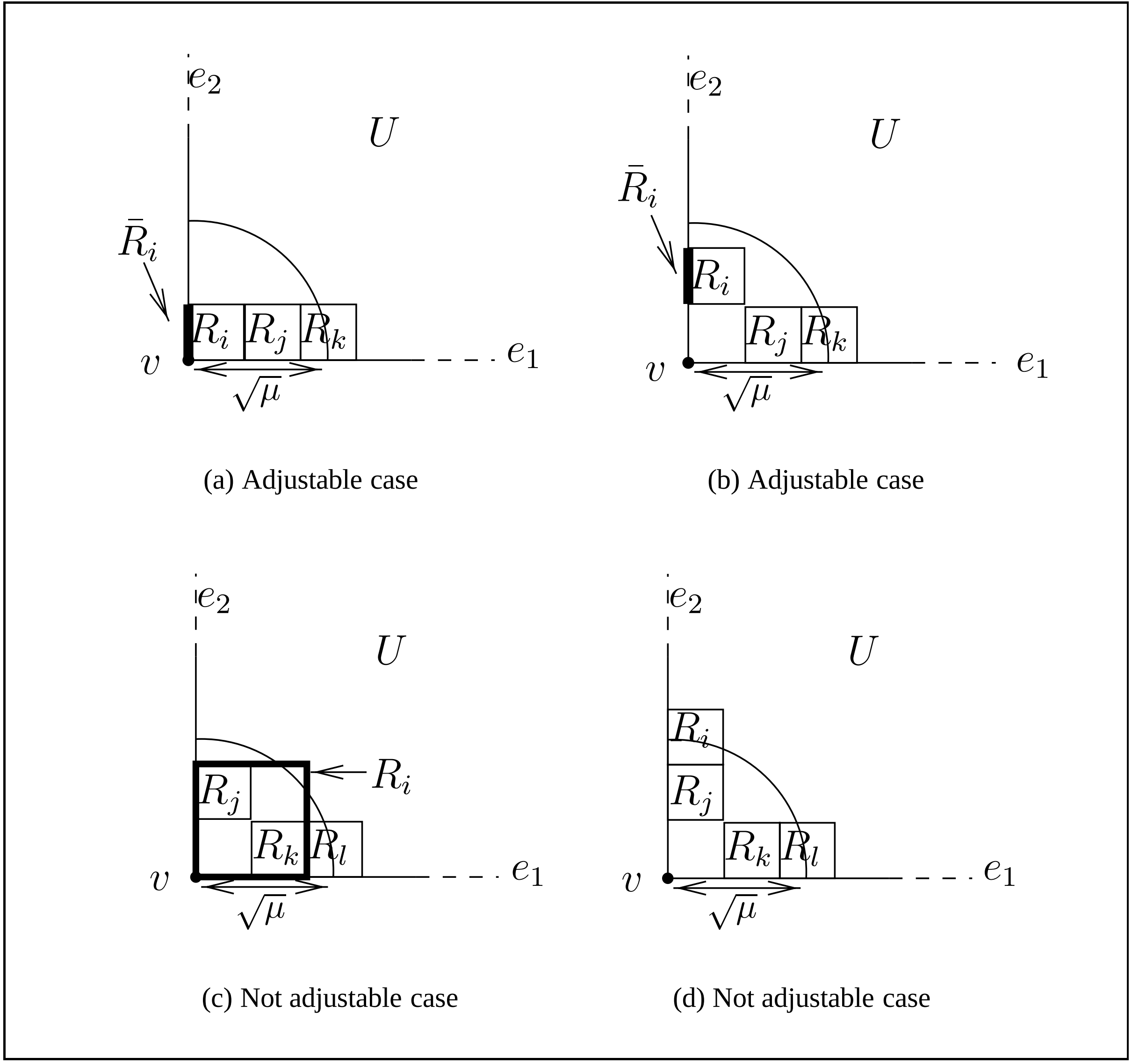
(a) Adjustable case (b) Adjustable case

If there holds, additionally, $R_2 \subset \{x \in U : \|x - v\|_2^2 < \eta\}$ (see Figure 5.29(b)), we can also reduce R_2 to an interval $\bar{R}_2$ by intersecting R_2 with the touched edge-line e_1 or e_2. Note that in this situation it is not possible that v is a member of an optimal solution x belonging to R. Thus, only Property (P3b) can be satisfied.

Case 4: $|\tilde{S}(v, \mu)| \geq 3$

If one and only one of the rectangles R_i belonging to $\tilde{S}(v, \mu)$ touches e_1 or e_2, we are in a comparable situation as in Case 3.2. We are able to shrink this rectangle to an interval $\bar{R}_i$ (see Figures 5.30(a) and 5.30(b)). In situations where at least two rectangles touch each edge-line e_1 and e_2 (see Figures 5.30(c) and 5.30(d)), we

FIGURE 5.30. Corner rules (Case 4)



(a) Adjustable case

(b) Adjustable case

(c) Not adjustable case

(d) Not adjustable case

do not reduce the dimension of a rectangle $R_i \in \tilde{S}(v, \mu)$. If v does not belong to the hyperrectangle R, as it is shown in Figure 5.30(d), only Property (P3b) can be fulfilled. Therefore, it could be possible to shrink some rectangles to intervals. Since it is not immediately clear which one we have to choose, we decided to do nothing in such a situation. Our numerical experience showed, moreover, that this situation almost never occurs.

5.6.2. Edge Rules. We are interested in optimal solutions $x^\star = (x_1^\star, \ldots, x_n^\star)^T$ satisfying (P3) and (P4). For such points it is obvious that there does not exist a segment of a boundary line of U with length greater than or equal to $2\sqrt{\mu} \geq 2\sqrt{t^\star(n)}$, which does not contain a member $x_k^\star$ ($k \in \{1, \ldots, n\}$) of $x^\star$. Using this fact we are able to reduce the dimension of more rectangles R_i ($i \in \{1, \ldots, n\}$) forming R than by using the previously described corner rules alone.

In order to explain the strategy applied in our approach let e be an arbitrary boundary line of U, i.e.,

$$e \in \{e_i^j : i \in \{1, 2\}, j \in \{0, 1\}\}$$

with $e_i^j = \{x \in U : x_i = j\}$. Furthermore let, for two different points $v, w \in e$, $e(\mu) = [v, w]$ be a line segment of e and assume that this segment has a length of $2\sqrt{\mu}$, i.e., $\|v - w\|_2^2 = 4\mu$. Denote by

$$\tilde{L}(e, \mu) = \{R_i : i \in \{1, \ldots, n\} \text{ and } R_i \cap e(\mu) \neq \emptyset\}$$

the set of all rectangles R_i ($i \in \{1, \ldots, n\}$) touching $e(\mu)$.

• If there holds $\tilde{L}(e, \mu) = \emptyset$, it follows that R contains no point fulfilling Property (P4). Hence, as in Case 1 in Subsection 5.6.1, it is not necessary to analyze R further, i.e., R can be pruned.

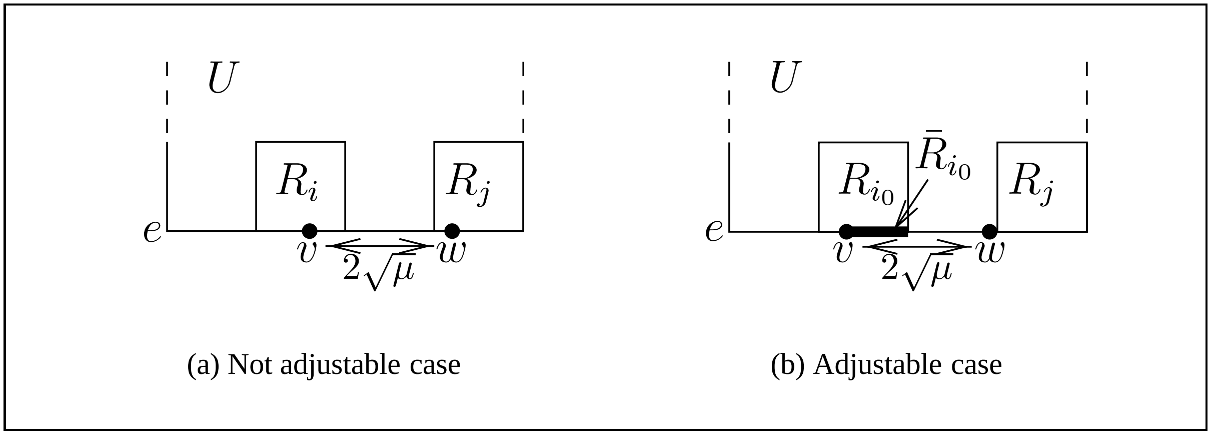
• In the cases where more than one rectangle R_i ($i \in \{1, \ldots, n\}$) touches $e(\mu)$, i.e., $\tilde{L}(e, \mu) > 1$ (see Figure 5.31(a)), it is not possible to reduce the size of a rectangle $R_i \in \tilde{L}(e, \mu)$ without running the risk of losing all optimal solutions.

• If there holds

$$|\tilde{L}(e, \mu)| = 1 ,$$

we can shrink the unique rectangle R_{i_0} touching $e(\mu)$, i.e., $R_{i_0} \cap e(\mu) \neq \emptyset$, to an interval (see Figure 5.31(b)). Indeed, as mentioned before, we know that any point

FIGURE 5.31. Edge rules



(a) Not adjustable case

(b) Adjustable case

$x \in R$ fulfilling (P3) and (P4) has to satisfy the relation

$$x_{i_0} \;\in\; R_{i_0} \cap e(\mu) \;.$$

Therefore, any element x of the hyperrectangle R, which we are interested in, belongs also to the set $R_1 \times \ldots \times R_{i_0-1} \times \bar{R}_{i_0} \times R_{i_0+1} \times \ldots \times R_n$ with

$$\bar{R}_{i_0} \;:=\; R_{i_0} \cap e(\mu) \;.$$

These edge rules leads to a change of R only if $\tilde{L}(e, \mu)$ contains less than 2 elements. With respect to the basic subdivision strategy it is hence not surprising that the edge rules can be applied more rarely than the corner rules. Nevertheless, this strategy improves the numerical performance of Algorithm 5.1. Note that the edge rules are not relevant as long as the current upper bound μ is not smaller than 0.25.

The corner as well as the edge rules use the current upper bound μ in connection with Property (P3) and Property (P4) in order to diminish the size of the considered hyperrectangle R via a reduction of its dimension. In the next subsection we will see how it is possible to further reduce the size of R by exploiting the knowledge of the best known value η, i.e., of the current lower bound.

5.6.3. Volume Reduction. The third size reduction strategy used in Step IV of Algorithm 5.1 is similar to an approach presented in [DGPWM91]. In contrast to the corner and the edge rules we do not diminish the size of the current hyperrectangle R by reducing the dimension of some rectangles R_i ($i \in \{1, \ldots, n\}$) forming R. In this method we reduce the volume of several rectangles R_i by constructing smaller rectangles $\bar{R}_i \subset R_i$ still containing all feasible points of Subproblem (SP) considered in Step V of Algorithm 5.1, i.e., we design rectangles $\bar{R}_i \subset U$

($i \in \{1, \dots, n\}$) with the properties

$$\bar{R}_i \subset R_i \qquad i = 1, \dots, n \tag{5.6.1.a}$$

and

$$F \subset [\eta, \mu] \times \bar{R}_1 \times \dots \times \bar{R}_n \tag{5.6.1.b}$$

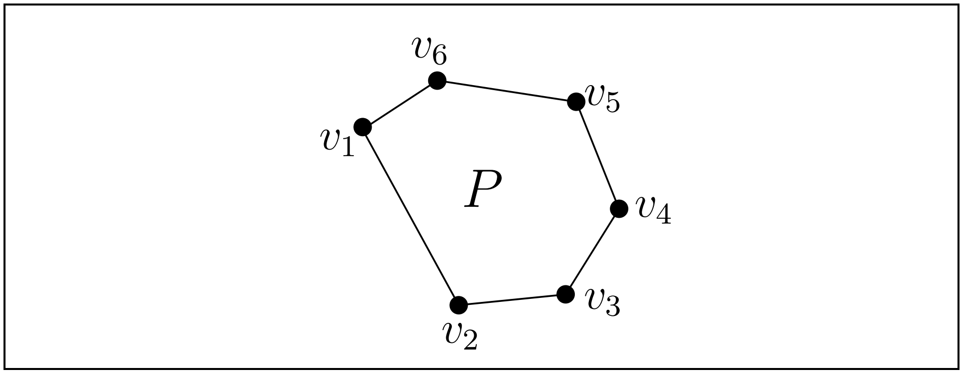
where $F = \{(t, x) \in [\eta, \mu] \times R_1 \times \dots \times R_n : t - \|x_i - x_j\|_2^2 \leq 0, 1 \leq i < j \leq n\}$ denotes the feasible region of (SP).

We compare the rectangles R_i ($i \in \{1, \dots, n\}$) pairwise and try to cut away a part of the infeasible areas of R_i while preserving the structure of a rectangle. The infeasible area of R_i with respect to F is characterized by the fact that, for each element x_i of such an area, there does exist an index $j \in \{1, \dots, n\} \setminus \{i\}$ such that each element of R_j has a squared distance to x_i smaller than η. In order to explain our volume reduction strategy in a general way let, for $i, j \in \{1, \dots, n\}$ with $i \neq j$, two polytopes $P \subset R_i$ and $Q \subset R_j$ be given by their vertex sets, i.e.,

$$\begin{aligned} P &= [v_1, \dots, v_{m_P}], \\ Q &= [w_1, \dots, w_{m_Q}]. \end{aligned}$$

Assume that the vertex lists of P and Q are ordered in such a way that v_{i+1} and w_{i+1} is a direct neighbor of v_i and w_i, respectively (see Figure 5.32). Assume further that $P \times Q$ is a superset of the projection of F on $R_i \times R_j$. We are interested in the set

FIGURE 5.32. Vertex numbering of P



$$\bar{F} := \{(x, y) \in P \times Q : \|x - y\|_2^2 \geq \eta\},$$

since the projection of F on $R_i \times R_j$ is a subset of $\bar{F}$. Denote by

$$C := (\ P \cap \{x \in \mathbb{R}^2 : \underbrace{\max\{\|x - y\|_2^2 : y \in Q\}}_{= \max\{\|x - w_i\|_2^2 : i = 1, \dots, m_Q\} =: f_Q(x)} < \eta\ )$$

the set of all points $x \in P$ with a maximal squared distance to each point in Q smaller than η, i.e., C is part of the infeasible area of P mentioned above. Obviously, there holds

$$\bar{F} \subset (P \setminus C) \times Q \,.$$

This means that we can cut away C from P without eliminating a feasible point of (SP). C is a convex set. If C is not empty, then we do not know whether $P \setminus C$ is still a polytope. However, we would like to preserve the linear structure of P. Therefore, we look for the smallest polytope $\bar{P}$ satisfying

$$P \setminus C \subset \bar{P} \subset P \,.$$

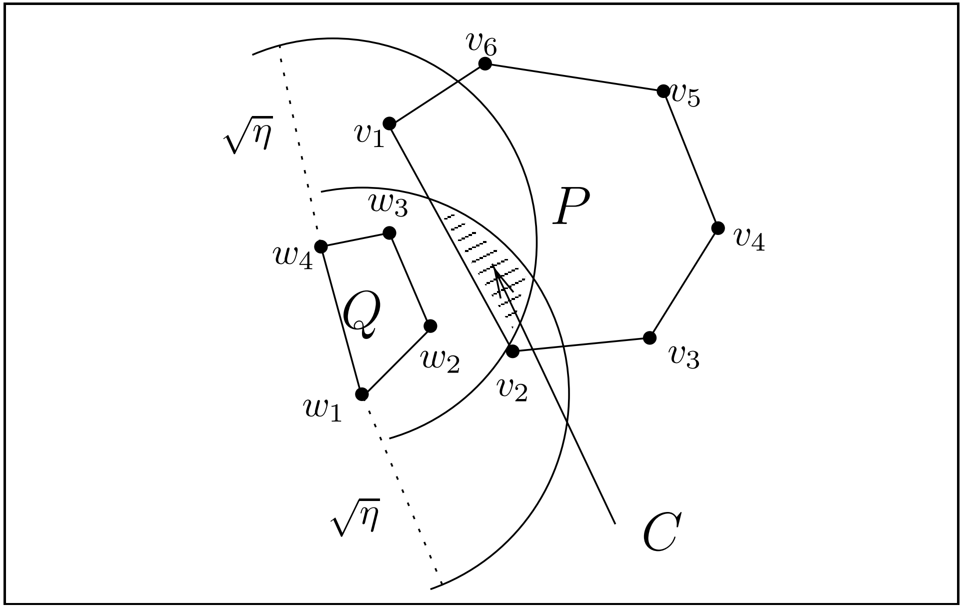
In the sequel we describe the construction of this polytope $\bar{P}$. Assume that C is not empty and denote by

$$C_v := \{v_1, \ldots v_{m_P}\} \cap C$$

the set of all vertices of P belonging to C. Depending on the structure of C_v we distinguish three cases.

• If there holds $C_v = \emptyset$, it follows that we have to take P for $\bar{P}$ itself (see Figure 5.33), i.e., if no vertex v of P has a maximum squared distance to all vertices of Q smaller than η, then we are not able to reduce the size of P without losing the convexity of P.

FIGURE 5.33. $\bar{P} = P$



• If there holds $C_v = \{v_1, \dots, v_{m_P}\}$, it follows immediately $P \setminus C = \emptyset$ and we can eliminate R. Note that in this situation the feasible region F of Subproblem (SP) is empty.

• If C_v is a non-empty real subset of $\{v_1, \dots, v_{m_P}\}$, we can adjust P in the following way. Assume that there exists an index $r \in \{1, \dots, m_P - 1\}$ with

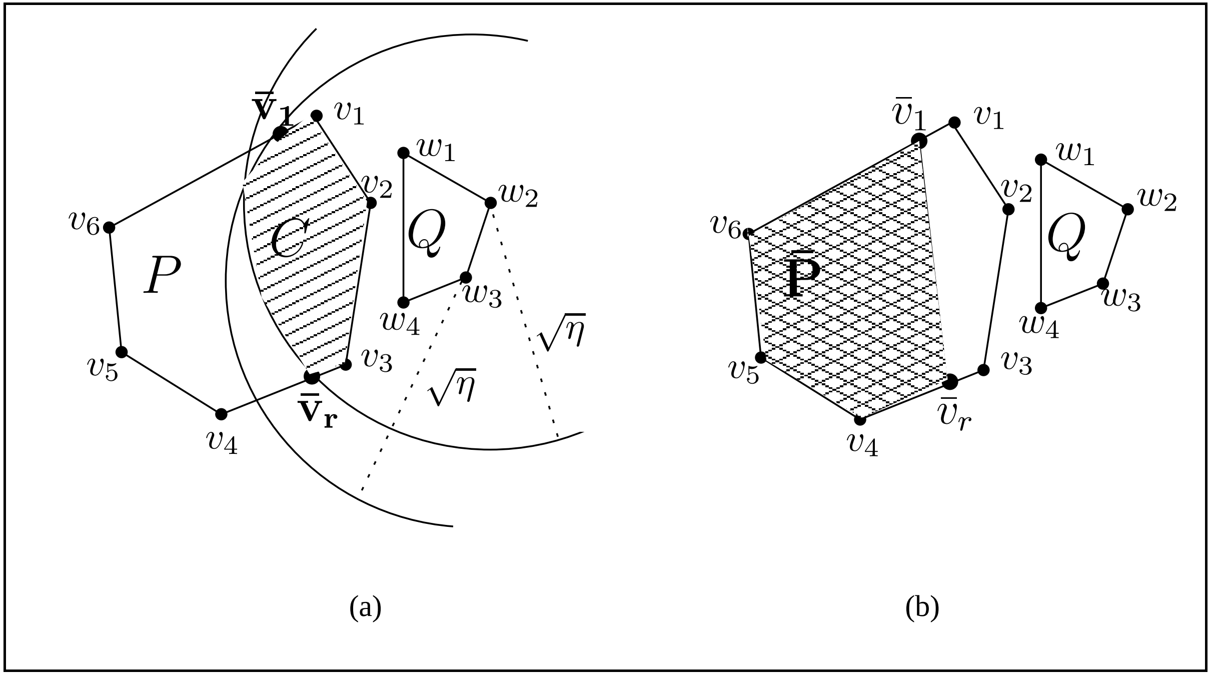
$$C_v \;=\; \{v_1, \dots, v_r\}\,, \tag{5.6.2}$$

i.e., all vertices v of P satisfying $f_Q(v) < \eta$ are neighboring. Since, for each $j \in \{1, \dots, r\}$, there holds

$$f_Q(v_j) \;=\; \max\{\|v_j - w_i\|_2^2 : i = 1, \dots, m_Q\} \;<\; \eta\,,$$

it follows that there exist a unique point $\bar{v}_1$ on the facet of P defined by the neighboring vertices v_{m_P} and v_1 with a maximal squared distance to the polytope Q equal to η, i.e., $f_Q(\bar{v}_1) = \eta$. Note that by construction there holds $f_Q(v_1) < \eta$ and $f_Q(v_{m_P}) \geq \eta$. Let $\bar{v}_r$ be the corresponding point with respect to the facet defined by v_r and v_{r+1} (see Figure 5.34(a)). The function f_Q is a maximum of convex functions and hence convex itself. It follows that, for any $x \in [\bar{v}_1, v_1, \dots, v_r, \bar{v}_r]$, there holds

FIGURE 5.34. $\bar{P} \neq P$



(a) (b)

$$f_Q(x) \;\leq\; \eta$$

and thus

$$[\bar{v}_1, v_1, \ldots, v_r, \bar{v}_r] \ \subset \ \mathrm{cl} C\ .$$

P is a two-dimensional polytope and, therefore, we obtain

$$P \setminus C \subset \bar{P} \ := \ [\bar{v}_1, \bar{v}_r, v_{r+1}, \ldots, v_{m_P}] \ \neq \ P$$

(compare Figure 5.34(b)). If Assumption (5.6.2) is not fulfilled, we adjust P in the same way by analyzing each subset of C_v consisting of a sequence of direct neighboring vertices.

With this general framework we are now able to describe the volume reduction strategy in a more detailed manner. We use the following iterative process.

```
INITIALIZATION
   Set, for $i \in \{1, \ldots, n\}$, $P_i \leftarrow R_i$, i.e.,
      $P_i = \left[\binom{l_{i_1}}{l_{i_2}}, \binom{l_{i_1}}{L_{i_2}}, \binom{L_{i_1}}{l_{i_2}}, \binom{L_{i_1}}{L_{i_2}}\right]$
LOOP
   For $i = 1$ To $n$ Do
      For $j = 1$ To $n$ Do
         If $j \neq i$ Then
            Compare $P_i$ and $P_j$ and construct $\bar{P}_i$ in the described way.
            $P_i \ \leftarrow \ \bar{P}_i$
            If $P_i = \emptyset$ Then STOP ($F$ is empty)
         EndIf
      EndFor
   EndFor
```

After the execution of this process we know either that F is empty, i.e., we can eliminate R from further considerations, or we obtain n two-dimensional polytopes P_i given by the list of their vertices. In this situation it is easy to generate, for each index $i \in \{1, \ldots, n\}$, the smallest rectangle $\bar{R}_i$ containing the polytope P_i. Setting

$$\bar{R} \ := \ \bar{R}_1 \times \ldots \times \bar{R}_n$$

we obtain a hyperrectangle $\bar{R}$, which is a subset of R and, additionally, has the property

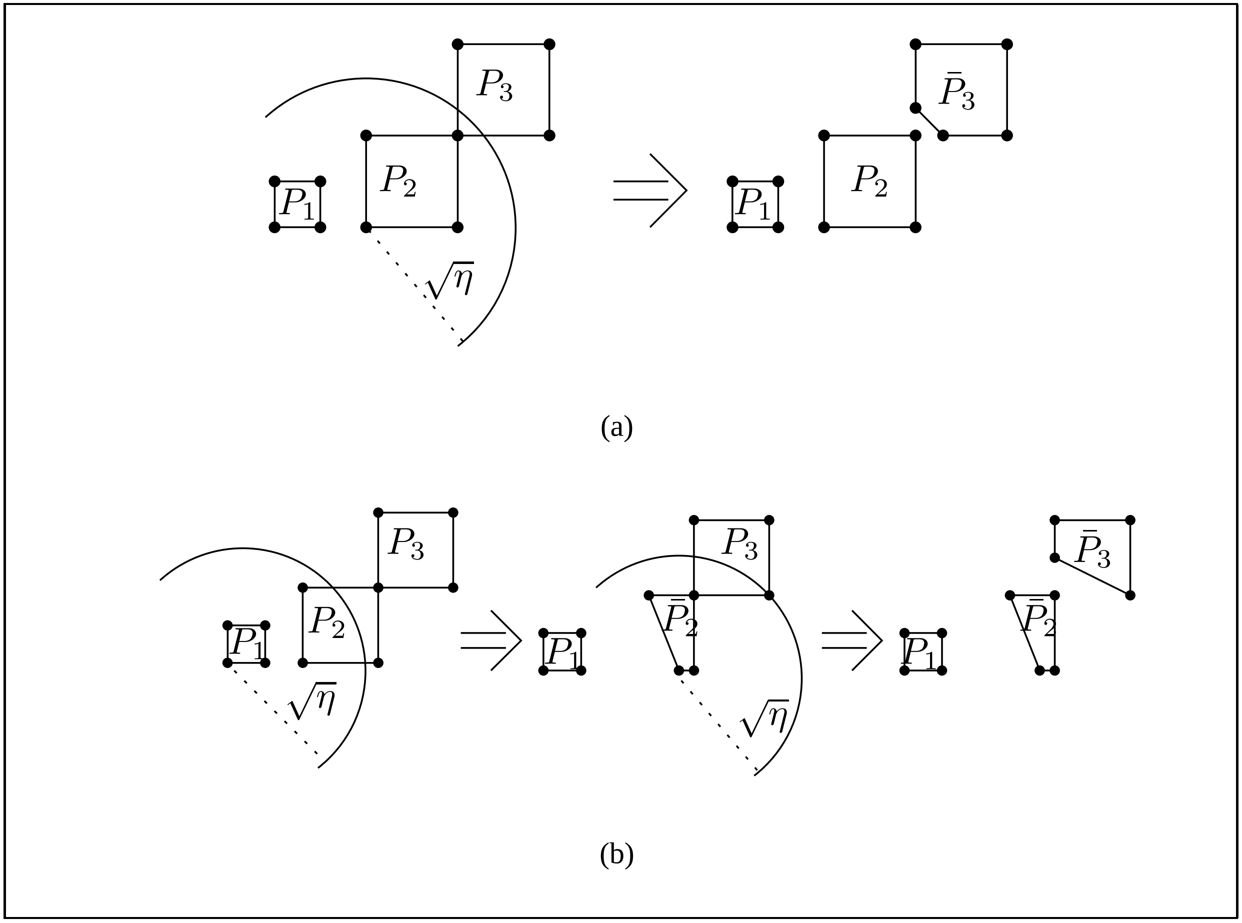
$$F \ \subset \ [\eta, \mu] \times \bar{R}$$

(compare with the required Properties (5.6.1.a) and (5.6.1.b)).

The reason for going back to rectangles $\bar{R}_i$ ($i \in \{1, \ldots, n\}$) instead of using the better approximation of the feasible region F by the polytopes P_i has different aspects. First of all, since the number of vertices describing the polytope P_i ($i \in \{1, \ldots, n\}$) can grow, a strategy using all information given by P_i could extremely increase the storage requirements in an implementation of Algorithm 5.1. A second reason is that our numerical experience for Algorithm 5.1 showed that we do not have a gain using the polytopes P_i instead of the rectangles $\bar{R}_i$ in the calculation of the upper bounds for Subproblem (SP). This seems to depend on the construction of the LP-relaxation for Problem (SP), which is needed in order to calculate upper bounds, as it is described in Section 5.4.

At this place we would like to pay some attention to a special effect, which could happen during the execution of the presented iterative process and which we would like to call the ***wave effect***. Let us illustrate this effect with an example. Figure 5.35(a) shows the possible adjustment of rectangle P_3, if we compare P_3 with

FIGURE 5.35. The wave effect



P_2. However, if we adjust first P_2 using P_1, we are able to cut away a larger part of P_3, as it is displayed in Figure 5.35(b). This effect justifies the substantial effort in executing our volume reduction strategy. Note that there are $n(n-1)$ comparisons between polytopes, and furthermore that the number of vertices describing a polytope P_i can grow and hence the effort for calculating the adjustments. Taking this effect into account it seems, moreover, possible that a repetition of the loop-phase of the iterative process is able to further reduce the size of the relevant hyperrectangle. As we will see in Section 5.7 considering some computational results, there is a trade-off – in the repetition of the volume reduction strategy – between the advantages of a better size reduction and the disadvantage of a growing running-time needed for doing this.

