Adaptive Algorithms for Semi-Infinite Programming
with Arbitrary Index Sets

Zur Erlangung des akademischen Grades
eines Doktors der Wirtschaftswissenschaften

(Dr. rer. pol)

an der
Fakultät für Wirtschaftswissenschaften
des Karlsruher Institut für Technologie (KIT)

genehmigte

DISSERTATION

von

Dipl.-Math. Heinz-Paul Steuermann

Referent: Prof. Dr. Oliver Stein
Koreferent: Prof. Dr. Karl-Heinz Waldmann

Karlsruhe 2011
Contents

1 Introduction 1

2 Background 7
  2.1 Stationarity conditions 7
  2.2 Overestimating techniques 8
    2.2.1 The unimodalization method 8
    2.2.2 The $\alpha BB$ method 14
    2.2.3 A reduced outer approximation 15

3 The Adaptive-Reduction Algorithm 17
  3.1 Relaxation and reformulation 17
  3.2 Algorithms 19
  3.3 Convergence results 24
  3.4 A first numerical example 32

4 The Adaptive-Convexification Algorithm 37
  4.1 Relaxation and reformulation 37
  4.2 Algorithms 40
  4.3 Convergence results 42
  4.4 A first numerical example 46

5 An $X$-adaptation method 51
  5.1 The Adaptive-Reduction Algorithm with $X$-adaptation 51
    5.1.1 Relaxation and reformulation 51
    5.1.2 Algorithms and $X$-Adaptation 53
    5.1.3 Convergence results 58
  5.2 The Adaptive-Convexification Algorithm with $X$-adaptation 63
    5.2.1 Relaxation and reformulation 63
    5.2.2 Algorithms and $X$-Adaptation 65
    5.2.3 Convergence results 69

6 The hybrid method 75
  6.1 Motivation and reformulation 75
  6.2 Algorithms and convergence results 77

7 Implementation Details 83
  7.1 General information 83
Contents

7.2 Regularization of MPCC ........................................... 85
7.3 A phase 1 algorithm ................................................. 88
7.4 X-adaptation strategies ............................................. 89
7.5 Reduction of complexity ............................................ 90

8 Numerical examples .................................................... 95

9 Final remarks ............................................................. 139

References ................................................................. 146

List of figures ............................................................. 149

List of algorithms ......................................................... 151

Curriculum Vitae .......................................................... 153
1 Introduction

We consider (standard) semi-infinite optimization problems of the form

$$\text{SIP : } \min_{x \in X} f(x) \quad \text{s.t. } g(x, y) \leq 0 \quad \text{for all } y \in Y$$

with $X = [x^\ell, x^u] \subset \mathbb{R}^n$, $x^\ell < x^u$, $Y \subset \mathbb{R}^m$ and $f \in C^p(\mathbb{R}^n, \mathbb{R})$, $g \in C^p(\mathbb{R}^n \times \mathbb{R}^m, \mathbb{R})$ with $p \geq 1$. The ideas presented here can easily be generalized to the case of more than one semi-infinite constraint.

There is a wide variety of applications of semi-infinite optimization. Examples for these applications are

- Chebyshev approximation,
- Robust optimization,
- Minimax problems,
- Design centering,
- Defect minimization for operator equations.

Here we will introduce two applications in more detail, that is, Chebyshev approximation and design centering problems. For details on the theory, more applications, not covered by the mentioned problem classes, and many references on semi-infinite optimization we refer to the excellent reviews [23, 40].

Chebyshev approximation deals with the approximation of a given function $F$ on a compact set $Y$ by a function $a(x, \cdot)$ with parameters $x \in M \subset \mathbb{R}^n$ so that the maximal error becomes small. That leads to the non-smooth problem

$$\text{CA : } \min_{x \in M} \| F(\cdot) - a(x, \cdot) \|_{\infty,Y} = \min_{x \in M} \max_{y \in Y} | F(y) - a(x, y) |.$$  

By an epigraph reformulation $\text{CA}$ can be equivalently rewritten as

$$\text{EPI}_{\text{CA}} : \min_{(x, z) \in M \times \mathbb{R}} z \quad \text{s.t. } \max_{y \in Y} | F(y) - a(x, y) | \leq z.$$  

It is not hard to see that $\max_{y \in Y} | F(y) - a(x, y) | \leq z$ holds if and only if $| F(y) - a(x, y) | \leq z$ holds for all $y \in Y$. Thus, we can rewrite $\text{EPI}_{\text{CA}}$ as the following semi-infinite problem

$$\text{SIP}_{\text{CA}} : \min_{(x, z) \in M \times \mathbb{R}} z \quad \text{s.t. } F(y) - a(x, y) \leq z \quad \text{for all } y \in Y$$

$$- F(y) + a(x, y) \leq z \quad \text{for all } y \in Y.$$
1 Introduction

We have to mention that the problem \( \text{SIP}_{CA} \) is a smooth problem if all defining functions are also smooth.

Given a parametrized body \( B(x) \subset \mathbb{R}^m \) and a container \( C \subset \mathbb{R}^m \) a design centering problem consists in maximizing some measure, for example the area or the volume, of \( B(x) \) so that it is contained in \( C \). That leads to the problem:

\[
DC: \max_{x \in \mathbb{R}^n} \text{Vol}(B(x)) \quad \text{s.t.} \quad B(x) \subset C.
\]

Problems of this type arise, for example, if one wishes to minimize the cutoff in a manufacturing process, cf. [52, 53]. In [18, 24] the maneuverability problem of a robot is formulated as a special design centering problem. And in [49] the problem of deciding whether the quality of a manufactured element, produced in a fabrication process with unavoidable fluctuations, is modeled as a design centering problem. For more details on design centering problems we refer to [20, 36, 39].

Let us assume that \( C \) is described by a finite number of functional constraints, that is,

\[
C = \{ y \in \mathbb{R}^m \mid c_1(y) \leq 0, \ldots, c_q(y) \leq 0 \},
\]

and

\[
B(x) = \{ y \in \mathbb{R}^m \mid y = T(x, z), z \in Z \}
\]

with some fixed compact set \( Z \) and a function \( T \). Then, we can reformulate \( DC \), cf. [29], as a semi-infinite problem:

\[
\text{SIP}_{DC}: \min_{x \in \mathbb{R}^n} \text{Vol}(B(x)) \quad \text{s.t.} \quad c_i(T(x, z)) \leq 0 \text{ for all } z \in Z, i \in \{1, \ldots, q\}.
\]

There exist a wide range of numerical solution methods for linear and nonlinear \( \text{SIPs} \). Overviews can be found in [25, 41, 42, 45]. Traditional solution methods for \( \text{SIPs} \) are, for example, discretization, exchange and reduction-based methods. These methods generate a sequence of finite problems to approximate \( \text{SIP} \). All of these methods have in common that, at the beginning, some finite subset \( T \) of \( Y \) has to be chosen. In discretization-based methods new points of \( Y \) are added successively to \( T \), while exchange-based methods replace some points in \( T \) by some new points from \( Y \). The reduction-based methods make use of the local reduction theory proposed in [22] to determine a sequence of finite subsets of \( Y \). However, these methods suffer from the major drawback that their approximation of the feasible set \( X \cap M \) of \( \text{SIP} \), with

\[
M = \{ x \in \mathbb{R}^n \mid g(x, y) \leq 0 \text{ for all } y \in Y \},
\]

may contain infeasible points for the original problem. Other methods that do not suffer from this drawback use bi-level strategies and a branch-and-bound framework to handle the infinite number of constraints. However, most of these methods with feasible iterates like in [5, 6, 33, 34] focus on the global solution of \( \text{SIP} \). Throughout these papers it is assumed that the index set \( Y \) is given by a Cartesian product of intervals, that is, a box.
The method presented in [31] does not make assumptions on the shape of the index set $Y$, but also focuses on the global solution of $SIP$. Thus, additional properties of the objective function like convexity must be assumed, or one has to establish an additional branch and bound framework to ensure the global optimality.

The algorithm discussed in [16] does not focus on the global solution of $SIP$. The solution concept of this method is that of stationary points, and, thus, no additional properties of the objective function are assumed. However, a drawback of this method is that it can only handle one-dimensional index sets $Y$.

In the first part of the present work two basic numerical solution methods for solving $SIP$ with an arbitrary dimensional and arbitrarily shaped index set $Y \subset \mathbb{R}^m$ are presented. The second method is an extension of the algorithm discussed in [16]. An additional $X$-adaptation strategy, enhancing the numerical performance, for both algorithms is discussed in the second part.

The solution concept for all algorithms will be that of stationary points. For this reason no global assumptions are made on the structure of the objective function or the constraints like linearity or convexity, neither in the decision variable, nor in the index variable. Moreover, the techniques presented in [31] for handling the set $Y$ differ from our method in the adaptive way of the used subdivision strategies.

In the sequel we will make two assumptions. The first of the assumptions is made to keep the exposition as simple as possible, the second is standard in semi-infinite programming.

**Assumption 1.** The box $X \subset \mathbb{R}^n$ contains all feasible points of $SIP$ in its interior.

**Assumption 2.** The index set $Y \subset \mathbb{R}^m$ is non-empty, compact and given by

$$Y = \{ y \in \mathbb{R}^m | v_l(y) \leq 0, l \in \mathcal{L} \},$$

where $\mathcal{L}$ is a finite index set and the functions $v_l, l \in \mathcal{L}$, are continuous.

It is well known, cf. [47], that $SIP$ can be reformulated as the Stackelberg game

$$SG : \min_{x,y} f(x) \quad s.t. \quad g(x,y) \leq 0, \quad y \text{ is a solution of } Q(x),$$

where the so-called lower level problem $Q(x)$ is

$$Q(x) : \max_{y \in \mathbb{R}^m} g(x,y) \quad s.t. \quad y \in Y.$$

Note that a point $\pi$ is feasible for $SIP$ if and only if the global optimal value of $Q(\pi)$ is non-positive. Thus, even checking feasibility of a point for $SIP$ results in solving a global optimization problem. If a smooth optimal value function $\overline{y}(x)$ of $Q(x)$ is known, that is a smooth function $\overline{y}(x)$ that solves $Q(x)$ for each $x$, then $SG$ can be reduced to a standard nonlinear problem by replacing $y$ by $\overline{y}(x)$. However, in general such a global solution of $Q(x)$ is not available.
1 Introduction

For some special functions $g$ and sets $Y$ checking feasibility of a point $x$ for $SIP$ becomes easier than in the general case, and, moreover, $SG$ can be further reformulated so that we may obtain a problem which can be handled easier. Here we consider two special cases, namely, unimodal or concave functions $g$ on convex sets $Y$.

**Definition 1.1.1.**

(i) A set $Y \subset \mathbb{R}^m$ is called convex if for each $y_1, y_2 \in Y$ we have $(1 - \lambda) y_1 + \lambda y_2 \in Y$ for each $\lambda \in (0, 1)$.

(ii) Let $Y \subset \mathbb{R}^m$ be a convex set. A function $g : Y \rightarrow \mathbb{R}$ is called unimodal on $Y$ if it possesses a unique global maximizer.

(iii) Let $Y$ be a convex set. A function $g : \mathbb{R}^m \rightarrow \mathbb{R}$ is called concave on $Y$ if the relation $g((1 - \lambda) y_1 + \lambda y_2) \geq (1 - \lambda) g(y_1) + \lambda g(y_2)$ holds for each $y_1, y_2 \in Y$ and each $\lambda \in (0, 1)$.

It is well known that a function $g \in C^2(\mathbb{R}^m, \mathbb{R})$ is concave if and only if $D^2 g(y) \preceq 0$.

For a function $g$ that is continuously differentiable on a dotted open set we have the following sufficient condition to check if it is unimodal.

**Lemma 1.1.2.** Let $Y \subset \mathbb{R}^m$ be a convex set, $c \in Y$ and $g \in C^1(\mathbb{R}^m \setminus \{c\}, \mathbb{R})$. If the relation

$$Dg(y)(c - y) > 0$$

holds for each $y \in Y \setminus \{c\}$, then $g$ is unimodal on $Y$ and $c$ is the global maximizer of $g$ on $Y$.

**Proof.** We give the proof by enforcing a contradiction. Let there be some $c \in Y$ so that the relation $Dg(y)(c - y) > 0$ holds for each $y \in Y \setminus \{c\}$, and let $\overline{y} \in Y$, $\overline{y} \neq c$, with $g(\overline{y}) \geq g(c)$. As $Y$ is convex we have that $\hat{y}_1(\lambda) = (1 - \lambda) \overline{y} + \lambda c \in Y$ and $\hat{y}_2(\lambda) = \lambda \overline{y} + (1 - \lambda) c \in Y$ holds for each $\lambda \in (0, 1)$. With the mean value theorem we obtain that the relations

$$0 \geq g(\hat{y}(1)) - g(\hat{y}(0))$$

$$= D\lambda (g(\hat{y}_1(\lambda)))|_{\lambda = \xi_1}$$

$$= Dg(\hat{y}_1(\xi_1))(c - \overline{y})$$

and

$$0 \geq g(\hat{y}(1)) - g(\hat{y}(0))$$

$$= D\lambda (g(\hat{y}_2(\lambda)))|_{\lambda = \xi_2}$$

$$= Dg(\hat{y}_2(\xi_2))(\overline{y} - c)$$

hold for some $\xi_1, \xi_2 \in (0, 1)$. With the intermediate value theorem we obtain that there is some $\xi_0 \in [\min(\xi_1, \xi_2), \max(\xi_1, \xi_2)]$ so that

$$0 = Dg(\hat{y}_1(\xi_0))(c - \overline{y})$$

4
holds. As for each \( \xi \in (0, 1) \) we have \( (1 - \xi) (c - \bar{y}) = (c - \hat{y}_1 (\xi)) \), we also have
\[
0 = D g (\hat{y}_1 (\xi_0)) (c - \bar{y}) \\
= D g (\hat{y}_1 (\xi_0)) (c - \hat{y}_1 (\xi_0))
\]
for some \( \xi_0 \in [\min(\xi_1, \xi_2), \max(\xi_1, \xi_2)] \). That is a contradiction.

Let us assume for the moment that \( Q (x) \) is a unimodal problem, that is, \( g \) is unimodal in the second argument on \( Y \) for each \( x \), and that we have the solution \( \bar{y} (x) \) of \( Q (x) \). Then we are in the situation suggested above and \( SG \) reduces to a nonlinear optimization problem.

\[
NLP : \min_{x \in X} f(x) \quad s.t. \quad g(x, \bar{y}(x)) \leq 0.
\]
That type of problems can be tackled, for example, by the methods presented in [50, 51]. That \( Q (x) \) is an unimodal problem is, of course, a very restrictive assumption and the knowledge of a solution makes it even worse.

If we assume now, for the moment, that \( v_l \in C^1 (\mathbb{R}^m, \mathbb{R}) \) for each \( l \in L \), that \( Q (x) \) is a convex problem, that is, \( Y \) is convex and \( g \) is concave in \( y \), and that \( Y \) possesses a Slater point, then the Karush-Kuhn-Tucker conditions are necessary and sufficient for optimality and \( SG \) can be reformulated as the following mathematical program with complementary constraints ([16]).

\[
MPCC : \min_{x, y, \gamma} f(x) \quad s.t. \quad g(x, y) \leq 0 \quad \nabla_y g(x, y) + \sum_{l \in L} \gamma_l \nabla_y v_l (y) = 0 \quad 0 \leq -v(y) \perp \gamma \geq 0.
\]
Hence, under our temporary convexity assumption on \( Q (x) \), checking feasibility of a point \( \bar{y} \) for \( SIP \) becomes easier than for non-convex \( Q (x) \).