REMARK 5.6.2. The presented size reduction strategies, i.e, the corner and the edge rules and the volume reduction, have two effects on the performance of Algorithm 5.1. On the one hand, they reduce the size of the linear part of the feasible region F of the current Subproblem (SP). Hence, they abate the effort for solving (SP). Note that the LP-relaxation of (SP) tends to be better if the relevant hyperrectangle R gets smaller (see Section 5.4).

On the other hand, they work like a *pruning*-rule (see Step VII of Algorithm 5.1 for the classical pruning rule in branch-and-bound methods). We can cut away many possible subdivision sets, since Algorithm 5.1 using our strategies recognize that in these sets there do not exist points satisfying (P3) and (P4) and with a minimum pairwise squared distance not smaller than η.

As mentioned before it is possible that the use of the corner and the edge rules eliminates optimal solutions of Problem (PP) from further considerations without detecting them. However, we never throw away all solutions by using these ideas. Therefore, the presented size reduction strategies are consistent in the sense of Condition (C3) (see Section 5.3). If we are careful in the implementation of our approach, especially if we pay attention to the possible interactions between our diverse subdivision set manipulation strategies (see Remark 5.5.5(b)), we are able to satisfy Condition (C1) and Condition (C3) required in Section 5.3. This ensures a correct functioning and particularly the convergence of Algorithm 5.1 (see Theorem 5.3.1 and, additionally, Lemma 5.4.1).

5.7. Computational Results

The description of Algorithm 5.1 is now complete. The missing details in the formulation of this approach (Step III - Step V) in Section 5.3 were described in the foregoing three sections. We derived several strategies exploiting the special structure of Problem (PP) for the calculation of the upper bounds (see Section 5.4) as well as for the splitting (Section 5.5) and the adjustment (Section 5.6) of the subdivision sets considered in this approach. This was necessary in order to obtain an efficient method for solving (PP) since general approaches fail to determine approximate solutions of this problem, as we pointed out in the introduction of this chapter. Even though we will introduce a modified basic partitioning strategy and some further improvements of Algorithm 5.1 in the next section, we would like to present first some computational results, which were obtained with Algorithm 5.1 using the strategies developed so far. These results correspond to the numerical tests reported in [LR98B].

As we did with all algorithms discussed in this thesis until now, Algorithm 5.1 was encoded in C++ with management of subdivision sets by AVL-trees. The occurring linear problems in Step V were solved with *MINOS 5.4* (see also Algorithm 3.1). Note that the LP-relaxation of Subproblem (SP) has a sparse structure and that *MINOS 5.4* is able to exploit sparsity. With this implementation of Algorithm 5.1 we solved Problem (PP) with $n \leq 27$ points. The tolerance ϵ was chosen as 10^{-5}. Using a *SUN ULTRA 60* workstation we were able to determine approximate solutions for each of these problems within less than two and a half hours.

In order to obtain these good running-time results we applied additional ideas, which are more heuristically motivated. Before discussing the numerical results in detail we would like to give some notes on these ideas.

- In the description of the iterative process for the volume reduction strategy in Section 5.6 we pointed out the existence of the so-called wave effect. Regarding this effect it seems to be reasonable to repeat the loop-phase of the iterative process in order to reduce the size of the relevant hyperrectangle as much as possible. However, we have to remember that this process could be expensive with respect to the running-time. Our numerical tests showed that it is efficient to repeat the process once, i.e., the advantage of a bigger size reduction outbalances the disadvantage of a growing running-time needed for doing this. If we repeat the process again, the disadvantage outbalances the advantage. Therefore, we decided to use the size reduction strategies in the following way. For each hyperrectangle R^{k_i} remaining after Step III (see the description of Algorithm 5.1 in Section 5.3), we apply at first the volume reduction strategy, where we repeat the iterative process once. After this we use the corner and the edge rules in order to diminish the dimension of the resulting hyperrectangles $\bar{R}^{k_i}$. If the dimension reduction is successful, we apply the volume reduction process again – now without a repetition. It might be possible to choose an implementation of the volume reduction strategy that is less time-consuming than the one we used. This could allow more than one repetition of the iterative process without increasing the running-time.

An interesting aspect in our numerical tests was that the combination of the dimension reduction strategies, i.e., the corner and edge rules, with the volume reduction strategy led to an extraordinary better running-time performance than the use of one of these strategies alone. There are at least two reasons for this improvement. First of all, single points or intervals, which can be the result of the corner and the edge rules, lead in general to a larger reduction of the size of neighboring rectangles. Hence – via the wave effect – they have an impact on the size of the whole hyperrectangle. On the other hand, smaller rectangles forming the relevant hyperrectangle can result in a successful dimension reduction at an earlier stage of the algorithm. Therefore, the volume reduction strategy and the dimension reduction strategies are not independent from each other, rather they interact.

- Our numerical tests showed, furthermore, that we need most of the time for solving the linear subproblems in Step V in order to calculate the upper bounds. Moreover, we observed that in many cases there holds $\mu_{R^{k_p}} = \mu_{R^k}$, i.e., the upper bound with respect to R^k is equal to the upper bound of its direct child R^{k_p}. For that reason we developed a criterion in order to decide whether is seems to be useless to calculate a new upper bound for R^{k_p} by solving a linear program instead of taking the old bound μ_{R^k}, or not. This criterion is as follows.

  Let $R^{k_p} = R_1^{k_p} \times \ldots \times R_n^{k_p} \subset U^n$ ($p \in \{1, \ldots, l\}$) be the hyperrectangle examined in Step V. An upper bound for Problem (SP) with respect to this set is obviously given by

$$\bar{\mu}_{R^{k_p}} = \min_{1 \leq i < j \leq n} \max_{\substack{x_i \in R_i^{k_p} \\ x_j \in R_j^{k_p}}} \|x_i - x_j\|_2^2 , \tag{5.7.1}$$

(see the proof of Lemma 5.4.1). Recognize that this value can be calculated by considering the vertices of $R_i^{k_p}$ ($i \in \{1, \ldots, n\}$). We have to decide whether it is useful to obtain a bound $\mu_{R^{k_p}}$ by solving the linear program

$$\begin{aligned} \max\ t & \\ h_{ij}(t, x_i, x_j) \leq 0 & \qquad 1 \leq i < j \leq n \\ x_i \in R_i^{k_p} & \qquad i = 1, \ldots, n \\ \eta^k \leq t \leq \min\{\mu^k, \bar{\mu}_{R^{k_p}}\} & \end{aligned} \tag{LSP'}$$

(see Section 5.4 for the construction of h_{ij} depending on $R_i^{k_p}$ and $R_j^{k_p}$), or whether we should simply set $\mu_{R^{k_p}} = \min\{\mu^k, \bar{\mu}_{R^{k_p}}\}$.

If we are able to construct a feasible point $(\bar{t}, \bar{x}) \in [\eta^k, \mu^k] \times R^{k_p}$ for Problem (LSP') combining the vertices of the two-dimensional rectangles $R_i^{k_p}$ ($i = 1, \ldots, n$), we do not solve (LSP'). In this situation it is very likely that we have to analyze R^{k_p} further in a later iteration, i.e., R^{k_p} will not be pruned – at least as long as the lower bound η^k is not improved. Therefore, it seems to be useless to calculate an upper bound for R^{k_p} by solving (LSP'), since we have to solve this LP-relaxation with respect to the relevant subsets of R^{k_p} in later iterations. The use of this criterion reduced significantly the running-time of Algorithm 5.1. Applying this criterion we needed less time for solving the linear subproblems than for executing the subdivision set manipulation strategies.

Note that in our implementation of this criterion we obtain the value $\bar{\mu}_{R^{k_p}}$ without additional effort. For that reason, we exploited the knowledge of this value throughout the verification of the described criterion, i.e., we checked whether it seems to be possible to improve the possibly better upper bound $\min\{\mu^k, \bar{\mu}_{R^{k_p}}\}$ instead of considering only μ^k.

In the description of Algorithm 5.1 in Section 5.3 we assumed that a point $\bar{x} \in U^n$ with $f(\bar{x}) = \min_{1 \leq i < j \leq n} \|\bar{x}_i - \bar{x}_j\|_2^2 > 0$ is given. We use this point in order to initialize the lower bound η^0 (see the initialization phase of Algorithm

5.1). Since the choice of the integer m in the basic part of our subdivision strategies depends on the value η^0 (see Subsection 5.5.1), the number of subdivision sets, which have to be analyzed during the execution of Algorithm 5.1, is very sensitive to changes in this value. In several papers (see, e.g., [MFP95, GL96, NO97]) good solutions for the point scattering problem are given. Therefore, we decided to choose the best known solution for Problem (PP) as starting point $\bar{x}$. If we were not able to reproduce the coordinates of $\bar{x}$ from a paper, we used a simple multi-start algorithm developed by Prof. Fabio Schoen at the University of Florence in order to generate good solutions (for the framework of stochastical approaches in global optimization we refer again to [BR95]). Because of this choice of $\bar{x}$ it was sufficient to set $m = 3$ for $n \leq 27$.

Another consequence of this choice was that Algorithm 5.1 did not substantially improve the known solutions. The slight improvements displayed in Table 5.3 seem to be rounding differences – for the case $n = 12$ we did not start with an optimal solution and, therefore, we had a larger improvement. In Table 5.3

TABLE 5.3. Improvements

n	η^0	μ^0	$\eta^\star$	$\mu^\star$	$\eta^\star - \eta^0$
10	0.177399	0.25	0.177468	0.177477	6.9e-5
11	0.158568	0.17743	0.158568	0.158568	0.0
12	0.146713	0.15857	0.151111	0.151121	4.4e-3
13	0.134021	0.15112	0.134021	0.134031	0.0
14	0.121739	0.13403	0.121742	0.121743	3.0e-6
15	0.116329	0.12174	0.116336	0.116338	7.0e-6
16	0.111111	0.11634	0.111111	0.111121	0.0
17	0.0937256	0.11111	0.0937279	0.0937379	2.3e-6
18	0.0902758	0.09425	0.0902778	0.0902876	2.0e-6
19	0.0838326	0.09061	0.0838326	0.0838419	0.0
20	0.0821442	0.08385	0.0821462	0.0821548	2.0e-6
21	0.0738791	0.08219	0.0738791	0.0738891	0.0
22	0.0717971	0.08219	0.0717971	0.0718059	0.0
23	0.0669872	0.07189	0.0669872	0.0669952	0.0
24	0.0646835	0.07189	0.0646853	0.0646950	1.8e-6
25	0.0625	0.067	0.0625	0.0625096	0.0
26	0.0569574	0.0625	0.056989	0.056999	3.16e-5
27	0.055625	0.0625	0.055625	0.055648	0.0

we use η^0 and μ^0 for the first lower respectively upper bound, which was set in the initialization phase of Algorithm 5.1. The columns $\eta^\star$ and $\mu^\star$ show the last lower and upper bound fulfilling the stopping criterion (Step I). The last column displays the improvements made by Algorithm 5.1. Note that $\eta^\star$ is the minimum pairwise squared distance of the members $x_i^\star$ $(i = 1, \dots, n)$ of the best solution $x^\star = (x_1^\star, \dots, x_n^\star)^T \in U^n$, which was determined by Algorithm 5.1. Note, moreover, that even though the values displayed in column $\eta^\star$ are slight improvements of the initial lower bounds, they are not better than those given in [MFP95].

Even though we did not calculate better points, the main advantage of Algorithm 5.1 is that this method can guarantee the ϵ-optimality of the determined solutions, which was not known at least for $n = 21 - 24$, 26, 27. Hence, as it is done in [DGPW90, DGPWM91], our method can be used as a computer aided proof for the optimality of detected solutions.

REMARK 5.7.1. We have to note that the current implementation of Algorithm 5.1 cannot be used unreserved as a computer aided proof. The main intention of our current implementation of this method was to show that it is possible to solve Problem (PP) with more than 20 points and acceptable computational effort. In the sequel we will see that we were able to determine approximate solutions of a global optimization problem in dimension 55 and with 351 concave quadratic constraints within less than one hour. However, if we would like to use Algorithm 5.1 for a computer aided proof, we have to pay more attention to calculation errors resulting from the machine precision.

We examined this problem in our volume reduction strategy, which might be the most sensitive part of our approach with respect to such errors. It is possible to adjust the implementation of the iterative process such that we can guarantee that no point $x \in U^n$ with a minimum squared distance larger than η^k is eliminated because of calculation errors. In order to ensure that we never lose ϵ-optimal points because of such errors, we have to examine each step of our implementation. This means that in order to obtain a *computer aided proof implementation* still a lot of work has to be done. Moreover, the proof of the numerical correctness of the resulting implementation is behind the scope of this thesis. Therefore and in view of the main intention of our implementation mentioned above, we did not invest this effort for the numerical results reported here. Only the adjustment of the volume reduction strategy was applied.

In Table 5.4 we show the effort for solving Problem (PP) with $n \in \{10, \dots, 27\}$ points. We use the abbreviation IT for the number of iterations. The column TT displays the total CPU-time in seconds necessary for determining the approximate solutions. NLP stands for the number of linear subproblems of type (LSP'), which had to be solved during the execution of Algorithm 5.1, and TLP shows the running-time needed for the solution of these linear programs by *MINOS 5.4.* The abbreviation NR is used for the number of hyperrectangles remaining after Step III, i.e., this is the number of hyperrectangles, which had to be analyzed by the size reduction strategies. It it interesting to note that these numbers are mostly smaller than two times the number of iterations. This means, that even though we had in each iteration $l \in \{4, m^2\}$ possible partition sets, there remained on average only two hyperrectangles after the application of the special features of our subdivision strategy. In the last column MNPS we report the maximal number of subdivision sets, which had to be stored in an iteration $k \in \mathbb{N}$ in the set $\mathcal{R}^k$. These numbers give us some insight into the storage requirements of our approach.

TABLE 5.4. Numerical effort

n	IT	TT	NLP	TLP	NR	MNPS
10	1,008	2.50	144	0.57	2,997	195
11	792	2.58	162	0.90	2,100	125
12	1,351	7.19	416	3.07	3,415	266
13	2,379	8.26	272	2.45	6,548	332
14	8,457	43.8	1,766	18.9	20,456	1,445
15	1,851	9.47	359	4.56	3,809	251
16	24,127	99.6	1,016	13.5	54,492	3,950
17	38,890	297	10,268	144	80,897	6,230
18	22,429	218	6,545	103	45,727	4,576
19	66,122	548	12,003	247	131,763	10,032
20	22,200	252	6,065	135	43,032	2,343
21	240,210	2,269	35,630	920	472,716	42,977
22	55,005	516	6,838	203	103,903	8,776
23	153,884	2,500	38,417	1,377	268,598	20,873
24	194,497	2,956	38,475	1,484	335,547	25,411
25	109,798	1,759	20,063	868	184,419	14,644
26	669,450	8,941	48,114	2,284	1,038,174	86,950
27	250,102	3,172	13,830	730	364,026	31,918

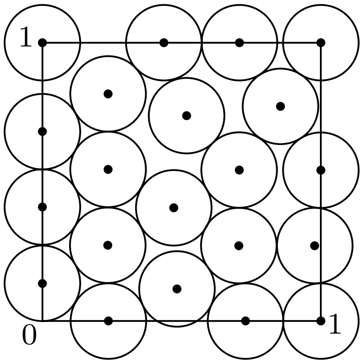
REMARK 5.7.2.

(a) In the next section we will see that a slight change of the selection rule for the current hyperrectangle R^k can significantly reduce the storage requirements.

(b) We use nearly optimal solution as starting points of our approach. Therefore, we do not substantially improve the lower bounds η^k ($k \in \mathbb{N}$) during the execution of Algorithm 5.1. This means that the standard pruning rule in Step VII of our method is not very successful. Recognize that the main task of the classical pruning rule is to cut away branches of the tree consisting of all possible subdivision sets. Hence, with respect to our good starting points we know that we have to examine almost the whole tree. Regarding this fact we can also use a *depth-first-search-strategy*. Applying such a strategy we will examine also the whole tree, but we are able to bound the storage requirements. We have to store at most the maximal length of one branch of the tree. Such a strategy was used in order to calculate an ϵ-optimal solution for Problem (PP) with $n > 27$, as we will see also in the next section.

In Figures 5.36-5.38 we present, finally, the arrangements of the calculated ϵ-optimal solutions for $n = 21 - 24$, 26, 27 together with their coordinates. These can be used for further research on this topic (see Remark 5.7.1). The highly symmetric structure for the solution of the prime number 23 is interesting to note.

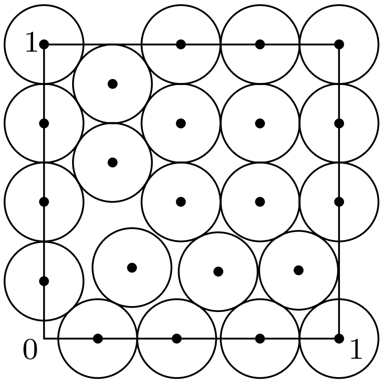
FIGURE 5.36. Solution for $n = 21$



$x_1 = (0.5176, 0.7384)$	$x_2 = (0.0000, 0.6805)$
$x_3 = (0.0000, 0.1349)$	$x_4 = (0.4702, 0.4077)$
$x_5 = (0.2354, 0.8165)$	$x_6 = (0.2354, 0.5446)$
$x_7 = (0.7077, 1.0000)$	$x_8 = (0.7062, 0.5426)$
$x_9 = (0.4821, 0.1154)$	$x_{10} = (0.0000, 0.4087)$
$x_{11} = (0.8539, 0.7708)$	$x_{12} = (0.0001, 0.9996)$
$x_{13} = (1.0000, 0.5416)$	$x_{14} = (0.4359, 1.0000)$
$x_{15} = (0.9768, 0.2708)$	$x_{16} = (0.7050, 0.7050)$
$x_{17} = (1.0000, 0.0000)$	$x_{18} = (0.7282, 0.0000)$
$x_{19} = (1.0000, 1.0000)$	$x_{20} = (0.2348, 0.2718)$
$x_{21} = (0.2360, 0.0000)$	

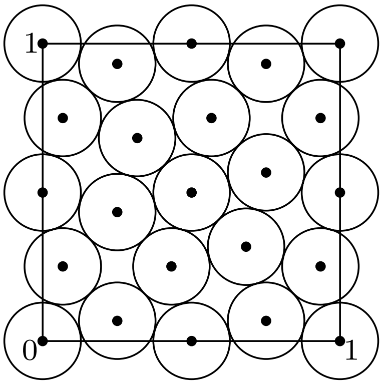
According to the best known solutions for the point scattering problem with more than 27 points we have to choose the integer m as 4 in the second level of our basic subdivision strategy. This leads to a substantial increase of the possible

FIGURE 5.37. Solutions for $n = 22, 23, 24$



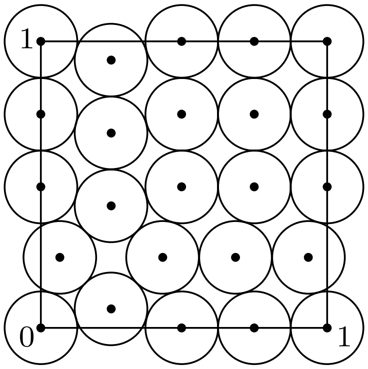
$x_1 = (0.0000, 0.1960)$	$x_2 = (1.0000, 0.0000)$
$x_3 = (1.0000, 0.4641)$	$x_4 = (0.1827, 0.0000)$
$x_5 = (0.0000, 1.0000)$	$x_6 = (0.4641, 1.0000)$
$x_7 = (0.4507, 0.0000)$	$x_8 = (0.7320, 0.0000)$
$x_9 = (0.2984, 0.2417)$	$x_{10} = (0.0000, 0.4640)$
$x_{11} = (0.4640, 0.4639)$	$x_{12} = (0.5913, 0.2281)$
$x_{13} = (0.8660, 0.8660)$	$x_{14} = (0.7320, 0.4641)$
$x_{15} = (0.2320, 0.5980)$	$x_{16} = (0.0000, 0.7320)$
$x_{17} = (0.4641, 0.7320)$	$x_{18} = (0.2320, 0.8660)$
$x_{19} = (0.7320, 0.7320)$	$x_{20} = (1.0000, 0.7320)$
$x_{21} = (0.7320, 1.0000)$	$x_{22} = (1.0000, 1.0000)$

(a) $n = 22$



$x_1 = (0.5000, 1.0000)$	$x_2 = (0.0670, 0.7500)$
$x_3 = (0.7500, 0.0670)$	$x_4 = (0.3170, 0.6830)$
$x_5 = (0.7500, 0.5670)$	$x_6 = (0.4330, 0.2500)$
$x_7 = (0.5000, 0.5000)$	$x_8 = (1.0000, 1.0000)$
$x_9 = (1.0000, 0.5000)$	$x_{10} = (0.9330, 0.7500)$
$x_{11} = (0.0000, 0.5000)$	$x_{12} = (0.0670, 0.2500)$
$x_{13} = (0.0000, 1.0000)$	$x_{14} = (0.6830, 0.3170)$
$x_{15} = (0.5670, 0.7500)$	$x_{16} = (0.0000, 0.0000)$
$x_{17} = (0.2500, 0.4330)$	$x_{18} = (0.9330, 0.2500)$
$x_{19} = (0.2500, 0.9330)$	$x_{20} = (0.5000, 0.0000)$
$x_{21} = (1.0000, 0.0000)$	$x_{22} = (0.2500, 0.0670)$
$x_{23} = (0.7500, 0.9330)$	

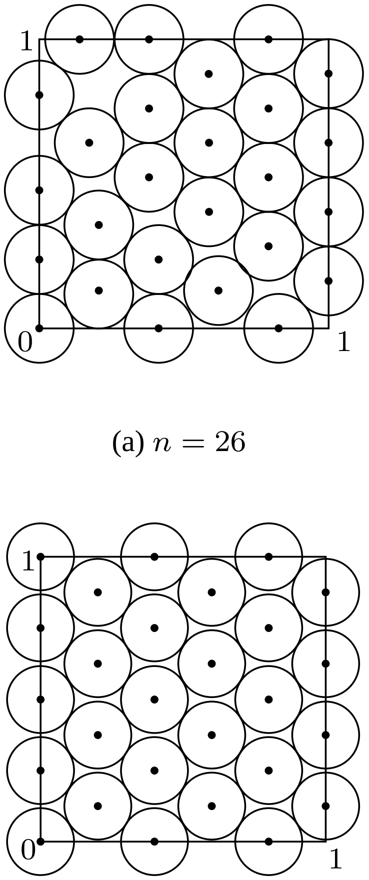
(b) $n = 23$



$x_1 = (0.0000, 0.0000)$	$x_2 = (0.0000, 0.4913)$
$x_3 = (1.0000, 0.4913)$	$x_4 = (1.0000, 0.0000)$
$x_5 = (0.4913, 0.0000)$	$x_6 = (0.4913, 1.0000)$
$x_7 = (0.0000, 1.0000)$	$x_8 = (0.2457, 0.0658)$
$x_9 = (0.0658, 0.2457)$	$x_{10} = (0.4255, 0.2457)$
$x_{11} = (0.2457, 0.4255)$	$x_{12} = (0.4913, 0.4913)$
$x_{13} = (0.7457, 0.0000)$	$x_{14} = (0.6798, 0.2457)$
$x_{15} = (0.9342, 0.2457)$	$x_{16} = (0.7457, 0.4913)$
$x_{17} = (0.0000, 0.7457)$	$x_{18} = (0.2457, 0.2457)$
$x_{19} = (0.4913, 0.7457)$	$x_{20} = (0.2457, 0.9342)$
$x_{21} = (0.7457, 0.7457)$	$x_{22} = (1.0000, 0.7457)$
$x_{23} = (0.7457, 1.0000)$	$x_{24} = (1.0000, 1.0000)$

(c) $n = 24$

FIGURE 5.38. Solutions for $n = 26, 27$



(a) $n = 26$

$x_1 = (0.0000, 0.8073)$	$x_2 = (1.0000, 0.8807)$
$x_3 = (1.0000, 0.1644)$	$x_4 = (0.7932, 1.0000)$
$x_5 = (0.8269, 0.0000)$	$x_6 = (0.4134, 0.0000)$
$x_7 = (0.0000, 0.0000)$	$x_8 = (0.3797, 1.0000)$
$x_9 = (0.2067, 0.1195)$	$x_{10} = (0.0000, 0.2388)$
$x_{11} = (0.4134, 0.2388)$	$x_{12} = (0.2067, 0.3582)$
$x_{13} = (0.6201, 0.1194)$	$x_{14} = (0.7933, 0.2838)$
$x_{15} = (0.5865, 0.4032)$	$x_{16} = (1.0000, 0.4032)$
$x_{17} = (0.1727, 0.6425)$	$x_{18} = (0.3798, 0.5226)$
$x_{19} = (0.3797, 0.7613)$	$x_{20} = (0.5865, 0.6419)$
$x_{21} = (0.7932, 0.5225)$	$x_{22} = (0.7932, 0.7613)$
$x_{23} = (1.0000, 0.6419)$	$x_{24} = (0.5864, 0.8806)$
$x_{25} = (0.0000, 0.4776)$	$x_{26} = (0.1409, 1.0000)$

(b) $n = 27$

$x_1 = (0.0000, 0.0000)$	$x_2 = (0.0000, 1.0000)$
$x_3 = (1.0000, 0.8750)$	$x_4 = (1.0000, 0.1250)$
$x_5 = (0.0000, 0.7500)$	$x_6 = (0.0000, 0.5000)$
$x_7 = (1.0000, 0.3750)$	$x_8 = (1.0000, 0.6250)$
$x_9 = (0.0000, 0.2500)$	$x_{10} = (0.8000, 1.0000)$
$x_{11} = (0.8000, 0.0000)$	$x_{12} = (0.4000, 0.0000)$
$x_{13} = (0.4000, 1.0000)$	$x_{14} = (0.2000, 0.1250)$
$x_{15} = (0.4000, 0.2500)$	$x_{16} = (0.2000, 0.3750)$
$x_{17} = (0.4000, 0.5000)$	$x_{18} = (0.6000, 0.1250)$
$x_{19} = (0.8000, 0.2500)$	$x_{20} = (0.6000, 0.3750)$
$x_{21} = (0.2000, 0.6250)$	$x_{22} = (0.4000, 0.7500)$
$x_{23} = (0.2000, 0.8750)$	$x_{24} = (0.6000, 0.6250)$
$x_{25} = (0.8000, 0.5000)$	$x_{26} = (0.8000, 0.7500)$
$x_{27} = (0.6000, 0.8750)$	

partition sets and also to an explosion of the running-times. We were not able to solve Problem (PP) with $n > 27$ and the current version of Algorithm 5.1 within several days.

5.8. Improvements of Algorithm 5.1

We complete the consideration of the packing problem with the description of some further improvements of Algorithm 5.1. Applying these new ideas we could significantly reduce the computational effort for solving Problem (PP) with $n \in \{14, \dots, 27\}$ points. Moreover, we were able to determine approximate solutions of Problem (PP) with more than 27 points.

At first we discuss a slightly modified basic subdivision strategy. The application of this new method results in a reduction of the effort for solving (PP) in all respects, i.e., with respect to the iteration number, the running-time as well as the storage requirement. After this we will shortly describe a new criterion in order to decide whether we should calculate a new upper bound by solving a linear program or whether we should take the old one. We will see that this strategy further reduced substantially the running-times for solving Problem (PP) with $n \leq 27$.

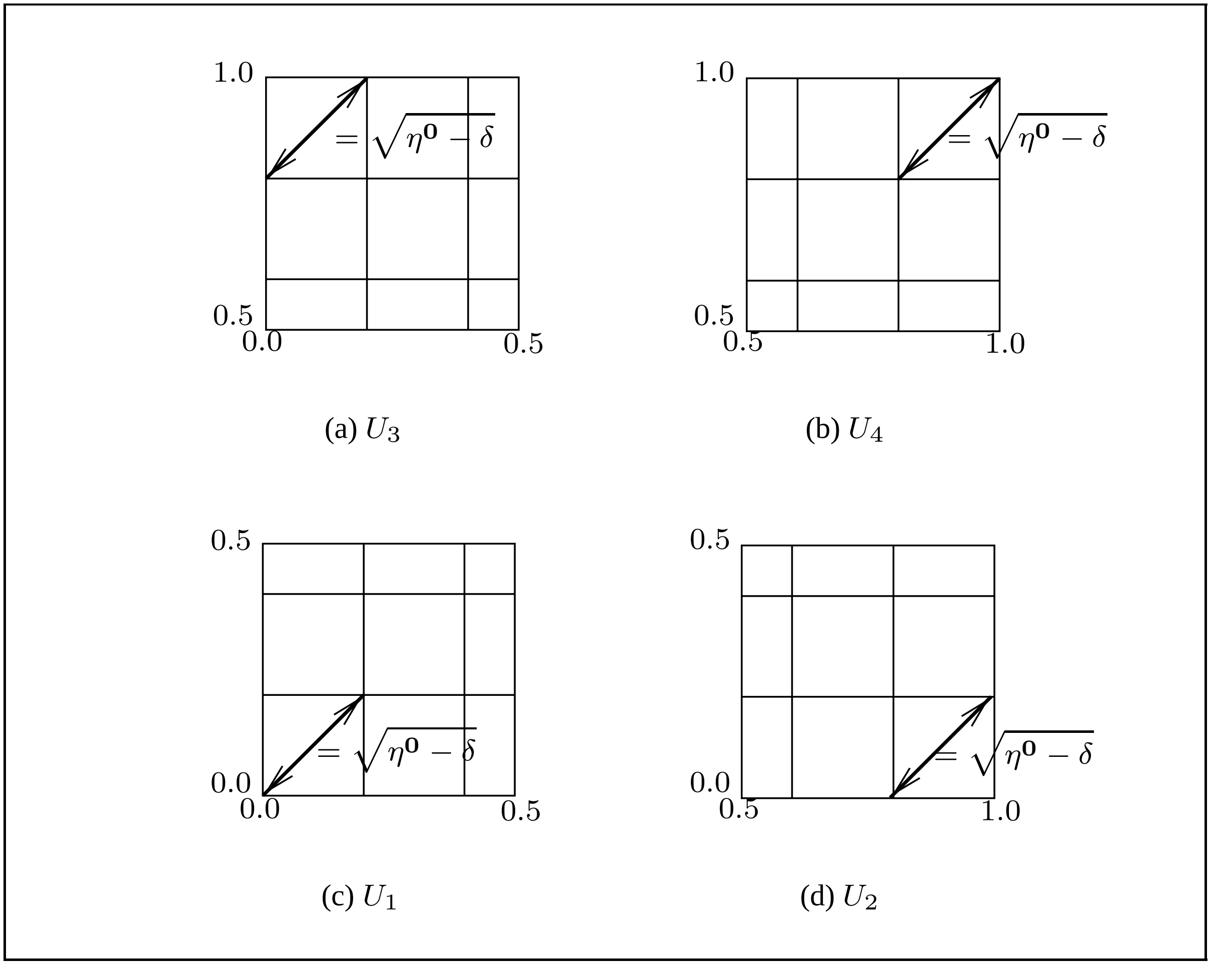
5.8.1. Another Basic Partitioning Strategy. As in the description of the old strategy in Subsection 5.5.1, let $R_j \subset U$ ($j \in \{1, \dots, n\}$) be the rectangle chosen in Step II of Algorithm 5.1. We pointed out that, taking the symmetry avoiding strategies into account, it is essential that in the first level, i.e., if R_j coincides with the unit square U, we use a partition of R_j consisting of squares with equal size. Therefore, we did not change the basic strategy in the first level.

However, in the second level, i.e., if R_j is equal to one of the squares U_i ($i \in \{1, \dots, n\}$) (see (5.5.1)), it is no longer necessary that we use squares with equal size. We only have to ensure that the partition of each square U_i ($i \in \{1, \dots, 4\}$) is invariant with respect to the relevant types of symmetry. Hence we can use the following subdivision of R_j in the second level.

The integer $m \geq 2$ is chosen as in Subsection 5.5.1 and R_j is partitioned into m^2 rectangles with a squared diameter less than η^0. This ensures again that each partition set contains at most one member of an optimal solution of (PP). The $(m-1)^2$ rectangles, which are nearest to the vertex v of U belonging to R_j, are chosen as squares with a squared diameter of $\eta^0 - \delta$, where $\delta > 0$ is a given small tolerance, i.e., we choose squares with edge-length $\sqrt{\frac{\eta^0-\delta}{2}}$. Assume that we have to select $m = 3$. The partition of each square U_i ($i \in \{1, \dots, 4\}$) is done as in Figure 5.39. Applying this strategy we obtain that the partition of the squares U_i ($i \in \{1, \dots, 4\}$) is still invariant with respect to rotations and the relevant reflections. Moreover, we choose among the m^2 partition sets $(m-1)^2$ squares as large as possible. This strategy has at least two effects on the performance of Algorithm 5.1.

First of all, we obtain $m^2 - (m-1)^2$ rectangles, which are smaller than those given by the old basic strategy. Taking the volume reduction strategy into account we can expect that this strategy is more successful. Indeed, on the one hand we can cut away more from neighboring rectangles and on the other hand, it is more likely that the smaller sets gets empty. A second effect of this new strategy is

FIGURE 5.39. New basic partitioning strategy


1.0
0.5
0.0
0.5
$= \sqrt{\eta^0 - \delta}$
(a) U_3
1.0
0.5
0.5
1.0
$= \sqrt{\eta^0 - \delta}$
(b) U_4
0.5
0.0
0.0
0.5
$= \sqrt{\eta^0 - \delta}$
(c) U_1
0.5
0.0
0.5
1.0
$= \sqrt{\eta^0 - \delta}$
(d) U_2

that the partition sets of U_i ($i \in \{1, \ldots, 4\}$), which are nearest to the vertices of the unit square, are larger. This can lead to a more successful application of the corner rules. Remember that, in addition, the dimension and the volume reduction strategies interact.

The application of this altered basic strategy led to a substantial reduction of the numerical effort for solving Problem (PP). In Table 5.5 the effort for solving the problems, where we had to choose m as 3, are displayed. It can be seen, especially, that we obtained an extraordinary reduction of the iterations for solving (PP) with 16 and 21 points. For Problem (PP) with $n = 26, 27$ the new basic strategy led in the second level nearly to the same partition as the old strategy. Note that in these situations we had $\sqrt{\frac{\eta^0 - \delta}{2}} = 0.1687$ for $n = 26$ and 0.1667 for $n = 27$, which is almost equal to $\frac{0.5}{3}$. Therefore, there was no improvement in these two cases. For the other cases we could nearly halve the effort for determining approximate solutions.

TABLE 5.5. Numerical effort with altered basic strategy

n	IT	TT	NLP	TLP	NR	MNPS
14	6,651	24.1	340	3.97	18,015	702
15	1,532	6.30	158	2.22	3,590	255
16	1,832	7.24	150	2.11	4,401	176
17	17,023	113	2,371	44.0	38,718	1,373
18	10,446	101	2,306	54.3	21,861	494
19	29,046	130	1,239	30.0	60,514	1,927
20	13,374	180	4,440	107	26,161	488
21	82,865	771	11,885	314	161,072	5,590
22	33,644	281	3,331	102	62,579	2,669
23	86,412	1,549	26,750	953	154,418	4,778
24	103,557	1,431	17,285	697	177,526	7,007
25	66,900	1,066	12,324	554	111,056	4,376
26	661,811	9,024	46,472	2,291	1,034,886	36,335
27	251,004	3,204	13,656	799	365,210	15,457

In the implementation of Algorithm 5.1, whose results are reported in Table 5.5, we also altered the selection rule of the current hyperrectangle R^{k+1} in Step VII of our method. Instead of choosing R^{k+1} among all hyperrectangles $R \in \mathcal{R}^{k+1}$ with a lower bound equal to μ^{k+1}, we select, if possible, a child of R^k with this attribute. Note that in the old strategy we applied the FIFO principle, i.e., first-in-first-out, and that the new strategy is related to the LIFO principle, i.e., last-in-first-out. Our numerical experience showed that this selection strategy nearly halved the maximal number of stored partition sets (see the cases $n = 26, 27$ in Table 5.5 and remember that in these cases the new basic strategy had almost no influence).

5.8.2. Altered Decision Criterion. In the previous section we described a criterion according to which we decide, whether we calculate an upper bound for the current hyperrectangle R^{k_p} ($p \in \{1, \ldots, l\}$) by solving a linear program or whether we choose the old upper bound μ^k, respectively the updated one $\mu_{R^{k_p}} = \min\{\mu^k, \bar{\mu}_{Rk_p}\}$ (see (5.7.1)). This criterion based on the check of the feasibility of a point $x \in U^n$ with respect to the linear subproblem (LSP') (see page 264), where the members of this point are vertices of the rectangles forming R^{k_p}. Taking the large number of possible points into account the verification of this criterion can be time-consuming. Nevertheless, the application of this strategy lead to an essential reduction of the running-times. The following cheap and simple decision criterion showed even better results.

As long as the squared diameter $d(R^{k_p})$ of the hyperrectangle R^{k_p} is not smaller than $\frac{\eta^0}{2}$, i.e, as long as there holds

$$d(R^{k_p}) \;=\; \sum_{i=1}^{n}\sum_{j=1}^{2}|L_{i_j}^{k_p} - l_{i_j}^{k_p}|^2 \;\geq\; \frac{\eta^0}{2}\,, \tag{5.8.1}$$

we do not calculate a new upper bound for Problem (SP) with respect to the set R^{k_p}. We simply set $\mu_{R^{k_p}} = \mu^k$. Moreover, if $d(R^{k_p})$ is smaller than $\frac{\eta^0}{2}$ we check additionally the old criterion.

At first glance this new criterion might be surprising, since in all strategies developed so far we never considered the hyperrectangle as a whole. We always analyzed the rectangles forming these sets. Nevertheless, this criterion worked very well and is thus at least a good heuristic. Applying this decision criterion we could almost halve again the running-times of Algorithm 5.1 for solving Problem (PP) with $n \in \{14, \ldots, 27\}$ points (see Table 5.6 and compare with the results in Table 5.5). For determining an approximate solution for the scattering problem with 16 points we had to solve only one linear program. It is interesting to note that the number of iterations did not grow substantially. Hence, the advantage of solving less linear programs was not outbalanced by the fact that we could obtain worse

TABLE 5.6. Numerical effort with altered basic strategy and altered decision criterion

n	IT	TT	NLP	TLP	NR	MNPS
14	6,777	17.0	3	0.01	18,403	691
15	1,534	3.07	2	0.01	3,596	224
16	1,878	3.49	1	0.01	4,536	177
17	17,176	56.9	52	0.37	39,057	1,478
18	11,229	32.7	27	0.16	23,733	599
19	29,149	85.5	14	0.14	60,717	1,955
20	13,491	41.2	12	0.09	26,838	706
21	83,799	371	993	11.7	162,838	6,057
22	34,772	147	14	0.21	64,851	2,759
23	87,667	382	8	0.13	154,418	5,431
24	104,684	554	29	0.25	179,524	7,853
25	67,742	412	4	0.03	112,534	4,405
26	669,709	6,107	1,883	36.7	1,051,445	40,517
27	252,517	2,187	1,021	24.6	367,854	15,648

upper bounds. Note that, if we do not solve (LSP'), we save time since we do not call *MINOS 5.4* and, additionally, we save time since we do not construct the LP-relaxation of Problem (SP).

A reason for the good performance of Algorithm 5.1 using this criterion might be the following. Our numerical experience showed that the subdivision set manipulation strategies are really successful in detecting areas of U^n, where no optimal solution exists. Moreover, they are able to significantly reduce the size of the remaining sets containing solutions of Problem (PP). However, if these sets get *small* and the best known solution is maybe not good enough, the manipulation strategies do not result in further progress. In this situation it is useful to calculate upper bounds by solving the LP-relaxation of (SP). Doing this we can obtain slight improvements of the best known solution and we diminish the distance between the lower and the upper bounds. The described criterion seems to be a good choice for detecting such situations, where the considered hyperrectangles are *small*.

We tested also several other criteria, which were not as keen as the above one. For instance, we required that each rectangle $R_i^{k_p}$ ($i \in \{1, \dots, n\}$) was subdivided twice, i.e., that there held

$$\max_{i=1,\dots,n} \max_{j=1,2} |L_{i_j}^{k_p} - l_{i_j}^{k_p}| \; < \; \sqrt{\frac{\eta^0}{2}} \, .$$

Using these criteria the running-times always increased on average.

On the other hand we tested additionally a variant of Algorithm 5.1, where we never calculate an upper bound by solving a linear program. We always chose the simple upper bound $\mu_{R^{k_p}} = \min\{\mu^k, \bar{\mu}_{R^{k_p}}\}$ with $\bar{\mu}_{R^{k_p}}$ given as in (5.7.1). Doing this the running-times explode for the most examples. Hence, even though we have in comparison with the number of considered hyperrectangles only a small number of linear problems to solve, the solution of these problems is necessary in order to obtain an efficient method for solving the point scattering problem.

5.8.3. Solutions of Problem (PP) with more than 27 Points. In the previous section we pointed out that we were not able to solve Problem (PP) with more than 27 points, at least with the version of Algorithm 5.1 used there. The improvements of Algorithm 5.1 developed in the present section enabled us to solve such problems. However, in comparison to the cases $n \leq 27$ the running-times still explode.

In order to obtain approximate solutions for Problem (PP) with $n > 27$ we used a *"parallelized"* variant of Algorithm 5.1. At first we generated all possible hyperrectangles $R = R_1 \times \ldots \times R_n$ with the property that each rectangle R_j was subdivided once, i.e., coincides with one of the squares U_j ($j \in \{1, \ldots, 4\}$, see (5.5.1)). After this we used each of these hyperrectangles as the initialization set for Algorithm 5.1. In this way we could use several machines in order to solve Problem (PP). Apart form the *SUN ULTRA 60* workstations used till now, we additionally applied *SUN Server 1000* workstations. These machines are – with our code – on average 4 times slower.

In order to avoid excessive storage requirements we used the depth-first-search-strategy mentioned in Remark 5.7.2(b). Since we did not know the coordinates of the best known solutions for Problem (PP) with $n > 27$ we did not initialize x^0. We initialized only η^0.

REMARK 5.8.1. The coordinates of $\bar{x}$, which is used for the initialization of x^0, are not substantial for Algorithm 5.1. This methods needs only a good approximation for η^0. Therefore, it is sufficient if we set η^0 without knowing the coordinates of $\bar{x}$. Moreover, if we initialize η^0 with $\bar{\eta} - \epsilon$, where $\bar{\eta}$ is the best published value, we can guarantee that Algorithm 5.1 delivers the coordinates of an ϵ-optimal solution.

Since we used the best known solutions given in [NO97] we had to choose $m = 4$ in the second level of our basic partitioning strategy in order to solve Problem (PP) with more than 27 points. According to our new partitioning strategy the rectangles U_i ($i \in \{1, \ldots, 4\}$) were hence partitioned into 9 squares with edge-length $\sqrt{\frac{\eta^0 - \delta}{2}}$ and 7 additional rectangles. Thus the number of possibilities for setting $R_j^{k_i}$ ($j \in \{1, \ldots, n\}$, $i \in \{1, \ldots, l\}$, $k \in \mathbb{N}$) in Step III of Algorithm 5.1 was substantially larger than by choosing $m = 3$, as it was the case for $n \leq 27$. With respect to our subdivision set manipulation strategies and the fact that there were 7 *small* rectangles we still hoped to be able to solve (PP) with acceptable computational effort. However, even the fastest one (see case $n = 30$ in Table 5.7) could not be solved within one day, and hence we could not expect to solve many cases with $n > 27$.

The numerical effort for solving Problem (PP) with $n \in \{28, 29, 30, 31\}$ is shown in Table 5.7. We used again the accuracy $\epsilon = 10^{-5}$, except for $n = 31$ we applied $\epsilon = 5 * 10^{-5}$. The abbreviations are the same as before. However, the running-times are given in hours. The additional column NP shows the number of possible hyperrectangles after the first level, i.e., the number of different problems we solved in order to obtain an approximate solution of the corresponding point scattering problem. Since we solved each problem using different machines we

TABLE 5.7. Numerical effort for solving (PP) with $n > 27$

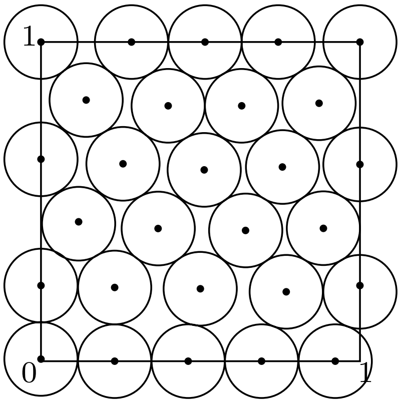
n	NP	IT	TT	NLP	TLP	NR	$\eta^\star$
28	29	86,848,406	205.44	7,149	0.04	206,979,450	0.0531427
29	21	38,423,801	103.23	109	0.01	96,339,168	0.0514739
30	16	11,034,381	27.71	8	0.00	24,955,286	0.0503987
31	9	76,263,071	229.39	4,802,901	42,57	164,774,004	0.047324

added the effort needed for each process. In order to obtain comparable running-times we scaled the times obtained with the *SUN Server 1000* workstations with the speedup-factor 4.0 mentioned before. Thus the running-times displayed in Table 5.7 are approximations of the necessary time for completely solving these problems on a *SUN ULTRA 60* workstation.

It is interesting to note that even though there are nearly double as much possibilities in the second partitioning level as for the cases $n \leq 27$, the number of considered hyperrectangles in Step IV is still less than three times the number of iterations. This corroborates again the success of the special features of the subdivision strategy. As in the cases considered in the previous section we did not determine solutions of Problem (PP) (see column $\eta^\star$) with a larger minimum squared pairwise distance than the best known so far.

Our numerical experience for the case $n = 31$ showed that in this case a symmetry avoiding strategy in the third level could improve the numerical performance of Algorithm 5.1. Note that for the cases $n \leq 27$ we pointed out that it was not useful to use this strategy in deeper levels than the second one. In Figures 5.40-5.42 we show again the arrangements of the calculated ϵ-optimal solutions together with

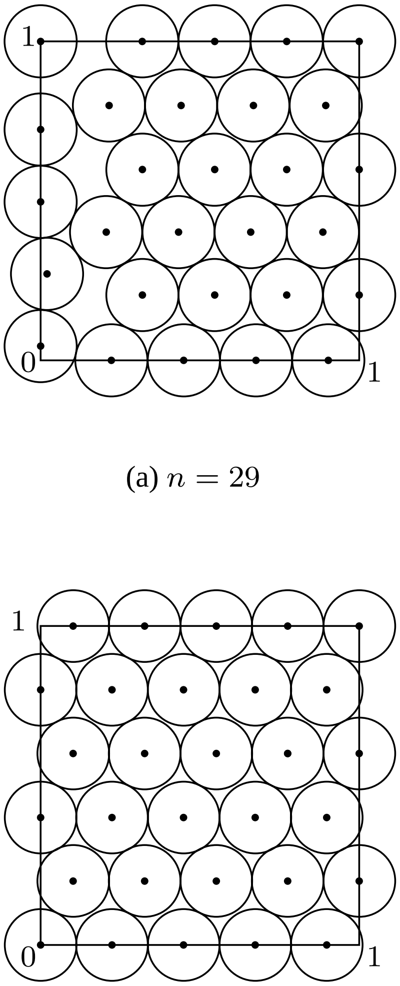
FIGURE 5.40. Solution for $n = 28$



$x_1 = (0.0000, 1.0000)$	$x_2 = (1.0000, 1.0000)$
$x_3 = (0.0000, 0.0060)$	$x_4 = (1.0000, 0.2170)$
$x_5 = (0.0000, 0.6333)$	$x_6 = (1.0000, 0.6163)$
$x_7 = (0.0000, 0.2365)$	$x_8 = (0.2305, 0.0000)$
$x_9 = (0.9221, 0.0000)$	$x_{10} = (0.2833, 1.0000)$
$x_{11} = (0.7444, 1.0000)$	$x_{12} = (0.2305, 0.2305)$
$x_{13} = (0.4601, 0.0000)$	$x_{14} = (0.4996, 0.2273)$
$x_{15} = (0.1175, 0.4349)$	$x_{16} = (0.3672, 0.4161)$
$x_{17} = (0.6915, 0.0000)$	$x_{18} = (0.7695, 0.2170)$
$x_{19} = (0.6417, 0.4088)$	$x_{20} = (0.8847, 0.4166)$
$x_{21} = (0.2569, 0.6185)$	$x_{22} = (0.1417, 0.8181)$
$x_{23} = (0.3986, 0.8003)$	$x_{24} = (0.5119, 0.5994)$
$x_{25} = (0.7569, 0.6085)$	$x_{26} = (0.6291, 0.8003)$
$x_{27} = (0.5139, 1.0000)$	$x_{28} = (0.8722, 0.8081)$

their coordinates.

FIGURE 5.41. Solutions for $n = 29, 30$



(a) $n = 29$

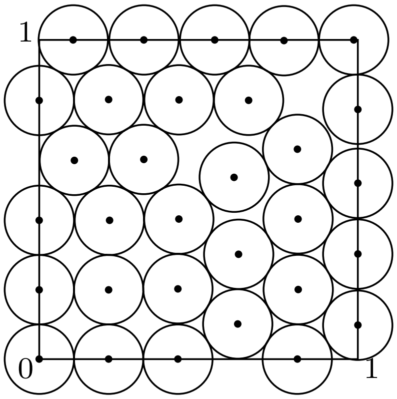
$x_1 = (0.0000, 1.0000)$	$x_2 = (0.0000, 0.0450)$
$x_3 = (1.0000, 0.2051)$	$x_4 = (1.0000, 0.5981)$
$x_5 = (1.0000, 1.0000)$	$x_6 = (0.0000, 0.7237)$
$x_7 = (0.0000, 0.4969)$	$x_8 = (0.2224, 0.0000)$
$x_9 = (0.9030, 0.0000)$	$x_{10} = (0.3193, 1.0000)$
$x_{11} = (0.0204, 0.2709)$	$x_{12} = (0.3194, 0.2051)$
$x_{13} = (0.4493, 0.0000)$	$x_{14} = (0.2059, 0.4016)$
$x_{15} = (0.4328, 0.4016)$	$x_{16} = (0.6761, 0.0000)$
$x_{17} = (0.5462, 0.2051)$	$x_{18} = (0.7731, 0.2051)$
$x_{19} = (0.6597, 0.4016)$	$x_{20} = (0.8860, 0.4016)$
$x_{21} = (0.3194, 0.5981)$	$x_{22} = (0.2140, 0.7990)$
$x_{23} = (0.4409, 0.7990)$	$x_{24} = (0.5462, 0.5981)$
$x_{25} = (0.7731, 0.5981)$	$x_{26} = (0.6678, 0.7790)$
$x_{27} = (0.5462, 1.0000)$	$x_{28} = (0.8947, 0.7790)$
$x_{29} = (0.7731, 1.0000)$	

(b) $n = 30$

$x_1 = (1.0000, 0.2000)$	$x_2 = (0.0000, 0.8000)$
$x_3 = (0.0000, 0.4000)$	$x_4 = (0.0000, 0.0000)$
$x_5 = (1.0000, 0.6000)$	$x_6 = (1.0000, 1.0000)$
$x_7 = (0.8980, 0.0000)$	$x_8 = (0.1020, 1.0000)$
$x_9 = (0.2245, 0.0000)$	$x_{10} = (0.1020, 0.2000)$
$x_{11} = (0.4490, 0.0000)$	$x_{12} = (0.3265, 0.2000)$
$x_{13} = (0.2245, 0.4000)$	$x_{14} = (0.4490, 0.4000)$
$x_{15} = (0.6735, 0.0000)$	$x_{16} = (0.5510, 0.2000)$
$x_{17} = (0.7755, 0.2000)$	$x_{18} = (0.6735, 0.4000)$
$x_{19} = (0.8980, 0.4000)$	$x_{20} = (0.1020, 0.6000)$
$x_{21} = (0.3265, 0.6000)$	$x_{22} = (0.2245, 0.8000)$
$x_{23} = (0.4490, 0.8000)$	$x_{24} = (0.3265, 1.0000)$
$x_{25} = (0.5510, 0.6000)$	$x_{26} = (0.7755, 0.6000)$
$x_{27} = (0.6735, 0.8000)$	$x_{28} = (0.5510, 1.0000)$
$x_{29} = (0.8980, 0.8000)$	$x_{30} = (0.7755, 1.0000)$

We also tried to solve even larger problems. However, the point scattering problem with 32 points could not be solved within two weeks. Hence we do not expect that the current version of Algorithm 5.1 is able to solve Problem (PP) with $n > 31$ within several days, and we did not try this until now. Nevertheless, in this section we saw that slight changes of some strategies can essentially improve the numerical performance of Algorithm 5.1. Consequently, we hope that it is possible to modify the presented strategies as well as to develop new strategies in order to

FIGURE 5.42. Solution for $n = 31$

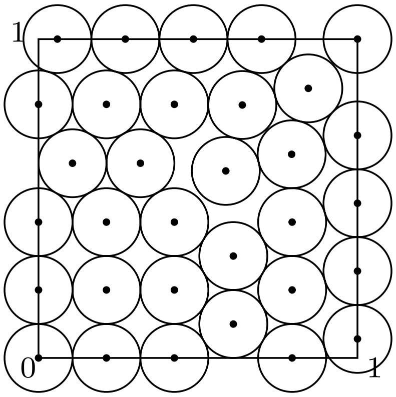


$x_1 = (1.0000, 0.7828)$	$x_2 = (1.0000, 0.1061)$
$x_3 = (0.0000, 0.8101)$	$x_4 = (0.0000, 0.4351)$
$x_5 = (1.0000, 0.5497)$	$x_6 = (1.0000, 0.3292)$
$x_7 = (0.9870, 1.0000)$	$x_8 = (0.8101, 0.0000)$
$x_9 = (0.1061, 1.0000)$	$x_{10} = (0.4351, 0.0000)$
$x_{11} = (0.5496, 1.0000)$	$x_{12} = (0.3290, 1.0000)$
$x_{13} = (0.0000, 0.0000)$	$x_{14} = (0.2175, 0.0000)$
$x_{15} = (0.0000, 0.2175)$	$x_{16} = (0.2175, 0.2175)$
$x_{17} = (0.4351, 0.2206)$	$x_{18} = (0.2206, 0.4351)$
$x_{19} = (0.4382, 0.4382)$	$x_{20} = (0.6226, 0.1103)$
$x_{21} = (0.8132, 0.2176)$	$x_{22} = (0.6257, 0.3279)$
$x_{23} = (0.8125, 0.4394)$	$x_{24} = (0.1103, 0.6226)$
$x_{25} = (0.3279, 0.6257)$	$x_{26} = (0.2175, 0.8132)$
$x_{27} = (0.4393, 0.8125)$	$x_{28} = (0.6115, 0.5696)$
$x_{29} = (0.8107, 0.6570)$	$x_{30} = (0.6568, 0.8107)$
$x_{31} = (0.7685, 0.9979)$	

further improve the suggested method. This might lead to an approach, which is even able to solve larger problems with acceptable effort. Recognize that Problem (PP) with $n > 30$ is – from a deterministic global optimization point of view – a huge problem. The small numbers of linear programs, which had to be solved during the execution of Algorithm 5.1 (see the corresponding columns in Table 5.6 and Table 5.7), let expect that a further modification of the upper bounds will not lead to a faster approach. The key for the acceleration of Algorithm 5.1 are the subdivision set manipulation strategies.

Let us finish this chapter with a good solution of Problem (PP) with 32 points

FIGURE 5.43. Good solution for $n = 32$



$x_1 = (1.0000, 1.0000)$	$x_2 = (1.0000, 0.0594)$
$x_3 = (0.0000, 0.7953)$	$x_4 = (1.0000, 0.6987)$
$x_5 = (0.0000, 0.4262)$	$x_6 = (0.7953, 0.0000)$
$x_7 = (0.0594, 1.0000)$	$x_8 = (0.6987, 1.0000)$
$x_9 = (0.4262, 0.0000)$	$x_{10} = (0.0000, 0.0000)$
$x_{11} = (0.2131, 0.0000)$	$x_{12} = (0.0000, 0.2131)$
$x_{13} = (0.2131, 0.2131)$	$x_{14} = (0.4262, 0.2131)$
$x_{15} = (0.2131, 0.4262)$	$x_{16} = (0.4262, 0.4262)$
$x_{17} = (0.6108, 0.1066)$	$x_{18} = (0.7953, 0.2131)$
$x_{19} = (1.0000, 0.2725)$	$x_{20} = (0.6108, 0.3197)$
$x_{21} = (0.7953, 0.4262)$	$x_{22} = (1.0000, 0.4856)$
$x_{23} = (0.1066, 0.6108)$	$x_{24} = (0.3197, 0.6108)$
$x_{25} = (0.2131, 0.7953)$	$x_{26} = (0.2725, 1.0000)$
$x_{27} = (0.4262, 0.7953)$	$x_{28} = (0.4856, 1.0000)$
$x_{29} = (0.5872, 0.5872)$	$x_{30} = (0.7938, 0.6393)$
$x_{31} = (0.6393, 0.7938)$	$x_{32} = (0.8459, 0.8459)$

detected during our numerical tests. The solution displayed in Figure 5.43 has a minimum squared pairwise distance of $\min_{1 \le i < j \le 32} \|x_i - x_j\|_2^2 = 0.0454068$, which is slightly better than the one given in [NO97] (0.04540409). This is the only case, where we detected by applying Algorithm 5.1 a better solution than the best known so far.

CHAPTER 6

Conclusion

We would like to complete this doctoral thesis with a short review of the topics we treated. What have we reached and which questions are still to be answered?