Next we take a look at a special semi-infinite problem. Choose \( B := [b, \tilde{b}] \subset \mathbb{R}^m \), \( b < \tilde{b} \in \mathbb{R}^m \), with \( Y \subset B \) and define the semi-infinite problem

\[
SIP_B : \min_{x \in X} f(x) \quad s.t. \quad g(x, y) \leq 0 \quad \text{for all } y \in B
\]
with the feasible set \( X \cap M_B \), where
\[
M_B = \{ x \in \mathbb{R}^n | g(x, y) \leq 0 \text{ for all } y \in B \}.
\]
Then, under the assumption that \( g \) is unimodal in \( y \) and a solution of the lower level problem is known, \( SIP_B \) can be equivalent reformulated as

\[
NLP_B : \min_{x \in X} f(x) \quad s.t. \quad g(x, \bar{y}(x)) \leq 0,
\]
1 Introduction

and, under the assumption that $g$ is concave in $y$, $SIP_B$ can be equivalently reformulated as

$$\text{MPCC}_B : \min_{x,y,\gamma} f(x) \quad \text{s.t.} \quad g(x,y) \leq 0$$

$$\nabla_y g(x,y) + \gamma - \gamma = 0$$

$$0 \leq (\bar{b} - y) \perp \gamma \geq 0$$

$$0 \leq (y - \bar{b}) \perp \gamma \geq 0.$$

The fact that $B$ is a superset of $Y$ obviously entails $M_B \subset M$. Using these observations and looking at $B$ as an outer approximation of $Y$, one obtains an upper bound on the optimal value of $SIP$ by solving $SIP_B$. The better the approximation $B$ of $Y$, the better should be the obtained upper bound. The first algorithm, introduced in the present work, uses this idea to construct a sequence of outer approximations of $Y$, combined with a unimodalization strategy for $g$ based on ideas from the optimal centered forms [4]. The second algorithm also constructs a sequence of outer approximations of $Y$, combined with the $\alpha BB$ method from [1, 2, 3, 14] as a concavification strategy for $g$. Both algorithms generate a sequence of points which tends to a stationary point of $SIP$.

In Chapter 2 a brief review of stationarity, the main ideas of two relaxation strategies of global optimization and the basic concept of a reduced outer approximation of a set are presented. Chapter 3 contains the adaptive reduction algorithm, and Chapter 4 the adaptive convexification algorithm. Both chapters contain first numerical examples. An additional adaptation strategy for the set $X$ for both algorithms is presented in Chapter 5. Chapter 6 contains a hybrid method and implementation details on all algorithms are introduced in Chapter 7. Numerical examples illustrating the performance of all methods are given in Chapter 8. Finally, in Chapter 9 we give final remarks and point out possible improvements.
2 Background

In Section 2.1 a short overview of first order optimality conditions for SIP and a natural constraint qualification are given. A brief review of overestimating techniques from global optimization, that is, the unimodalization strategy and the αBB method, are given in Section 2.2. In the last section of this chapter we introduce a special approximation of a set. It is called reduced outer approximation.

2.1 Stationarity conditions

Let $\partial M$ be the topological boundary of $M$. For $\bar{x} \in \partial M$ define the active index set

$$Y_0(\bar{x}) := \{y \in Y | g(\bar{x}, y) = 0\}.$$ 

It is easy to see that $Y_0(\bar{x})$ is non-empty, compact and coincides with the set of global maximizers of $Q(\bar{x})$. Let $\sigma^{n+1} = \{s \in \mathbb{R}^{n+2} | s \geq 0, \sum_{i=1}^{n+2} s_i = 1\}$ be the $(n+1)$-dimensional standard simplex.

**Theorem 2.1.1** (Theorem of John, [26]). Let $\bar{x} \in M$ be a local minimizer of SIP. Then there exist $y^k \in Y_0(\bar{x})$, $k = 1, \ldots, n+1$, and $(\kappa, \lambda) \in \sigma^{n+1}$ with

$$\kappa \nabla f(\bar{x}) + \sum_{k=1}^{n+1} \lambda_k \nabla_x g(\bar{x}, y^k) = 0$$

$$\lambda_k g(\bar{x}, y^k) = 0, \quad k = 1, \ldots, n+1.$$ 

Note that for $\bar{x} \in \text{int}(M)$ the set $Y_0(\bar{x})$ may be empty, and the conditions from Theorem 2.1.1 reduce to $\nabla f(\bar{x}) = 0$. If a point $\bar{x} \in M$ satisfies the Extended Mangasarian-Fromovitz Constraint Qualification (EMFCQ), that is, if there exists a $d \in \mathbb{R}^n$ with

$$\langle \nabla_x g(\bar{x}, y), d \rangle < 0$$

for all $y \in Y_0(\bar{x})$, then one can choose $\kappa > 0$ in Theorem 2.1.1, and $\bar{x}$ is called a Karush-Kuhn-Tucker point (KKT point) of SIP. Here $\langle .., \rangle$ denotes the standard inner product.
2 Background

2.2 Overestimating techniques

In this section we present two techniques to construct an overestimator for a given function on a box set. Let \( B := [\underline{b}, \overline{b}] \subset \mathbb{R}^m \) be a box. For a given function \( g : \mathbb{R}^m \to \mathbb{R} \) we call a function \( \tilde{g} : \mathbb{R}^m \to \mathbb{R} \) an overestimator for \( g \) on \( B \) if the relation \( \tilde{g}(y) \geq g(y) \) holds for each \( y \in B \). In the sequel let \( \mathbb{R}^m \) denote the set of nonempty compact boxes in \( \mathbb{R}^m \). For some \( B \in \mathbb{R}^m \) we use \( \underline{b} \) and \( \overline{b} \) to denote the vectors of lower and upper bounds, that is, \( B = [\underline{b}, \overline{b}] \). For some mapping \( L \) whose image is as subset of \( \mathbb{R}^m \) we use \( L \) and \( \overline{L} \) to denote the vectors of lower and upper bounds, that is, \( L = [L, \overline{L}] \).

It is possible to calculate with intervals, interval vectors and interval matrices. We refer to [35] and the references there in for details on the arithmetic.

2.2.1 The unimodalization method

With the unimodalization method from global optimization one can construct unimodal overestimators of a given function on a box set. Since this method is based on the choice of an optimal center and some concepts used in optimal centered forms, we will first give a brief review of optimal centered forms discussed in [4]. After that we present the unimodalization method which was first discussed in [17].

Let \( g \in C^1(B, \mathbb{R}) \) be some function on \( B \in \mathbb{R}^m \). A mapping \( L : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^m \) is called Lipschitz function for \( g \) on \( B \) if the inclusion

\[
g(y) - g(c) \in L(B, c)^T (y - c)
\]

holds for each \( y, c \in B \). The expression

\[
\tilde{G}(B, c) := g(c) + L(B, c)^T (B - c)
\]

is called a centered form of \( g \) on \( B \) with center \( c \). The inclusion property (2.1) immediately implies that the centered forms bound the function values of \( g \) on \( B \). In general we may not expect to find a Lipschitz function \( L \) so that the bounds are exact. Since \( g \) is continuously differentiable on the compact set \( B \), \( g \) is also Lipschitz continuous on \( B \) and the Lipschitz constant can be used as a Lipschitz function, independent from the center \( c \), to construct a centered form. Instead of the Lipschitz constant we use a slightly different Lipschitz function in the sequel. Let \( L_\mathbb{R} \mathbb{R}^m \to \mathbb{R}^m \) be defined by

\[
L(B) := \begin{pmatrix}
\min(0, L_1(B)) \\
\vdots \\
\min(0, L_m(B))
\end{pmatrix}
\quad \text{and} \quad
\overline{L}(B) := \begin{pmatrix}
\max(0, \overline{L}_1(B)) \\
\vdots \\
\max(0, \overline{L}_m(B))
\end{pmatrix}
\]

with

\[
L_j(B) < \min_{y \in B} \frac{\partial}{\partial y_j} g(y) \quad \text{and} \quad \overline{L}_j(B) > \max_{y \in B} \frac{\partial}{\partial y_j} g(y)
\]
for \( j = 1, \ldots, m \). It is not hard to see that \( L = [L(B), \overline{L}(B)] \) is a Lipschitz function for \( g \) on \( B \). Since only some bounds for the gradient of \( g \) on \( B \) are needed to construct \( L \), one may determine it by methods of interval arithmetic (cf., e.g., [21, 35]). Note that \( \max (\|L(B)\|_{\infty}, \|\overline{L}(B)\|_{\infty}) \) is at least as large as the original Lipschitz constant.

Since \( \hat{G} \) still depends on the choice of the center \( c \), a centered form is not uniquely defined. Thus, naturally, the question arises if there is a center so that the upper or lower bound for the function values become tightened. Such a center is called optimal center. Since we have a maximization problem, we only have to find a center so that the value for the upper bound for the values of \( g \) on \( B \) is the smallest among all possible centers.

**Theorem 2.2.1** ([4]). Let \( L(B) = [L(B), \overline{L}(B)] \) be a Lipschitz function for \( g \) on \( B \). For \( j = 1, \ldots, m \) let

\[
c_j^+ := \frac{L_j(B)b_j - \overline{L}_j(B)b_j}{L_j(B) - \overline{L}_j(B)}.
\]

Then it holds \( c^+ \in B \) and \( \max(\hat{G}(B, c^+)) \leq \max(\hat{G}(B, c)) \) for each \( c \in B \). \( c^+ \) is called optimal center and \( \hat{G}(B, c^+) \) is called optimal centered form for \( g \) on \( B \).

Another useful way to express \( \hat{G} \) is using slopes. For \( j = 1, \ldots, m \) let

\[
\hat{g}_j(y; c) := L_j(B)(y_j - c_j) \quad \text{and} \quad \hat{g}_j(y; c) := \overline{L}_j(B)(y_j - c_j)
\]

and set \( \hat{g}(y; c) = (\hat{g}_1(y; c), \ldots, \hat{g}_m(y; c))^T \), \( \hat{g}(y; c) = (\hat{g}_1(y; c), \ldots, \hat{g}_m(y; c))^T \). Then we have that

\[
\hat{G}(B, c) = g(c) + \left[ \min_{y \in B}(\hat{g}(y; c), \hat{g}(y; c)), \max_{y \in B}(\hat{g}(y; c), \hat{g}(y; c)) \right].
\]

The next example illustrates our latter discussions.

**Example 2.2.2.** Using the slopes, for \( g(y) = \sin(y) \) on \( B = [0, 2\pi] \) a centered form with center \( c = \frac{\pi}{2} \) and an optimal centered form is illustrated in Figure 2.1. We used \( L(B) = -1.1, \overline{L}(B) = 1.1 \) and obtain \( c^+ = \pi \) for the optimal center.

Not only in Example 2.2.2, but also in general, it is not hard to see that even the upper bound of \( \hat{G}(B, c^+) \) is not unimodal, and that for each vertex \( v \) of \( B \) we have

\[
\max(\hat{G}(B, c^+)) = g(c^+) + \max(\hat{g}(v, c^+), \hat{g}(v, c^+)).
\]

Moreover, the graph of \( g \) on \( B \) is contained within the area enclosed by the slopes constructed with \( \hat{g}(y; c) \) and \( \hat{g}(y; c) \). That observation leads to the unimodalization method.

The main idea of unimodalization is to add some piecewise linear term, similarly to \( \hat{g}(y; c) \) or \( \hat{g}(y; c) \), but not to a special value of the function, but to the function \( g \) in each point. More precisely this is performed as follows:
2 Background

Let \( J_\ell(y,c) \) and \( J_u(y,c) \) be defined as:

\[
J_\ell(y,c) = \{ j \in \{1, \ldots, m\} | y_j < c_j \} \quad \text{and} \quad J_u(y,c) = \{ j \in \{1, \ldots, m\} | y_j \geq c_j \}.
\]

Let \( \phi(y,c,B,L(B),\overline{L}(B)) = \sum_{j \in J_\ell(y,c)} L(B)_j (b_j - y_j) + \sum_{j \in J_u(y,c)} \overline{L}(B)_j (\overline{b}_j - y_j) \)

and define a unimodal form

\[
\hat{g}_B(y,c) = \hat{g}(y,c,B,L(B),\overline{L}(B)) = g(y) + \phi(y,c,B,L(B),\overline{L}(B)).
\]

Before we give two examples for \( \hat{g}(y,c) \) we point out some properties. The straightforward proofs are omitted.

**Lemma 2.2.3.**

(i) For all vertices \( v \) of \( B \) and \( c \in B \) we have \( \hat{g}_B(v,c) = g(v) \).

(ii) For all \( y,c \in B \) the relation \( \hat{g}_B(y,c) \geq g(y) \) holds.

(iii) If \( L(B),\overline{L}(B) \neq 0 \) holds, then we have the relation \( \hat{g}_B(y,c) > g(y) \) for each \( y,c \in \text{int}(B) \).

(iv) Let \( L(B),\overline{L}(B) \neq 0 \). For each vertex \( y \) of \( B \) we have \( \hat{g}_B(y,c) = g(y) \).
2.2 Overestimating techniques

Example 2.2.4. As a first example we look again at \( g(y) = \sin(y) \) on \( B = [0, 2\pi] \) with \( c = \frac{\pi}{2} \) and \( L(B) = -1.1, U(B) = 1.1 \). With \( J^L(y, c) \neq \emptyset \) if \( y < \frac{\pi}{2} \) and \( J^U(y, c) \neq \emptyset \) if \( y \geq \frac{\pi}{2} \), we obtain

\[
\hat{g}_B(y; c) = \sin(y) + \begin{cases} 
1.1y, & y < \frac{\pi}{2} \\
1.1(2\pi - y), & y \geq \frac{\pi}{2}
\end{cases}
\]

for \( y \in B \). This function is illustrated in Figure 2.2.

From the Example 2.2.4 one can see that for an arbitrarily chosen \( c \in B \) the function \( \hat{g}_B(y; c) \) does not have to be continuous. The next example shows that the maximum of \( \hat{g}_B(y; c) \) over \( B \) may not be attained.

Example 2.2.5. Let \( g(y) = \sin(y_1) + (y_2 - 1)^2 \), \( B = [0, 2\pi] \times [0, 3] \) and \( c = (\frac{\pi}{2}, \frac{5}{2})^T \). With \( L(B) = (-1.1, -2.1)^T, U(B) = (1.1, 4.1)^T \) we obtain

\[
\hat{g}_B(y; c) = \begin{cases} 
\sin(y_1) + (y_2 - 1)^2 + 1.1y_1 + 2.1y_2 & , y_1 < \frac{\pi}{2}, y_2 < \frac{5}{2} \\
\sin(y_1) + (y_2 - 1)^2 + 1.1(2\pi - y_1) + 2.1y_2 & , y_1 < \frac{\pi}{2}, y_2 \geq \frac{5}{2} \\
\sin(y_1) + 1.1(2\pi - y_1) + 4.1(3 - y_2) & , y_1 \geq \frac{\pi}{2}, y_2 < \frac{5}{2} \\
\sin(y_1) + (y_2 - 1)^2 + 1.1(2\pi - y_1) + 4.1(3 - y_2) & , y_1 \geq \frac{\pi}{2}, y_2 \geq \frac{5}{2}
\end{cases}
\]

for \( y \in B \). This function is illustrated in Figure 2.3. It is not hard to see that \( \hat{g}_B(y; c) \) is also not continuous and has a supremum with the value \( \frac{17+3.3\pi}{2} \).

Even though the latter examples are not very promising, we obtain the following result for a special choice for \( c \).
2 Background

Figure 2.3: $\hat{g}_B(y; c)$ for $g(y) = \sin(y_1) + (y_2 - 1)^2$ on $B = [0, 2\pi] \times [0, 3]$ with $c = \left(\frac{\pi}{2}, \frac{5}{2}\right)^T$ and $L(B) = (-1.1, -2.1)^T$, $\mathcal{L}(B) = (1.1, 4.1)^T$. The red dot marks the supremum of $\hat{g}_B(y; c)$.

Theorem 2.2.6. Let $\hat{g}_B(y) = \hat{g}_B(y; c^+)$ with $c^+$ from Theorem 2.2.1. We have that $\hat{g}(y)$ is continuous and unimodal on $B$, and that $\hat{g}(c^+) = \max_{y \in B} \hat{g}(y)$ holds.

Proof. To show that $\hat{g}_B(y)$ is continuous it is sufficient to show that $\hat{g}_B(y)$ is continuous in $c^+$. Let $L > \max\left(-L, L\right) \in \mathbb{R}$ and let $\epsilon, \delta_0 > 0$ so that $|g(y) - g(c^+)| \leq \frac{\epsilon}{2}$ if $\|y - c^+\| \leq \delta_0$. Let $\delta = \min\left(\delta_0, \frac{\epsilon}{2L}\right)$. From $J^l(y, c^+) \cup J^u(y, c^+) = \{1, \ldots, m\}$ for each $y \in B$ and the definition of $c^+$ we have that

$$|\hat{g}_B(y) - \hat{g}_B(c^+) | \leq |g(y) - g(c^+)| + \sum_{j \in J^l(y, c^+)} L_j \left(y_j - c^+_j\right) + \sum_{j \in J^u(y, c^+)} \mathcal{L}_j \left(y_j - c^+_j\right) | \leq |g(y) - g(c^+)| + 2L\|y - c^+\| \leq \epsilon.$$

Next we show that $\hat{g}_B(y)$ is unimodal on $B$ and that $\hat{g}_B(c^+) = \max_{y \in B} \hat{g}_B(y)$ holds. Let $j \in \{1, \ldots, m\}$. If on the one hand for some $y^* \in B \setminus \{c^+\}$ the relation $y^*_j < c_j^+$ holds, we have from the definition of $\mathcal{L}_j(B)$ that

$$\frac{\partial \hat{g}_B}{\partial y_j}(y^*) = \frac{\partial g}{\partial y_j}(y^*) - L_j(B) > 0.$$
holds. If on the other hand the relation \( y_j^* > c_j^+ \) holds, we have from the definition of \( \mathcal{L}(B) \) that

\[
\frac{\partial \hat{g}_B}{\partial y_j}(y^*) = \frac{\partial g}{\partial y_j}(y^*) - \mathcal{L}(B)_j < 0
\]

holds. In summary we obtain the relation \( \langle \nabla \hat{g}_B(y), c^+ - y \rangle > 0 \) for each \( y \in B \setminus \{c^+\} \).

With Lemma 1.1.2 we have that \( \hat{g}_B \) is unimodal on \( B \), and \( c^+ \) is the global maximizer.

Before we continue our discussion we pick up Example 2.2.4 and 2.2.5 to illustrate the latter result.

**Example 2.2.7.** For the function \( g(y) = \sin(y) \) on \( B = [0, 2\pi] \) with \( \mathcal{L}(B) = [-1.1, 1.1] \), we obtain \( c^+ = \pi \). Thus, the overestimator is

\[
\hat{g}_B(y) = \sin(y) + \begin{cases} 
1.1y & , y < \pi \\
1.1(2\pi - y) & , y \geq \pi
\end{cases}.
\]

For \( g(y) = \sin(y_1) + (y_2 - 1)^2 \) on \( B = [0, 2\pi] \times [0, 3] \) with \( \mathcal{L}(B) = (-1.1, -2.1) \), \( \mathcal{L}(B) = (1.1, 4.1) \), we obtain \( c^+ = (\pi, \frac{123}{62}) \). Thus, the overestimator is

\[
\hat{g}_B(y) = \begin{cases} 
sin(y_1) + (y_2 - 1)^2 + 1.1y_1 + 2.1y_2 & , y_1 < \pi, y_2 < \frac{123}{62} \\
sin(y_1) + (y_2 - 1)^2 + 1.1y_1 + 4.1(3 - y_2) & , y_1 < \pi, y_2 \geq \frac{123}{62} \\
sin(y_1) + (y_2 - 1)^2 + 1.1(2\pi - y_1) + 2.1y_2 & , y_1 \geq \pi, y_2 < \frac{123}{62} \\
sin(y_1) + (y_2 - 1)^2 + 1.1(2\pi - y_1) + 4.1(3 - y_2) & , y_1 \geq \pi, y_2 \geq \frac{123}{62}
\end{cases}.
\]

Both overestimators are illustrated in Figure 2.4. It can be seen that these overestimators are continuous, unimodal and that they have a maximum in \( c^+ \).

![Figure 2.4](image-url)
A measure for the quality of the overestimator is the separation distance $d_{UF}$ between $\hat{g}_B$ and $g$. It is defined by

$$d_{UF}(y; c^+, B, L(B), L(B)) := \hat{g}_B(y) - g(y) = \phi(y; c^+, B, L(B), L(B)).$$

It is not hard to see that $\phi(y; c^+, B, L(B), L(B))$ attains its maximum on $B$ for $y = c^+$. Thus, with the definition of $c^+$ we have that

$$\max_{y \in B} d_{UF}(y; c^+, B, L(B), L(B)) = \langle L, b - c^+ \rangle = \langle L, \bar{b} - c^+ \rangle. \quad (2.4)$$

On the one hand, it immediately follows that for a fixed box $B$ one may improve the overestimator by determining tighter bounds $L(B), L(B)$ for the gradient of $g$ on $B$. On the other hand, the maximum separation distance linearly tends to zero if the box diameter $\|\bar{b} - b\|_2$ goes to zero.

### 2.2.2 The $\alpha BB$ method

Let $g \in C^2(B, \mathbb{R})$ be some general non-concave function on $B$. With the $\alpha BB$ method of global optimization one can construct concave relaxations of $g$ on $B$ by adding a quadratic term. More precisely this is performed as follows:

Let $\psi(y; \alpha, B) = \frac{\alpha^2}{2} (y - \bar{b}, \bar{b} - y)$ and define

$$\tilde{g}_B(y) = \tilde{g}(y; \alpha, B) := g(y) + \psi(y; \alpha, B).$$

For $\alpha \geq 0$ the function $\tilde{g}_B$ is an overestimator of $g$ on $B$ and coincides with $g$ at the vertices of $B$. Moreover the Hessian of $\tilde{g}_B$ is $D^2\tilde{g}_B(y) = D^2g(y) - \alpha I$, where $I$ denotes the unit matrix. Let $\lambda_{\text{max}}(y)$ denote the maximal eigenvalue of $D^2g(y)$. Then $\tilde{g}_B$ is concave on $B$ if we choose

$$\alpha \geq \max_{y \in B} \lambda_{\text{max}}(y).$$

Since only some upper bound for $\max_{y \in B} \lambda_{\text{max}}(y)$ is needed one may determine it by methods of interval arithmetic (cf., e.g., [14, 21, 35]). Note that determining the exact value of $\max_{y \in B} \lambda_{\text{max}}(y)$ would lead to a global optimization problem. Altogether it follows that $\tilde{g}_B$ is a concave overestimator of $g$ on $B$ if

$$\alpha \geq \max \{0, \max_{y \in B} \lambda_{\text{max}}(y)\}.$$ 

That is illustrated by the next example.

**Example 2.2.8.** Figure 2.5 shows two overestimators of the form $\tilde{g}_B(y; \alpha, B)$ for $g(y) = \sin(y)$ on $B = [0, 2\pi]$. For one of these overestimators we used $\alpha = \frac{1}{2}$ which leads to a non-concave function, since $\max_{y \in B} \lambda_{\text{max}}(y) = 1$. For the other, concave, overestimator we used $\alpha = 1$. 

14
2.2 Overestimating techniques

The separation distance $d_{\alpha BB}$ between $\tilde{g}_B$ and $g$ is

$$d_{\alpha BB} (y; \alpha, B) := \tilde{g} (y; \alpha, B) - g (y) = \psi (y; \alpha, B).$$

The separation distance is maximal at the barycenter $\frac{1}{2} (\bar{b} + \bar{b})$ of $B$ with the value

$$\max_{y \in B} d_{\alpha BB} (y; \alpha, B) = \frac{\alpha}{8} \| \bar{b} - \bar{b} \|_2^2. \quad (2.5)$$

Similarly to the observations in the last subsection we have that for a fixed box $B$ one may improve the overestimator by determining tighter upper bounds $\alpha$ for $\max_{y \in B} \lambda_{\max} (y)$. But for $\tilde{g}$ the maximum separation distance quadratically tends to zero if the box diameter $\| \bar{b} - \bar{b} \|_2$ goes to zero.

2.2.3 A reduced outer approximation

Now let $Y \subset \mathbb{R}^m$ be some arbitrary compact set with $Y \subset B$. Then $\hat{g}_B$ is an unimodal and $\tilde{g}_B$ is a concave overestimator of $g$ on $B$, and, thus, also on $Y$. In this case, however, the separation distance of $\hat{g}_B$, respectively $\tilde{g}_B$, and $g$ is not the only error which must be taken into account to measure the quality of the overestimator on $Y$. Another error is due to the quality of the approximation of $Y$ by $B$. This motivates the following definitions.

**Definition 2.2.9.** Let $N \in \mathbb{N}$, $B^k = \left[ \bar{b}^k, \bar{b}^k \right] \subset \mathbb{R}^m$, $\bar{b}^k < \bar{b}^k$, $k = 1, \ldots, N$, and $\mu^n$ be the Lebesgue measure on $\mathbb{R}^m$. The $N$-tuple of sets $\mathcal{B}^N = (B^k, k = 1, \ldots, N)$ is called a tessellation of $B$ if $\bigcup_{k=1}^N B^k = B$ and $\mu (B^k \cap B^j) = 0$ hold for all $k, j = 1, \ldots, N$, $k \neq j$. 

Figure 2.5: $\hat{g} (y; \alpha, B)$ for $g (y) = \sin (y)$ on $B = [0, 2\pi]$ with $\alpha = \frac{1}{2}$ and with $\alpha = 1$. 

The separation distance $d_{\alpha BB}$ between $\tilde{g}_B$ and $g$ is

$$d_{\alpha BB} (y; \alpha, B) := \tilde{g} (y; \alpha, B) - g (y) = \psi (y; \alpha, B).$$

The separation distance is maximal at the barycenter $\frac{1}{2} (\bar{b} + \bar{b})$ of $B$ with the value

$$\max_{y \in B} d_{\alpha BB} (y; \alpha, B) = \frac{\alpha}{8} \| \bar{b} - \bar{b} \|_2^2. \quad (2.5)$$

Similarly to the observations in the last subsection we have that for a fixed box $B$ one may improve the overestimator by determining tighter upper bounds $\alpha$ for $\max_{y \in B} \lambda_{\max} (y)$. But for $\tilde{g}$ the maximum separation distance quadratically tends to zero if the box diameter $\| \bar{b} - \bar{b} \|_2$ goes to zero.

**Definition 2.2.9.** Let $N \in \mathbb{N}$, $B^k = \left[ \bar{b}^k, \bar{b}^k \right] \subset \mathbb{R}^m$, $\bar{b}^k < \bar{b}^k$, $k = 1, \ldots, N$, and $\mu^n$ be the Lebesgue measure on $\mathbb{R}^m$. The $N$-tuple of sets $\mathcal{B}^N = (B^k, k = 1, \ldots, N)$ is called a tessellation of $B$ if $\bigcup_{k=1}^N B^k = B$ and $\mu (B^k \cap B^j) = 0$ hold for all $k, j = 1, \ldots, N$, $k \neq j$. 

15
For a tessellation \( B^N \) of \( B \) and \( k \in \{1, \ldots, N\} \) let \( \hat{g}^k (y) = \hat{g} (y; c^k, B^k, \underline{L}^k, \overline{L}^k) \) be the unimodal relaxation and let \( \hat{g}^k (y) = \hat{g} (y; \alpha_k, B^k) \) be the \( \alpha \text{BB} \) relaxation of \( g(y) \) on \( B^k \) with \( \alpha_k \geq \max \{ 0, \max_{y \in B^k} \lambda_{\max} (y) \} \),

\[
\begin{align*}
\underline{L}^k &= \left( \min(0, \underline{L}_1 (B^k)) \right) \quad \text{and} \quad \overline{L}^k = \left( \max(0, \overline{L}_1 (B^k)) \right)
\end{align*}
\]

and

\[
\underline{L}_j (B^k) < \min_{y \in B^k} \left( \frac{\partial}{\partial y_j} g(y) \right) \quad \text{and} \quad \overline{L}_j (B^k) > \max_{y \in B^k} \left( \frac{\partial}{\partial y_j} g(y) \right)
\]

for \( j = 1, \ldots, m \). It is easy to see that for all \( k \) the values of \( \underline{L}^k \), \( \overline{L}^k \) and \( \alpha_k \) can be chosen so that \( \underline{L}(B) \leq \underline{L}^k \), \( \overline{L}(B) \geq \overline{L}^k \) and \( \alpha \geq \alpha_k \) holds. Moreover we have that

\[
\max_{y \in B} d_{\text{UF}} (y; c^+, B, \underline{L}(B), \overline{L}(B)) \geq \max_{k \in \{1, \ldots, N\}} \max_{y \in B^k} \max_{k \in \{1, \ldots, N\}} d_{\text{UF}} (y; c^k, B^k, \underline{L}^k, \overline{L}^k)
\]

and that

\[
\max_{y \in B} d_{\alpha \text{BB}} (y; \alpha, B) \geq \max_{k \in \{1, \ldots, N\}} \max_{y \in B^k} d_{\alpha \text{BB}} (y; \alpha_k, B^k)
\]

holds.

**Definition 2.2.10.** Let \( Y \subset B \), let \( B^N = (B_k, k = 1, \ldots, N) \) be a tessellation of \( B \) and let \( \mathcal{N}_Y \subset \{1, \ldots, N\} \). \( B^N = (B_k, k \in \mathcal{N}_Y) \) is called a reduced outer approximation of \( Y \) if \( Y \subset \bigcup_{k \in \mathcal{N}_Y} B^k \) and \( Y \cap B^k \neq \emptyset \) holds for all \( k \in \mathcal{N}_Y \).

The approximation error of \( Y \) by a reduced outer approximation should become smaller, the smaller all sets \( B^k \) and the larger \( N \) become. A simple but useful observation is that not all sets \( B^k \) have to become small to reach a good approximation of \( Y \), but only those sets \( B^k \) whose intersection with the topological boundary of \( Y \) is non-empty. Moreover only the reduced outer approximation of \( Y \) must be taken into account to construct piecewise unimodal or concave overestimators of \( g \) on \( Y \). These ideas will be used to construct unimodal relaxations of the lower level problems in the adaptive reduction algorithm, and convex relaxations of the lower level problems in the adaptive convexification algorithm. Later it will even turn out, in both algorithms, that the approximation of the index set \( Y \) of \( \text{SIP} \) must only be good locally around certain points of interest.
3 The Adaptive-Reduction Algorithm

In this chapter we present the Adaptive-Reduction Algorithm. The chapter is organized as follows. In Section 3.1 we discuss the relaxation and reformulation techniques used within the algorithm which is presented in Section 3.2. After that, in Section 3.3, we give the convergence results for the algorithm. In Section 3.4, we present a first numerical example.