The main aim of this dissertation was the development and the theoretical as well as numerical examination of solution methods for so-called *nonconvex all-quadratic optimization problems*, i.e., for problems of type

$$\begin{aligned} \min\ x^TQ^0x + (d^0)^Tx & \\ x^TQ^lx + (d^l)^Tx + c^l \ \leq\ 0 & \qquad l = 1, \ldots, p \\ x\ \in P\ , & \end{aligned} \tag{QP}$$

with $Q^l \in \mathbb{R}^{n\times n}$ symmetric, $d^l \in \mathbb{R}^n$ ($l = 0, \ldots, p$), $c^l \in \mathbb{R}$ ($l = 1, \ldots, p$) and $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ a non-empty, full-dimensional polytope with $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. We proposed two, respectively three new approaches for solving this class of global optimization problems.

The first method was developed for the solution of so-called *unary problems*. This class of optimization problem was of interest, since each problem of type (QP) can be transformed into a unary program. With some technical effort we derived several convergent solution approaches for this type of global optimization problems. Hence we overcome the theoretical deficiency of an outer approximation scheme given by Ramana [RAM93], which was the only solution approach for unary problems known so far. Our algorithms are combinations of outer approximations and – branch-and-bound like – successive subdivisions of the feasible region. One variant of these new methods uses a regular n-simplex with all its vertices on the boundary of the Euclidean unit ball. Even though the properties of such an n-simplex are known in the literature, to the author's knowledge there is no explicit construction of such a set – except in the present work and in [HR98].

Unfortunately, the indirect approach for solving (QP) via the solution of the corresponding unary program is only of theoretical interest. The numerical results showed that this method is not able to solve all-quadratic problems with acceptable computational effort. Excessive numerical effort was needed in order to solve the unary problems resulting from the transformation of the all-quadratic problems belonging to our randomly generated test set.

The numerical effort for solving a unary problem depends substantially on the structure of the affine matrix mapping forming the single nonlinear constraint. For unary problems resulting from the transformation of all-quadratic problems this mapping has an unpleasant structure. If unary problems with an easier matrix mapping are considered it is likely that our methods show a substantially better numerical performance. Moreover, in such a case it could even be interesting to change some features of Algorithm 2.3 in order to further improve its numerical performance (see the considerations at the end of Chapter 2). However, another application of unary problems, which could lead to a simpler matrix mapping, is not known to our knowledge.

As mentioned before, we developed a new method for the solution of unary problems since the convergence of the outer approximation scheme proposed by Ramana is not provable. This is a theoretical problem of several algorithms, which base on cutting planes, developed for global optimization problems (see, e.g., [HT96B]). It might be that the ideas used in Chapter 2 in order to obtain a convergent algorithm can also be applied for other problem classes, where the convergence of corresponding outer approximation schemes is not known. Hence, there are some theoretically interesting aspects of the content of Chapter 2, even though we did not reach our main goal to obtain practicable solution methods for problems of type (QP).

The second solution approach for Problem (QP) suggested in Chapter 3 was more successful with respect to this main intention of the present thesis. The presented simplicial branch-and-bound method showed a good numerical performance, at least for the solution of arbitrary problems of type (QP) with a dimension less than 10. Beside the rectangular branch-and-bound algorithm introduced in [AKLV95] our simplicial method belongs to the rare approaches in the literature, which consider Problem (QP) directly. Other solution approaches for all-quadratic problems mostly interpret this type of programs as a special instance of a more general problem class, like bilinear problems [AK92], polynomial problems [ST92], problems involving biconvex functions [FV93B], general d.c. problems (see Chapter 4) or – as we did in Chapter 2 – unary problems [RAM93].

As long as an exhaustive subdivision rule for the n-simplices considered in our algorithm is used the convergence of our method can be ensured. Hence, this simplicial algorithm has the same theoretical properties as the comparable rectangular approach by Al-Khayyal et al. [AKLV95]. Moreover, numerically, our method often outperforms this rectangular approach, in particular when all-quadratic problems with more quadratic constraints than the dimensions, i.e., $p > n$, have to be solved. The complexity of the LP-relaxations used in our approach, i.e., their dimension and number of linear constraints, depends linearly on p and n. In contrast to this the LP-relaxations applied in Al-Khayyal et al.'s method for the calculation of lower bounds have a dimension of $(p+2)n$ and $4(p+1)n+p+m$ linear constraints. The less complex relaxations in our approach are the main reason for the better numerical performance.

If we are interested in a practicable approach for solving (QP), convergence of such an algorithm is not sufficient. In addition we need that the method is finite. However, this can only be obtained, if we are satisfied with approximate solutions. In particular, we have to be satisfied with solutions of Problem (QP), which are approximately feasible, i.e., fulfill the constraints up to a prespecified tolerance, and which are, additionally, approximately optimal. The solution methods considered in this thesis can determine such approximate solutions in finite time. In contrast to other global optimization problems, like concave minimization, it is in general not possible to require that the determined approximate solution of Problem (QP) is at least feasible. Note that the problem of detecting a feasible point for this type of problems is as hard as the solution of (QP) itself. However, under additional assumptions the feasibility of calculated solutions could be guaranteed. For instance, if we require that the feasible region of (QP) contains a ball with known radius, then it is possible to choose the accuracies in our simplicial method such that Algorithm 3.1 determines a feasible point in finite time. Our first method and the generalization of the simplicial branch-and-bound approach considered in Chapter 4 can also be adapted in this way. However, the verification of such a strict assumption is again a hard problem, unless the examined instance of Problem (QP) has a special structure.

In the definition of our simplicial branch-and-bound method there are still some features, which could be modified or changed in order to affect the numerical behavior of this approach. At the end of Chapter 3 we saw, for example, that the exploitation of the fact that the LP-relaxations applied in our approach are not uniquely determined can improve the performance of Algorithm 3.1. Another feature of this method, which could be changed, is the subdivision rule. We use *bisection*, where a partition of the current n-simplex is performed by a radial subdivision with respect to the midpoint of the longest edge. This subdivision rule is exhaustive. The same holds for the *generalized bisection* mentioned on page 127. Hence, this rule could also be applied without altering the theoretical properties of our approach.

In the context of simplicial branch-and-bound methods for the minimization of concave functions with respect to polytopes, i.e., for *concave minimization problems*, the so-called *ω-subdivision* is favored by some authors. In this rule the current n-simplex S is partitioned into up to $n + 1$ subsimplices by applying a radial subdivision with respect to the point, where the optimal solution of the LP-relaxation on the set S is attained. Using this rule one hopes to obtain better numerical results, since the subdivision point is more related to the problem than the midpoint of the longest edge used in the bisection rule. However, it was an open question, whether simplicial branch-and-bound methods employing only ω-subdivisions are convergent, since this subdivision rule is not necessarily exhaustive.

We were first of all interested in convergent solution approaches for Problem (QP). Therefore, we had to answer this open question, if we wanted to apply the ω-subdivision rule in our simplicial branch-and-bound method. The ideas used in Chapter 3 in order to develop a solution approach for (QP) could analogously be used for deriving a simplicial branch-and-bound algorithm for the solution of so-called *generalized d.c. problems* containing the class of all-quadratic problems as well as the class of concave minimization problems. Therefore, we tried to answer the open question mentioned above for the generalized algorithm introduced in Chapter 4, which is able to solve such d.c. programs.

We proved that for general d.c. problems and in particular for general all-quadratic problems our algorithm can fail to converge, if only ω-subdivisions are employed. However, if this method is applied for concave minimization problems or for problems with a d.c. objective function – consisting of a quadratic convex part and a strictly concave part – and with concave and linear constraints, the convergence even with ω-subdivisions can be guaranteed. This was the main result in Chapter 4. Note that the convergence of this method could only be guaranteed in the sense that this approach detects for arbitrary accuracies ϵ, $\delta > 0$ either the emptiness of the feasible region or an (ϵ, δ, 0)-solution of the considered problem in finite time. This convergence result was theoretically weaker than the other convergence results examined in this thesis. Nevertheless, – from a practical point of view – all results have the same quality. They ensure the finiteness of our methods, if we are satisfied with approximate solutions.

Apart from these convergence results we were, furthermore, able to prove that this method with ω-subdivisions delivers in finite time even the optimal solution of a concave minimization problem, if two additional assumptions are fulfilled. It does not seem that this finiteness result can be extended to more general problem classes, since for the proofs it was essential that the feasible set is a polytope. On the other hand, it is an interesting question whether the additional assumptions could be weakened without losing the finiteness result.

We also examined the numerical performance of the introduced generalized simplicial branch-and-bound Algorithm 4.1 with respect to problems of type (QP). Since this algorithm uses convex relaxations instead of LP-relaxations applied by Algorithm 3.1, we were at first interested in a numerical comparison of both approaches only using bisections. Note that the generalized algorithm is always convergent, if an exhaustive subdivision rule is used – as it is the case for Algorithm 3.1. We observed that the version of Algorithm 4.1, which employs only bisections, can be expected to be numerically more efficient than Algorithm 3.1, at least as long as a solver for the convex relaxations is used, which exploits the quadratic structure of the involved functions.

The main aim for considering ω-subdivisions in Chapter 4 was the hope to obtain an algorithm showing a better numerical performance than a method, which simply chooses bisection. As mentioned before, we know that in general the convergence of an approach, which employs only ω-subdivisions, cannot be ensured. However, our theoretical results derived in Chapter 4 enabled us to develop a mixed subdivision strategy (MGWSR) – consisting of bisections and ω-subdivisions – leading to a convergent approach for solving general d.c. problems. Note that (MGWSR) is different form the mixed strategy used in the so-called *normal* simplicial branch-and-bound algorithms. Unfortunately, the numerical tests using this mixed strategy (MGWSR) and variants of it were really disappointing. Even though Algorithm 4.1 using (MGWSR) had a good performance for a few test examples, on average the application of this rule results in a substantially worse numerical performance than the use of bisection. We did not find a strategy based on

(MGWSR), which has the best numerical performance in all test examples, i.e., which is the fastest one, if (MGWSR) leads faster to a solution than bisection as well as if bisection is faster than (MGWSR). It is hence still an open question whether there exists a subdivision rule for simplicial branch-and-bound methods, which always shows the best numerical behavior. We believe that such a rule does not exist in general. Taking the structure of special problem instances into account it might be non the less possible to develop a subdivision rule, which has this property, at least as long as the resulting algorithm is applied for the solution of these instances.

Till Chapter 4 we developed two new approaches for the solution of the general form of Problem (QP). Note that Algorithm 3.1 can be interpreted as a special case of Algorithm 4.1. In the introduction of this thesis (see Section 1.1) we saw that there are many applications of this type of global optimization problems. Consequently, it was interesting to examine such a special instance of Problem (QP). We chose the problem of packing $n \in \mathbb{N}$ equal and non-overlapping circles with maximum radius into the two-dimensional unit square, which we called the *packing problem*. The optimal solutions of this problem with up to 20 circles are reported to be known. Hence, we had to solve an all-quadratic problem (see page 189) with a dimension higher than 40, if we wanted to determine new global optimal solutions. From a global optimization point of view, such problems are very large. Thus, it was not really surprising that our general methods for (QP) developed so far fail to solve the packing problem with more than 20 circles and acceptable effort.

In the considerations of our general schemes we often claimed that the exploitation of the structure of special problem instances can improve their numerical performance. This was particularly the case with the packing problem. We suggested a rectangular branch-and-bound algorithm, which was able to solve the packing problem with up to 27 circles within two hours.

We exploited the special structure of the packing problem, respectively of the all-quadratic formulation (PP) of the equivalent *point scattering problem* in different ways. First of all we derived some new theoretical results showing the existence of a solution of this problem with a special behavior on the boundary of the unit square. In particular, we proved that there is an optimal solution $x^\star$ consisting of n two-dimensional members $x_i^\star$ ($i = 1, \dots, n$) such that for each vertex v of the unit square there holds: Either the vertex v is itself a member of $x^\star$ or there are two members of this solution belonging to the boundary lines of the unit square forming this vertex, which have exactly the optimal distance from each other. Even though we do not know an optimal solution of the point scattering problem with the property that no vertex of the unit square is a member of this solution, we were not able to prove this fact. We could not show that at least one vertex has to belong to an optimal solution of the point scattering problem. This is still an open question. The existence of optimal solutions with the proven behavior on the boundary of the unit square could be used in order to reduce the number of possible solutions of the considered problem, which our algorithm has to look for.

Apart from the derivation of these theoretical results we exploited the structure of the packing problem in the construction of LP-relaxations, which are needed for the calculation of bounds. These special relaxations are better than those obtained by general approaches. We could also use the structure of the considered instance of (QP) in order to develop special subdivision strategies for the relevant hyperrectangles. Moreover, we were able to derive further powerful subdivision set manipulation strategies. Note that the derivation of such strategies is mostly not possible, if general problem classes are considered.

The combination of all these adjustments of a general rectangular branch-and-bound scheme to a special problem instance led to a really successful solution approach for the packing problem. The proposed algorithm is able to prove the ϵ-optimality of determined approximate solutions. Hence, this approach could be used as a computer aided proof, since the optimal solutions of the packing problem for more than 20 circles are mostly not known and since our numerical experience showed that Algorithm 5.1 is able to solve problems with these sizes and acceptable effort. We pointed out that our current implementation of Algorithm 5.1 cannot be used without reservation as a computer aided proof. Nevertheless, it is possible to adapt this implementation such that the required precision can be reached.

The performance of the proposed method depends essentially on the quality of the solutions of the examined problem known in advance. Algorithm 5.1 is not applicable in order to determine new solutions of the packing problem without knowing good initial approximations. The main advantage of our method is the possible guarantee of the ϵ-optimality of determined points mentioned before. Nevertheless, this approach can also detect better solutions than the best known so far, as we saw for the case with 32 circles.

The suggested strategies for improving the performance of Algorithm 5.1 are surely not yet the best ones. We believe that there are still further possible improvements, especially for the subdivision set manipulation strategies, such that even larger examples of the packing problem can be solved globally.

The last chapter of this thesis showed that an adjustment of a general solution scheme to a special problem instance can significantly improve the numerical performance of the method applied for the solution of this instance. Moreover, we saw that in global optimization general approaches are – from a practical point of view – often not able to solve problems resulting from applications, since the sizes of these problems are too large. Another interesting aspect of Chapter 5 is that the key for the acceleration of the branch-and-bound Algorithm 5.1 was not the development of good relaxations for calculating bounds. The subdivision set manipulation strategies were decisive. Also for other problem instances it is possible that the examination of the structure of the feasible region and the resulting derivation of subdivision set manipulation strategies is – with respect to the numerical performance of a branch-and-bound method – more successful than the development of special bounds.

APPENDIX A

Proofs for Section 4.4

Before expatiating the longer and more technical proofs of some results proposed in Section 4.4 we first establish a lemma, which will ease our work. The statement of this lemma does not depend on the problem class which we would like to solve with Algorithm 4.1.

LEMMA A.1. *Let $\{S^k\}_{k\in\mathbb{N}}$ be an infinite nested simplex sequence generated by Algorithm 4.1 with Properties (4.4.2.a) and (4.4.2.b), and let $\{x^k\}_{k\in\mathbb{N}}$ be a point sequence with $x^k \in S^k$ ($k \in \mathbb{N}$). Choose $\lambda^k \in B_n$ such that, for all $k \in \mathbb{N}$,*

$$x^k \;=\; \sum_{i=0}^{n} \lambda_i^k v_i^k \;. \tag{A.0.1}$$

Assume that there is an index $i' \in \{0, \dots, n\}$ such that the vertices $v_{i'}^k$ ($k \in \mathbb{N}$) change infinitely often, i.e.,

$$|\{k \in \mathbb{N} :\; v_{i'}^{k+1} \neq v_{i'}^k\}| \;=\; \infty \,, \tag{A.0.2}$$

and such that there holds

$$\lambda_{i'}^k \;\not\to\; 0 \;\; (k \to \infty) \,. \tag{A.0.3}$$

Then there exist an index $l \in \{0, \dots, p\}$, a positive real number τ and a subsequence $\{k_q\}_{q\in\mathbb{N}}$ of $\{k\}_{k\in\mathbb{N}}$ satisfying, for all $q \in \mathbb{N}$,

$$\varphi^l_{S^{k_q}}(x^{k_q}) \;\geq\; \varphi^l_{S^{k_q-1}}(x^{k_q}) + \tau \,. \tag{A.0.4}$$

In particular, there holds

$$\varphi^0_{S^{k_q}}(x^{k_q}) \;\geq\; \varphi^0_{S^{k_q-1}}(x^{k_q}) + \tau \,,$$

if all elements of the vertex sequence $\{v_{i'}^k\}_{k\in\mathbb{N}}$ are (δ, 0)-feasible.

PROOF FOR (DCP$_1$) AND (DCP$_2$): The boundedness of the sequence $\{\lambda_{i'}^{k}\}_{k\in\mathbb{N}}$ implies that there is a convergent subsequence $\{\lambda_{i'}^{k_q}\}_{q\in\mathbb{N}}$ satisfying

$$\lambda_{i'}^{k_q} \ \rightarrow \ 2\nu \ \ (q\rightarrow\infty)$$

for a positive real value ν. It follows that there exists a number $Q\in\mathbb{N}$ with the property that, for all $q\geq Q$, there holds

$$\lambda_{i'}^{k_q} > \nu \, .$$

We assume, without loss of generality, that $Q = 1$. Regarding (A.0.2) we are able to choose the subsequence $\{\lambda_{i'}^{k_q}\}_{q\in\mathbb{N}}$ in a way such that each member of the corresponding vertex sequence $\{v_{i'}^{k_q}\}_{q\in\mathbb{N}}$ is different from his successor, i.e.,

$$\forall q\in\mathbb{N} \ \ v_{i'}^{k_{q+1}} \neq v_{i'}^{k_q} \ .$$

Therefore, we know that, for all $q\in\mathbb{N}$, there exists an index $k_q(i') < k_q$, $k_q(i')\geq k_{q-1}$ – not necessarily belonging to the subsequence $\{k_q\}_{q\in\mathbb{N}}$ – with

$$v_{i'}^{k_q} = \omega(S^{k_q(i')}) \, .$$

Thus, for each $q\in\mathbb{N}$, there holds

$$\begin{aligned} S^{k_q(i')+1} = [v_0^{k_q(i')},\dots,v_{i'-1}^{k_q(i')},v_{i'}^{k_q},v_{i'+1}^{k_q(i')},\dots,v_n^{k_q(i')}] \quad \text{and} \\ S^{k_q}\subseteq S^{k_q(i')+1} \, . \end{aligned} \qquad \text{(A.0.5)}$$

It follows that each vertex $v_j^{k_q}$ $(j\in\{0,\dots,n\}\backslash\{i'\})$ of the n-simplex S^{k_q} can be represented as a convex combination of the vertices of $S^{k_q(i')+1}$, i.e., there exists a vector $\gamma_j^{k_q(i')}\in B_n$ satisfying

$$v_j^{k_q} = \sum_{i=0,i\neq i'}^{n} (\gamma_j^{k_q(i')})_i v_i^{k_q(i')} + (\gamma_j^{k_q(i')})_{i'} v_{i'}^{k_q} \, . \qquad \text{(A.0.6)}$$

By substituting each $v_j^{k_q}$ $(j\in\{0,\dots,n\}\backslash\{i'\})$ in (A.0.1) with (A.0.6) we obtain a vector $\alpha^{k_q}\in B_n$ satisfying

$$x^{k_q} = \sum_{i=0,i\neq i'}^{n} \alpha_i^{k_q} v_i^{k_q(i')} + \alpha_{i'}^{k_q} v_{i'}^{k_q}$$

with

$$\alpha_{i'}^{k_q} = \lambda_{i'}^{k_q} + \underbrace{\sum_{j=0, j\neq i'}^{n} \lambda_j^{k_q} (\gamma_j^{k_q(i')})_{i'}}_{\geq 0} \ \geq\ \nu \, .$$

Denote by

$$Fea \ :=\ \{v_{i'}^{k_q} \,:\, v_{i'}^{k_q} \text{ is } (\delta,\, 0)\text{-feasible, } q \in \mathbb{N}\}$$

the set of all $(\delta,\, 0)$-feasible elements of the sequence $\{v_{i'}^{k_q}\}_{q\in\mathbb{N}}$. Note that there holds $Fea = \{v_{i'}^{k_q} \,:\, q \in \mathbb{N}\}$ in the case of problem type (DCP$_1$). If the set Fea contains an infinite number of elements, then we can assume, without loss of generality, that each vertex $v_{i'}^{k_q}$ ($q \in \mathbb{N}$) is $(\delta,\, 0)$-feasible. From Lemma 4.4.1 we obtain

$$\begin{aligned} \varphi^0_{S^{k_q}}(x^{k_q}) & \underset{\text{(A.0.5) and (4.4.3)}}{\geq} \varphi^0_{S^{k_q(i')+1}}(x^{k_q}) \\ & \underset{\text{Lemma 4.4.1}}{\geq} \varphi^0_{S^{k_q(i')}}(x^{k_q}) + \underbrace{\alpha_{i'}^{k_q}}_{\geq \nu} \epsilon \\ & \underset{S^{k_q-1} \supset S^{k_q(i')}}{\geq} \varphi^0_{S^{k_q-1}}(x^{k_q}) + \underbrace{\nu\epsilon}_{:=\tau} \, . \end{aligned}$$

If Fea contains a finite number of elements, then there exists an index $l \in \{1, \ldots, p\}$ such that the constraint

$$g^l(x) + f^l(x) \ \leq\ \delta \tag{A.0.7}$$

is violated infinitely often by the elements of the sequence $\{v_{i'}^{k_q}\}_{q\in\mathbb{N}}$. In this case we can assume, again without loss of generality, that each vertex $v_{i'}^{k_q}$ ($q \in \mathbb{N}$) violates the constraint (A.0.7) for a fixed $l \in \{1, \ldots, p\}$. Taking Lemma 4.4.1 into account (see, in particular, the proof of this lemma) we obtain now in an analogous way

$$\varphi^l_{S^{k_q}}(x^{k_q}) \ \geq\ \varphi^l_{S^{k_q-1}}(x^{k_q}) + \underbrace{\nu\delta}_{:=\tau} \, ,$$

which completes the proof. ■

REMARK A.1. The proof of Lemma A.1 in connection with the proof of Lemma 4.4.1 shows that, in addition, there exist a further subsequence $\{k_q(i')\}_{q\in\mathbb{N}}$ of

$\{k\}_{k\in\mathbb{N}}$ and a positive real value ν with the properties, for all $q \in \mathbb{N}$,

$$k_{q-1} \leq k_q(i') < k_q \quad , \quad v_{i'}^{k_q} = \omega(S^{k_q(i')}) = v_{i'}^{k_q(i')+1} \tag{A.0.8.a}$$

and

$$\begin{aligned} \lambda_{i'}^{k_q} &\geq \nu \quad , \\ \varphi^0_{S^{k_q(i')+1}}(x^{k_q}) &\geq \varphi^0_{S^{k_q(i')}}(x^{k_q}) + \lambda_{i'}^{k_q}(f^0(v_{i'}^{k_q}) - \varphi^0_{S^{k_q(i')}}(v_{i'}^{k_q})) \ . \end{aligned} \tag{A.0.8.b}$$

This additional subsequence and the real value ν are useful in the proof of Lemma 4.4.5 for (DCP$_2$).

A.1. Proof of Lemma 4.4.2 for (DCP$_1$) and (DCP$_2$)

PROOF: Since $\{S^k\}_{k\in\mathbb{N}}$ is a nested sequence of compact and non-empty sets $S^k = [v_0^k, \dots, v_n^k]$ ($k \in \mathbb{N}$) we know that the set $\bigcap_{k\in\mathbb{N}} S^k$ is not empty. Choose a point $x \in \bigcap_{k\in\mathbb{N}} S^k$ and, for each $k \in \mathbb{N}$, an $(n+1)$-dimensional vector $\lambda^k \in B_n$ satisfying

$$x \ = \ \sum_{i=0}^{n} \lambda_i^k v_i^k \ .$$

Denote by

$$I \ := \ \{i \in \{0, \dots, n\} : |\{k \in \mathbb{N} : v_i^{k+1} \neq v_i^k\}| = \infty\}$$

the index set of the vertices which change infinitely often. In the following we show that, for each $k \in I$, there holds

$$\lambda_i^k \ \to \ 0 \ \ (k \to \infty) \ . \tag{A.1.1}$$

Assume, by contradiction, that there exists an index $i' \in I$ with the property

$$\lambda_{i'}^k \ \not\to \ 0 \ \ (k \to \infty) \ .$$

It follows by Lemma A.1 that there exist an index $l \in \{0, \dots, p\}$, a real number $\tau > 0$ and a subsequence $\{k_q\}_{q\in\mathbb{N}}$ of $\{k\}_{k\in\mathbb{N}}$ satisfying, for all $q \in \mathbb{N}$,

$$\varphi^l_{S^{k_q}}(x) \ \geq \ \varphi^l_{S^{k_{q-1}}}(x) + \tau \ .$$

Therefore, we obtain, for each $q \in \mathbb{N}$,

$$\varphi^l_{S^{k_q}}(x) \ \geq \ \varphi^l_{S^{k_1}}(x) + (q-1)\tau \ ,$$

and, in particular,

$$\varphi^l_{S^{k_q}}(x) \ \to \ \infty \ \ (q \to \infty) \ . \tag{A.1.2}$$

Because of $x \in S^k$ ($k \in \mathbb{N}$) we know – in view of the properties of the convex envelope – that, for all $k \in \mathbb{N}$, there holds

$$\varphi^l_{S^k}(x) \;\leq\; f^l(x) \;<\; \infty \,,$$

which contradicts (A.1.2) and proves (A.1.1).

By construction we know that $\lambda^k \in B_n$ and, hence, it is not possible that (A.1.1) is satisfied by each index $i \in \{0,\ldots,n\}$, i.e., it must exist an index j, which is not contained in I. Using an adequate numbering of the vertices of S^k ($k \in \mathbb{N}$) we are able to assume, without loss of generality, that there holds $I = \{r+1,\ldots,n\}$ for some $r \in \{0,\ldots,n-1\}$. Hence, we obtain the existence of a number $K \in \mathbb{N}$ and an integer $0 \leq r < n$ satisfying (4.4.5). Moreover, we obtain that there holds

$$x \;\in\; [v_0,\ldots,v_r] \;=\; S \tag{A.1.3}$$

and, in particular, for $k \geq K$ and $i \in \{r+1,\ldots,n\}$, that λ^k_i vanishes, i.e., $\lambda^k_i = 0$. Note that the representation of a point in a simplex as a convex combination of its vertices is unique. With (A.1.3) we see

$$\bigcap_{k\in\mathbb{N}} S^k \;\subset\; S \,,$$

and, on the other hand, for $k \geq K$, we know

$$S \;\subset\; S^k = [v_0,\ldots,v_r,v^k_{r+1},\ldots,v^k_n] \,,$$

which proves (4.4.6). ■

A.2. Proof of Lemma 4.4.3 for (DCP$_1$) and (DCP$_2$)

PROOF: Let $\bar{\omega}$ be an accumulation point of $\{\omega(S^k)\}_{k\in\mathbb{N}}$, and let $\{\omega(S^{k_q})\}_{q\in\mathbb{N}}$ be a subsequence converging to $\bar{\omega}$ with the additional property $k_1 \geq K$. Since $\{S^{k_q}\}_{q\in\mathbb{N}}$ is a nested sequence of compact, non-empty sets it follows immediately that

$$\bar{\omega} \;\in\; \bigcap_{q\in\mathbb{N}} S^{k_q} \;=\; \bigcap_{k\in\mathbb{N}} S^k \;=\; S \,.$$

Assume, by contradiction, that there is an index $i' \in \{0,\ldots,r\}$ with

$$v_{i'} \;=\; \bar{\omega} \,. \tag{A.2.1}$$

Using the properties of a convex envelope (see, especially, Relation (4.4.3)) we obtain, for $l \in \{0, \dots , p\}$ and $q \in \mathbb{N}$,

$$\varphi^l_{S^{k_1}}(\omega(S^{k_q})) \;\leq\; \varphi^l_{S^{k_q}}(\omega(S^{k_q})) \;\leq\; f^l(\omega(S^{k_q}))\,.$$

The functions $\varphi^l_{S^{k_1}}$ and f^l ($l = 0, \dots , p$) are continuous (for continuity of f^l see, e.g., [ROC70, Theorem 10.1]) and, furthermore, we know from Assumption (A.2.1) that there holds $\varphi^l_{S^{k_1}}(\bar{\omega}) = f^l(\bar{\omega})$. Therefore, we see that, for each $l \in \{0, \dots , p\}$,

$$\varphi^l_{S^{k_q}}(\omega(S^{k_q})) \;\rightarrow\; f^l(\bar{\omega}) \quad (q \rightarrow \infty)\,. \tag{A.2.2}$$

The point $\omega(S^{k_q})$ is feasible for the convex subproblem (DCP$^{S^{k_q}}$) ($q \in \mathbb{N}$). With (A.2.2) we obtain, for $l \in \{1, \dots , p\}$, by continuity of g^l

$$\begin{array}{ccccc} g^l(\omega(S^{k_q})) & + & \varphi^l_{S^{k_q}}(\omega(S^{k_q})) & \leq & 0 \\ \downarrow & (q \rightarrow \infty) & \downarrow & & \\ g^l(\bar{\omega}) & + & f^l(\bar{\omega}) & \leq & 0 \;, \end{array}$$

i.e., $\bar{\omega}$ is feasible. It follows that there exists an integer $Q \in \mathbb{N}$ such that, for any $q \geq Q$, $\omega(S^{k_q})$ is (δ, 0)-feasible. Though in the case of problem class (DCP$_1$) we assumed $\delta = 0$, we know that each point $\omega(S^k)$ ($k \in \mathbb{N}$) is (0, 0)-feasible, and hence the (δ, 0)-feasibility of $\omega(S^{k_q})$ ($q \geq Q$) is also guaranteed in this case.