3.1 Relaxation and reformulation

Let \( Y \subset B \subset \mathbb{R}^m \) and let \( B^N = (B^k, k = 1, \ldots, N) \) be a reduced outer approximation of \( Y \). According to the formulas 2.2 and 2.3, let \( \mathcal{L}_i, \mathcal{T}_i : \mathbb{R}^m \times \mathbb{R}^m \mapsto \mathbb{R}^m \) be defined by

\[
\mathcal{L}_i(X, B) := \begin{pmatrix}
\min(0, L^i_1(X, B)) \\
\vdots \\
\min(0, L^i_m(X, B))
\end{pmatrix}
\]

and

\[
\mathcal{T}_i(X, B) := \begin{pmatrix}
\max(0, T^i_1(X, B)) \\
\vdots \\
\max(0, T^i_m(X, B))
\end{pmatrix}
\]

with

\[
L^i_j(X, B) < \min_{(x,y)\in X \times B} \left( \frac{\partial}{\partial y_i} g(y) \right) \quad \text{and} \quad T^i_j(X, B) > \max_{(x,y)\in X \times B} \left( \frac{\partial}{\partial y_i} g(y) \right),
\]

and let \( c \in \mathbb{R}^m \) with \( c_j := \frac{L^i_j b_j - T^i_j b_j}{L^i_j - T^i_j} \) for each \( j = 1, \ldots, m \). Define an unimodal overestimator (w.r.t. \( y \)) of \( g \) on \( X \times B \) by

\[
\hat{g} : X \times B \mapsto \mathbb{R}, \quad (x,y) \mapsto g(x,y) + \phi \left( y; c, B, \mathcal{L}, \mathcal{T} \right)
\]

as well as an unimodal overestimator of the restriction of \( g \) to \( X \times B^k \) by

\[
\hat{g}^k : X \times B^k \mapsto \mathbb{R}, \quad (x,y) \mapsto g(x,y) + \phi \left( y; c^k, B^k, \mathcal{L}^k, \mathcal{T}^k \right)
\]

with \( \mathcal{L}^k = \mathcal{L}(X, B^k), \mathcal{T}^k = \mathcal{T}(X, B^k) \) and \( c^k_j := \frac{L^k_j b_j - T^k_j b_j}{L^k_j - T^k_j} \) for each \( j = 1, \ldots, m \). Since

\[
\min_{(x,y)\in X \times B} \left( \frac{\partial}{\partial y_i} g(y) \right) \leq \min_{(x,y)\in X \times B^k} \left( \frac{\partial}{\partial y_i} g(y) \right)
\]

and

\[
\max_{(x,y)\in X \times B} \left( \frac{\partial}{\partial y_i} g(y) \right) \geq \max_{(x,y)\in X \times B^k} \left( \frac{\partial}{\partial y_i} g(y) \right),
\]
3 The Adaptive-Reduction Algorithm

Figure 3.1: The overestimator \( \hat{g} \) for \( g(x, y) = \sin(xy) \) on \( X \times B \) with \( X = B = [0, 2\pi], L = -\frac{63}{10}, L = \frac{63}{10} \) and \( c = \pi \).

both, \( L^k \) and \( \bar{L}^k \) can always be chosen so that \( L \leq L^k \) and \( \bar{L} \geq \bar{L}^k \) hold for each \( k = 1, \ldots, N \). Before we continue we give an example for \( \hat{g} \).

Example 3.1.1. Let \( X = B = [0, 2\pi] \) and \( g(x, y) = \sin(xy) \). It is not hard to see that the choices \( L = -\frac{63}{10} \) and \( L = \frac{63}{10} \) satisfy the requirements. Using \( L \) and \( \bar{L} \) we obtain \( c = \pi \). Thus, we have

\[
\hat{g}(x, y) = \sin(xy) + \begin{cases} 
\frac{63}{10} y, & y < \pi \\
\frac{63}{10} (2\pi - y), & y \geq \pi
\end{cases}
\]

\( \hat{g} \) is illustrated in Figure 3.1. One can see that \( \hat{g} \) is an overestimator for \( g \) on \( X \times B \) and unimodal in the second argument \( y \) for each \( x \) but neither unimodal in general nor in the first argument.

The next results are consequences from the discussion in Chapter 2, Subsection 2.2.1.

Lemma 3.1.2.

(i) For all \( k = 1, \ldots, N \) and \( (x, y) \in X \times B^k \) the relation \( g(x, y) \leq \hat{g}^k(x, y) \) holds.

(ii) For \( x \in X \) and \( k = 1, \ldots, N \) the relaxation \( \hat{g}^k(x, y) \) is unimodal in the second argument on \( B^k \).

(iii) For all \( k = 1, \ldots, N \) the maximum separation distance between \( \hat{g}^k \) and \( g \) on \( X \times B^k \) is \( \langle \bar{L}^k, \bar{b}^k - c^k \rangle = \langle L^k, \bar{b}^k - c^k \rangle \).

(iv) For each \( x \in X \) the point \( c^k \) is the solution of \( \max_{y \in B^k} \hat{g}^k(x, y) \) for each \( k = 1, \ldots, N \).

(v) If \( \pi \in X \) and \( \hat{g}^k(\pi, c^k) \leq 0 \) holds for all \( k = 1, \ldots, N \), then we have \( \pi \in M \).
3.2 Algorithms

Define the set

\[ M_{UF} (B^N) := \{ x \in \mathbb{R}^n | \hat{g}^k (x, y) \leq 0 \text{ for all } y \in B^k, \ k = 1, \ldots, N \} \]

and the semi-infinite problem

\[ SIP_{UF} (B^N) : \min_{x \in X} f(x) \quad s.t. \quad x \in M_{UF} (B^N). \]

For \( M_{UF} \) and \( SIP_{UF} \) we obtain the following results. The straightforward proofs are omitted.

**Lemma 3.1.3.**

(i) We have \( M_{UF} (B^N) \subset M \).

(ii) Let \( SIP_{UF} (B^N) \) be consistent. Then every global, local solution or stationary point of \( SIP_{UF} (B^N) \) is a feasible point of \( SIP \).

From Lemma 3.1.2 (iv) we have that for each \( k = 1, \ldots, N \) it holds \( \hat{g}^k (x, y) \leq 0 \) for all \( y \in B^k \) if and only if \( \hat{g}^k (x, c^k) \leq 0 \) is true. Thus we obtain

\[ M_{UF} (B^N) = \{ x \in \mathbb{R}^n | \hat{g}^k (x, c^k) \leq 0 \text{ for all } k = 1, \ldots, N \}. \]

Define

\[ G_{UF} (x) := \left( \hat{g}^k (x, c^k) \right)_{k=1,\ldots,N}. \]

Now \( SIP_{UF} (B^N) \) can be equivalently reformulated as the nonlinear problem

\[ P_{UF}(B^N) : \min_{x} f(x) \quad s.t. \quad G_{UF}(x) \leq 0. \]

This problem may be tackled by different solvers like those in [50, 51]. For more details we refer to Chapter 7, Section 7.1.

3.2 Algorithms

Following the discussions in [16], the algorithm presented in this section computes a stationary point of \( SIP_{UF} (B^N) \) with active indices and terminates if it is also a stationary point of \( SIP \) within a given tolerance on stationarity and additional tolerances on the feasibility of the indices. If the computed point is not approximately stationary or the active indices are not approximately feasible, the reduced outer approximation \( B^N \) of \( Y \) is refined and a respective refined \( SIP_{UF} \) is solved. Before the algorithms are presented some notation must be introduced.

For a point \( \bar{x} \in M_{UF} (B^N) \) and points \( c^k \in B^k, \ k = 1, \ldots, N \), define \( K^U_{0} (\bar{x}) := \{ k \in \{ 1, \ldots, N \} | \hat{g}^k (\bar{x}, c^k) = 0 \} \). As in Theorem 2.1.1, a point \( \bar{x} \in M_{UF} (B^N) \) is stationary
3 The Adaptive-Reduction Algorithm

for $SIP_{UF} (\mathcal{B}^N)$ in the sense of John, if there is a subset $J \subset K_0^{UF}, |J| = n + 1$, and $(\kappa, \lambda) \in \sigma^{n+1}$, so that

$$\nabla f(\pi) + \sum_{k \in J} \lambda_k \nabla x g(\pi, c^k) = 0$$

$$\lambda_k \check{g}^k(\pi, c^k) = 0, \quad k \in J.$$

Notice here that $\nabla x \check{g}^k(\pi, c^k) = \nabla x g(\pi, c^k)$. In view of the semi-infinite problem and to point out some similarities of all presented algorithms, we call a $c^k \in B^k$ with $k \in K_0^{UF}(\pi)$ an active index of $SIP_{UF}(\mathcal{B}^N)$. For the algorithms some concepts like stationarity and feasibility must be relaxed.

**Definition 3.2.1.** Let $\epsilon_{act}, \epsilon_{stat}, \epsilon_Y > 0$.

(i) $y \in B^k$ with $k \in \{1, \ldots, N\}$ is called $\epsilon_{act}$-active for $\check{g}^k$ at $\pi$, if $\check{g}^k(\pi, y) \in [-\epsilon_{act}, 0]$.

(ii) $\pi$ is called $\epsilon_{stat}$-stationary for SIP with $\epsilon_{act}$-active indices $y^k, k = 1, \ldots, n + 1$, if $\pi \in M$ and $y^k \in Y, k = 1, \ldots, n + 1$, and there exist some $(\kappa, \lambda) \in \sigma^{n+1}$, so that

$$\|\kappa \nabla f(x) + \sum_{k=1}^{n+1} \lambda_k \nabla x g(x, y^k)\| \leq \epsilon_{stat}$$

$$(3.3)$$

$$\lambda_k g(x, y^k) \in [-\lambda_k \epsilon_{act}, 0], \quad k = 1, \ldots, n + 1.$$  

$$(3.4)$$

(iii) $c^k \in B^k$ is called $\epsilon_Y$-feasible with respect to the reduced outer approximation $\mathcal{B}^N$ of $Y$ if $c^k \in Y$ or if $c^k \notin Y$ and $B^k \cap Y \neq \emptyset$ and $\|\check{y}^k - \check{b}^k\|_{\infty} < \epsilon_Y$.

(iv) $\pi$ is called $\epsilon_{stat}$-stationary for SIP with $\epsilon_{act}$-active and $\epsilon_Y$-feasible indices $y^k, k = 1, \ldots, n + 1$, with respect to $\mathcal{B}^N$, if $\pi \in M$ and there exist $\epsilon_Y$-feasible indices $y^k$ with respect to the reduced outer approximation $\mathcal{B}^N$ of $Y$ and $(\kappa, \lambda) \in \sigma^{n+1}$ so that

$$\|\kappa \nabla f(x) + \sum_{k=1}^{n+1} \lambda_k \nabla x g(x, y^k)\| \leq \epsilon_{stat}$$

$$(3.5)$$

$$\lambda_k g(x, y^k) \in [-\lambda_k \epsilon_{act}, 0], \quad k = 1, \ldots, n + 1.$$  

$$(3.6)$$

holds.

The basic idea for the adaptive refinement of a given reduced outer approximation of $Y$ is to split the boxes which contain the computed active indices. To reduce the approximation error uniformly on a box, it can be split through its barycenter. This, however, would make no sense in situations where a computed active index $c^k$ in $Y$ is a vertex of the given box, since then it is also an active index of the original problem and no further refinement is necessary. This leads to the idea to split the adaptively chosen boxes also adaptively, namely through the computed active indices, whenever possible.

The following definition deals with the choice of the appropriate coordinate along which a box is split, and with situations in which splitting is not necessary or would lead to numerically degenerate boxes.
Definition 3.2.2. For a reduced outer approximation $B^N$ let $\eta \in B^k$ with some $k \in \{1, \ldots, N\}$, and let $\epsilon_{\text{split}} > 0$ be given. Define the index set of coordinate directions along which the distance of $\eta$ from the boundary $\partial B^k$ is sufficiently large by

$$P^k := P^k(\eta) = \{l \in \{1, \ldots, m\} \mid \min\{\eta_l - b^k_l, b^k_l - \eta_l\} > \epsilon_{\text{split}}(b^k_l - b^k_l)\}.$$ 

In the case $P^k \neq \emptyset$ choose the coordinate direction $l \in P^k$ with

$$\left(b^k_l - b^k_l\right) = \max_{j \in P^k} \left(b^k_j - b^k_j\right),$$

that is, the direction corresponding to the longest edges of $B^k$ among the directions in $P^k$. Choose the hyperplane normal to the coordinate direction $l$ through $\eta$ to split $B^k$ into two boxes $B^{k,(1)}$ and $B^{k,(2)}$. For $P^k = \emptyset$ the box $B^k$ is not split. In this way, the following splitting operator is defined:

$$S(B^k, \eta) := \begin{cases} (B^{k,(1)}, B^{k,(2)}), & \text{if } P^k(\eta) \neq \emptyset \\ B^k, & \text{if } P^k(\eta) = \emptyset. \end{cases}$$

Note that, in Definition 3.2.2, for $P^k(\eta) = \emptyset$ the distance of $\eta$ from some vertex of $B^k$ is so small that the box is not split. Moreover, choosing a coordinate direction $l \notin P^k$ as the normal of a splitting hyperplane would generate one very ‘thin’ and, thus, numerically degenerate box. Also note that the choice $\epsilon_{\text{split}} \geq \frac{1}{2}$ would imply $P^k = \emptyset$ for any $\eta \in B^k$ so that in the following we will always assume $\epsilon_{\text{split}} < \frac{1}{2}$. The choice of the barycenter $S^k$ of a box $B^k$ as a splitting point, that is $\eta = S^k$, always entails $P^k = \{1, \ldots, m\}$.

A splitting algorithm merely based on Definition 3.2.2 may lead to the following problem. In our convergence proof it will be necessary that the maximum edge lengths in the sequence of boxes generated in the refinement steps tend to zero. This may not be the case, as the index of the coordinate direction corresponding the longest edge of a box $B^k$ may never be contained in $P^k$. This is illustrated by the following example.

Example 3.2.3. Let $B^0 := [0, 1]^2$ and $(\eta^\nu)_\nu$ be a sequence of splitting points with

$$\eta^\nu = (\eta^\nu_1, \eta^\nu_2) := \left(\epsilon_{\text{split}}, \frac{1}{2^\nu+1}\right).$$

Due to $P^\nu = \{2\}$ one box in each tessellation of $B^0$ generated by splitting through $\eta^\nu$ is given by

$$B^\nu = [0, 1] \times \left[0, \frac{1}{2^\nu}\right].$$

That is illustrated in Figure 3.2. Looking at the limit leads to

$$\lim_{\nu \to \infty} B^\nu = [0, 1] \times \{0\}.$$

This means that the boxes degenerate in the sense that their surface area is vanishing but the length of the longest edge does not tend to zero. The generalization of this example to an arbitrary dimension is straightforward.
3 The Adaptive-Reduction Algorithm

Figure 3.2: Sequence of splitting points \((\eta^\nu)\) with \(\eta^\nu = (\epsilon_{\text{split}}, \frac{1}{2^\nu+1})\) and the corresponding sequence of boxes tending to a degenerated box.

To avoid this kind of degeneration it is sufficient to enforce the index of the coordinate of the longest edges to be contained in infinitely many \(P^\nu\) in the generated sequence. For a given box \(B^\nu\), comparing the edge lengths whose coordinate index is contained in \(P^\nu\) and the length of the longest edge of the box and replacing \(\eta^\nu\) by the barycenter of \(B^\nu\) if necessary gives one possibility of treating the problem. To express this more precisely let \(\eta^\nu\) be given, \(\epsilon > 0\) and

\[
Q^\nu = \min_{l \in P^\nu} \left( b_l^\nu - b_l^\nu \right) \frac{\| b_l^\nu - b_l^\nu \|_\infty}{\| b_l^\nu - b_l^\nu \|_\infty}
\]

be the relation of the shortest edge length with coordinate index in \(P^\nu\) to the longest edge length of \(B^\nu\). In the case \(Q^\nu < \epsilon\) a degeneration of a box \(B^\nu\) may occur. By setting \(\eta^\nu = S^\nu\) the coordinate index of the longest edges is always contained in \(P^\nu\), and a degeneration can be avoided.

Taking all this into account one achieves the splitting algorithm stated in Algorithm 1 which also describes how the feasible set of a refined problem \(SIP_U F (B^N)\) is constructed.

It is not hard to see that each \(N\)-tuple \((B^k, k = 1, \ldots, N)\) generated by Algorithm 1 is a reduced outer approximation of \(Y\): \(Y\) is covered by \(\bigcup_{k=1}^N B^k\), only those boxes are taken into account which have a non-empty intersection with \(Y\), and the intersection of pairwise different boxes has measure zero. Notice here that the last 'else' condition in Algorithm 1 is included for numerical reasons. We refer to Chapter 7 for details on how the needed constants are determined in Algorithm 1 and on checking if a generated box has a nonempty intersection with the index set \(Y\). Using the splitting algorithm, the adaptive reduction algorithm is stated in Algorithm 2.

If for a given reduced outer approximation \(B^N\) the problem \(SIP_U F (B^N)\) is not consistent in Algorithm 2, a phase 1 algorithm like that described in [16] can be performed. In
Algorithm 1 Splitting step - refineUF(η)

Let η ∈ B^k*, k* ∈ {1, . . . , N}, and let S^k* be the barycenter of B^k* and

Q^k* = \min_{b \in \partial B^k*} \left( \frac{b - b^k*}{\|b - b^k*\|_\infty} \right).

if η ∉ Y or Q^k* < δ then
    Set η = S^k*.
else if P^k* = ∅ then
    Compute (B^k*,1), B^k*,(2) = S(B^k*, η).
    Compute \( \underline{L}^k*,(1), \underline{L}^k*,(2) \geq L^k* \) and \( \overline{L}^k*,(1), \overline{L}^k*,(2) \leq \overline{L}^k* \) on \( X \times B^k*,(1), X \times B^k*,(2) \)
    and set

    \[
    c^k*,(1) = \left( \frac{L^k*,(1) - L^k*,(1)\hat{b}^k*,(1)}{L^k*,(2) - L^k*,(2)\hat{b}^k*,(2)} \right) \ldots \left( \frac{L^k*,(1) - L^k*,(1)\hat{b}^k*,(1)}{L^k*,(2) - L^k*,(2)\hat{b}^k*,(1)} \right) T
    \]

    \[
    c^k*,(2) = \left( \frac{L^k*,(1) - L^k*,(1)\hat{b}^k*,(1)}{L^k*,(2) - L^k*,(2)\hat{b}^k*,(2)} \right) \ldots \left( \frac{L^k*,(1) - L^k*,(1)\hat{b}^k*,(1)}{L^k*,(2) - L^k*,(2)\hat{b}^k*,(2)} \right) T
    \]

    \[
    \hat{g}^k*,(1)(x, y) = g(x, y) + \phi \left( y; c^k*,(1), B^k*,(1), \underline{L}^k*,(1), \overline{L}^k*,(1) \right)
    \]

    \[
    \hat{g}^k*,(2)(x, y) = g(x, y) + \phi \left( y; c^k*,(2), B^k*,(2), \underline{L}^k*,(2), \overline{L}^k*,(2) \right)
    \]

    \[
    M^{(1)} = \{ x \in \mathbb{R}^n \mid \hat{g}^k*,(1)(x, c^k*,(1)) \leq 0 \}
    \]

    \[
    M^{(2)} = \{ x \in \mathbb{R}^n \mid \hat{g}^k*,(2)(x, c^k*,(2)) \leq 0 \}
    \]

    \[\tilde{M} = \{ x \in \mathbb{R}^n \mid \hat{g}^k(x, c^k) \leq 0 \; \text{for all } k \in \{1, \ldots, N\} \setminus \{k^*\} \}
    \]

    if Y ∩ B^k*,(1) = ∅ and Y ∩ B^k*,(2) = ∅ then
        Replace B^k* by B^k*,(1) and B^k*,(2) and replace \( \underline{L}^k*, \overline{L}^k* \) by \( \underline{L}^k*,(1), \overline{L}^k*,(2) \)
        Set M_{UF}(B^N) = \tilde{M} ∩ M^{(1)} \cap M^{(2)}.
        Set N = N + 1.
    else if Y ∩ B^k*,(1) = ∅ then
        Replace B^k* by B^k*,(1) and \( \underline{L}^k*, \overline{L}^k* \) by \( \underline{L}^k*,(1), \overline{L}^k*,(1) \)
        Set M_{UF}(B^N) = \tilde{M} ∩ M^{(1)}.
    else if Y ∩ B^k*,(2) = ∅ then
        Replace B^k* by B^k*,(2) and \( \underline{L}^k*, \overline{L}^k* \) by \( \underline{L}^k*,(2), \overline{L}^k*,(2) \)
        Set M_{UF}(B^N) = \tilde{M} ∩ M^{(2)}.
    else
        Delete B^k* and \( \underline{L}^k*, \overline{L}^k* \).
        Set M_{UF}(B^N) = \tilde{M}.
        Set N = N - 1.
    end if
### 3.3 Convergence results

Before stating the main convergence result some useful lemmata will be given to keep the final proof simple. First of all it will be shown that it is always possible to find a tessellation $B^N$ of the box $B$ so that the problem $SIP_{U_F}(B^N)$ is consistent if $SIP$ has a Slater point. Here the tessellation is not chosen in an adaptive way but temporarily uniform.

**Lemma 3.3.1.** Let $B = [b, \bar{b}] \subset \mathbb{R}^m$ be furnished with a Cartesian grid equally spaced along every coordinate axis, and $B^N = \left( B^k, \ k = 1, \ldots, N \right)$ the induced tessellation of $B$. Let $\bar{x} \in \{ x \in \mathbb{R}^m \mid g(x, y) < 0, y \in Y \}$ and $\bar{y} \in Y$ so that $g(\bar{x}, \bar{y}) = \max_{y \in Y} g(\bar{x}, y) < 0$. Let $L = \max_{(x,y) \in X \times B} \| \nabla_y g(x, y) \|_\infty$ and let $(N_1 + 1, \ldots, N_m + 1)$ be the number of grid points in the spatial directions. If for all $i \in \{1, \ldots, m\}$

$$N_i \geq L \frac{\| \bar{b} - b \|_2}{|g(\bar{x}, \bar{y})|}$$

holds, then $\bar{x}$ is feasible for $SIP_{U_F}(B^N)$.

**Proof.** Let $L^k = \max_{(x,y) \in X \times B^k} \| \nabla_y g(x, y) \|_\infty$ and let $d = \bar{b} - b$. Let $B^k \subset B$ be an arbitrary box. Then its side lengths are $\frac{d_j}{N_j}$, $j = 1, \ldots, m$, and with Lemma 3.1.2 it

---

**Algorithm 2 Adaptive reduction algorithm - ara**

Choose $X \subset \mathbb{R}^n$ with $M \subset X$, $B = [b, \bar{b}] \subset \mathbb{R}^m$ with $Y \subset B$ and compute $L, \bar{L}$ on $B$. Set $L = \max(\| L \|_\infty, \| \bar{L} \|_\infty) \in \mathbb{R}$.

Determine a reduced outer approximation $B^N$ of $Y$ with some $N \in \mathbb{N}$ as well as $L^k \geq L, \bar{L}^k \leq \bar{L}$ and $c^k$ on $B^k$, $k = 1, \ldots, N$, so that $SIP_{U_F}(B^N)$ is consistent.

Choose $\epsilon_{act}, \epsilon_{stat}, \epsilon_Y > 0$ and $\epsilon_{split} \in (0, \frac{1}{2})$ with $\epsilon_{split} \leq \epsilon_{act}/(L\| \bar{b} - b \|_1)$.

Compute a stationary point $x$ of $SIP_{U_F}(B^N)$ with $\epsilon_{act}$-active indices $c^k$, $k = 1, \ldots, n + 1$, and multipliers $(\kappa, \lambda)$ by solving $P_{U_F}(B^N)$.

**while** $x$ is not a stationary point of $SIP$ with $2 \epsilon_{act}$-active, $\epsilon_Y$-feasible indices $c^k$, $k = 1, \ldots, n + 1$, with respect to $B^N$, and multipliers $(\kappa, \lambda)$ **do**

for $k = 1$ to $n + 1$ **do**

refine$_{U_F}(c^k)$

**end for**

Compute a stationary point $x$ of $SIP_{U_F}(B^N)$ with $\epsilon_{act}$-active indices $c^k$, $k = 1, \ldots, n + 1$, and multipliers $(\kappa, \lambda)$ by solving $P_{U_F}(B^N)$.

**end while**

Section 7.3 we will discuss a phase 1 algorithm.

In the next section it will be shown that, if $SIP$ has a Slater point, there always exists a reduced outer approximation so that $SIP_{U_F}(B^N)$ is consistent.
follows for all $y \in B^k$:

$$
\hat{g}^k(x, y) = g(x, y) + \phi \left( y; c^k, B^k, L^k, T^k \right)
\leq g(x, \bar{y}) + (L^k, \bar{b} - c^k)
\leq g(x, \bar{y}) + L^k \| \bar{b} - b^k \|_2
= g(x, \bar{y}) + L^k \sqrt{\sum_{j=1}^{m} \frac{d_j^2}{N_j^2}}
\leq g(x, \bar{y}) + \max_{k=1, \ldots, n+1} L^k \sqrt{\sum_{j=1}^{m} \frac{d_j^2}{\min_{i=1, \ldots, m} N_i^2}}.
$$

Due to the condition on $N_i$ it holds

$$
\min_{i=1, \ldots, m} N_i^2 \geq L^2 \frac{\| \bar{b} - b \|_2^2}{g(x, \bar{y})^2}
$$

which implies $\hat{g}^k(x, y) \leq 0$ and, thus, the assertion. \hfill \Box

Next we show that Algorithm 2 is well defined. In fact, if $P^k = \emptyset$ holds at each active index $c^k$ of an approximating problem $\text{SIP}_{UF}(B^N)$, the reduced outer approximation of $Y$ is not refined any further and Algorithm 2 might loop. However, the following lemma shows that the algorithm terminates in this case.

**Lemma 3.3.2.** Let $\bar{x}$ be a stationary point of $\text{SIP}_{UF}(B^N)$ with $\epsilon_{\text{act}}$-active indices $c^k$, $k = 1, \ldots, n + 1$, and multipliers $(\kappa, \lambda)$. Furthermore, let the refined reduced outer approximation arising from applying Algorithm 1 to each $c^k$, $k = 1, \ldots, n + 1$, coincide with $B^N$. Then Algorithm 2 terminates at $\bar{x}$.

**Proof.** We show that $\bar{x}$ is a stationary point of $\text{SIP}$ with $2\epsilon_{\text{act}}$-active, $\epsilon_Y$-feasible indices $c^k$, $k = 1, \ldots, n + 1$, with respect to $B^N$, and multipliers $(\kappa, \lambda)$.

As a stationary point of $\text{SIP}_{UF}(B^N)$, $\bar{x}$ is also feasible for $\text{SIP}_{UF}(B^N)$, that is, $\bar{x} \in M_{UF}(B^N)$. In view of Lemma 3.1.3 we have that $x \in M$ holds, too. From the stationarity condition for $\text{SIP}_{UF}(B^N)$ we have

$$
\| \kappa \nabla f(\bar{x}) + \sum_{k=1}^{n+1} \lambda_k \nabla_x g(\bar{x}, c^k) \| \leq \epsilon_{\text{stat}}
$$

with the multipliers $(\kappa, \lambda)$ from $\text{SIP}_{UF}(B^N)$ and with any $\epsilon_{\text{stat}}$. Now choose an arbitrary
3 The Adaptive-Reduction Algorithm

c\(^k\), \(k = 1, \ldots, n + 1\). Since \(c^k\) is \(\epsilon_{\text{act}}\)-active we have

\[
\begin{align*}
g(\overline{x}, c^k) + \phi \left( c^k, c^k, B^k, L^k \right) &= g(\overline{x}, c^k) + \sum_{j=1}^{m} L^k_j \left( \overline{b}^k_j - c^k_j \right) \\
&= \hat{g}^k(\overline{x}, c^k) \\
&\in [-\epsilon_{\text{act}}, 0]
\end{align*}
\]

and hence

\[
-\sum_{j=1}^{m} L^k_j \left( \overline{b}^k_j - c^k_j \right) \geq g(\overline{x}, c^k) \geq -\epsilon_{\text{act}} - \sum_{j=1}^{m} L^k_j \left( \overline{b}^k_j - c^k_j \right).
\]

The first inequality shows \(g(\overline{x}, c^k) \leq 0\). By our assumption the reduced outer approximation is not refined, so that \(P^k(c^k) = \emptyset\) must hold. That means that \(\min \left( \overline{b}^k_j - c^k_j, c^k_j - \overline{b}^k_j \right)\) is bounded above by \(\epsilon_{\text{split}} \left( \overline{b}^k_j - \overline{b}^k_j \right)\). Furthermore, the choice of \(c^k\) implies

\[
L^k_j \left( \overline{b}^k_j - c^k_j \right) = L^k_j \left( \overline{b}^k_j - c^k_j \right),
\]

and, thus, the second inequality implies

\[
g(\overline{x}, c^k) \geq -\epsilon_{\text{act}} - L \epsilon_{\text{split}} \|\overline{b} - \overline{b}\|_1,
\]

with \(L = \max(\|L\|_\infty, \|T\|_\infty)\). Using the upper bound \(\frac{\epsilon_{\text{act}}}{L \|\overline{b} - \overline{b}\|_1}\) on \(\epsilon_{\text{split}}\) from Algorithm 2, this leads to

\[
g(\overline{x}, c^k) \geq -\epsilon_{\text{act}} - L \epsilon_{\text{split}} \|\overline{b} - \overline{b}\|_1 \geq -2\epsilon_{\text{act}}.
\]

We have thus shown \(g(\overline{x}, c^k) \in [-2\epsilon_{\text{act}}, 0]\), that is, \(c^k\) is \(2\epsilon_{\text{act}}\)-active.

Finally we have to prove that \(c^k\) is \(\epsilon_Y\)-feasible with respect to \(B^N\). In fact, we even have \(c^k \in Y\) as otherwise in Algorithm 1 \(c^k\) would be replaced by the barycenter of \(B^k\), in contradiction to \(P^k = \emptyset\). Consequently \(c^k\) is \(\epsilon_Y\)-feasible with respect to \(B^N\) for any \(\epsilon_Y > 0\).

The next lemma shows that refining a given reduced outer approximation \(B^N\) of \(Y\) by means of Algorithm 1 enlarges the feasible set of \(\text{SIP}_{UF}(B^N)\).

**Lemma 3.3.3.** Let a reduced outer approximation \(B^N\) of \(Y\) be given, and for a splitting point \(c^k\) let \(B^N\) be the refinement of \(B^N\) by means of Algorithm 1. Then \(M_{UF}(B^N) \subset M_{UF}(B^N)\) holds.

**Proof.** We have \(c^k \in B^k\) with \(k \in \{1, \ldots, N\}\). In the case \(P^k = \emptyset\) the assertion follows immediately, as \(B^k\) is not split. Otherwise, let \(B^k, B^k, T^k, L^k, L^k\) and \(c^k, c^k, c^k\) be constructed as in Algorithm 1. For \(i = 1, 2\) the inequalities \(L^k(i) \geq \overline{L}(i)\) and \(0 \leq \overline{L}(i)\)
3.3 Convergence results

\( \overline{b}^{k,(i)} - \underline{c}^{k,(i)} \leq \overline{b}^k - \underline{c}^{k,(i)} \) hold true. Hence, for all \( x \in M_{UF}(B^N), i = 1, 2, \) and all \( y \in B^{k,(i)} \) it follows that

\[
\hat{g}^{k,(i)}(x, y) \leq \hat{g}^{k,(i)}(x, \underline{c}^{k,(i)}) \\
= g(x, \underline{c}^{k,(i)}) + \sum_{j=1}^{m} L_j^{k,(i)} \left( \overline{b}_j^{k,(i)} - \underline{c}^{k,(i)} \right) \\
\leq g(x, \underline{c}^{k,(i)}) + \sum_{j=1}^{m} L_j^k \left( \overline{b}_j^k - \underline{c}^{k,(i)} \right) \\
\leq g(x, \underline{c}^k) + \sum_{j=1}^{m} L_j^k \left( \overline{b}_j^k - \underline{c}^k \right) \\
= \hat{g}^k(x, \underline{c}^k) \leq 0.
\]

To show some approximation properties of the reduced outer approximations especially on the topological boundary of the index set \( Y \) in the main convergence proofs of the algorithms presented in this and the following chapters, it will be needed that the lengths of the longest edges of the generated sequence of boxes tend to zero. The following lemma discusses this property for a sequence of boxes generated by using the barycenters as splitting points.

**Lemma 3.3.4.** Let \( B^0 = [\overline{b}^0, \underline{b}^0] \subset \mathbb{R}^m \) and \( (B^\nu)_\nu \) be a sequence of boxes where \( B^\nu \) is one of the two boxes generated by the splitting operator \( S(B^{\nu-1}, S^{\nu-1}) \), that is, by splitting \( B^{\nu-1} \) at its barycenter. Then with \( D^\nu := \|\overline{b}^\nu - \underline{b}^\nu\|_2 \) it holds:

(i) The relation \( D^{\nu+1} \leq \sqrt{1 - \frac{3}{4m}} D^\nu \) holds for all \( \nu \in \mathbb{N} \).