This means that $\omega(S^{k_q})$ was used for updating the upper bound η^{k_q}. With respect to Property (4.4.2.b) of the simplex sequence we obtain, for $q \geq Q$,

$$\begin{aligned} \mu^{k_q} \;=\; \mu(S^{k_q}) \;=\; & g^0(\omega(S^{k_q})) + \varphi^0_{S^{k_q}}(\omega(S^{k_q})) \\ & < \; \eta^{k_q} - \epsilon \;\leq\; g^0(\omega(S^{k_q})) + f^0(\omega(S^{k_q})) - \epsilon\,. \end{aligned}$$

This contradicts (A.2.2) for $l = 0$. Thus we have proved

$$\bar{\omega} \;\notin\; \{v_0, \dots , v_r\}\,.$$

■

REMARK A.2. As we pointed out at the end of Section 4.4 it is possible to prove some of the results of this section also for problems of the general type (DCP$_3$). Therefore, as long as we do not need more technical effort, we do not eliminate the convex parts g^l ($l \in \{1, \dots , p\}$) of the nonlinear constraints in the proofs in this appendix, even though there holds $g^l \equiv 0$ ($l \in \{1, \dots , p\}$) in the proper relevant cases (DCP$_1$) and (DCP$_2$).

A.3. Proof of Lemma 4.4.5

The proof of this lemma is different, depending on the considered problem class. In the proof for problem class (DCP$_1$) we are able to exploit the feasibility of each generated point $\omega(S^k)$. In the case of problems of type (DCP$_2$) we do not have this property. However, exploiting the strict concavity of f^0 in connection with the result of the foregoing Lemma 4.4.3 we are still able to show the required result.

PROOF FOR (DCP$_1$): From Lemma 4.4.2 we know that, for each $k \geq K$,

$$S^k \;=\; [v_0, \dots, v_r, v^k_{r+1}, \dots, v^k_n]\,,$$

where, for each $j \in \{r+1, \dots, n\}$, the vertices v^k_j $(k \geq K)$ change infinitely often, i.e.,

$$|\{k \in \mathbb{N} : v^{k+1}_j \;\neq\; v^k_j\}| \;=\; \infty\,. \tag{A.3.1}$$

For each $k \in \mathbb{N}$ with $k \geq K$, choose $\lambda^k \in B_n$ such that

$$\omega(S^k) = \sum_{j=0}^{r} \lambda^k_j v_j + \sum_{j=r+1}^{n} \lambda^k_j v^k_j\,. \tag{A.3.2}$$

Assume, by contradiction, that there exists an index $i' \in \{r+1, \dots, n\}$ with

$$\lambda^k_{i'} \not\to 0 \;\; (k \to \infty)\,. \tag{A.3.3}$$

In the considered situation there holds $g^0 \equiv 0$, $\rho = 0$ and $\delta = 0$. Therefore, we know that each solution $\omega(S^k)$ of the linear subproblem (DCP$_1^{S^k}$) is feasible for (DCP$_1$). It follows that we can assume, without loss of generality, that each vertex $v^k_{i'}$ $(k \geq K)$ is feasible. Using Lemma A.1 we obtain the existence of a positive real value τ and a subsequence $\{k_q\}_{q\in\mathbb{N}}$ of $\{k\}_{k\in\mathbb{N}}$ satisfying, for all $q \in \mathbb{N}$,

$$\varphi^0_{S^{k_q}}(\omega(S^{k_q})) \;\geq\; \varphi^0_{S^{k_{q-1}}}(\omega(S^{k_q})) + \tau\,.$$

Because of the feasibility of $\omega(S^{k_q})$, i.e., $\omega(S^{k_q}) \in S^{k_{q-l}} \cap F$ $(l \in \mathbb{N}, l < q)$, and the optimality of $\omega(S^{k_{q-1}})$ with respect to the subproblem (DCP$_1^{S^{k_{q-1}}}$) we obtain, for each $q \in \mathbb{N}$,

$$\begin{aligned}
\mu^{k_q} \;=\; \mu(S^{k_q}) \;&=\; \varphi^0_{S^{k_q}}(\omega(S^{k_q})) \\
&\geq\; \varphi^0_{S^{k_{q-1}}}(\omega(S^{k_q})) + \tau \\
&\geq\; \varphi^0_{S^{k_{q-1}}}(\omega(S^{k_{q-1}})) + \tau \\
&=\; \mu(S^{k_{q-1}}) + \tau \;=\; \mu^{k_{q-1}} + \tau\,.
\end{aligned}$$

It follows that

$$\mu^{k_q} \;\to\; \infty \;\; (q \to \infty)$$

and, in particular,

$$\mu^{k_q} \geq \eta^{k_q}$$

for q big enough, since $\{\eta^k\}_{k\in\mathbb{N}}$ is by construction a non-increasing sequence. This contradicts Property (4.4.2.b) of the nested simplex sequence $\{S^k\}_{k\in\mathbb{N}}$. ■

PROOF FOR (DCP$_2$): As in the foregoing proof for (DCP$_1$) let $\lambda^k \in B_n$ be chosen such that we have the representation (A.3.2) of $\omega(S^k)$ $(k \geq K)$, and assume, by contradiction, that there is an index $i' \in \{r+1,\ldots,n\}$ with Property (A.3.3). Since we do not know anything about the feasibility of $v^k_{i'}$ $(k \in \mathbb{N})$ Lemma A.1 only delivers the existence of an index $\bar{l} \in \{0,\ldots,p\}$, a positive real value τ and a subsequence $\{k_q\}_{q\in\mathbb{N}}$ of $\{k\}_{k\in\mathbb{N}}$ satisfying, for all $q \in \mathbb{N}$

$$\varphi^{\bar{l}}_{S^{k_q}}(\omega(S^{k_q})) \;\geq\; \varphi^{\bar{l}}_{S^{k_{q-1}}}(\omega(S^{k_q})) + \tau\;. \tag{A.3.4}$$

There holds $\rho = 0$ and from Property (4.4.3) of the convex envelopes we see that $\omega(S^{k_q})$ is feasible for the convex optimization problem (DCP$_2^{S^{k_{q-1}}}$). Therefore, if there holds $\bar{l} = 0$ in Relation (A.3.4), we obtain, for $q \in \mathbb{N}$, by the same arguments as in the proof for the case (DCP$_1$)

$$\begin{aligned}
\mu^{k_q} \;=\; \mu(S^{k_q}) \;&=\; g^0(\omega(S^{k_q})) + \varphi^0_{S^{k_q}}(\omega(S^{k_q})) \\
&\geq\; g^0(\omega(S^{k_q})) + \varphi^0_{S^{k_{q-1}}}(\omega(S^{k_q})) + \tau \\
&\geq\; g^0(\omega(S^{k_{q-1}})) + \varphi^0_{S^{k_{q-1}}}(\omega(S^{k_{q-1}})) + \tau \\
&=\; \mu(S^{k_{q-1}}) + \tau \;=\; \mu^{k_{q-1}} + \tau\;.
\end{aligned}$$

In this situation it follows again that $\mu^{k_q} \;\geq\; \eta^{k_q}$ for q big enough, contradicting Property (4.4.2.b) of the sequence $\{S^k\}_{k\in\mathbb{N}}$.

In the case $\bar{l} > 0$ it is necessary to exploit the strict concavity of f^0 in order to obtain the required result. In view of Remark A.1 we know that there exist a further subsequence $\{k_q(i')\}_{q\in\mathbb{N}}$ of $\{k\}_{k\in\mathbb{N}}$ and a real value $\nu > 0$ with Properties (A.0.8.a) and (A.0.8.b). Let $\bar{v}_{i'}$ be an accumulation point of the sequence $\{v^{k_q}_{i'}\}_{q\in\mathbb{N}}$ and assume, without loss of generality, that this sequence converges to $\bar{v}_{i'}$. For each $l \in \{1,\ldots,p\}$, $q \in \mathbb{N}$ and $k \in \mathbb{N}$ with $k \leq k_q(i')$ we obtain by Property (4.4.3) of the convex envelopes and by using the feasibility of $\omega(S^k)$ with respect to the convex subproblem (DCP$^{S^k}$) that

$$\begin{aligned} g^l(v_{i'}^{k_q}) + \varphi_{S^k}^l(v_{i'}^{k_q}) \;&\leq\; g^l(v_{i'}^{k_q}) + \varphi_{S^{k_q(i')}}^l(v_{i'}^{k_q}) \\ &=\; g^l(\omega(S^{k_q(i')})) + \varphi_{S^{k_q(i')}}^l(\omega(S^{k_q(i')})) \;\leq\; 0\,. \end{aligned} \tag{A.3.5}$$

The functions g^l and $\varphi_{S^k}^l$ ($k \leq k_q(i')$, $l \in \{1, \dots, p\}$) are continuous (for continuity of g^l see again [ROC70, Theorem 10.1]). Therefore, for each $k \in \mathbb{N}$, it follows

$$g^l(\bar{v}_{i'}) + \varphi_{S^k}^l(\bar{v}_{i'}) \;\leq\; 0\,,$$

i.e., $\bar{v}_{i'}$ is feasible with respect to (DCP$^{S^k}$). The point $\bar{v}_{i'}$ is obviously an accumulation point of the sequence $\{\omega(S^k)\}_{k\in\mathbb{N}}$. From Lemma 4.4.3 we know that

$$\bar{v}_{i'} \;\in\; [v_0, \dots, v_r] \setminus \{v_0, \dots, v_r\}\,.$$

Therefore, it follows by the strict concavity of f^0 that there exists a real value $\varsigma > 0$ satisfying

$$f^0(\bar{v}_{i'}) \;\geq\; \sum_{i=0}^{r} \lambda_i f^0(v_i) + \varsigma \tag{A.3.6}$$

with $\lambda \in B_r$, $\bar{v}_{i'} = \sum_{i=0}^r \lambda_i v_i$. For each $k \geq K$, we know that the functions $\varphi_{S^k}^0$ have the same function values on the r-simplex $S = [v_0, \dots, v_r]$ independent of k, i.e.,

$$\varphi_{S^k}^0(\bar{v}_{i'}) \;=\; \sum_{i=0}^{r} \lambda_i f^0(v_i)\,. \tag{A.3.7}$$

Since $\bar{v}_{i'}$ is feasible we obtain with (A.3.6) and (A.3.7)

$$\begin{aligned} g^0(\omega(S^{k_q(i')})) + \varphi_{S^{k_q(i')}}^0(\omega(S^{k_q(i')})) \;&\leq\; g^0(\bar{v}_{i'}) + \varphi_{S^{k_q(i')}}^0(\bar{v}_{i'}) \\ &\leq\; g^0(\bar{v}_{i'}) + f^0(\bar{v}_{i'}) - \varsigma\,. \end{aligned} \tag{A.3.8}$$

The functions g^0 and f^0 are continuous, thus, there exists a number $Q \in \mathbb{N}$ such that, for all $q \geq Q$,

$$|g^0(v_{i'}^{k_q}) + f^0(v_{i'}^{k_q}) \;-\; g^0(\bar{v}_{i'}) - f^0(\bar{v}_{i'})| \;\leq\; \frac{\varsigma}{2}\,.$$

Hence we obtain, for all $q \geq Q$, by using Relation (A.3.8)

$$g^0(\omega(S^{k_q(i')})) + \varphi^0_{S^{k_q(i')}}(\omega(S^{k_q(i')})) \;\leq\; g^0(v_{i'}^{k_q}) + f^0(v_{i'}^{k_q}) - \tfrac{\varsigma}{2} \,,$$

$$\text{i.e., } f^0(v_{i'}^{k_q}) - \varphi^0_{S^{k_q(i')}}(v_{i'}^{k_q}) \;\geq\; \tfrac{\varsigma}{2} \,.$$

With (A.0.8.b) it follows, for each $q \geq Q$,

$$\varphi^0_{S^{k_q(i')+1}}(\omega(S^{k_q})) \;\geq\; \varphi^0_{S^{k_q(i')}}(\omega(S^{k_q})) + \underbrace{\lambda_{i'}^{k_q}\frac{\varsigma}{2}}_{\geq \nu\frac{\varsigma}{2} =: \bar{\tau} > 0} \,.$$

Thus, by using the relations $k_q \;\geq\; k_q(i') + 1$ and $k_{q-1} \;\leq\; k_q(i')$ and the fact that $\omega(S^{k_q})$ is feasible for (DCP$_2^{S_{k_{q-1}}}$) there holds

$$\begin{aligned}
\mu^{k_q} \;=\; \mu(S^{k_q}) \;&=\; g^0(\omega(S^{k_q})) + \varphi^0_{S^{k_q}}(\omega(S^{k_q})) \\
&\geq\; g^0(\omega(S^{k_q})) + \varphi^0_{S^{k_q(i')+1}}(\omega(S^{k_q})) \\
&\geq\; g^0(\omega(S^{k_q})) + \varphi^0_{S^{k_q(i')}}(\omega(S^{k_q})) + \bar{\tau} \\
&\geq\; g^0(\omega(S^{k_q})) + \varphi^0_{S^{k_{q-1}}}(\omega(S^{k_q})) + \bar{\tau} \\
&\geq\; g^0(\omega(S^{k_{q-1}})) + \varphi^0_{S^{k_{q-1}}}(\omega(S^{k_{q-1}})) + \bar{\tau} \\
&=\; \mu(S^{k_{q-1}}) + \bar{\tau} \;=\; \mu^{k_{q-1}} + \bar{\tau} \,.
\end{aligned}$$

By the same arguments as in the case $\bar{l} = 0$, we see that the last relation is a contradiction to Property (4.4.2.b) of the nested simplex sequence $\{S^k\}_{k\in\mathbb{N}}$, and the proof is complete. ■

A.4. Proof of Lemma 4.4.7

The proof of Lemma 4.4.7 depends again on the considered problem class. A part of the proof is the same for both classes and a part is different. In order to make this proof more structured we split it. In Lemma A.3 we prove a technical result, which is the substantial part of the proof of Lemma 4.4.7. Only for this result we need different argumentation depending on the problem type. After proving this lemma for both classes we are able to show the existence of a point $r^k \in S$ with Properties (4.4.14.a)-(4.4.14.c) in one proof, i.e., independent of the considered problem class.

However, first of all, we present and prove a result in Lemma A.2, which will ease the proof of the technical Lemma A.3 in the case of (DCP$_2$). This lemma does not depend on Algorithm 4.1. It is a general result concerning strictly concave functions over simplices.

LEMMA A.2. *Let $S = [v_0, \dots, v_r] \subset \mathbb{R}^n$ be an r-simplex ($r \leq n$), $f : \mathbb{R}^n \to \mathbb{R}$ be a strictly concave function, and $\{x^k\}_{k\in\mathbb{N}}$ be a sequence in S. Let further $\gamma^k \in B_r$ be the barycentric coordinates of x^k ($k \in \mathbb{N}$) with respect to S. Assume that each accumulation point $\bar{x}$ of the sequence $\{x^k\}_{k\in\mathbb{N}}$ is not a member of the vertex set of S, i.e.,*

$$\bar{x} \in S \setminus \{v_0, \dots, v_r\} .$$

Then there exist an integer $K \in \mathbb{N}$ and a positive real value ν such that, for all $k \geq K$, there holds

$$f(x^k) \geq \sum_{i=0}^{r} \gamma_i^k f(v_i) + \nu . \tag{A.4.1}$$

PROOF: Assume that there do not exist a number $K \in \mathbb{N}$ and a real value $\nu > 0$ with the required property. Let $\{\nu^q\}_{q\in\mathbb{N}}$ be a non-increasing sequence of positive real values converging to 0. It follows that, for each $q \in \mathbb{N}$, there exists a number $k_q \in \mathbb{N}$ with

$$f(x^{k_q}) \leq \sum_{i=0}^{r} \gamma_i^{k_q} f(v_i) + \nu^q .$$

We can assume, without loss of generality, that the sequence $\{k_q\}_{q\in\mathbb{N}}$ is monotonously increasing. The corresponding subsequence $\{x^{k_q}\}_{q\in\mathbb{N}}$ is bounded. Therefore, there exists an accumulation point $\bar{x}$ of this sequence. Assume, again without loss of generality, that the sequence $\{x^{k_q}\}_{q\in\mathbb{N}}$ converges to $\bar{x}$, and, moreover, that the sequence of the barycentric coordinates $\{\gamma^{k_q}\}_{q\in\mathbb{N}}$ is converging to a point $\bar{\gamma} \in B_r$, i.e., $\bar{x} = \sum_{i=0}^{r} \bar{\gamma}_i v_i$. Then, we obtain by continuity and concavity of f (for continuity of a concave function see, e.g., [ROC70, Theorem 10.1])

$$\begin{array}{ccccccc} \sum_{i=0}^{r} \gamma_i^{k_q} f(v_i) & \leq & f(x^{k_q}) \leq & \sum_{i=0}^{r} \gamma_i^{k_q} f(v_i) & + & \nu^q & \\ \downarrow & & \downarrow & \downarrow & & \downarrow & (q \to \infty) \\ \sum_{i=0}^{r} \bar{\gamma}_i f(v_i) & \leq & f(\bar{x}) \quad \leq & \sum_{i=0}^{r} \bar{\gamma}_i f(v_i) & + & 0 & . \end{array} \tag{A.4.2}$$

The accumulation point $\bar{x}$ of the subsequence $\{x^{k_q}\}_{q\in\mathbb{N}}$ is obviously an accumulation point of the whole sequence $\{x^k\}_{k\in\mathbb{N}}$. Therefore, we know that $\bar{x}$ does not belong to the vertex set of S, and we obtain by the strict concavity of f

$$f(\bar{x}) \ > \sum_{i=0}^{r} \bar{\gamma}_i f(v_i) \, .$$

This contradicts (A.4.2) and completes the proof. ■

As mentioned before, we are now able to show a technical result, which will be substantial for the final proof of Lemma 4.4.7. In the proof of this lemma we have to distinguish between both problem classes.

LEMMA A.3. *Let $\{S^k\}_{k\in\mathbb{N}}$ be an infinite nested sequence of simplices generated by Algorithm 4.1 with Properties (4.4.2.a) and (4.4.2.b). Let $\tilde{K} \in \mathbb{N}$ be chosen as in Corollary 4.4.6 and let $0 \leq r < n$ be chosen as in Lemma 4.4.2, i.e.,*

$$S^k = [v_0, \dots, v_r, v^k_{r+1}, \dots, v^k_n] \quad (k \geq K)$$

and

$$|\{k \in \mathbb{N} : v^{k+1}_j \neq v^k_j\}| = \infty \quad (j \in \{r+1, \dots, n\}) \, .$$

Then there exist an integer $\bar{K} \in \mathbb{N}$, a positive real value σ and, for each index $j \in \{r+1, \dots, n\}$, an integer sequence $\{k(j)\}_{k\in\mathbb{N}} \subset \{k\}_{k\in\mathbb{N}}$, point sequences $\{\gamma^{k(j)}\}_{k\in\mathbb{N}}$ and $\{\varsigma^{k(j)}\}_{k\in\mathbb{N}}$, and, additionally, for each index $l \in \{0, \dots, p\}$, real value sequences $\{\tau^k_{l,j}\}_{k\in\mathbb{N}}$ such that, for all $k \geq \bar{K}$ and $j \in \{r+1, \dots, n\}$, there holds

$$v^k_j = \omega(S^{k(j)}) \, , \, \gamma^{k(j)} \in B_r \, , \, \varsigma^{k(j)} \in \mathbb{R}^n \, , \, v^k_j = \sum_{i=0}^{r} \gamma^{k(j)}_i v_i + \varsigma^{k(j)} \, , \quad \text{(A.4.3.a)}$$

$$f^0(v^k_j) \ \geq \ \tau^k_{0,j} + \sum_{i=0}^{r} \gamma^{k(j)}_i f^0(v_i) + \sigma \qquad \text{(A.4.3.b)}$$

and

$$f^l(v^k_j) \ = \ \tau^k_{l,j} + f^l\left(\sum_{i=0}^{r} \gamma^{k(j)}_i v_i\right) \qquad l = 1, \dots, p \, . \qquad \text{(A.4.3.c)}$$

Furthermore, the involved sequences have, for each $j \in \{r+1, \dots, n\}$, the following convergence properties

$$k(j) \ \to \ \infty \quad (k \to \infty) \, , \qquad \text{(A.4.3.d)}$$

$$\|\varsigma^{k(j)}\|_2 \ \to \ 0 \quad (k \to \infty) \qquad \text{(A.4.3.e)}$$

and

$$\tau_{l,j}^k \ \to\ 0 \quad (k \to \infty) \qquad l = 0, \ldots, p\,. \tag{A.4.3.f}$$

PROOF FOR (DCP$_1$) AND (DCP$_2$): Even though the proof of Lemma A.3 is different for the considered problem classes, the choice of the involved sequences $\{k(j)\}_{k\in\mathbb{N}}$, $\{\gamma^{k(j)}\}_{k\in\mathbb{N}}$, $\{\varsigma^{k(j)}\}_{k\in\mathbb{N}}$ and $\{\tau_{l,j}^k\}_{k\in\mathbb{N}}$ ($j \in \{r+1, \ldots, n\}$, $l \in \{0, \ldots, p\}$) is the same. Therefore, the beginning of the proof holds for both problem classes.

Choose $\hat{K} \in \mathbb{N}$ such that, for all $k \geq \hat{K}$ and $j \in \{r+1, \ldots, n\}$, there exists an integer $k(j) \geq \tilde{K}$, $k(j) < k$ with

$$v_j^k \ =\ \omega(S^{k(j)})\,.$$

Since, for each index $j \in \{r+1, \ldots, n\}$, the vertices v_j^k ($k \in \mathbb{N}$) change infinitely often Property (A.4.3.d) of the sequence $\{k(j)\}_{k\in\mathbb{N}}$ follows immediately.

Select now an arbitrary, but fixed index $j \in \{r+1, \ldots, n\}$. From result (4.4.12) of Corollary 4.4.6 we know that, for each $k \geq \hat{K}$, there exist a point $\gamma^{k(j)} \in B_r$ and a residual $\varsigma^{k(j)} \in \mathbb{R}^n$ satisfying

$$v_j^k \ =\ \sum_{i=0}^{r} \gamma_i^{k(j)} v_i + \varsigma^{k(j)} \tag{A.4.4}$$

and, additionally,

$$\|\varsigma^{k(j)}\|_2 \ \to\ 0 \qquad (k \to \infty)\,. \tag{A.4.5}$$

The functions f^l ($l \in \{0, \ldots, p\}$) are continuous. Therefore, there must exist, for each index $l \in \{0, \ldots, p\}$, a sequence $\{\tau_{l,j}^k\}_{k\in\mathbb{N}}$ with the properties

$$f^l(v_j^k) \ =\ f^l\left(\sum_{i=0}^{r} \gamma_i^{k(j)} v_i\right) + \tau_{l,j}^k\,, \tag{A.4.6}$$

and

$$\tau_{l,j}^k \ \to\ 0 \qquad (k \to \infty)\,. \tag{A.4.7}$$

The choice of the positive real value σ and the verification of Property (A.4.3.b) depends now on the considered problem class.

> PROOF FOR (DCP$_1$): In this situation we know that each point $\omega(S^k)$ ($k \in \mathbb{N}$) is used for calculating the current upper bound η^k, i.e., for each $k \geq \hat{K}$, there holds

$$\eta^k \;\le\; f^0(\omega(S^{k(j)})) \;=\; f^0(v_j^k) \;=\; f^0\left(\sum_{i=0}^{r}\gamma_i^{k(j)}v_i\right) + \tau_{0,j}^k \,. \qquad \text{(A.4.8)}$$

Moreover, in view of Property (4.4.2.b) of the simplex sequence $\{S^k\}_{k\in\mathbb{N}}$, we know $\mu^k = \mu(S^k) < \eta^k - \epsilon$. Therefore, by using result (4.4.13) of Corollary 4.4.6 we obtain, for each $k \ge \hat{K}$, a real value σ_j^k satisfying

$$\begin{aligned}\sum_{i=0}^{r}\gamma_i^{k(j)}f^0(v_i) \;&=\; \varphi^0_{S^{k(j)}}(\sum_{i=0}^{r}\gamma_i^{k(j)}v_i) \;=\; \underbrace{\varphi^0_{S^{k(j)}}(\omega(S^{k(j)}))}_{=\mu^{k(j)}} - \sigma_j^k \\ &< \;\eta^{k(j)} - \epsilon - \sigma_j^k \;, \qquad \text{(A.4.9)}\end{aligned}$$

and, furthermore,

$$\sigma_j^k \;\to\; 0 \qquad (k\to\infty)\,. \qquad \text{(A.4.10)}$$

From (A.4.7) and (A.4.10) we know that there exists a number $\bar{K}(j) \in \mathbb{N}$, $\bar{K}(j) \ge \hat{K}$ such that, for all $k \ge \bar{K}(j)$, there holds

$$|\tau_{0,j}^k| \;\le\; \frac{\epsilon}{4} \text{ and } |\sigma_j^k| \;\le\; \frac{\epsilon}{4}\,.$$

If we set $\sigma(j) := \frac{\epsilon}{2}$, then we obtain, for each $k \ge \bar{K}(j)$,

$$\begin{aligned} f^0\left(\sum_{i=0}^{r}\gamma_i^{k(j)}v_i\right) \;&\underset{\text{(A.4.8)}}{\ge}\; \eta^k - \tau_{0,j}^k \\ &\underset{\text{(A.4.9)}}{>}\; \sum_{i=0}^{r}\gamma_i^{k(j)}f^0(v_i) \;+\underbrace{\epsilon + \sigma_j^k - \tau_{0,j}^k}_{\ge\frac{\epsilon}{2}} \\ &\ge\; \sum_{i=0}^{r}\gamma_i^{k(j)}f^0(v_i) + \sigma(j)\,. \qquad \text{(A.4.11)}\end{aligned}$$

Combining this result and (A.4.8) we see that Property (A.4.3.b) is fulfilled for index $j \in \{r+1,\dots,n\}$ in the case of problems of type (DCP$_1$). □

PROOF FOR (DCP$_2$): Because of the possible infeasibility of $\omega(S^k)$ the left-hand side of Relation (A.4.8) is no longer true. In order to prove the existence of a positive real value σ with Property (A.4.11) in this situation, it is necessary to exploit the strict concavity of f^0. Denote for $k \ge \hat{K}$ by

$$x^k \;:=\; \sum_{i=0}^{r}\gamma_i^{k(j)}v_i$$

the part of the representation (A.4.4) of v_j^k contained in $S = [v_0,\dots,v_r]$.

Since the residual $\varsigma^{k(j)}$ vanishes, if k tends to infinity, it is obvious, that each accumulation point of the sequence $\{x^k\}_{k\geq\hat{K}}$ is also an accumulation point of $\{v_j^k\}_{k\geq\hat{K}}$. The sequence $\{v_j^k\}_{k\geq\hat{K}}$ is by construction a subsequence of $\{\omega(S^k)\}_{k\in\mathbb{N}}$. Therefore, according to Lemma 4.4.3 we know that, for each accumulation point $\bar{x}$ of $\{x^k\}_{k\geq\hat{K}}$, there holds

$$\bar{x} \ \in \ S\setminus\{v_0,\ldots,v_r\}\ .$$

Using Lemma A.2 we obtain a number $\bar{K}(j)\in\mathbb{N}$, $\bar{K}(j)\geq\hat{K}$ and a real value $\sigma(j)>0$ satisfying, for all $k\geq\bar{K}(j)$,

$$f^0(x^k) \ \geq \ \sum_{i=0}^{r}\gamma_i^{k(j)}f^0(v_i)+\sigma(j)\ . \tag{A.4.12}$$

Combining (A.4.12) with Relation (A.4.6) for $l=0$ we see that Property (A.4.3.b) is also satisfied for $j\in\{r+1,\ldots,n\}$ in the case of problems of type (DCP$_2$) with a strictly concave part of the objective function. □

Setting $\bar{K}:=\max_{j=r+1,\ldots,n}\bar{K}(j)$ and $\sigma:=\min_{j=r+1,\ldots,n}\sigma(j)$ completes the proof. ■

REMARK A.3.

(a) In order to prove Relation (A.4.3.b) we needed a *special* strict concavity result for f^0 at the points $x^k=\sum_{i=0}^{r}\gamma_i^{k(j)}v_i$ ($k\in\mathbb{N}$; $j=r+1,\ldots,n$), i.e., we need the existence of a positive real value $\sigma(j)$ satisfying

$$f^0(x^k) \ \geq \ \sum_{i=0}^{r}\gamma_i^{k(j)}f^0(v_i)+\sigma(j)\ .$$

As long as the points $\omega(S^k)$ ($k\in\mathbb{N}$) are feasible for the original problem (DCP) and hence used for updating the upper bound η^k, as it is the case for the class (DCP$_1$), this result follows by the definition of Algorithm 4.1 and the existence of an infinite nested sequence of simplices with Properties (4.4.2.a) and (4.4.2.b). By applying Algorithm 4.1 to problems of type (DCP$_2$) we do not have the guaranteed feasibility of $\omega(S^k)$ ($k\in\mathbb{N}$) anymore. In order to ensure also in this case the above relation we need the additional requirement that f^0 is strictly concave itself.

(b) In the part of the proof of the previous lemma concerning (DCP$_1$) the value σ is chosen in a constructive way, i.e., σ can be determined in advance, where in case of (DCP$_2$) we were only able to show the existence of such a value with the required properties.

After the verification of the statements of the technical Lemma A.3 we are now able to prove Lemma 4.4.7 independent of the considered problem class.

PROOF OF LEMMA 4.4.7 FOR (DCP$_1$) AND (DCP$_2$): In view of Lemma A.3 we know that there exist sequences $\{k(j)\}_{k\in\mathbb{N}}$, $\{\gamma^{k(j)}\}_{k\in\mathbb{N}}$, $\{\varsigma^{k(j)}\}_{k\in\mathbb{N}}$, $\{\tau^k_{l,j}\}_{k\in\mathbb{N}}$ ($j \in \{r+1,\dots,n\}$ and $l \in \{0,\dots,p\}$), an integer $\bar{K} \in \mathbb{N}$ and a real value $\sigma > 0$ with Properties (A.4.3.a)-(A.4.3.f). In particular, for $k \geq \bar{K}$ and $j \in \{r+1,\dots,n\}$, there holds

$$v_j^k \;=\; \sum_{i=0}^{r} \gamma_i^{k(j)} v_i + \varsigma^{k(j)} \,. \qquad \text{(A.4.13)}$$

By substituting (A.4.13) in the representation (4.4.10) of $\omega(S^k)$ we obtain, for each $k \geq \bar{K}$,

$$\omega(S^k) \;=\; \underbrace{\sum_{i=0}^{r}\left(\lambda_i^k + \sum_{j=r+1}^{n} \lambda_j^k \gamma_i^{k(j)}\right) v_i}_{=:\, r^k} + \sum_{i=r+1}^{n} \lambda_i^k \varsigma^{k(i)} \,.$$

It follows that r^k belongs to $[v_0,\dots,v_r]$. With Property (A.4.3.e) of the sequences $\{\varsigma^{k(j)}\}_{k\in\mathbb{N}}$ ($j \in \{r+1,\dots,n\}$) we obtain further

$$\sum_{i=r+1}^{n} \lambda_i^k \|\varsigma^{k(i)}\|_2 \;=\; o(\Lambda^k) \,,$$

with Λ^k defined as in (4.4.11). Thus, (4.4.14.c) is proved, i.e.,

$$\|\omega(S^k) - r^k\|_2 \;=\; o(\Lambda^k) \,. \qquad \text{(A.4.14)}$$

Using Properties (A.4.3.b) and (A.4.3.f) it follows, for each $k \geq \bar{K}$,

$$\begin{aligned}
\varphi^0_{S^k}(\omega(S^k)) \;&=\; \sum_{i=0}^{r} \lambda_i^k f^0(v_i) + \sum_{i=r+1}^{n} \lambda_i^k f^0(v_i^k) \\
&\geq\; \sum_{i=0}^{r} \lambda_i^k f^0(v_i) + \sum_{i=r+1}^{n} \lambda_i^k \left(\tau^k_{0,i} + \sum_{j=0}^{r} \gamma_j^{k(i)} f^0(v_j) + \sigma\right) \\
&=\; \sum_{i=0}^{r}\left(\lambda_i^k + \sum_{j=r+1}^{n} \lambda_j^k \gamma_i^{k(j)}\right) f^0(v_i) + \sum_{i=r+1}^{n} \lambda_i^k \tau^k_{0,i} + \Lambda^k \sigma \\
&=\; \varphi^0_{S^k}(r^k) + \Lambda^k \sigma + o(\Lambda^k) \,. \qquad \text{(A.4.15)}
\end{aligned}$$

The function g^0 is convex on the whole space $\mathbb{R}^n$. Therefore, we know that for each compact, non-empty subset $C \subset \mathbb{R}^n$, g^0 is Lipschitz continuous on C with Lipschitz constant

$$L_C \;=\; \sup\{\|\xi\|_2 : \xi \in \partial g^0(x), x \in C\}\;,$$

where $\partial g^0(x)$ denotes the subdifferential of g^0 at the point x (see, e.g., [ROC70, Theorem 24.7]). Taking Relation (A.4.14) into account it follows that

$$\begin{aligned} g^0(r^k) \;&\leq\; g^0(\omega(S^k)) + L_P\|\omega(S^k) - r^k\|_2 \\ &=\; g^0(\omega(S^k)) + o(\Lambda^k)\;. \end{aligned} \tag{A.4.16}$$

Combining this result with (A.4.15) we obtain the postulated result (4.4.14.a) of Lemma 4.4.7.

In order to complete the proof, we have to show that (4.4.14.b) is also true. The point $\omega(S^k)$ is feasible with respect to (DCP$^{S^k}$). Therefore, it follows from (A.4.3.c), for $l \in \{1, \ldots, p\}$,

$$\begin{aligned} \varphi^l_{S^k}(r^k) \;&=\; \varphi^l_{S^k}(r^k) - \varphi^l_{S^k}(\omega(S^k)) + \underbrace{\varphi^l_{S^k}(\omega(S^k))}_{\leq 0} \\ &\leq\; \sum_{i=0}^{r}\left(\lambda_i^k + \sum_{j=r+1}^{n}\lambda_j^k\gamma_i^{k(j)}\right) f^l(v_i) - \sum_{i=0}^{r}\lambda_i^k f^l(v_i) - \sum_{i=r+1}^{n}\lambda_i^k f^l(v_i^k) \\ &=\; \sum_{i=0}^{r}\left(\sum_{j=r+1}^{n}\lambda_j^k\gamma_i^{k(j)}\right) f^l(v_i) - \sum_{i=r+1}^{n}\lambda_i^k\left(\tau_{l,i}^k + f^l(\sum_{j=0}^{r}\gamma_j^{k(i)}v_j)\right)\;. \end{aligned}$$

In view of the concavity of f^l we obtain

$$f^l(\sum_{j=0}^{r}\gamma_j^{k(i)}v_j) \;\geq\; \sum_{j=0}^{r}\gamma_j^{k(i)} f^l(v_j)\;,$$

and from Property (A.4.3.f) of the sequences $\{\tau_{l,j}^k\}_{k\in\mathbb{N}}$ we can conclude

$$\varphi^l_{S^k}(r^k) \;\leq\; -\sum_{i=r+1}^{n}\lambda_i^k\tau_{l,i}^k \;=\; o(\Lambda^k)\;.$$

■

A.5. Proof of Lemma 4.4.8

The proof of Lemma 4.4.8, which does not depend explicitly on the considered problem class, is again a technical one. Therefore, we decided to split this proof in three steps. First of all, we establish a technical result concerning finite point sets and the cones generated by these sets. We will obtain this result in Corollary A.6 after introducing two lemmata, where each lemma itself is of some interest.

In the proof of Lemma 4.4.7 the precedent Lemma A.3 contained the substantial and most technical part. We repeat here this strategy in order to obtain a more structured proof. We show again the essential part of the proof of Lemma 4.4.8 in the independent technical Lemma A.7. Then we derive the results of Lemma 4.4.8 in a short and clear way.

LEMMA A.4. *Let $L = \{y_1, \dots , y_q\} \subset \mathbb{R}^n$ ($q \in \mathbb{N}$) be an arbitrary finite set of n-dimensional points. Then there exist two positive real values τ_1 and τ_2 with the property that, for each linear independent subset $LI = \{y_{i_0}, \dots , y_{i_r}\}$ ($r < n$) of L, there holds*

$$\tau_1 \ \leq \ \|x\|_2 \ \leq \ \tau_2 \qquad \forall x \in S_{LI} \ , \tag{A.5.1}$$

where $S_{LI} := [y_{i_0}, \dots , y_{i_r}] \subset \mathbb{R}^n$ denotes the r-simplex with the vertices $y_{i_0}, \dots , y_{i_r}$.

PROOF: Let $LI \subset L$ be an arbitrary linear independent subset of L. Since S_{LI} is a compact set not-containing the origin, it follows immediately that there exist real values $\tau_1(LI), \tau_2(LI) > 0$ satisfying, for all $x \in S_{LI}$,

$$\tau_1(LI) \ \leq \ \|x\|_2 \ \leq \ \tau_2(LI) \, .$$

The set L is finite. Therefore, we know that there is only a finite number of linear independent subsets of L. By setting

$$\begin{aligned} \tau_1 \ &:= \ \min\{\tau_1(LI) : LI \subset L \, , \, LI \text{ linear independent} \} \, , \\ \tau_2 \ &:= \ \max\{\tau_2(LI) : LI \subset L \, , \, LI \text{ linear independent} \} \end{aligned}$$

we obtain that τ_1 and τ_2 have Property (A.5.1) for each linear independent subset LI, and, in particular, there holds

$$\tau_1 \, , \tau_2 \ > \ 0 \, .$$

■

LEMMA A.5. *Let $L = \{y_1, \ldots, y_q\} \subset \mathbb{R}^n$ ($q \in \mathbb{N}$) be an arbitrary finite set of n-dimensional points. Denote by*

$$CO \;:=\; \{x \in \mathbb{R}^n : x = \sum_{i=1}^{q} \gamma_i y_i \,,\; \gamma \in \mathbb{R}^q_+ \,,\; i = 1, \ldots, q\}$$

the cone generated by the elements of L. Denote further by

$$CO_{LI} \;:=\; \{x \in \mathbb{R}^n : x = \sum_{j=0}^{r} \gamma_j y_{i_j} \,,\; \gamma \in \mathbb{R}^{r+1}_+ \,,\; j = 0, \ldots, r\}$$

the cone generated by the elements of a linear independent (l.i.) subset $LI = \{y_{i_0}, \ldots, y_{i_r}\}$ ($r < n$) of L. Then there holds

$$CO \;=\; \bigcup_{\substack{LI \,\subset\, L \\ LI \text{ l.i.}}} CO_{LI} \,. \tag{A.5.2}$$

PROOF: The relation $\bigcup_{\substack{LI \,\subset\, L \\ LI \text{ l.i.}}} CO_{LI} \subset CO$ is immediately clear. In order to prove the inverse relation, we choose an arbitrary point $x \in CO$. From definition of CO we know that there exists a vector $\gamma \in \mathbb{R}^q$ satisfying

$$x \;=\; \sum_{i=1}^{q} \gamma_i y_i \tag{A.5.3}$$

and

$$\gamma_i \;\geq\; 0 \qquad i = 1, \ldots, q \,.$$

Denote by $I := \{i \in \{1, \ldots, q\} : \gamma_i > 0\}$ the index set of all positive γ_i‘s. Assume that the set $\{y_i : i \in I\}$ is linear dependent. It follows that there exists a vector $\beta \in \mathbb{R}^{|I|}$, $\beta \neq 0$, with

$$0 \;=\; \sum_{i \in I} \beta_i y_i \,,$$

i.e., there exists a non-trivial representation of the origin by the elements of $\{y_i : i \in I\}$. We assume further, without loss of generality, that

$$\max_{i \in I} \beta_i \;>\; 0 \,.$$

Set $\alpha := \min_{i \in I}\{\frac{\gamma_i}{\beta_i} : \beta_i > 0\}$ and let $i' \in I$ be one of the indices where this minimum is attained, i.e., $\alpha = \frac{\gamma_{i'}}{\beta_{i'}}$. We obtain

$$\gamma_i - \alpha\beta_i \;\geq\; 0 \quad i \in I \,, \tag{A.5.4}$$

$$\gamma_{i'} - \alpha\beta_{i'} \;=\; 0 \tag{A.5.5}$$

and

$$x \;=\; \sum_{i\in I}\gamma_i y_i \;=\; \sum_{i\in I}\gamma_i y_i - \alpha\sum_{i\in I}\beta_i y_i \;=\; \sum_{i\in I\setminus\{i'\}}(\gamma_i - \alpha\beta_i)y_i \;. \qquad \text{(A.5.6)}$$

This means that x is an element of the cone generated by the points y_i ($i \in I\setminus\{i'\}$). In this way we see that as long as the set $\{y_i : i \in I\}$ is linear dependent we are able to reduce the number of necessary elements in the representation (A.5.3) of x, i.e., of elements y_i with $\gamma_i > 0$, by at least one. Therefore, the required relation can be deduced by induction. ■

With these two lemmas we are now able to develop the pronounced result of the following Corollary A.6. This result, though really technical, will ease the proof of the subsequent Lemma A.7.

COROLLARY A.6. *Let $L = \{y_1, \ldots, y_q\} \subset \mathbb{R}^n$ ($q \in \mathbb{N}$) be an arbitrary finite set of n-dimensional points, and let $CO \subset \mathbb{R}^n$ be defined as in Lemma A.5. Let further $\{x^k\}_{k\in\mathbb{N}}$ be a point sequence in CO. Then there exist a positive real value τ, and, for each $k \in \mathbb{N}$, a linear independent subset $LI^k = \{y^k_{i_0}, \ldots, y^k_{i_{\bar{q}(k)}}\}$, a point $w^k \in [y^k_{i_0}, \ldots, y^k_{i_{\bar{q}(k)}}]$ and a real value $\beta^k \geq 0$ satisfying*

$$x^k \;=\; \beta^k w^k \qquad \text{(A.5.7)}$$

and

$$\|w^k\|_2 \;\geq\; \tau \;. \qquad \text{(A.5.8)}$$

PROOF: Choose a fixed $k \in \mathbb{N}$. Since x^k is contained in the cone CO we know, in view of Lemma A.5, that there exists a linear independent subset $LI^k = \{y^k_{i_0}, \ldots, y^k_{i_{\bar{q}(k)}}\}$ ($\bar{q}(k) < n$) of L such that

$$x^k \;\in\; CO_{LI^k} = \{x \in \mathbb{R}^n : x = \sum_{j=0}^{\bar{q}(k)} \gamma_j y_{i_j}, \gamma \in \mathbb{R}^{\bar{q}(k)+1}_+ \} \;.$$

Therefore, there is a vector $\gamma^k \in \mathbb{R}^{\bar{q}(k)+1}_+$ with

$$x^k \;=\; \sum_{j=0}^{\bar{q}(k)} \gamma^k_j y_{i_j} \;.$$

Lemma A.4 yields the existence of a real value $\tau > 0$ independent of the set LI^k satisfying, for all $x \in [y_{i_0}, \ldots, y_{i_{\bar{q}(k)}}]$,

$$\|x\|_2 \;\geq\; \tau \;. \qquad \text{(A.5.9)}$$

If there holds $\sum_{j=0}^{\bar{q}(k)} \gamma_j^k = 0$, then we can choose for w^k an arbitrary element of the $\bar{q}(k)$-simplex $[y_{i_0}, \dots, y_{i_{\bar{q}(k)}}]$. By setting $\beta^k = 0$ we obtain the required value, and in view of (A.5.9) it follows that w^k is a point satisfying Properties (A.5.7) and (A.5.8). In the case that $\sum_{j=0}^{\bar{q}(k)} \gamma_j^k = \bar{\gamma}^k > 0$ is true, we obtain by setting

$$w^k \; := \; \sum_{j=0}^{\bar{q}(k)} \frac{\gamma_j^k}{\bar{\gamma}^k} y_{i_j} \; \in \; [y_{i_0}, \dots, y_{i_{\bar{q}(k)}}]$$

and

$$\beta^k \; := \; \bar{\gamma}^k \; > \; 0$$

the postulated results. ■

Using this corollary we prove now the substantial part for the proof of Lemma 4.4.8. This will be done with the next lemma.

LEMMA A.7. *Let $S = [v_0, \dots, v_r] \subset \mathrm{I\!R}^n$ be an r-simplex with $1 \le r < n$ and $\bar{P} := \{x \in \mathrm{I\!R}^n : \bar{A}x \le \bar{b}\}$ be a polytope with $\bar{A} = (\bar{a}_1, \dots, \bar{a}_{\bar{m}})^T \in \mathrm{I\!R}^{\bar{m} \times n}$ and $\bar{b} \in \mathrm{I\!R}^{\bar{m}}$. Let further $\{r^k\}_{k \in \mathrm{I\!N}}$ be a point sequence satisfying, for each $k \in \mathrm{I\!N}$,*

$$r^k \; \in \; S \tag{A.5.10.a}$$

and

$$\bar{a}_j^T r^k \; \le \; \bar{b}_j + \nu^k \qquad j = 1, \dots, \bar{m} \tag{A.5.10.b}$$

with a positive real-valued sequence $\{\nu^k\}_{k \in \mathrm{I\!N}}$ converging to 0.

Then there exist a real value $C > 0$ and a point sequence $\{\bar{r}^k\}_{k \in \mathrm{I\!N}}$ with the properties, for all $k \in \mathrm{I\!N}$,

$$\bar{r}^k \; \in \; S \cap \bar{P} \tag{A.5.11.a}$$

and

$$\|r^k - \bar{r}^k\|_2 \; \le \; C\nu^k \; . \tag{A.5.11.b}$$

Moreover, there is an affine r-dimensional subspace $\mathcal{H}$ containing S and a matrix $H \in \mathrm{I\!R}^{(n-r) \times n}$ with linear independent rows $h_{r+i} \in \mathrm{I\!R}^n$ ($i = 1, \dots, n-r$) satisfying

$$\mathcal{H} \; = \; \{x \in \mathrm{I\!R}^n : Hx = Hv_0\} \tag{A.5.12.a}$$

and, for $x \in S$,

$$h_{r+i}^T x \; = \; 0 \qquad i = 1, \dots, n-r \; . \tag{A.5.12.b}$$

PROOF: We will show that the projection of r^k on the set $S \cap \bar{P}$ has Properties (A.5.11.a) and (A.5.11.b). However, first of all, we have to show that such a projection exists, i.e., that the set $S \cap \bar{P}$ is not empty. The set S is compact. Therefore, there exists a convergent subsequence $\{r^{k_q}\}_{q \in \mathbb{N}}$ of $\{r^k\}_{k \in \mathbb{N}}$, i.e.,

$$r^{k_q} \ \rightarrow \ r \ \in \ S \qquad (q \rightarrow \infty)\ .$$

With Property (A.5.10.b) of the sequence $\{r^k\}_{k \in \mathbb{N}}$ we obtain, for each $j \in \{1, \ldots, \bar{m}\}$,

$$\begin{array}{ccc} \bar{a}_j^T r^{k_q} & \leq & \bar{b}_j + \nu^{k_q} \\ \downarrow & (q \rightarrow \infty) & \downarrow \\ \bar{a}_j^T r & \leq & \bar{b}_j + \ 0 \ , \end{array}$$

and, thus, r is contained also in $\bar{P}$, i.e.,

$$S \cap \bar{P} \ \neq \ \emptyset\ .$$

In order to prove that the projection $\bar{r}^k$ on the set $S \cap \bar{P}$ has Property (A.5.11.b) we use the Karush-Kuhn-Tucker (KKT) optimality conditions of the convex optimization problem, which delivers this projection as its solution. Therefore, we need a representation of this optimization problem. This will be done in the following.

Let $\mathcal{H} = \{x \in \mathbb{R}^n : x = \sum_{i=0}^r \lambda_i v_i, \lambda \in \mathbb{R}^{r+1}, \sum_{i=0}^r \lambda_i = 1\}$ be the r-dimensional affine subspace of $\mathbb{R}^n$ containing the simplex $S = [v_0, \ldots, v_r]$ and $\{h_1, \ldots, h_r\}$ be a base of $\mathcal{H}$. Let further $\mathcal{H}^\perp$ be the orthogonal complement of $\mathcal{H}$ with base $\{h_{r+1}, \ldots, h_n\}$. If we denote by $H \in \mathbb{R}^{(n-r)\times n}$ the matrix with rows h_{r+i} ($i = 1, \ldots, n-r$), there holds

$$\mathcal{H} \ = \ \{x \in \mathbb{R}^n : Hx = Hv_0\}\ . \tag{A.5.13}$$

In order to describe the r-simplex S by a system of linear equalities and inequalities let, for each $i \in \{0, \ldots, r\}$, $\bar{v}_i \in \mathbb{R}^n$ be the normed normal of the facet $S_i = [v_0, \ldots, v_{i-1}, v_{i+1}, \ldots, v_r]$ with respect to the subspace $\mathcal{H}$, i.e., the point $\bar{v}_i$ is the unique solution of the following system

$$(v_j - v_{i'})^T \bar{v}_i \ = \ 0 \qquad j \in \{0, \ldots, r\} \setminus \{i, i'\}\ , \tag{A.5.14.a}$$

$$h_{r+j}^T \bar{v}_i \ = \ 0 \qquad j \in \{1, \ldots, n-r\} \tag{A.5.14.b}$$

and

$$\|\bar{v}_i\|_2 \ = \ 1 \quad , \quad \bar{v}_i^T v_i \ < \ \bar{v}_i^T v_{i'} =: c_i \tag{A.5.14.c}$$

for an arbitrary, but fixed $i' \in \{0,\dots,r\}\setminus\{i\}$. Note that there holds $r \geq 1$ and thus $\{0,\dots,r\} \setminus \{i\} \neq \emptyset$. A solution of this system always exists. Indeed, the solution of the systems (A.5.14.a) and (A.5.14.b) of linear equations is a line, since the set $\hat{L} = \{(v_j - v_i), j \in \{0,\dots,r\} \setminus \{i,i'\}\ ,\ h_{r+j}, j \in \{1,\dots,n-r\}\}$ is linear independent with $|\hat{L}| = n-1$. The constraints (A.5.14.c) guarantees the uniqueness of the solution and, additionally, that, for all $x \in S$, there holds $\bar{v}_i^T x \leq c_i$ ($i \in \{0,\dots,r\}$).

With these normal vectors and the representation (A.5.13) of $\mathcal{H}$ we obtain a description of S by linear equalities and inequalities. There holds

$$S \;=\; \{x \in \mathbb{R}^n :\ Hx = Hv_0\ ,\ \bar{v}_i^T x \leq c_i\, ,\ i = 0,\dots,r\}\ .$$

Now we are able to formulate in the following way the convex optimization problem (OPP), which has the orthogonal projection of r^k on the set $S \cap \bar{P}$ as its solution.

$$\begin{aligned} \min\ \|r^k - x\|_2^2 & \\ \bar{A}x \;&\leq\; \bar{b} \\ Hx \;&=\; Hv_0 \\ \bar{v}_i^T x \;&\leq\; c_i \qquad i = 0,\dots,r \\ x \;&\in\; \mathbb{R}^n \end{aligned} \tag{OPP}$$

Using the KKT optimality conditions for the optimal solution $\bar{r}^k$ of (OPP) (see, for example, [HOR79, FLE87, MAN94]) we obtain that there exist index sets $I_1^k \subset \{1,\dots,\bar{m}\}$ and $I_2^k \subset \{1,\dots,r\}$ satisfying

$$\begin{aligned} \bar{a}_i^T \bar{r}^k \;&=\; \bar{b}_i \qquad i \in I_1^k\ , \\ \bar{v}_i^T \bar{r}^k \;&=\; c_i \qquad i \in I_2^k\ , \end{aligned}$$

and, additionally, there are real vectors $\gamma^1 \in \mathbb{R}_+^{|I_1^k|}$, $\gamma^2 \in \mathbb{R}_+^{|I_2^k|}$ and $\gamma^3 \in \mathbb{R}^{n-r}$ with

$$r^k - \bar{r}^k \;=\; \sum_{i \in I_1^k} \gamma_i^1 \bar{a}_i \;+\; \sum_{i \in I_2^k} \gamma_i^2 \bar{v}_i \;+\; \sum_{i=1}^{n-r} \gamma_i^3 h_{r+i}\ .$$

It follows that the vector $r^k - \bar{r}^k$ is contained in the cone generated by the elements of the finite set

$$\begin{aligned} L^k \;=\; &\{\bar{a}_i, i \in I_1^k\} \cup \{\bar{v}_i, i \in I_2^k\} \\ &\qquad \cup \{h_{r+i}, i = 1,\dots,n-r\} \cup \{-h_{r+i}, i = 1,\dots,n-r\}\ . \end{aligned}$$

From Corollary A.6 we know that there exists a linear independent subset $LI^k = \{y_0^k, \dots, y_{\bar{q}(k)}^k\} \subset L^k$ ($\bar{q}(k) < n$), a point $w^k = [y_0^k, \dots, y_{\bar{q}(k)}^k]$ and a real value β^k with

$$r^k - \bar{r}^k \;=\; \beta^k w^k \;,$$

in particular,

$$\beta^k \;=\; \frac{\|r^k - \bar{r}^k\|_2}{\|w^k\|_2} \;. \tag{A.5.15}$$

Moreover, since there is only a finite number of possibilities for the set L^k, Corollary A.6 yields the existence of a positive real value τ, independent of k, satisfying

$$\|w^k\|_2 \;\geq\; \tau \;. \tag{A.5.16}$$

Select now a point $\bar{\lambda} \in B_{\bar{q}(k)}$ with $w^k = \sum_{i=0}^{\bar{q}(k)} \bar{\lambda}_i y_i^k$ and set $J_1 := \{i : y_i^k \in \{\bar{a}_j, j \in I_1^k\}\}$, $J_2 := \{i : y_i^k \in \{\bar{v}_j, j \in I_2^k\}\}$, $J_3 = \{i : y_i^k \in \{h_{r+j}, j = 1, \dots, n-r\}\}$ and $J_4 = \{i : y_i^k \in \{-h_{r+j}, j = 1, \dots, n-r\}\}$. It follows

$$\begin{aligned}
0 \;\leq\; \|r^k - \bar{r}^k\|_2^2 \;&=\; \beta^k (w^k)^T (r^k - \bar{r}^k) \\
&=\; \beta^k \sum_{i=0}^{\bar{q}(k)} \bar{\lambda}_i (y_i^k)^T (r^k - \bar{r}^k) \\
&=\; \beta^k \Bigg(\sum_{i \in J_1} \bar{\lambda}_i (\underbrace{\bar{a}_i^T r^k}_{\leq \bar{b}_i + \nu^k} - \underbrace{\bar{a}_i^T \bar{r}^k}_{= \bar{b}_i}) + \sum_{j \in J_2} \bar{\lambda}_i (\underbrace{\bar{v}_i^T r^k}_{\leq c_i} - \underbrace{\bar{v}_i^T \bar{r}^k}_{= c_i}) \\
&\qquad + \sum_{i \in J_3} \bar{\lambda}_i \underbrace{h_{r+i}^T (r^k - \bar{r}^k)}_{= 0} - \sum_{i \in J_4} \bar{\lambda}_i \underbrace{h_{r+i}^T (r^k - \bar{r}^k)}_{= 0} \Bigg) \\
&\leq\; \beta^k \underbrace{\sum_{i \in J_1} \bar{\lambda}_i}_{\leq 1} \nu^k \;\leq\; \beta^k \nu^k \;.
\end{aligned}$$

By substituting β^k with (A.5.15) we obtain

$$\|r^k - \bar{r}^k\|_2 \|w^k\|_2 \;\leq\; \nu^k \;.$$

Hence, with (A.5.16) and $C := \frac{1}{\tau}$ it follows, for each $k \in \mathbb{N}$,

$$\|r^k - \bar{r}^k\|_2 \;\leq\; C\nu^k \;.$$

■

Now we complete the proof of Lemma 4.4.8 for both considered problem classes.