(ii) For all \( \nu \in \mathbb{N} \) there is some \( \tau \leq m \) with \( \|\overline{b}^{\nu+\tau} - \underline{b}^{\nu+\tau}\|_\infty \leq \frac{1}{2} \|\overline{b}^\nu - \underline{b}^\nu\|_\infty \).

**Proof.** To see (i) notice that in every splitting step \( 2^{m-1} \) parallel edges are bisected. That means there is some \( l^* \in \{1, \ldots, m\} \) so that \( \left( \overline{b}^{\nu+1} - \underline{b}^{\nu+1} \right)_{l^*} = \frac{1}{2} \left( \overline{b}^\nu - \underline{b}^\nu \right)_{l^*} \), and the other edges remain unchanged. So one obtains

\[
D^{\nu+1} = \sqrt{\sum_{l=1}^{m} \left( \overline{b}^{\nu+1} - \underline{b}^{\nu+1} \right)^2_l + \left( \overline{b}^{\nu+1} - \underline{b}^{\nu+1} \right)^2_{l^*}} \\
= \sqrt{\sum_{l=1}^{m} \left( \overline{b}^{\nu} - \underline{b}^{\nu} \right)^2_l + \left( \overline{b}^{\nu} - \underline{b}^{\nu} \right)^2_{l^*} \frac{1}{4}}.
\]
3 The Adaptive-Reduction Algorithm

\[ = \sqrt{(D')^2 - \frac{3}{4} (\frac{\bar{b}'}{b'})^2}. \]

As \( l^* \) is the index of the longest edge of \( B' \), we have

\[ (D')^2 = \sum_{l=1}^{m} \left( \frac{\bar{b}'}{b'} \right)_l^2 \leq m \left( \frac{\bar{b}'}{b'} \right)_{l^*}^2 \]

and thus

\[ D_{l+1} \leq \sqrt{(D')^2 - \frac{3(D')^2}{4m}} = \sqrt{1 - \frac{3}{4m} D'}. \]

To see part (ii) let without loss of generality

\[ \bar{b}'_1 - b'_1 \geq \ldots \geq \bar{b}'_m - b'_m \]

so that \( \| \bar{b}' - b' \|_\infty = \bar{b}'_1 - b'_1 \). Then in the following step \( \bar{b}'_1' - b'_1' \) is replaced by \( \bar{b}'_1' - b'_1' = \frac{1}{2} \left( \bar{b}'_1 - b'_1 \right) \). If the latter length is still the largest, one can choose \( \tau = 1 \). Otherwise, in the next step \( \bar{b}'_2' - b'_2' \) is replaced by \( \bar{b}'_2' - b'_2' = \frac{1}{2} \left( \bar{b}'_2 - b'_2 \right) \). Again, if this length stays maximal after halving, one can choose \( \tau = 2 \), as \( \frac{1}{2} \left( \bar{b}'_1 - b'_1 \right) \leq \frac{1}{2} \left( \bar{b}'_2 - b'_2 \right) \). Continuing in this way one may either choose some \( \tau \in \{1, \ldots, m-1\} \), or finally all original side lengths are halved, which allows to choose \( \tau = m \). This shows the assertion.

By construction in Algorithm 1, we have that for each splitting point the quotient \( Q \) is larger then a given constant \( \epsilon > 0 \). By Lemma 3.3.2 we have that for the sequence of splitting points the sets \( P \) are nonempty, unless an approximately stationary point with approximately active and feasible indices is computed. For a sequence of boxes arising from applying the splitting operator to a sequence of boxes and splitting points fulfilling the latter conditions, the next lemma discusses a property similar to that of Lemma 3.3.4.

**Lemma 3.3.5.** Let \( 0 < \epsilon_{\text{split}} < \frac{1}{2}, \epsilon > 0 \), \( B^{N,0} = B \) and \( (\eta')_\nu \) be a sequence of points with \( \eta' \in B \) so that \( P' \neq \emptyset \) in the sequence of tessellations \( (B^{N',\nu})_\nu \) arising from applying the splitting operator \( S \) to a box in \( B^{N',\nu} \) and a point \( \eta \) of that box. Provided that \( Q' \geq \epsilon \) for that box and \( \eta' \), choose \( \eta' = \eta' \), and otherwise \( \eta = S^{\nu} \). Let \( \eta^* \in B \) be an accumulation point of \( (\eta')_\nu \). Then \( \| b^{N',\nu} - b^{\nu} \|_\infty \) tends to zero for all boxes \( B^{\nu} \in B^{N',\nu} \) with \( \eta^* \in B^{\nu} \).

**Proof.** First note that the tessellation \( B^{N',\nu} \) consists of finitely many boxes for each fixed \( \nu \). Thus there are only finitely many boxes \( B^{\nu} \in B^{N',\nu} \) with \( \eta^* \in B^{\nu} \) for each fixed \( \nu \). As \( \eta^* \) is an accumulation point of \( (\eta')_\nu \), there must exist infinitely many \( B^{\nu} \in B^{N',\nu} \) in the sequence of tessellations \( (B^{N',\nu})_\nu \) with \( \eta^* \in B^{\nu} \).
By construction, for each box $B^{*,\nu}$ and $\nu > 0$ there is at least one box $B^{*,\mu}$ with $\mu < \nu$ and $B^{*,\nu} \subset B^{*,\mu}$. For each $\nu$ let $l' \in P^\nu$ so that $b_{l'}^{*,\nu} - b_{l'}^{*,\nu} = \max_{l \in P^\nu} b_{l}^{*,\nu} - b_{l}^{*,\nu}$. For a given $\nu$ and a box $B^{*,\nu}$ let $\mu(\nu)$ be the maximal number so that $\mu(\nu) < \nu$, $l^{\mu(\nu)} = l'$ and $B^{*,\nu} \subset B^{*,\mu(\nu)}$. Since the edges of the box $B^{*,\mu(\nu)}$ with the coordinate direction $l^{\mu(\nu)} = l'$ are split and $B^{*,\mu(\nu)}$ is replaced by two new boxes in step $\mu(\nu) + 1$ in the sequence of tessellations, there can only be at most two different boxes $B^{*,l}$ and $B^{*,\tilde{\nu}}$, $\tilde{\nu} \neq \tilde{\nu}$, in the sequence so that $\mu(\tilde{\nu}) = \mu(\tilde{\nu})$ holds. Furthermore, from the definition of $P^\nu$ we obtain that

$$
\left((b_{l'}^{*,\mu(\nu)} - b_{l'}^{*,\nu}) - (b_{l'}^{*,\nu} - b_{l'}^{*,\nu})\right) > \epsilon_{\text{split}}\left(b_{l'}^{*,\mu(\nu)} - b_{l'}^{*,\nu}\right) \quad (3.7)
$$

is true for each $\nu$. Now we have to discuss two cases.

The first case is that each coordinate direction $l \in \{1, \ldots, m\}$ is the direction of the longest edge of infinitely many boxes $B^{*,\nu}$, that is, for infinitely many $\nu$ we have that $l'$ takes every value in $\{1, \ldots, m\}$. In that case (3.7) implies that for each $\nu$ and $l \in \{1, \ldots, m\}$ the inequality $b_{l}^{*,\nu} - b_{l}^{*,\nu} < (1 - \epsilon_{\text{split}})(b_{l'}^{*,\nu} - b_{l'}^{*,\nu})$ holds. Since for each $\nu$ all numbers for $\mu(\nu)$ can only appear twice, for each $\xi \in \mathbb{N}$ and $l \in \{1, \ldots, m\}$ there is some $\nu_0$ so that $b_{l}^{*,\nu} - b_{l}^{*,\nu} < (1 - \epsilon_{\text{split}})^{\xi}(b_{l}^{*} - b_{l}^{*})$ holds for $\nu > \nu_0$. Due to $0 < \epsilon_{\text{split}} < \frac{1}{2}$ it follows that $||b_{l}^{*,\nu} - b_{l}^{*,\nu}||_{\infty}$ tends to zero.

The second case is that there is a set $\hat{L} \subset \{1, \ldots, m\}$ so that the edges of all boxes $B^{*,\nu}$ with coordinate directions $\hat{l} \in \hat{L}$ are split only finitely many times, that is, there is some $\nu_0$ so that $l' \notin \hat{L}$ for all $\nu > \nu_0$. Then (3.7) implies, like in the first case, that for each $\xi \in \mathbb{N}$ and $l \in \{1, \ldots, m\} \setminus \hat{L}$ there is some $\nu_1$ so that $b_{l}^{*,\nu} - b_{l}^{*,\nu} < (1 - \epsilon_{\text{split}})^{\xi}(b_{l}^{*} - b_{l}^{*})$ holds for $\nu > \nu_1$. Since $l' \in \hat{L}$ for finitely many $\nu$, there is some $\nu_2 > \nu_0$ so that $b_{l}^{*,\nu} - b_{l}^{*,\nu} = b_{l}^{*,\mu(\nu)+1} - b_{l}^{*,\mu(\nu)+1}$ holds for each $\nu > \nu_2$ and $\hat{l} \in \hat{L}$, and by the definition of $P^\nu$ we have that $\frac{b_{l_1}^{*,\mu(\nu)+1} - b_{l_1}^{*,\mu(\nu)+1}}{b_{l_1}^{*,\mu(\nu)+1} - b_{l_1}^{*,\mu(\nu)+1}} \geq \epsilon_{\text{split}}(b_{l_1}^{*,\nu} - b_{l_1}^{*,\nu})$ for each $\hat{l} \in \hat{L}$. By applying these results again to $B^{*,\mu(\nu)}$ we obtain that there is some $\nu_2$ and a fixed number $j \in \mathbb{N}$, so that $b_{l}^{*,\nu} - b_{l}^{*,\nu} \geq \epsilon_{\text{split}}(b_{l}^{*} - b_{l}^{*})$ holds for each $\nu > \nu_2$ and $\hat{l} \in \hat{L}$. For any fixed $j \in \mathbb{N}$, and due to $0 < \epsilon_{\text{split}} < \frac{1}{2}$, there is some $\xi \in \mathbb{N}$ so that the inequality $(1 - \epsilon_{\text{split}})^{\xi}(b_{l_1}^{*} - b_{l_1}^{*})$ holds for each $l \in \{1, \ldots, m\} \setminus \hat{L}$ and $\hat{l} \in \hat{L}$. As a consequence of the latter result and the results for $l \in \{1, \ldots, m\} \setminus \hat{L}$ and $\hat{l} \in \hat{L}$, there is some $\nu_3 > \max\{\nu_1, \nu_2\}$ and some $\hat{l} \in \hat{L}$ so that $b_{l}^{*,\nu} - b_{l}^{*,\nu} = b_{l_1}^{*,\nu} - b_{l_1}^{*,\nu}$ holds for each $\nu > \nu_3$. Thus we obtain that for each $\nu > \nu_3$

$$
Q^{\nu} = \frac{\min_{l \in P^\nu} (b_{l}^{*,\nu} - b_{l}^{*,\nu})}{\|b_{l}^{*,\nu} - b_{l}^{*,\nu}\|_{\infty}} = \frac{(b_{l}^{*,\nu} - b_{l}^{*,\nu})}{\|b_{l}^{*,\nu} - b_{l}^{*,\nu}\|_{\infty}}
$$

29
3 The Adaptive-Reduction Algorithm

\[
\frac{(1 - \epsilon_{\text{split}})^\xi (b_l - b_i)}{b_{i^*}^{\nu} - b_{i^*}^{\nu}} \leq \frac{(1 - \epsilon_{\text{split}})^\xi (b_l - b_i)}{b_{i^*}^{\nu} - b_{i^*}^{\nu}},
\]

holds. As \( b_l^\nu - b_{i^*}^{\nu} \) is bounded for each \( \bar{l} \) and \( \hat{l}^* \), and as \( j \) is fixed for \( \nu > \nu_3 \), for every \( \epsilon > 0 \) there is some \( \xi \in \mathbb{N} \) so that \( \frac{(1 - \epsilon_{\text{split}})^\xi (b_l - b_i)}{b_{i^*}^{\nu} - b_{i^*}^{\nu}} \leq \epsilon \). Accordingly there is some \( \nu > \nu_4 \) so that there is some \( \xi \in \mathbb{N} \) with

\[
Q^{\nu} < \frac{(1 - \epsilon_{\text{split}})^\xi (b_l - b_i)}{b_{i^*}^{\nu} - b_{i^*}^{\nu}} < \epsilon
\]

for \( \nu > \nu_4 \). Thus for some \( \nu > \nu_4 \) the point \( S^{\nu} \) is chosen as a splitting point and hence \( l^{\nu} \in \hat{L} \) for some \( \nu > \nu_4 \). That is a contradiction. So in the same manner as in the first case \( \|b_{\nu,\nu}^* - b_{\nu,\nu}^*\|_{\infty} \) tends to zero.

By construction in Algorithm 1, we have that for each splitting point the quotient \( Q \) is larger than a given constant \( \epsilon > 0 \). By Lemma 3.3.2 we have that for the sequence of splitting points the sets \( P \) are nonempty, unless an approximately stationary point with approximately active and feasible indices is computed. Now we state the main convergence result.

**Theorem 3.3.6.** Algorithm 2 terminates after finitely many steps.

**Proof.** We give the proof by enforcing a contradiction. Assume that Algorithm 2 does not terminate. Then there exist sequences of reduced outer approximations \( (B_{\nu,\nu}^N) \) of \( Y \), \( (x_{\nu}) \), \( (c_{k,\nu})_{k=1,\ldots,n+1} \), and \( (\kappa_{\nu}, \lambda_{\nu}) \), so that for each \( \nu \) the point \( x_{\nu} \) is stationary for \( \text{SIP}_{UF}(B_{\nu,\nu}^N) \) with \( \epsilon_{\text{act}} \)-active indices \( (c_{k,\nu})_{k=1,\ldots,n+1} \) and multipliers \( (\kappa_{\nu}, \lambda_{\nu}) \), while \( x_{\nu} \) with all data is not \( \epsilon_{\text{stat}} \)-stationary for \( \text{SIP} \) with \( 2\epsilon_{\text{act}} \)-active, \( \epsilon_Y \)-feasible indices with respect to \( B_{\nu,\nu}^N \).

To derive a contradiction, in steps 1. to 3. the existence of accumulation points of the sequences \( (x_{\nu})_{\nu}, \left((c_{k,\nu})_{k=1,\ldots,n+1}\right)_{\nu} \) and \( (\kappa_{\nu}, \lambda_{\nu})_{\nu} \) is shown. After that it is shown in steps 4. to 6. that there exists a \( \nu_0 \in \mathbb{N} \) so that \( x_{\nu_0} \) with all data is an \( \epsilon_{\text{stat}} \)-stationary point for \( \text{SIP} \) with \( 2\epsilon_{\text{act}} \)-active, \( \epsilon_Y \)-feasible indices with respect to \( B_{\nu,\nu}^N \).

1. **Assertion:** The sequence \( (x_{\nu})_{\nu} \) has an accumulation point \( x^* \in X \cap M \).

Since \( (x_{\nu})_{\nu} \) is contained in the compact set \( X \), it possesses an accumulation point \( x^* \in X \). Let \( M^\nu := M_{UF}(B_{\nu,\nu}^N) \). By Lemma 3.1.3 and Lemma 3.3.3 we have \( M^0 \subset M^1 \subset M^2 \subset \ldots \subset M \), so that \( (x_{\nu})_{\nu} \) is also contained in the closed set \( M \). From this the assertion immediately follows.
2. **Assertion:** The sequence \( \left( (c^{k,\nu})_{k=1,\ldots,n+1}\right)_{\nu} \) has an accumulation point \( (c^{k,*})_{k=1,\ldots,n+1} \in Y^{n+1} \).