PROOF OF LEMMA 4.4.8 FOR (DCP$_1$) AND (DCP$_2$): If we denote by $S = [v_0, \dots, v_r]$ the r-simplex, which is the fixed face of the residual simplices $\{S^k\}_{k \geq \bar{K}}$, then – in view of Lemma 4.4.7 – we know that, for each $k \geq \bar{K}$, there is a point $r^k \in S$ with Properties (4.4.14.a)-(4.4.14.c). The set

$$\bar{P} := \{x \in P : \textstyle\sum_{i=0}^{r} \lambda_i f^l(v_i) \leq 0 \ , \ l = 1, \dots, p \\ \text{with } \lambda \in B_r \ , \ x = \textstyle\sum_{i=0}^{r} \lambda_i v_i\} \ .$$

is a polytope. In order to describe the functions $\sum_{i=0}^{r} \lambda_i f^l(v_i)$ $(l = 0, \dots, p)$ by an inner product of $x = \sum_{i=0}^{r} \lambda_i v_i \in \mathrm{I\!R}^n$ and a vector $s^l \in \mathrm{I\!R}^n$ we use the representation (A.5.12.a) of the affine subspace $\mathcal{H}$ containing S, which is given by Lemma A.7. Let $s^l \in \mathrm{I\!R}^n$, for $l \in \{0, \dots, p\}$, be the unique solution of the following system of linear equations

$$\begin{aligned} (s^l)^T (v_i - v_0) &= f^l(v_i) - f^l(v_0) \qquad && i = 1, \dots, r \\ (s^l)^T h_{r+i} &= 0 && i = 1, \dots, n - r \end{aligned}$$

with h_{r+i} $(i = 1, \dots, n - r)$ given by Lemma A.7. Note that the set $\{(v_i - v_0), i = 1, \dots, r\}$ is a base of $\mathcal{H}$ and $\{h_{r+i}, i = 1, \dots, r\}$ is a base of the orthogonal complement of $\mathcal{H}$ (see Property (A.5.12.b)). Then we obtain, for $l \in \{0, \dots, p\}$ and $x \in S$,

$$(s^l)^T (x - v_0) + f^l(v_0) = \sum_{i=0}^{r} \lambda_i f^l(v_i) \tag{A.5.17}$$

with $\lambda \in B_r$, $x = \sum_{i=0}^{r} \lambda_i v_i$. With (A.5.17) we are able to describe the polytope $\bar{P}$ in the following way

$$\bar{P} = \{x \in \mathrm{I\!R}^n : a_i^T x \leq b_i \ , \ i = 1, \dots, m \ , \\ (s^l)^T x \leq f^l(v_0) + (s^l)^T v_0 \ , \ l = 1, \dots, p\} \ .$$

Because of Lemma 4.4.2 we know that, for each $k \geq \bar{K} \geq K$, the vertices $v_0, \dots, v_r$ of the simplex S^k are fixed. Therefore, for each $x \in S = [v_0, \dots, v_r]$, there holds that the function value of the convex envelope $\varphi^l_{S^k}$ $(l = 0, \dots, p)$ does not depend on k.

Actually, for $\lambda \in B_r$ with $x = \sum_{i=0}^{r} \lambda_i v_i$ and $k \geq \bar{K}$, there holds

$$\varphi^l_{S^k}(x) \;=\; \sum_{i=0}^{r} \lambda_i f^l(v_i)\;. \tag{A.5.18}$$

Combining (A.5.17) and (A.5.18) and by using Property (4.4.14.b) of r^k we obtain, for $k \geq \bar{K}$ and $l \in \{1, \dots, p\}$,

$$(s^l)^T r^k \;\leq\; f^0(v_0) + (s^l)^T v_0 + o(\Lambda^k)\;. \tag{A.5.19}$$

Furthermore, for $i \in \{1, \dots, m\}$, it follows from Property (4.4.14.c) of r^k that

$$\begin{aligned} a_i^T r^k \;&=\; a_i^T(r^k - \omega(S^k)) + \underbrace{a_i^T \omega(S^k)}_{\leq b_i \text{ since } \omega(S^k) \in P} \\ &\leq\; \|a_i\|_2 \|r^k - \omega(S^k)\|_2 + b_i \\ &\leq\; o(\Lambda^k) + b_i\;. \end{aligned} \tag{A.5.20}$$

From (A.5.19) and (A.5.20) we see, that the sequence $\{r^k\}_{k \geq \bar{K}}$ fulfills the assumptions of Lemma A.7 with respect to the polytope $\bar{P}$ and the r-simplex S. Thus, this lemma provides the existence of a sequence $\{\bar{r}^k\}_{k \in \mathbb{N}}$ satisfying, for all $k \geq \bar{K}$,

$$\bar{r}^k \;\in\; S \cap \bar{P}$$

and, additionally,

$$\|\omega(S^k) - \bar{r}^k\|_2 \;\leq\; \|\omega(S^k) - r^k\|_2 + \|r^k - \bar{r}^k\|_2 \;=\; o(\Lambda^k)\;,$$

i.e., $\bar{r}^k$ is an element of $\bar{F}$ and fulfills condition (4.4.16.b).

In order to prove that $\bar{r}^k$ has also Property (4.4.16.a) we use Relations (A.5.17) and (A.5.18) for $l = 0$, and exploit again the Lipschitz continuity of g^0 on the compact set P (see the proof of Lemma 4.4.7). If L_P is a Lipschitz constant of g^0 on P, then, for $k \geq \bar{K}$, there holds

$$|g^0(r^k) - g^0(\bar{r}^k)| \;\leq\; L_P \|r^k - \bar{r}^k\|_2 \;=\; o(\Lambda^k)$$

and

$$|\varphi^0_{S^k}(r^k) - \varphi^0_{S^k}(\bar{r}^k)| \;=\; |(s^0)^T(r^k - \bar{r}^k)| \;\leq\; \|s^0\|_2 \|r^k - \bar{r}^k\|_2 \;=\; o(\Lambda^k)\;.$$

Therefore, it follows

$$g^0(\bar{r}^k) + \varphi^0_{S^k}(\bar{r}^k) \;=\; g^0(r^k) + \varphi^0_{S^k}(r^k) + o(\Lambda^k)\;,$$

and the use of Property (4.4.14.a) concludes the proof. ■

APPENDIX B

Solution Methods for (DCP^S)

In this appendix we are interested in some solution methods for the convex optimization problem

$$\begin{aligned} &\min\ g^0(x) \\ &g^l(x)\ \leq 0 \qquad l=1,\dots,p \qquad\qquad \text{(CP)} \\ &x \in P \cap S\ , \end{aligned}$$

where $g^l\ :\ \mathbb{R}^n \to \mathbb{R}$ ($l = 0,\dots,p$) are convex functions, $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ with $A = (a_1,\dots,a_m)^T \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$ is a non-empty full-dimensional polytope, and $S = [v_0,\dots,v_n] = \{x \in \mathbb{R}^n : (\bar{v}_i^S)^T x \leq c_i^S,\ i=0,\dots,n\}$ is an n-simplex (see Problem (DCP^S) in Section 4.2 and see also (4.2.3.a), (4.2.3.b) for the construction of $\bar{v}_i^S$ and c_i^S ($i = 0,\dots,n$)). These solution methods should detect in finite time either the emptiness of the feasible region $F := \{x \in P \cap S : g^l(x) \leq 0\ ,\ l=1,\dots,p\}$ or an (ϵ, δ, 0)-solution $\bar{x} \in \mathbb{R}^n$ (see Definition 4.2.1), i.e., a point with the properties

$$\bar{x}\ \in\ P \cap S\,, \qquad \text{(B.a)}$$

$$g^l(\bar{x})\ \leq\ \delta \qquad l=1,\dots,p \qquad \text{(B.b)}$$

and

$$g^0(\bar{x}) - \epsilon\ \leq\ \min_{x\in F} g^0(x) \qquad \text{(B.c)}$$

for prespecified tolerances $\epsilon, \delta > 0$ (see also the definition of a CONVEXSOLVER$_{\epsilon,\delta,0}$ in Section 4.2).

In the first section of this appendix we use the concept of subgradients of convex functions (see, e.g., [ROC70, ROC81, SHO85]) to develop a solution method following the cutting-plane approach given, for example, in [KEL60] (see Remark 4.2.1(c)). In Section B.2 we assume that a solution method for (CP) is given, which detects in finitely many iterations either the emptiness of F or an ($\tilde{\epsilon}$, $\tilde{\delta}$, $\tilde{\rho}$)-solution $\tilde{x} \in \mathbb{R}^n$ (see again Definition 4.2.1), i.e., a point with the properties

$$\begin{aligned} a_j^T \tilde{x} - b_j \ &\leq \ \tilde{\rho} \qquad j = 1, \dots, m\,, \\ (\bar{v}_i^S)^T \tilde{x} - c_i^S \ &\leq \ \tilde{\rho} \qquad i = 0, \dots, n\,, \end{aligned} \tag{B.a‘}$$

$$g^l(\tilde{x}) \ \leq \ \tilde{\delta} \qquad l = 1, \dots, p \tag{B.b‘}$$

and

$$g^0(\tilde{x}) - \tilde{\epsilon} \ \leq \ \min_{x \in F} g^0(x) \tag{B.c‘}$$

for arbitrary accuracies $\tilde{\epsilon}$, $\tilde{\delta}$, $\tilde{\rho} > 0$. We show that in this situation it is possible to construct an (ϵ, δ, 0)-solution $\bar{x} \in \mathbb{R}^n$ of (CP) by using the orthogonal projection of $\tilde{x}$ on the set $P \cap S$, and, in particular, we are able to specify the necessary values $\tilde{\epsilon}$, $\tilde{\delta}$ and $\tilde{\rho}$.

B.1. The Kelley-Cheney-Goldstein Cutting-Plane Approach

A **subgradient** ξ of a convex function $g : \mathbb{R}^n \to \mathbb{R}$ at a point $y \in \mathbb{R}^n$ is defined as an n-dimensional vector with the property

$$g(x) \ \geq \ g(y) + \xi^T(x - y)\ , \ \forall x \in \mathbb{R}^n\ . \tag{B.1.1}$$

As customary we denote by $\partial g(y)$ the set of all subgradients of g at the point y. This set is called the **subdifferential** of g at y. It is known that, for an arbitrary convex function g and an arbitrary point $y \in \mathbb{R}^n$, the subdifferential $\partial g(y)$ is non-empty, bounded, convex and closed [SHO85, Theorem 1.7]. In general it can be a hard problem to calculate a subgradient, but for some interesting classes of convex functions the subgradients are known (see, e.g., [SHO85, Section 1.3]). If the function g is differentiable, then there holds $\partial g(y) = \{\nabla g(y)\}$ for each $y \in \mathbb{R}^n$ [ROC70, Theorem 25.1].

In the following we assume that the linear set $P \cap S$ is not empty. This assumption can be verified by the first phase of the Simplex-Algorithm. We do not require that a feasible point $\bar{x} \in F$ for Problem (CP) exists, since in our branch-and-bound Algorithm 4.1 in Section 4.2 we are not able to verify such an assumption for all convex subproblems of the form (DCP^S). This is the main reason for the description of a CONVEXSOLVER$_{\epsilon,\delta,0}$ for Problem (CP) in this appendix, even though this method is very similar to the KCG-method [CG59, KEL60] or to other outer approximation methods for optimization problems with convex feasible sets given, for example, in [HTT87, HT96B].

The algorithm is as follows.

ALGORITHM B.1 (***A CONVEXSOLVER***$_{\epsilon,\delta,0}$ ***for (CP)***).

```
Initialization
    Choose real numbers $\epsilon$, $\delta \geq 0$ and a point $x^0 \in P \cap S$.
    Compute a vector $\xi^{0,0} \in \partial g^0(x^0)$ and set
    $F^1 \leftarrow \{\binom{x}{t} \in \mathbb{R}^{n+1} : x \in P \cap S\ ,\ g^0(x^0) + (\xi^{0,0})^T(x - x^0) - t \leq 0\}$,
    STOP $\leftarrow$ False , $k \leftarrow 1$
While STOP $=$ False Do
    If $F^k = \emptyset$ Then                                                  (SC1)
        STOP $\leftarrow$ True ($F = \emptyset$)
    Else
        Solve the linear optimization problem $\min_{\binom{x}{t} \in F^k} t$ and let $\binom{x^k}{t^k}$ be
        an optimal solution.
        If ( $g^l(x^k) \leq \delta$ , $l = 1, \ldots, p$ ) AND ( $g^0(x^k) - t^k \leq \epsilon$ ) Then     (SC2)
            STOP $\leftarrow$ True ($x^k$ is an ($\epsilon$, $\delta$, 0)-solution of (CP))
        Else
            $F^{k+1} \leftarrow F^k$
            For $l = 1$ To $p$ Do
                If $g^l(x^k) > 0$ Then
                    Compute $\xi^{k,l} \in \partial g^l(x^k)$.
                    If $\xi^{k,l} = 0$ Then                                    (SC3)
                        STOP $\leftarrow$ True ($F = \emptyset$)
                    Else
                        $F^{k+1} \leftarrow F^{k+1} \cap \{\binom{x}{t} \in \mathbb{R}^{n+1} : g^l(x^k) + (\xi^{k,l})^T(x - x^k) \leq 0\}$
                    EndIf
                EndIf
            EndFor
            If $g^0(x^k) - t^k > 0$ Then
                Compute $\xi^{k,0} \in \partial g^0(x^k)$ and set
                $F^{k+1} \leftarrow F^{k+1} \cap \{\binom{x}{t} \in \mathbb{R}^{n+1} : g^0(x^k) + (\xi^{k,0})^T(x - x^k) - t \leq 0\}$.
            EndIf
        EndIf
    EndIf
    $k \leftarrow k + 1$
EndWhile
```

Algorithm B.1 does not solve Problem (CP) directly. Indeed, this method solves the equivalent problem

$$\begin{aligned} &\min t \\ g^0(x) \ &\le \ t \\ g^l(x) \ &\le \ 0 \qquad l = 1, \ldots, p \\ x &\in P \cap S \ , \end{aligned} \tag{$\overline{\text{CP}}$}$$

which has a linear objective function. If we denote by $\bar{F}$ the feasible region of ($\overline{\text{CP}}$), i.e.,

$$\bar{F} \ = \ \{\tbinom{x}{t} \in \mathrm{I\!R}^{n+1} : x \in P \cap S \ , \ g^0(x) \le t \ , \ g^l(x) \le 0 \ , \ l = 1, \ldots, p\} \ ,$$

then it follows immediately from Property (B.1.1) of the subgradients that, for each $k \in \mathrm{I\!N}$, the set F^k is an outer approximation of $\bar{F}$, i.e.,

$$\bar{F} \ \subset \ F^{k+1} \ \subset \ F^k \qquad (k \in \mathrm{I\!N}) \ . \tag{B.1.2}$$

Therefore, the emptiness of $\bar{F}$ follows if F^k is empty, and because of

$$\bar{F} \ = \ \emptyset \quad \Leftrightarrow \quad F \ = \ \emptyset$$

we obtain, in particular, the emptiness of F (see stopping criterion (SC1) in Algorithm B.1).

If a point $x^k \in P \cap S$ is infeasible with respect to a convex constraint $g^l(x) \le 0$ ($l \in \{1, \ldots, p\}$) and there holds $0 \in \partial g^l(x^k)$, then it follows by (B.1.1), for each $x \in \mathrm{I\!R}^n$,

$$g^l(x) \ \ge \ g^l(x^k) \ > \ 0 \ .$$

This is the second possibility to detect the emptiness of F (see stopping criterion (SC3) in Algorithm B.1).

As long as F^k is not empty, we know in view of (B.1.2) that t^k ($k \in \mathrm{I\!N}$), which is the solution value of the linear optimization problem $\min_{\binom{x}{t} \in F^k} t$, is a lower bound for the optimal value $t^\star$ of Problem ($\overline{\text{CP}}$). Note that the optimal value of ($\overline{\text{CP}}$) is ∞, if no feasible point $\binom{x}{t} \in \bar{F}$ exists. If the stopping criterion (SC2) is satisfied, it is clear that x^k is a (δ, 0)-feasible point for Problem (CP). Furthermore, we know

$$g^0(x^k) - \epsilon \ \le \ t^k \ \le \ \min_{\binom{x}{t} \in \bar{F}} t \ = \ \min_{x \in F} g^0(x) \ , \tag{B.1.3}$$

and, therefore, x^k fulfills Conditions (B.a)-(B.c), i.e., x^k is an (ϵ, δ, 0)-solution of Problem (CP).

Because of the previous notes we know that, when finite, Algorithm B.1 will solve Problem (CP) in the required way. In order to obtain the correctness of the presented method, we still have to prove that Algorithm B.1 is always finite, if the tolerances ϵ and δ are chosen greater than 0. This will be done by showing the convergence of Algorithm B.1 for $\epsilon = \delta = 0$, which is the result of the next theorem.

THEOREM B.1. *Assume that $\epsilon = \delta = 0$ and that Algorithm B.1 generates an infinite point sequence $\{\binom{x^k}{t^k}\}_{k\in\mathbb{N}}$. Then each accumulation point $x^\star \in \mathbb{R}^n$ of the sequence $\{x^k\}_{k\in\mathbb{N}}$ is an optimal solution of Problem (CP).*

PROOF: Let $x^\star$ be an accumulation point of the sequence $\{x^k\}_{k\in\mathbb{N}}$ and let $\{x^{k_q}\}_{q\in\mathbb{N}}$ be a subsequence of $\{x^k\}_{k\in\mathbb{N}}$ converging to $x^\star$. Since there holds, for each $q \in \mathbb{N}$, that x^{k_q} is an element of $P \cap S$ we obtain $x^\star \in P \cap S$. Denote by

$$I \ := \ \{l \in \{1,\dots,p\} : \ |\{q \in \mathbb{N} : g^l(x^{k_q}) > 0\}| = \infty\}$$

the index set of all convex constraints of (CP), which are infinitely often violated by the sequence $\{x^{k_q}\}_{q\in\mathbb{N}}$. If I is empty, then it follows immediately that $x^\star$ is a feasible point. Indeed, in this case, there must exist a number $Q \in \mathbb{N}$ with

$$g^l(x^{k_q}) \ \le \ 0 \qquad l = 1,\dots,p\ ,\ q \ge Q\ ,$$

and, in particular, because of the continuity of the convex functions g^l ($l = 1\dots,p$) (see, e.g., [ROC70, Theorem 10.1]) it follows

$$g^l(x^\star) \ \le \ 0 \qquad l = 1,\dots,p\ .$$

If I is not empty, then we are still able to prove the feasibility of $x^\star$. Indeed, choose $l \in I$ and assume, without loss of generality, that there holds $g^l(x^{k_q}) > 0$, for all $q \in \mathbb{N}$. The set $\{x^{k_q} : q \in \mathbb{N}\}$ is bounded. Therefore, it follows by [ROC70, Theorem 24.7] that the set $\{\xi^{k_q,l} : q \in \mathbb{N}\}$ is also bounded. We assume further that the sequence $\{\xi^{k_q,l}\}_{q\in\mathbb{N}}$ is convergent to a point $\xi^{\star,l} \in \mathbb{R}^n$. (There always exists a subsequence of $\{\xi^{k_q,l}\}_{q\in\mathbb{N}}$ with this property.) With respect to [ROC70, Theorem 24.4] there even holds $\xi^{\star,l} \in \partial g^l(x^\star)$. Because of $x^{k_{q+1}} \in F^{k_{q+1}-1} \subset F^{k_q}$ ($q \in \mathbb{N}$) we obtain

$$\begin{array}{ccccccc} g^l(x^{k_q}) & + & (\xi^{k_q,l})^T & (x^{k_{q+1}} & - x^{k_q}) & \leq 0 & \\ \downarrow & & \downarrow & \downarrow & \downarrow & & (q \to \infty) \\ g^l(x^\star) & + & (\ \xi^{\star,l})^T & (\ x^\star & - \ \ x^\star) & \leq 0\,. & \end{array} \tag{B.1.4}$$

Relation (B.1.4) is true for each $l \in I$. This shows the feasibility of $x^\star$.

In order to complete the proof we have to show the optimality of $x^\star$. If there holds

$$|\{q \in \mathbb{N}:\ g^0(x^{k_q}) - t^{k_q}\ >\ 0\}|\ <\ \infty\,, \tag{B.1.5}$$

then there exists a number $Q \in \mathbb{N}$ such that, for each $q \geq Q$,

$$g^0(x^{k_q})\ \leq\ t^{k_q}\ \leq\ \min_{x\in F} g^0(x)\ \leq\ g^0(x^\star)$$

(compare with (B.1.3)). Using the continuity of g^0 we obtain the optimality of $x^\star$. If (B.1.5) is not fulfilled, then it follows by the same argumentation as before

$$\begin{array}{cccccccc} g^0(x^{k_q}) & + & (\xi^{k_q,0})^T & (x^{k_{q+1}} & - x^{k_q}) & \leq t^{k_q} \leq & \min\limits_{x\in F} g^0(x) & \\ \downarrow & & \downarrow & \downarrow & \downarrow & (q \to \infty) & & \\ g^0(x^\star) & + & (\ \xi^{\star,0})^T & (\ x^\star & - \ \ x^\star) & \leq & \min\limits_{x\in F} g^0(x)\ \leq\ g^0(x^\star) & . \end{array}$$

Therefore, in each case $x^\star$ is an optimal solution of Problem (CP). ■

As a direct consequence of the previous proof we are able to conclude that Algorithm B.1 with $\epsilon = \delta = 0$ will stop after a finite number of iterations, if no feasible point exists, i.e., if $F = \emptyset$. A second consequence is that the presented solution method for Problem (CP) is always finite, if the tolerances ϵ and δ are chosen greater than 0. Therefore, Algorithm B.1 can be used as a CONVEXSOLVER$_{\epsilon,\delta,0}$ for solving the convex subproblems of the form (DCP^S) in Algorithm 4.1 proposed in Section 4.2.

B.2. Another Approach for Obtaining an (ϵ, δ, 0)-solution

We assume now that a CONVEXSOLVER$_{\tilde{\epsilon},\tilde{\delta},\tilde{\rho}}$ is given, i.e., a solution method for Problem (CP) which detects in finite time either the emptiness of F or a point $\tilde{x} \in \mathbb{R}^n$ with properties (B.a‘)-(B.c‘), and we assume again that $P \cap S$ is not empty. In this section we show that it is possible to choose the accuracies $\tilde{\epsilon}$, $\tilde{\delta}$, $\tilde{\rho} > 0$ such that the orthogonal projection $\bar{x}$ of $\tilde{x}$ on the set $P \cap S$ is an (ϵ, δ, 0)-solution of Problem (CP) (see (B.a)-(B.c)) for arbitrary tolerances ϵ, $\delta > 0$. Since $\bar{x}$ is the optimal solution of a quadratic convex optimization problem with linear constraints, this point can be calculated exactly in finite time (see Remark 4.2.1(b)). Therefore, by combining the given CONVEXSOLVER$_{\tilde{\epsilon},\tilde{\delta},\tilde{\rho}}$, with appropriate accuracies, with a finite solver for convex quadratic optimization problems we obtain a CONVEXSOLVER$_{\epsilon,\delta,0}$.

In order to prove that the orthogonal projection $\bar{x}$ of $\tilde{x}$ on the set $P \cap S$ has the required properties we use an analogous argumentation as in the proof of Lemma A.7. In the proof of this lemma it is sufficient that we have the existence of Lipschitz constants for the involved convex functions and the existence of a positive real value τ delivered by Corollary A.6. In the present section we would like to quantify the values of $\tilde{\epsilon}$, $\tilde{\delta}$ and $\tilde{\rho}$ depending on ϵ and δ. Therefore, we have to show that it is possible to calculate Lipschitz constants for the convex functions g^l ($l \in \{0, \dots, p\}$) and, furthermore, that we are able to calculate the value τ.

Due to [ROC70, Theorem 24.7] we know that the convex functions g^l ($l \in \{0, \dots, p\}$) are Lipschitz continuous on each compact set $C \subset \mathrm{I\!R}^n$ with Lipschitz constant

$$L^l_C \;=\; \max\{\|\xi\|_2 : \xi \in \partial g^l(y)\,,\, y \in C\}\,. \tag{B.2.1}$$

Note that the set $\{\xi : \xi \in \partial g(y), y \in C\}$ is compact (see also [ROC70, Theorem 24.7]). If g^l ($l \in \{0, \dots, p\}$) is differentiable and the set C is a polytope with known vertex set $V(C)$, then, as long as $\|\nabla g^l(x)\|_2$ is a convex function, we are able to calculate the value L^l_C by using the following relation

$$L^l_C \;=\; \max_{x \in C} \|\nabla g^l(x)\|_2 \;=\; \max_{x \in V(C)} \|\nabla g^l(x)\|_2\,.$$

In general, we do not know how to solve the optimization problem given in (B.2.1). Nevertheless, the following lemma yields an upper bound for L^l_C, which is computable if a CONVEXSOLVER$_{\tilde{\epsilon},\tilde{\delta},\tilde{\rho}}$ is known.

LEMMA B.2. *Let $g : \mathrm{I\!R}^n \to \mathrm{I\!R}$ be a convex function, Q,$Z \subset \mathrm{I\!R}^n$ be polytopes with known vertex sets $V(Q)$ and $V(Z)$ and the additional property that Z contains the unit ball $B = \{x \in \mathrm{I\!R}^n : \|x\|_2 \leq 1\}$. Let further $\bar{x} \in \mathrm{I\!R}^n$ be an ($\bar{\epsilon}$, 0, $\bar{\rho}$)-solution of the convex optimization problem $\min_{x \in Q} g(x)$ with $\bar{\epsilon}$, $\bar{\rho} \geq 0$. Then an upper bound for the Lipschitz constant L_Q of g on the set Q is given by*

$$\bar{L}_Q \;:=\; \max_{x \in V(Q)} \max_{z \in V(Z)} \; g(x+z) \,-\, g(\bar{x}) + \bar{\epsilon}\,. \tag{B.2.2}$$

PROOF: In view of Property (B.1.1) of a subgradient we know that, for each $y \in \mathrm{I\!R}^n$ and $\xi \in \partial g(y)$, there holds

$$\xi^T z \;\leq\; g(y+z) - g(y) \qquad \forall z \in \mathrm{I\!R}^n$$

and, therefore,

$$\max_{\xi \in \partial g(y)} \xi^T z \;\le\; g(y+z) - g(y) \qquad \forall z \in \mathbb{R}^n \,. \tag{B.2.3}$$

Furthermore, it is immediately clear that the following relation holds for the Euclidean norm. Let $x \in \mathbb{R}^n$ be an arbitrary point. Then we know that

$$\|x\|_2 \;=\; \max_{\substack{z \,\in\, \mathbb{R}^n \\ \|z\|_2 = 1}} \; x^T z \,. \tag{B.2.4}$$

Combining Relation (B.2.3) with Relation (B.2.4) we obtain an upper bound for L_Q.

$$\begin{aligned} L_Q \;=\; \max_{\substack{\xi \,\in\, \partial g(y) \\ y \,\in\, Q}} \|\xi\|_2 \;&=\; \max_{y \in Q} \max_{\xi \in \partial g(y)} \max_{\substack{z \,\in\, \mathbb{R}^n \\ \|z\|_2 = 1}} \xi^T z \\ &=\; \max_{y \in Q} \max_{\substack{z \,\in\, \mathbb{R}^n \\ \|z\|_2 = 1}} \underbrace{\max_{\xi \in \partial g(y)} \xi^T z}_{\le g(y+z) - g(y)} \\ &\le\; \max_{y \in Q} \max_{\substack{z \,\in\, \mathbb{R}^n \\ \|z\|_2 = 1}} g(y+z) - \underbrace{\min_{y \in Q} g(y)}_{\ge g(\bar{x}) - \bar{\epsilon}} \\ &\le\; \max_{y \in Q} \max_{z \in Z} g(y+z) - g(\bar{x}) + \bar{\epsilon} \;=\; \bar{L}_Q \,. \end{aligned}$$

The function $g(y + \cdot) : \mathbb{R}^n \to \mathbb{R}$ is convex for $y \in \mathbb{R}^n$. Using the facts that a convex function attains its maximum over a polytope in a vertex of this polytope [HPT95, Theorem 1.19] and that the maximum over an arbitrary family of convex functions is again a convex function, it follows

$$\max_{y \in Q} \max_{z \in Z} g(y+z) \;=\; \max_{y \in Q} \underbrace{(\max_{z \in V(Z)} g(y+z))}_{\text{convex in } y} \;=\; \max_{y \in V(Q)} \max_{z \in V(Z)} g(y+z) \,,$$

which proves (B.2.2). ∎

REMARK B.1. A polytope which contains the unit ball B is obviously the hypercube

$$Z \;=\; \{x \in \mathbb{R}^n : \|x\|_\infty \le 1\} \,.$$

This polytope is a good approximation of $B_{0,1}$, but it has 2^n vertices. Another possible polytope which has the required properties and which has only $n+1$ vertices is the regular simplex given in Chapter 2 (see, in particular, Subsection

2.6.2). It is immediately clear using the same ideas as in Section 2.6 that this regular simplex can be enlarged such that it contains B. Admittedly, this simplex is in general a worse approximation of the set B than the hypercube Z.

Apart from the Lipschitz constants we also need a way to quantify a value $\tau > 0$ with the property that, for each linear independent subset $\{y_0, \dots, y_q\}$ ($q < n$) of the set

$$L \;=\; \{a_j, j = 1, \dots, m\} \cup \{\bar{v}_i^S, i = 0, \dots, n\}\,,$$

there holds

$$\|x\|_2 \;\geq\; \tau \qquad \forall x \in [y_0, \dots, y_q]\,. \tag{B.2.5}$$

This value can be calculated in the following way. Let e_i ($i \in \{1, \dots, n\}$) be the i-th unit vector and denote by

$$\begin{aligned}\mathcal{R} \;:=\; \{&B = (b_1, \dots, b_n)^T \in \mathrm{I\!R}^{n\times n} : b_i \in L, i = 1, \dots, r\,,\\ &b_i \in \{e_1, \dots, e_n\}, i = r+1, \dots, n\,, B \text{ regular }\,,\ 1 \leq r \leq n\}\end{aligned}$$

the finite set of all regular matrices with at least one row $b_i \in L$ ($i \in \{1, \dots, n\}$). Set

$$\tau \;:=\; \frac{1}{\sqrt{n}} \min_{B\in\mathcal{R}} \frac{1}{\|B^{-1}\|}\,, \tag{B.2.6}$$

where $\|\cdot\| : \mathrm{I\!R}^{n\times n} \to \mathrm{I\!R}$ denotes an arbitrary norm on the space of ($n\times n$)-matrices, which is compatible with the Euclidean norm. Then there holds the following.