To show the assertion it is sufficient to show the existence of an accumulation point in \( Y \) for every component of the sequence of the \((n+1)\)-tuple. Let \( (c^{k,\nu})_{\nu} \), \( k^* \in \{ 1, \ldots, n + 1 \} \), be a sequence of such components. We will distinguish two cases. For the first case assume that \( c^{k,\nu} \in Y \) holds for infinitely many \( \nu \in \mathbb{N} \). By the compactness of \( Y \), the subsequence of these points possesses an accumulation point in \( Y \).

In the second case there is some \( \nu_0 \in \mathbb{N} \) with \( c^{k,\nu} \notin Y \) for all \( \nu \geq \nu_0 \). Let \( (B^{c^{k,\nu}})_{\nu} \) be the sequence of boxes with \( c^{k,\nu} \in B^{c^{k,\nu}} \).

As for each \( \nu \) the reduced outer approximation \( B^{N^{\nu}} \) only contains finitely many boxes, after possibly switching to a subsequence we may assume that \( (B^{k,\nu})_{\nu} \) satisfies \( B^{k,\nu} \supset B^{k,\nu+1} \), \( \nu \in \mathbb{N} \). As Algorithm 1 subdivides each box \( B^{c^{k,\nu}} \) with \( c^{k,\nu} \notin Y \) at its barycenter, Lemma 3.3.4 implies that the maximum edge lengths of the \( B^{k,\nu} \), \( \nu \in \mathbb{N} \), tend to zero. Due to \( B^{k,\nu} \cap Y \neq \emptyset \), \( \nu \in \mathbb{N} \), we arrive at

\[
\lim_{\nu \to \infty} \min_{y \in Y} \| c^{k,\nu} - y \|_2 = 0
\]

and, thus, the assertion.

3. **Assertion:** The sequence \( (\kappa^\nu, \lambda^\nu)_{\nu} \) has an accumulation point \( (\kappa^*, \lambda^*) \in \sigma^{n+1} \).

This is clear from the compactness of \( \sigma^{n+1} \).

4. **Assertion:** There exists a \( \nu_0 \in \mathbb{N} \) so that each \( (c^{k,\nu})_{\nu} \) with \( k = 1, \ldots, n + 1 \) is \( \epsilon_Y \)-feasible with respect to \( B^{N^{\nu}} \) for \( Y \) for all \( \nu \geq \nu_0 \).

By step 2. each sequence \( (c^{k,\nu})_{\nu} \) has an accumulation point \( c^{k,*} \) for each \( k = 1, \ldots, n + 1 \). If \( c^{k,*} \in \text{int}(Y) \) for some \( k^* \in \{ 1, \ldots, n + 1 \} \), the assertion follows immediately for that part of the \((n+1)\)-tuple of the sequence. If \( c^{k,*} \notin \text{int}(Y) \) for some \( k^* \in \{ 1, \ldots, n + 1 \} \), then the assertion follows from Lemma 3.3.5.

5. **Assertion:** There is some \( \nu_0 \in \mathbb{N} \) so that for all \( \nu \geq \nu_0 \) the point \( x^\nu \) is \( \epsilon_{\text{stat}} \)-stationary for SIP with multipliers \( (\kappa^\nu, \lambda^\nu) \).

Since for all \( \nu \) the point \( x^\nu \) is stationary for SIP_{UF}(B^{N^{\nu}}) with \( \epsilon_{\text{act}} \)-active indices \( (c^{k,\nu})_{k=1,\ldots,n+1} \) and multipliers \( (\kappa^\nu, \lambda^\nu) \), parts 1. to 3. of this proof together with a continuity argument entail

\[
\kappa^* \nabla f(x^*) + \sum_{k=1}^{n+1} \lambda^*_k \nabla_x g(x^*, c^{k,*}) = 0.
\]

Hence there is some \( \nu_0 \in \mathbb{N} \) so that for \( \nu \geq \nu_0 \)

\[
\| \kappa^\nu \nabla f(x^\nu) + \sum_{k=1}^{n+1} \lambda^\nu_k \nabla_x g(x^\nu, c^{k,\nu}) \| \leq \epsilon_{\text{stat}}
\]
3 The Adaptive-Reduction Algorithm

holds.

6. **Assertion:** There is some \( \nu_0 \in \mathbb{N} \) so that for all \( \nu \geq \nu_0 \) and each \( k \in \{1, \ldots, n+1\} \) the index \( c^{k,\nu} \) is \( 2\epsilon_{act} \)-active for \( g \).

For any \( k \in \{1, \ldots, n+1\} \) and all \( \nu \in \mathbb{N} \) by assumption we have \( \hat{g}^k(x^{\nu}, c^{k,\nu}) \in [-\epsilon_{act}, 0] \). The overestimating property of \( \hat{g}^k \) ensures \( g(x^{\nu}, c^{k,\nu}) \leq 0 \). Furthermore

\[
\hat{g}^k(x^{\nu}, c^{k,\nu}) = g(x^{\nu}, c^{k,\nu}) + (L^{k,\nu}, \tilde{b}^{k,\nu} - c^{k,\nu}) \leq g(x^{\nu}, c^{k,\nu}) + L \| \tilde{b}^{k,\nu} - b^{k,\nu} \|_2
\]

holds with \( L = \max(\|L\|_\infty, \|\bar{L}\|_\infty) \). In view of Lemma 3.3.5 we have \( \lim_{\nu \to \infty} \| \tilde{b}^{k,\nu} - b^{k,\nu} \|_2 = 0 \), so that there is some \( \nu_0 \in \mathbb{N} \) with \( g(x^{\nu}, c^{k,\nu}) \geq -2\epsilon_{act} \) for all \( \nu \geq \nu_0 \).

Altogether this leads to

\[
g(x^{\nu}, c^{k,\nu}) \in [-2\epsilon_{act}, 0], \quad k = 1, \ldots, n+1
\]

for all \( \nu \geq \nu_0 \), which shows the assertion.

In summary we have shown that there exists a \( \nu_0 \in \mathbb{N} \) so that \( x^{\nu_0} \) with all data is an \( \epsilon_{stat} \)-stationary point of \( SIP \) with \( 2\epsilon_{act} \)-active, \( \epsilon_Y \)-feasible indices with respect to \( B^{N_0,\nu_0} \).

Hence, Algorithm 2 would terminate with \( x^{\nu_0} \), in contradiction to the assumption.

3.4 A first numerical example

In this section we give a first numerical examples motivating our further approaches. We implemented the presented algorithms in Matlab 7.10.0 (R2010a). The parameters \( L \) and \( \bar{L} \) were computed using the gradients of the restrictions \( g \) and the Matlab toolbox Intlab 5.5, [43]. The nonlinear subproblems were solved using fmincon from the Matlab Optimization Toolbox Version 5.0 with default tolerances. The example was run on a 2.4 GHz AMD Athlon 64 X2 processor with 4 GB RAM under Ubuntu 10.04 (lucid).

For more implementation details we refer to Chapter 7.

As a first example we investigate a problem from Chebychev approximation (cf. Chapter 1), that is

\[
CA : \min_{x \in \mathbb{R}^4} \max_{y \in Y} | F(y) - a(x, y) |.
\]

With \( F(y) = \sin(\pi y), a(x, y) = x_3 y^2 + x_2 y + x_1 \) and \( Y = [0, 1] \) we obtain the semi-infinite problem

\[
SIP_{CA} : \min_{x \in \mathbb{R}^4} x_4 \quad s.t. \quad \sin(\pi y) - (x_3 y^2 + x_2 y + x_1) - x_4 \leq 0, \quad y \in [0, 1]
\]

\[
- \sin(\pi y) + x_3 y^2 + x_2 y + x_1 - x_4 \leq 0, \quad y \in [0, 1].
\]

For the set \( X \) we choose \( X = [-1, 1] \times [3, 5] \times [-5, -3] \times [0, 3] \) and we set \( B = Y \).

As a termination tolerance on the stationarity and on the activities of the constraints
we choose $\epsilon = \epsilon_{\text{act}} = 10^{-3}$. For the feasibility of the indices we choose $\epsilon_Y = 0.035$. Furthermore we set $\epsilon_{\text{split}} = \min \left( \frac{\epsilon_{\text{act}}}{\max(\|L\|_\infty, \|L\|_\infty)}, \|z - b\|_1, 0.5 - \epsilon_Y \right)$. One has to notice that we do not need a reduced outer approximation for the set $Y$. It can be handled exactly.

As a starting point we used $x^0 = (1, 5, -3, 2)$. After 1 iteration (0.66 CPU seconds) we obtain the feasible starting point

$$x^1 = (-0.14968, 4.2195, -3.5291, 3)^T.$$ 

After 37 more iterations (383455.38 CPU seconds) we obtain the approximately stationary point

$$x^* = (-0.028467, 4.007, -4.0069, 0.028467)^T$$

with the objective value 0.028467. The error functions are illustrated in Figure 3.3 and the approximating function in Figure 3.4. As termination criteria we used the norm of the stationarity condition, and, additionally, we required that the change in the value of the objective function is less then $10^{-3}$ and the norm of the change in the iterates is less then $10^{-2}$. At $x^*$ the norm of the stationarity condition is less then $10^{-16}$, the change in the value of the objective function is $2.7102 \cdot 10^{-4}$, and the norm of the change in the iterates is 0.00558.

It is particularly noticeable that the numerical performance of the algorithm is that poor, even though the set $Y = [0, 1]$ must not be approximated. A first reason for this may be the fact that, as a result of the reformulation, the number of constraints in the subproblems increases with the number of boxes arising from splitting $B$ into smaller boxes. The overall number of boxes generated during the iterations is displayed in Figure 3.5.
It can be seen that the number of boxes, generated during the iterations, increases exponentially. In the last iteration the total amount of boxes is 2108626 and each box corresponds to one constraint. Thus, a nonlinear problem with 4 variables and 2108626 constraints must be solved. Another criterion to benchmark the numerical effort is the error made by the unimodalization procedure. From the previous discussions we have that the error is $\langle L, \tilde{b}^k - c^k \rangle = \langle \mathcal{L}, \tilde{b}^k - c^k \rangle$. Thus, the error is highly related to the diameter of the generated boxes in the tessellation of $Y$ and the size of $\mathcal{L}, \mathcal{L}$. The latitude of boxes containing active indices and the maximum of the constants $L, \mathcal{L}$, as well as the error made by the unimodalization procedure on boxes containing active indices is illustrated in Figure 3.6.

It can be seen that during the first iterations the error, as well as the constants $L, \mathcal{L}$ decreases very fast, while it gets stuck in the later iterations. In the last iteration the error is around $9.5 \cdot 10^{-3}$. The peak in the values of the constants $L, \mathcal{L}$ can be explained...
3.4 A first numerical example

Figure 3.6: a) The maximum and minimum diameter of the boxes containing active indices generated by Algorithm 2 in each iteration. b) The black line is the maximum of the constants $T, L$ on the boxes containing active indices and the red line is the error made by the unimodalization procedure in each iteration.

by the fact that we only displayed the values for those boxes which contained active indices, and they change in each iteration. From the error bound we know that, in the worst case, we can only expect that the error tends linearly to zero with the size of the boxes. Thus, it is not surprising that the algorithm may have a bad numerical performance, as, particularly, the maximum value for $L, \overline{L}$ stays large. The constants $L, \overline{L}$ are computed by evaluating the derivatives with respect to $y$ of the constraints on the set $X$ and each generated box in the tessellation of $Y$. As the absolute value of these derivatives is $| - \pi \sin (\pi y) + 2x_3y + x_2 |$, one can see that the values for $L, \overline{L}$ are in some kind bounded below by the latitude of $X$, and this set stays unchanged during the iterations.

Notice that the index set $Y$ is an interval in this example, and, thus, the algorithm can handle it exactly and does not need a reduced outer approximation. However, the example points out a first drawback of the algorithm, that is, the error made by relaxing the constraints, and the associated increase in the number of generated boxes. Since the error is linked to the size of the boxes generated during the algorithm and the computed constants, one may try to use another relaxation strategy with better error bounds. Another possibility may be an adaptation of the set $X$ so that the constants $L, \overline{L}$ become smaller, and, thus, the error. Motivated by these observations, we will, at first, discuss another relaxation strategy in the next chapter whose error bound tends quadratically to zeros with the size of the generated boxes. After that, in Chapter 5 we will discuss adaptation strategies of the set $X$ for both algorithms. More numerical examples can be found in Chapter 8.
4 The Adaptive-Convexification Algorithm

The algorithm presented in this chapter has been published in [48]. It is an extension of the algorithm discussed in [16] to arbitrary dimensional and arbitrarily shaped index sets Y.

In Section 4.1 we discuss the relaxation and reformulation techniques used within the algorithm. Then the algorithm is presented in Section 4.2. After that, in Section 4.3, we give the convergence results. At last, in Section 4.4 we present a first numerical example.

4.1 Relaxation and reformulation

Let $Y \subset B \subset \mathbb{R}^m$ and let $B^N = (B^k, k = 1, \ldots, N)$ be a reduced outer approximation of $Y$. Let $\lambda_{max}(x, y)$ be the maximal eigenvalue of $D^2_y g(x, y)$. Define a concave overestimator (w.r.t. $y$) of $g$ on $X \times B$ by

$$\bar{g} : X \times B \longrightarrow \mathbb{R}, \quad (x, y) \longmapsto g(x, y) + \psi(y; \alpha, B)$$

with $\alpha > \max\{0, \max_{(x, y) \in X \times B} \lambda_{max} (x, y)\}$ as well as a concave overestimator of the restriction of $g$ to $X \times B^k$ by

$$\bar{g}^k : X \times B^k \longrightarrow \mathbb{R}, \quad (x, y) \longmapsto g(x, y) + \psi(y; \alpha_k, B^k)$$

with $\alpha_k > \max\{0, \max_{(x, y) \in X \times B^k} \lambda_{max} (x, y)\}$. Due to

$$\max\{0, \max_{(x, y) \in X \times B} \lambda_{max} (x, y)\} \geq \max\{0, \max_{(x, y) \in X \times B^k} \lambda_{max} (x, y)\}$$

$\alpha_k$ can always be chosen so that $\alpha_k \leq \alpha$ holds for each $k = 1, \ldots, N$. The next example illustrates the overestimator $\bar{g}$.

Example 4.1.1. Let $X = B = [0, 2\pi]$ and $g(x, y) = \sin(xy)$. It is not hard to see that we have $\alpha > \max\{0, \max_{(x, y) \in X \times B} \lambda_{max} (x, y)\}$ for $\alpha = \frac{79}{2}$. Thus, we obtain

$$\bar{g} (x, y) = \sin(xy) + \frac{79}{4} y (2\pi - y).$$

$\bar{g}$ is illustrated in Figure 4.1. It can be seen that $\bar{g}$ is an overestimator for $g$ on $X \times B$ and concave in the second argument $y$ for each $x$ but neither concave in general nor in the first argument.
The next results immediately follow from the discussion in Chapter 2, Subsection 2.2.2.

**Lemma 4.1.2.**

(i) For all $k = 1,\ldots,N$ and $(x,y) \in X \times B$ the relation $g(x,y) \leq \hat{g}^k(x,y)$ holds.

(ii) For $x \in X$ and $k = 1,\ldots,N$ the relaxation $\hat{g}^k(x,y)$ is concave in the second argument on $B^k$.

(iii) For all $k = 1,\ldots,N$ the maximum separation distance between $\hat{g}^k$ and $g$ on $X \times B^k$ is $rac{\alpha}{\pi} \|[\hat{b}^k - b^k]\|^2_2$.

(iv) If $\bar{x} \in X$ and $\hat{g}^k(\bar{x},y) \leq 0$ holds for all $y \in B^k$, $k = 1,\ldots,N$, then we have $\bar{x} \in M$.

Define the set

$$M_{\alpha BB}(B^N) := \{x \in \mathbb{R}^n | \hat{g}^k(x,y) \leq 0 \text{ for all } y \in B^k, k = 1,\ldots,N\}$$

and the semi-infinite problem

$$\text{SIP}_{\alpha BB}(B^N) : \min_{x \in X} f(x) \quad \text{s.t.} \quad x \in M_{\alpha BB}(B^N)$$

as well as the lower level problems

$$Q^k(x) : \max_{y \in \mathbb{R}^m} \hat{g}^k(x,y) \quad \text{s.t.} \quad y \in B^k$$

for $k = 1,\ldots,N$. For $M_{\alpha BB}$, $\text{SIP}_{\alpha BB}$ and $Q^k$ we obtain the following results. As in the former chapter, we omit the straightforward proofs.
4.1 Relaxation and reformulation

Lemma 4.1.3.