Let $\{y_0, \dots, y_q\}$ ($q \leq n$) be a linear independent subset of L, and let x be an element of the q-simplex $[y_0, \dots, y_q]$. Then there exists a matrix $B \in \mathcal{R}$ with $b_i = y_{i-1}$ ($i = 1, \dots, q+1$) and $b_i \in \{e_1, \dots, e_n\}$ ($i = q+2, \dots, n$), and there is a vector $\bar{\lambda} \in B_{n-1}$ with

$$x \;=\; \sum_{i=1}^{q+1} \bar{\lambda}_i y_{i-1} \;=\; B\bar{\lambda}$$

and

$$\bar{\lambda}_i \;=\; 0 \qquad i = q+2, \dots, n\,.$$

Using the facts that there hold $\bar{\lambda} = B^{-1}x$ and $\min_{\lambda\in B_{n-1}} \|\lambda\|_2 = \frac{1}{\sqrt{n}}$ we obtain

$$\|x\|_2 \;\geq\; \frac{\|\bar{\lambda}\|_2}{\|B^{-1}\|} \;\geq\; \frac{1}{\sqrt{n}}\frac{1}{\|B^{-1}\|} \;\geq\; \tau\,.$$

Therefore, the value τ defined in (B.2.6) has Property (B.2.5) for each linear independent subset $\{y_0, \dots, y_q\}$ $(q < n)$ of L. The final theorem quantifies now the necessary values of $\tilde{\epsilon}$, $\tilde{\delta}$ and $\tilde{\rho}$.

THEOREM B.3. *Let ϵ, $\delta > 0$. Let further $L^l_{\bar{S}}$ be a Lipschitz constant of g^l $(l \in \{0, \dots, p\})$ on the simplex $\bar{S} := \{x \in \mathbb{R}^n : (\bar{v}^S_i)^T x \leq c^S_i + 1,\ i = 0, \dots, n\} \supset S$ and let the real value $\tau > 0$ be given by (B.2.6). Assume that $\tilde{x}$ is an ($\tilde{\epsilon}$, $\tilde{\delta}$, $\tilde{\rho}$)-solution of (CP) with*

$$\tilde{\rho} := \min\{1\,,\ \frac{\epsilon\tau}{2L^0_{\bar{S}}}\,,\ \frac{\delta\tau}{2L^l_{\bar{S}}}\,,\ l = 1, \dots, p\} \ > \ 0\,,$$

$$\tilde{\delta} := \ \delta - \frac{\tilde{\rho}}{\tau} \max_{l=1,\dots,p} L^l_{\bar{S}} \ \geq \ \frac{\delta}{2}$$

and

$$\tilde{\epsilon} := \ \epsilon - \frac{\tilde{\rho}}{\tau} L^0_{\bar{S}} \ \geq \ \frac{\epsilon}{2}\,.$$

Then the orthogonal projection $\bar{x}$ of $\tilde{x}$ on the set $P \cap S$ is an (ϵ, δ, 0)-solution of Problem (CP).

PROOF: The orthogonal projection $\bar{x}$ of $\tilde{x}$ on the set $P \cap S$ is the solution of the following convex optimization problem (compare with the proof of Lemma A.7)

$$\begin{aligned} \min\ \|\tilde{x} - x\|_2^2 & \\ a_j^T x \ \leq \ b_i \qquad & j = 1, \dots, m \\ (\bar{v}^S_i)^T x \ \leq \ c_i \qquad & i = 0, \dots, n \\ x \ \in \ \mathbb{R}^n\,. & \end{aligned} \tag{OP}$$

Using the same argumentation as in the proof of Lemma A.7 (see pages 311f.) we know that there exist two index sets $I_1 \subset \{1, \dots, m\}$ and $I_2 \subset \{0, \dots, n\}$ with

$$a_i^T \bar{x} \ = \ b_i \quad i \in I_1 \qquad , \qquad (\bar{v}^S_i)^T \bar{x} = c^S_i \quad i \in I_2\,,$$

and, additionally, a linear independent subset $\{y_0, \dots, y_q\}$ $(q < n)$ of the set $L := \{a_i, i \in I_1\} \cup \{\bar{v}^S_i, i \in I_2\}$, a point $w \in [y_0, \dots, y_q]$ and a real value β with

$$\tilde{x} - \bar{x} \ = \ \beta w \quad , \quad \beta \ = \ \frac{\|\tilde{x} - \bar{x}\|_2}{\|w\|_2}\,. \tag{B.2.7}$$

Since τ has Property (B.2.5) with respect to each linear independent subset of L, we obtain

$$\|w\| \;\geq\; \tau\,. \tag{B.2.8}$$

Select $\bar{\lambda} \in B_q$ with $w = \sum_{i=0}^{q} \bar{\lambda}_i y_i$, and set $J_1 := \{j : y_j \in \{a_i, i \in I_1\}\}$ and $J_2 := \{j : y_j \in \{\bar{v}_i^S, i \in I_2\}\}$. By using the fact that $\tilde{x}$ is an ($\tilde{\epsilon}$, $\tilde{\delta}$, $\tilde{\rho}$)-solution of (CP) it follows

$$\begin{aligned}
\|\tilde{x} - \bar{x}\|_2^2 \;&=\; \beta w^T(\tilde{x} - \bar{x}) \\
&=\; \beta \left(\sum_{i=0}^{q} \bar{\lambda}_i y_i^T (\tilde{x} - \bar{x}) \right) \\
&=\; \beta \left(\sum_{i \in J_1} \bar{\lambda}_i (\underbrace{a_i^T \tilde{x}}_{\leq b_i + \tilde{\rho}} - \underbrace{a_i^T \bar{x}}_{= b_i}) + \sum_{i \in J_2} \bar{\lambda}_i (\underbrace{(\bar{v}_i^S)^T \tilde{x}}_{\leq c_i^S + \tilde{\rho}} - \underbrace{(\bar{v}_i^S)^T \bar{x}}_{= c_i^S}) \right) \\
&\leq\; \beta \underbrace{\sum_{i=0}^{q} \bar{\lambda}_i}_{=1} \;\tilde{\rho} \;=\; \beta \tilde{\rho}\,.
\end{aligned}$$

Substituting β with (B.2.7) and using (B.2.8) we obtain

$$\|\tilde{x} - \bar{x}\|_2 \;\leq\; \frac{\tilde{\rho}}{\|w\|_2} \;\leq\; \frac{\tilde{\rho}}{\tau}\,.$$

The real value $L_{\bar{S}}^l$ is by assumption a Lipschitz constant of g^l ($l \in \{0, \ldots, p\}$) on the set $\bar{S}$. Therefore, using the definition of $\tilde{\epsilon}$ and $\tilde{\delta}$ and the fact that there holds $\tilde{x}, \bar{x} \in \bar{S}$, it follows

$$g^0(\bar{x}) - \epsilon \;\leq\; g^0(\tilde{x}) + \underbrace{L_{\bar{S}}^0 \frac{\tilde{\rho}}{\tau} - \epsilon}_{= -\tilde{\epsilon}} \;\leq\; \min_{x \in F} g^0(x)$$

and, for $l \in \{1, \ldots, p\}$,

$$g^l(\bar{x}) \;\leq\; g^l(\tilde{x}) + L_{\bar{S}}^l \frac{\tilde{\rho}}{\tau} \;\leq\; \tilde{\delta} + \frac{L_{\bar{S}}^l \tilde{\rho}}{\tau} \;\leq\; \delta\,.$$

This means that $\bar{x}$ is an (ϵ, δ, 0)-solution of Problem (CP). ∎

This theorem shows that it is always possible to adjust each CONVEXSOLVER$_{\tilde{\epsilon},\tilde{\delta},\tilde{\rho}}$ in order to obtain a solver for Problem (CP), which delivers in finite time a solution with the same quality as Algorithm B.1 does. Whether this is numerically practicable depends on the effort which is necessary for calculating the Lipschitz constants and the value τ. Note that the calculation of these values must be done only with respect to the start simplex S^0, if we apply such an adjusted CONVEXSOLVER$_{\tilde{\epsilon},\tilde{\delta},\tilde{\rho}}$ for solving the subproblem (DCP^S) in Algorithm 4.1. Nevertheless, these calculations are in general expensive.

Bibliography

[AK92] Faiz A. Al-Khayyal. Generalized Bilinear Programming: Part I. Models, Applications and Linear Programming Relaxation. *European Journal of Operational Research*, 60:306–314, 1992.

[AKF83] Faiz A. Al-Khayyal and J.E. Falk. Jointly Constrained Biconvex Programming. *Annals of Operations Research*, 25:169–180, 1983.

[AKHP92] Faiz A. Al-Khayyal, R. Horst, and P.M. Pardalos. Global Optimization of Concave Functions subject to Quadratic Constraints: An Application in Nonlinear Bilevel Programming. *Annals of Operations Reserach*, 34:125–147, 1992.

[AKLV95] Faiz A. Al-Khayyal, C. Larsen, and T. van Voorhis. A Relaxation Method for Nonconvex Quadratically Constrained Quadratic Programs. *Journal of Global Optimization*, 6:215–230, 1995.

[AKV96] Faiz A. Al-Khayyal and T. van Voorhis. Accelerating Convergence of Branch-and-Bound Algorithms for Quadratically Constrained Optimization Problems. In C.A. Floudas, editor, *State of the Art in Global Optimization: Computational Methods and Applications*. Kluwer Academic Publishers, Dordrecht/Boston/London, 1996.

[Ali95] F. Alizadeh. Interior Point Methods in Semidefinite Programming with Applications to Combinatorial Optimization. *SIAM Journal in Optimization*, 5(1):13–51, 1995.

[AT98] Le Thi An and Pham Dinh Tao. A Branch-and-Bound Method via D.C. Optimization Algorithms and Ellipsoidal Technique for Box Constrained Nonconvex Quadratic Problems. *Journal of Global Optimization*, 13(2):171–206, 1998.

[Bar72] D.P. Baron. Quadratic Programming with Quadratic Constraints. *Naval Research Quarterly*, 19:253–260, 1972.

[Ben85] H.P. Benson. A Finite Algorithm for Concave Minimization over a Polyhedron. *Naval Research Logistics Quaterly*, 32:165–177, 1985.

[Ben95] H.P. Benson. Concave Minimization: Theory, Applications and Algorithms. In R. Horst and P.M. Pardalos, editors, *Handbook of Global Optimization*. Kluwer Academic Publishers, Dordrecht/Boston/London, 1995.

[Bom97] I.M. Bomze. Global Escape Strategies for Maximizing Quadratic Forms over a Simplex. *Journal of Global Optimization*, 11(2):325–338, 1997.

[BR95] C.G.E. Boender and H.E. Romeijn. Stochastic Methods. In R. Horst and P.M. Pardalos, editors, *Handbook of Global Optimization*. Kluwer Academic Publishers, Dordrecht/Boston/London, 1995.

[BS94] H.P. Benson and S. Sayin. A Finite Concave Minimization Algorithm using Branch-and-Bound and Neighbor Generation. *Journal of Global Optimization*, 5:1–14, 1994.

[CG59] E.W. Cheney and A.A. Goldstein. Newton's Method of Convex Programming and Tchebycheff Approximation. *Numerische Mathematik*, 1:253–268, 1959.

[CPS92] R.W. Cottle, Jong-Shi Pang, and R.E. Stone. *The Linear Complementarity Problem*. Academic Press, Inc., San Diego, USA, 1992.

[Dan63] G.B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, Princeton, 1963.

[DAPT97] P.L. De Angelis, P.M. Pardalos, and G. Toraldo. Quadratic Programming with Box Constraints. In I.M. Bomze, T. Csendes, R. Horst, and P.M. Pardalos, editors, *Developments in Global Optimization*. Kluwer Academic Publishers, Dordrecht, 1997.

[dGPW90] C. de Groot, R. Peikert, and D. Würtz. The Optimal Packing of Ten Equal Circles in a Square. IPS Research Report 90-12, ETH Zürich, 1990.

[dGPWM91] C. de Groot, R. Peikert, D. Würtz, and M. Monagan. Packing Circles in a Square: A Review and New Results. In P. Kall, editor, *System Modelling and Optimization*, pages 45–54. Proc. 15th IFIP Conf. Zürich, 1991.

[DT92] E.V. Donardo and C.S. Tang. Linear Control of a Markov Production System. *Operations Research*, 40(2):259–278, 1992.

[EN75] J.G. Ecker and R.D. Niemi. A Dual Method for Quadratic Programs with Quadratic Constraints. *SIAM Journal on Applied Mathematics*, 28(3):568–576, 1975.

[Eva63] D.H. Evans. Modular Design – A Special Case in Nonlinear Programming. *Operations Research*, 11:637–647, 1963.

[Eva70] D.H. Evans. A Note on "Modular Design – A Special Case in Nonlinear Programming". *Operations Research*, 18:562–564, 1970.

[FK97] T. Fujie and M. Kojima. Semidefinite Programming Relaxation for Nonconvex Quadratic Programs. *Journal of Global Optimization*, 10:367–380, 1997.

[Fle87] R. Fletcher. *Practical Methods of Optimization*. Princeton University Press, John Wiley and Sons, 2nd edition, 1987.

[FM68] A.V. Fiacco and G.P. McCormick. *Nonlinear Programming*. John Wiley, New York, 1968.

[FS69] J.E. Falk and R.M. Soland. An Algorithm for Separable Nonconvex Programming Problems. *Management Science*, 15:550–569, 1969.

[FS87] J.A. Filar and T.A. Schultz. Bilinear Programming and Structured Stochastic Games. *Journal of Optimization Theory and Applications*, 53(1):85–104, 1987.

[FV90a] C.A. Floudas and V. Visweswaran. A Global Optimization Algorithm (GOP) for Certain Classes of Nonconvex NLP's: II. Applications of Theory and Test Problems. *Computers and Chemical Engineering*, 14:1417–1434, 1990.

[FV90b] C.A. Floudas and V. Visweswaran. A Global Optimization Algorithm (GOP) for Certain Classes of Nonconvex NLP's: I. Theory. *Computers and Chemical Engineering*, 14:1397–1417, 1990.

[FV93a] C.A. Floudas and V. Visweswaran. New Properties and Computational Improvement of the GOP Algorithm for Problems with Quadratic Objective Function and Constraints. *Journal of Global Optimization*, 3:439–462, 1993.

[FV93b] C.A. Floudas and V. Visweswaran. Primal-Relaxed Dual Global Optimization Approach. *Journal of Optimization Theory and Applications*, 78(2):187–225, 1993.

[FV95] C.A. Floudas and V. Visweswaran. Quadratic Optimization. In R. Horst and P.M. Pardalos, editors, *Handbook of Global Optimization*. Kluwer Academic Publishers, Dordrecht/Boston/London, 1995.

[GKL95] P. Gritzmann, V. Klee, and D. Larman. Largest j-Simplices in n-Polytopes. *Discrete Comput. Geom.*, 13:477–513, 1995.

[GL96] R.L. Graham and B.D. Lubachevsky. Repeated Patterns of Dense Packings of Equal Disks in a Square. *The Electronic J. of Comb.*, 3:1–16, 1996.

[GMW81] P.E. Gill, W. Murray, and M.H. Wright. *Practical Optimization*. Academic Press, Inc., San Diego, USA, 1981.

[Gol70] M. Goldberg. The Packing of Equal Circles in a Square. *Math. Mag.*, 43:24–30, 1970.

[GVL89] G.H. Golub and C.F. Van Loan. *Matrix Computations*. John Hopkins University Press, Baltimore, 2nd edition, 1989.

[Her94] D.den Hertog. *Interior Point Approach to Linear, Quadratic and Convex Programming: Algorithms and Complexity*. Kluwer Academic Publishers, Dordrecht/Boston/London, 1994.

[HJ85] R. Horn and C.R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, 1985.

[HJ92] P. Hansen and B. Jaumard. Reduction of Indefinite Quadratic Programs to Bilinear Programs. *Journal of Global Optimization*, 2:41–60, 1992.

[Hor76] R. Horst. An Algorithm for Nonconvex Programming Problems. *Mathematical Programming*, 10:312–321, 1976.

[Hor79] R. Horst. *Nonlinear Optimization*. Carl Hanser Verlag, München, 1979. in German.

[Hor84] R. Horst. On the Global Minimization of Concave Functions. Introduction and Survey. *Operations Research Spektrum*, 6:195–205, 1984.

[Hor97] R. Horst. On Generalized Bisection of n-Simplices. *Mathematics of Computation*, 66(218):691–698, 1997.

[HPT95] R. Horst, P.M. Pardalos, and N.V. Thoai. *Introduction to Global Optimization*. Kluwer Academic Publishers, Dordrecht/Boston/London, 1995.

[HR98] R. Horst and U. Raber. Convergent Outer Approximation Algorithms for Solving Unary Problems. *Journal of Global Optimization*, 13:123–149, 1998.

[HT96a] R. Horst and N.V. Thoai. A new Algorithm for Solving the General Quadratic Programming Problem. *Computational Optimization and Applications*, 5:39–48, 1996.

[HT96b] R. Horst and H. Tuy. *Global Optimization: Deterministic Approaches*. Springer, Heidelberg, 3rd enlarged edition, 1996.

[HT99] R. Horst and N.V. Thoai. D.C. Programming: An Overview with New Results. *Journal of Optimization Theory and Applications*, 1999. to appear.

[HTT87] R. Horst, N.V. Thoai, and H. Tuy. Outer Approximation by Polyhedral Convex Sets. *OR Spektrum*, 9:153–159, 1987.

[HW53] J. Hoffman and H.W. Wielandt. The Variation of the Spectrum of a Normal Matrix. *Duke Mathematical Journal*, 20:37–39, 1953.

[ILM88] H. Idrissi, P. Loridan, and C. Michelot. Approximation of Solutions for Location Problems. *Journal of Optimization Theory and Applications*, 56:127–143, 1988.

[Jar96] F. Jarre. Interior-Point Methods for Classes of Convex Programming. In T. Terlaky, editor, *Interior Point Methods of Mathematical Programming*. Kluwer Academic Publishers, Dordrecht/Boston/London, 1996.

[JM98] B. Jaumard and C. Meyer. A Simplified Convergence Proof for the Cone Partitioning Algorithm. *Journal of Global Optimization*, 13(4):407–416, 1998.

[JRA93] L.W. Johnson, R.D. Riess, and J.T. Arnold. *Introduction to Linear Algebra*. Addison-Wesley Publishing Company, 3rd edition, 1993.

[Kea78] B. Kearfott. A Proof of Convergence and an Error Bound for the Method of Bisection in $\mathbb{R}^n$. *Mathematics of Computation*, 32:1147–1153, 1978.

[Kel60] J.E. Kelley. The Cutting-Plane Method for Solving Convex Programs. *SIAM Journal on Applied Mathematics*, 8:703–712, 1960.

[Loc97] M. Locatelli. Finiteness of Conical Algorithms with ω-Subdivisions. submitted, 1997.

[LR97a] M. Locatelli and U. Raber. A Finiteness Result for the Simplicial Branch-and-Bound Algorithm based on ω-Subdivisions. Technical Note, submitted, 1997.

[LR97b] M. Locatelli and U. Raber. On the Convergence of the Simplicial Branch-and-Bound Algorithm based on ω-Subdivisions. submitted, 1997.

[LR98a] M. Locatelli and U. Raber. Packing Equal Circles in a Square: I. Theoretical Results. 1998. submitted.

[LR98b] M. Locatelli and U. Raber. Packing Equal Circles in a Square: II. A Deterministic Global Optimization Approach. 1998. submitted.

[Man94] O. L. Mangasarian. Nonlinear programming. In *Classics in Applied Mathematics*, volume 10. SIAM, Philadelphia, 1994.

[MFP95] C.D. Maranas, C.A. Floudas, and P.M. Pardalos. New Results in the Packing of Equal Circles in a Square. *Discrete Mathematics*, 142:287–293, 1995.

[Nas96] M. Nast. Subdivision of Simplices Relative to a Cutting Plane and Finite Concave Minimization. *Journal of Global Optimization*, 9:65–93, 1996.

[Nes98] Y. Nesterov. Semidefinite Relaxation and Nonconvex Quadratic Optimization. *Optimization Methods & Software*, 9(1-3):141–160, 1998.

[NN94] J.E. Nesterov and A. Nemirovskii. *Interior Point Polynomial Algorithms in Convex Programming*. SIAM, Philadelphia, 1994.

[NO97] K.J. Nurmela and P.R.J. Oestergard. Packing up to 50 Equal Circles in a Square. *Discrete Comput. Geom.*, 18:111–120, 1997.

[NS92] V.H. Nguyen and J.J. Strodiot. Computing a Global Optimal Solution to a Design Centering Problem. *Mathematical Programming*, 53:111–123, 1992.

[PhH82] E. Phan-huy Hao. Quadratically Constrained Quadratic Programming: Some Applications and a Method for Solution. *Zeitschrift für Operations Research*, 26:105–119, 1982.

[PR86] P.M. Pardalos and J.B. Rosen. Methods for Global Concave Optimization: A Bibliographic Survey. *SIAM Review*, 26:367–379, 1986.

[PRW95] S. Poljak, F. Rendl, and H. Wolkowicz. A Recipe for Semidefinite Relaxation for $(0, 1)$-Quadratic Programming. *Journal of Global Optimization*, 7:51–73, 1995.

[PS76] P.M. Pardalos and G. Schnitger. Connections between Nonlinear and Integer Programming Problems. *Symposia Mathematica*, 19:161–176, 1976.

[PS88] P.M. Pardalos and G. Schnitger. Checking Local Optimality in Constrained Quadratic Programming is $\mathcal{NP}$-hard. *Operations Research Letters*, 7:33–35, 1988.

[PTA94] Thai Quynh Phing, Pham Dinh Tao, and Le Thi Hoai An. A Method for Solving D.C. Programming Problems, Application to Fuel Mixture Nonconvex Optimization Problems. *Journal of Global Optimization*, 6:87–105, 1994.

[PV91] P.M. Pardalos and S.A. Vavasis. Quadratic Programming with one Negative Eigenvalue is $\mathcal{NP}$-hard. *Journal of Global Optimization*, 1:15–22, 1991.

[QdKRT98] A.J. Quist, E. de Klerk, C. Roos, and T. Terlaky. Copositive Relaxation for General Quadratic Programming. *Optimization Methods & Software*, 9(1-3):185–208, 1998.

[Rab98] U. Raber. A Simplicial Branch-and-Bound Method for Solving Nonconvex All-Quadratic Programs. *Journal of Global Optimization*, 13:417–432, 1998.

[Ram93] M. Ramana. *An Algorithmic Analysis of Multiquadratic and Semidefinite Programming Problems*. PhD thesis, The John Hopkins University, Baltimore, 1993.

[Ree75] G.R. Reeves. Global Minimization in Nonconvex All-Quadratic Programming. *Management Science*, 22(1):76–86, 1975.

[Roc70] R.T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, New Jersey, 1970.

[Roc81] R.T. Rockafellar. *The Theory of Subgradients and its Applications to Problems of Optimization and Nonconvex Functions*. Heldermann Verlag, Berlin, 1981.

[RS71] D.P. Rutenberg and T.L. Shaftel. Product Design: Subassemblies for Multiple Markets. *Management Science*, 18(4):B–220–B–231, 1971.

[SA92] H.D. Sherali and A. Alameddine. A new Reformulation-Linearization Technique for Bilinear Programming Problems. *Journal of Global Optimization*, 2:379–410, 1992.

[SA99] H.D. Sherali and W.P. Adams. *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Programs*. Kluwer, Dordrecht/Boston/London, 1999.

[Sch65] J. Schaer. The Densest Packing of Nine Circles in a Square. *Canad.Math.Bull.*, 8:273–277, 1965.

[Sho85] N.Z. Shor. *Minimization Methods for Non-Differentiable Functions*. Springer, Heidelberg, 1985.

[Sho87] N.Z. Shor. Quadratic Optimization Problems. *Soviet J. Computer and Systems Sciences*, 25:1–11, 1987.

[Sho98] N.Z. Shor. *Nondifferentiable Optimization and Polynomial Problems*. Kluwer Academic Publishers, Dordrecht/Boston/London, 1998.

[Sle69] D. Slepan. The Content of some Extreme Simplices. *Pacific J. Math.*, 31:795–808, 1969.

[SM65] J. Schaer and A. Meir. On a Geometric Extremum Problem. *Canad.Math.Bull.*, 8:21–27, 1965.

[Sol71] R.M. Soland. An Algorithm for Separable Nonconvex Programming Problems II: Nonconvex Constraints. *Management Science*, 17(11):759–773, 1971.

[Som29] D.M.Y. Sommerville. *An Introduction to the Geometry of N Dimensions*. Methuen, London, 1929.

[ST92] H.D. Sherali and C.H. Tuncbilek. A Global Optimization Algorithm for Polynomial Programming Problems Using a Reformulation-Linearization Technique. *Journal of Global Optimization*, 2:101–112, 1992.

[Ste98] I. Stewart. Mathematical Recreations. *Scientific American*, pages 80–82, February 1998.

[TB85] B.T. Tam and V.T. Ban. Minimization of a Concave Function under Linear Constraints. *Economika i Mathematicheskie Metody*, 21:709–714, 1985. in Russian.

[Tha88] P.T. Thach. The Design Centering Problem as a D.C. Programming Problem. *Mathematical Programming*, 41:229–248, 1988.

[TT85] N.V. Thuong and H. Tuy. Minimizing a Convex Function over the Complement of a Convex Set. *Methods of Operations Research*, 49:85–89, 1985.

[Tuy64] H. Tuy. Concave Programming under Linear Constraints. *Soviet Mathematics*, 5:1437–1440, 1964.

[Tuy91a] H. Tuy. Effect of the Subdivision Strategy on Convergence and Efficiency of some Global Optimization Algorithms. *Journal of Global Optimization*, 1:23–36, 1991.

[Tuy91b] H. Tuy. Normal Conical Algorithm for Concave Minimization over Polytopes. *Mathematical Programming*, 51:229–245, 1991.

[Tuy95] H. Tuy. D.C. Optimization: Theory, Methods and Applications. In R. Horst and P.M. Pardalos, editors, *Handbook of Global Optimization*. Kluwer Academic Publishers, Dordrecht/Boston/London, 1995.

[VB96] L. Vandenberghe and S. Boyd. Semidefinite Programming. *SIAM Review*, 38:49–95, 1996.

[vdP66] C. van de Panne. Programming with a Quadratic Constraint. *Management Science*, 12(11):798–815, 1966.

[VS82] L. Vidigal and Director S. A Design Centering Algorithm for Nonconvex Regions of Acceptability. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pages 13–24, 1982.

[Wen83] G. Wengerodt. Die dichteste Packung von 16 Kreisen in einem Quadrat. *Beiträge Algebra Geom.*, 16:173–190, 1983.

[Wen87a] G. Wengerodt. Die dichteste Packung von 14 Kreisen in einem Quadrat. *Beiträge Algebra Geom.*, 25:25–46, 1987.

[Wen87b] G. Wengerodt. Die dichteste Packung von 25 Kreisen in einem Quadrat. *Ann.Univ.Sci.Budapest Eötvös Sect. Math.*, 30:3–15, 1987.

[WK87] G. Wengerodt and K. Kirchner. Die dichteste Packung von 36 Kreisen in einem Quadrat. *Beiträge Algebra Geom.*, 25:147–159, 1987.

[WV91] A. Weintraub and J. Vera. A Cutting Plane Approach for Chance Constrained Linear Programs. *Operations Research*, 39(5):776–785, 1991.

[YF98] Y. Yajima and T. Fujie. A Polyhedral Approach for Nonconvex Quadratic Programming Problems with Box Constraints. *Journal of Global Optimization*, 13(2):151–170, 1998.

[Zur64] R. Zurmühl. *Matrices*. Springer, Heidelberg, 4th edition, 1964. in German.

List of Tables

List of Figures

Tabellarischer Bildungsweg

Name:	Ulrich Raber
Geburtstag:	30.12.1971
Geburtsort:	Losheim am See
09/1978 –08/1982:	Grundschule Losheim
09/1982 –05/1991:	Hochwald–Gymnasium Wadern
10/1991 –11/1996:	Studium der Wirtschaftsmathematik an der Universität Trier mit Abschluß Diplom
01/1997 –09/1999:	Wissenschaftlicher Mitarbeiter in der Abteilung Mathematik im Fachbereich IV der Universität Trier
09/99:	Promotion zum Dr. rer. nat.