(i) We have $M_{\alpha BB}(B^N) \subset M$.

(ii) Let $SIP_{\alpha BB}(B^N)$ be consistent. Then every global, local solution or stationary point of $SIP_{\alpha BB}(B^N)$ is a feasible point of $SIP$.

(iii) For $\alpha_k > \max \left\{ 0, \max_{(x,y) \in X \times B^k} \lambda_{\text{max}}(x,y) \right\}$ the solution of $Q^k(x)$ is unique for each $k = 1, \ldots, N$.

In [47] it is shown that the problem $SIP_{\alpha BB}(B^N)$ is equivalent to the Stackelberg game $SG_{\alpha BB}(B^N) : \min_{x,y^1,\ldots,y^N} f(x) \text{ s.t. } \tilde{g}^k(x,y^k) \leq 0$.

For every $k \in \{1, \ldots, N\}$ the function $\tilde{g}^k$ is concave in the second argument, and $B^k$ is convex and possesses a Slater point. For this reason the Karush-Kuhn-Tucker conditions are necessary and sufficient for $y^k$ to be a solution of $Q^k(x)$. Thus for each $k \in \{1, \ldots, N\}$ the point $y^k$ is a solution of $Q^k(x)$ if and only if there exist multipliers $\gamma^k, \pi^k \in \mathbb{R}^m$ with

$$
\nabla_y \tilde{g}^k(x, y^k) + \gamma^k - \pi^k = 0,
$$

$$
0 \leq y^k - b^k \perp \gamma^k \geq 0,
$$

$$
0 \leq b^k - y^k \perp \pi^k \geq 0.
$$

Define

$$
\omega := \left( x, \left( y^k \right)_{k=1,\ldots,N}, \left( \gamma^k \right)_{k=1,\ldots,N}, \left( \pi^k \right)_{k=1,\ldots,N} \right)
$$

$$
F(\omega) := f(x)
$$

$$
G(\omega) := \left( \tilde{g}^k \left( x, y^k \right) \right)_{k=1,\ldots,N}
$$

$$
C^1(\omega) := \left( \gamma^k, \pi^k \right)_{k=1,\ldots,N}
$$

$$
C^2(\omega) := \left( \left( y^k - b^k \right), \left( b^k - y^k \right) \right)_{k=1,\ldots,N}
$$

$$
H(\omega) := \left( \nabla_y \tilde{g}^k \left( x, y^k \right) + \gamma^k - \pi^k \right)_{k=1,\ldots,N}.
$$

Now $SG_{\alpha BB}(B^N)$ can be equivalently reformulated as the MPCC

$$
P_{\alpha BB}(B^N) : \min_{\omega} F(\omega) \text{ s.t. } G(\omega) \leq 0, \quad H(\omega) = 0, \quad 0 \leq C^1(\omega) \perp C^2(\omega) \geq 0.
$$

This problem may be tackled by different solvers like those in [9, 44, 46]. For more details we, again, refer to Chapter 7, Section 7.2.
4 The Adaptive-Convexification Algorithm

4.2 Algorithms

Similar to the algorithm presented in the latter chapter, the algorithm presented in this section computes a stationary point of $SIP_{\alpha BB}(B^N)$ with active indices and terminates if it is also a stationary point of $SIP$ within a given tolerance on stationarity and additional tolerances on the feasibility of the indices. If the computed point is not approximately stationary or the active indices are not approximately feasible, the reduced outer approximation $B^N$ of $Y$ is refined and a respective refined $SIP_{\alpha BB}$ is solved. In the following chapter most of the notations used in the former chapter are adopted. But, since the indices are not predetermined as with the unimodal relaxations, when convex relaxations are used, we have to introduce some new notations.

For a point $\overline{x} \in M_{\alpha BB}(B^N)$ and solutions $y^k(\overline{x})$ of $Q^k(\overline{x})$, $k = 1, \ldots, N$, define

$$K^0_{\alpha BB}(\overline{x}) := \{k \in \{1, \ldots, N\} | \check{g}^k(\overline{x}, y^k(\overline{x})) = 0\} \quad \text{and} \quad B^0_{\alpha BB}(\overline{x}) := \{y^k(\overline{x}) | k \in K^0_{\alpha BB}(\overline{x})\}.$$ 

As in Theorem 2.1.1, a point $\overline{x} \in M_{\alpha BB}(B^N)$ is stationary for $SIP_{\alpha BB}(B^N)$ in the sense of John, if there exist $y^k \in B^0_{\alpha BB}(\overline{x})$, $k = 1, \ldots, n + 1$, and $(\kappa, \lambda) \in \sigma^{n+1}$, so that

$$\kappa \nabla f(\overline{x}) + \sum_{k=1}^{n+1} \lambda_k \nabla x g(\overline{x}, y^k) = 0,$$

$$\lambda_k \check{g}^k(\overline{x}, y^k) = 0, \quad k = 1, \ldots, n + 1.$$

Notice here that $\nabla_x \check{g}^k(\overline{x}, y^k) = \nabla x g(\overline{x}, y^k)$.

In contrast to the unimodal relaxations, as mentioned before, the indices $y^k$ are not predetermined in the current relaxation scheme. Thus, we have to introduce a slightly different concept for the $\epsilon_Y$-feasibility of the indices.

**Definition 4.2.1.** Let $\epsilon_Y > 0$. $y$ is called $\epsilon_Y$-feasible with respect to the reduced outer approximation $B^N$ of $Y$ if $y \in Y$ or if $y \notin Y$ and there exists a box $B^k$ so that $y \in B^k$, $B^k \cap Y \neq \emptyset$ and $\|b^k - y^k\|_{\infty} < \epsilon_Y$.

Using the splitting function and following the ideas of the former chapter one achieves the splitting algorithm stated in Algorithm 3. The algorithm also describes how the feasible set of a refined problem $SIP_{\alpha BB}(B^N)$ is constructed.

It is not hard to see that each $N$-tuple $(B^k, k = 1, \ldots, N)$ generated by Algorithm 3 is a reduced outer approximation of $Y$: $Y$ is covered by $\bigcup_{k=1}^N B^k$, only those boxes are taken into account which have a non-empty intersection with $Y$, and the intersection of pairwise different boxes has measure zero. Notice here that the last ‘else’ condition in Algorithm 3 is included for numerical reasons. We refer to Chapter 7 for details on how the needed constants are determined in Algorithm 3 and on checking if a generated box has a nonempty intersection with the index set $Y$. Using the splitting algorithm, the adaptive convexification algorithm is stated in Algorithm 4.

If for a given reduced outer approximation $B^N$ the problem $SIP_{\alpha BB}(B^N)$ is not consistent in Algorithm 4, a phase 1 algorithm like that described in [16] can be performed.
Algorithm 3 Splitting step - refine\(\alpha_{BB}(\eta)\)

Let \(\eta \in B^{k^*}, k^* \in \{1, \ldots, N\}\), and let \(S^{k^*}\) be the barycenter of \(B^{k^*}\) and
\[
Q^{k^*} = \min_{\eta \in P^{k^*}} \frac{\|b^{k^*} - \tilde{b}^{k^*}\|_\infty}{\|c^{k^*} - \tilde{c}^{k^*}\|_\infty}.
\]

if \(\eta \notin Y\) or \(Q^{k^*} < \epsilon\) then
Set \(\eta = S^{k^*}\).
end if
if \(P^{k^*} \neq \emptyset\) then
Compute \(\langle B^{k^*},(1), B^{k^*},(2) \rangle = S(B^{k^*}, \eta)\).
Compute \(\alpha^{(1)}_{k^*}, \alpha^{(2)}_{k^*} \leq \alpha_{k^*}\) on \(B^{k^*},(1), B^{k^*},(2)\) and set
\[
\tilde{g}^{k^*,(1)}(x, y) = g(x, y) + \frac{\alpha^{(1)}_{k^*}}{2} (y - \tilde{b}^{k^*,(1)}, \tilde{b}^{k^*,(1)} - y)
\]
\[
\tilde{g}^{k^*,(2)}(x, y) = g(x, y) + \frac{\alpha^{(2)}_{k^*}}{2} (y - \tilde{b}^{k^*,(2)}, \tilde{b}^{k^*,(2)} - y)
\]
\[
M^{(1)} = \{ x \in \mathbb{R}^n \mid \tilde{g}^{k^*,(1)}(x, y) \leq 0 \text{ for all } y \in B^{k^*,(1)} \}
\]
\[
M^{(2)} = \{ x \in \mathbb{R}^n \mid \tilde{g}^{k^*,(2)}(x, y) \leq 0 \text{ for all } y \in B^{k^*,(2)} \}
\]
\[
\tilde{M} = \{ x \in \mathbb{R}^n \mid \tilde{g}^k(x, y) \leq 0 \text{ for all } y \in B^{k}, k \in \{1, \ldots, N\} \setminus \{k^*\} \}
\]
if \(Y \cap B^{k^*},(1) \neq \emptyset\) and \(Y \cap B^{k^*},(2) \neq \emptyset\) then
Replace \(B^{k^*}\) by \(B^{k^*},(1)\) and \(B^{k^*},(2)\) and replace \(\alpha_{k^*}\) by \(\alpha^{(1)}_{k^*}\) and \(\alpha^{(2)}_{k^*}\).
Set \(M_{\alpha_{BB}}(B^N) = \tilde{M} \cap M^{(1)} \cap M^{(2)}\).
Set \(N = N + 1\).
else if \(Y \cap B^{k^*},(1) \neq \emptyset\) then
Replace \(B^{k^*}\) by \(B^{k^*},(1)\) and \(\alpha_{k^*}\) by \(\alpha^{(1)}_{k^*}\).
Set \(M_{\alpha_{BB}}(B^N) = \tilde{M} \cap M^{(1)}\).
else if \(Y \cap B^{k^*},(2) \neq \emptyset\) then
Replace \(B^{k^*}\) by \(B^{k^*},(2)\) and \(\alpha_{k^*}\) by \(\alpha^{(2)}_{k^*}\).
Set \(M_{\alpha_{BB}}(B^N) = \tilde{M} \cap M^{(2)}\).
else
Delete \(B^{k^*}\) and \(\alpha_{k^*}\).
Set \(M_{\alpha_{BB}}(B^N) = \tilde{M}\).
Set \(N = N - 1\).
end if
end if

For more details on a phase 1 algorithm we refer to Section 7.3. In the next section it will be shown that, if \(SIP\) has a Slater point, there always exists a reduced outer approximation so that \(SIP_{\alpha_{BB}}(B^N)\) is consistent.
Algorithm 4 Adaptive convexification algorithm - aca

Choose $X \subset \mathbb{R}^n$ with $M \subset X$ and compute $\alpha$ on $B_k$, $k = 1, \ldots, N$, so that $SIP_{aBB}(B^N)$ is consistent.

Determine a reduced outer approximation $B_N$ of $Y$ with some $N \in \mathbb{N}$ as well as $\alpha_k \leq \alpha$ on $B_k$, $k = 1, \ldots, N$, and multipliers $(\kappa, \lambda)$ by solving $P_{aBB}(B^N)$. 

while $x$ is not a stationary point of $SIP$ with $2\epsilon_{act}$-active, $\epsilon_Y$-feasible indices $y^k$, $k = 1, \ldots, n + 1$, with respect to $B^N$, and multipliers $(\kappa, \lambda)$ do

for $k = 1$ to $n + 1$ do

refine $aBB(y^k)$

end for

Compute a stationary point $x$ of $SIP_{aBB}(B^N)$ with $\epsilon_{act}$-active indices $y^k$, $k = 1, \ldots, n + 1$, and multipliers $(\kappa, \lambda)$ by solving $P_{aBB}(B^N)$.

end while

4.3 Convergence results

As in Chapter 3, Section 3.3 we will first give some useful lemmata to keep the final proof simple. Following the assembling of Section 3.3 it will be shown first that it is always possible to find a tessellation $B_N$ of the box $B$ so that the problem $SIP_{aBB}(B^N)$ is consistent if $SIP$ has a Slater point. Here the tessellation is not chosen in an adaptive way but temporarily uniform.

Lemma 4.3.1. Let $B = [\underline{b}, \overline{b}] \subset \mathbb{R}^m$ be furnished with a Cartesian grid equally spaced along every coordinate axis, and $B^N = (B_k, k = 1, \ldots, N)$ the induced tessellation of $B$. Let $\overline{x} \in \{ x \in \mathbb{R}^n \mid g(x, y) < 0, y \in Y \}$ and $\overline{y} \in Y$ so that $g(\overline{x}, \overline{y}) = \max_{y \in Y} g(\overline{x}, y) < 0$. Let $(N_1 + 1, \ldots, N_m + 1)$ be the number of grid points in the spatial directions. If for all $i \in \{ 1, \ldots, m \}$

$$N_i \geq \sqrt{\frac{\max_{k=1,\ldots,N} \alpha_k \|\overline{b} - b\|_2^2}{8|g(\overline{x}, \overline{y})|}}$$

holds, then $\overline{x}$ is feasible for $SIP_{aBB}(B^N)$.

Proof. Let $d = \overline{b} - \underline{b}$ and let $B^k \subset B$ be an arbitrary box. Then its side lengths are $\frac{d_j}{N_j}$, $j = 1, \ldots, m$, and with Lemma 4.1.2 it follows for all $y \in B^k$:

$$\tilde{g}^k(\overline{x}, y) = g(\overline{x}, y) + \frac{\alpha_k}{2} (y - \overline{b}^k, \overline{b}^k - y) \leq g(\overline{x}, \overline{y}) + \frac{\alpha_k}{8} \|\overline{b}^k - \underline{b}^k\|_2^2$$
4.3 Convergence results

\[ g(\bar{x}, \bar{y}) + \frac{\alpha_k}{8} \sum_{j=1}^{m} \frac{d_j^2}{N_j^2} \]

\[ \leq g(\bar{x}, \bar{y}) + \frac{\max_{k=1, \ldots, N} \alpha_k}{8} \sum_{j=1}^{m} \min_{i=1, \ldots, m} \frac{d_j^2}{N_i^2}. \]

Due to the condition on \( N_i \) it holds

\[ \min_{i=1, \ldots, m} N_i^2 \geq \frac{\max_{k=1, \ldots, N} \alpha_k}{8} \sum_{j=1}^{m} \frac{d_j^2}{|g(\bar{x}, \bar{y})|} \]

which implies \( \bar{g}^k(\bar{x}, y) \leq 0 \) and, thus, the assertion.

Now we show that Algorithm 4 is well defined. In fact, since we are in a similar situation as in the latter chapter, if \( P^k = \emptyset \) holds at each active index \( y^k \) of an approximating problem \( SIP_{aBB}(B^N) \), the reduced outer approximation of \( Y \) is not refined any further and Algorithm 4 might loop. The following lemma shows that the algorithm terminates in this case.

**Lemma 4.3.2.** Let \( \bar{x} \) be a stationary point of \( SIP_{aBB}(B^N) \) with \( \epsilon_{\text{act}} \)-active indices \( y^k \), \( k = 1, \ldots, n + 1 \), and multipliers \((\kappa, \lambda)\). Furthermore, let the refined reduced outer approximation arising from applying Algorithm 3 to each \( y^k \), \( k = 1, \ldots, n + 1 \), coincide with \( B^N \). Then Algorithm 4 terminates at \( \bar{x} \).

**Proof.** We show that \( \bar{x} \) is a stationary point of \( SIP \) with \( 2\epsilon_{\text{act}} \)-active, \( \epsilon_Y \)-feasible indices \( y^k \), \( k = 1, \ldots, n + 1 \), with respect to \( B^N \), and multipliers \((\kappa, \lambda)\).

First, as a stationary point of \( SIP_{aBB}(B^N) \), \( \bar{x} \) is in \( M_{\text{aBB}}(B^N) \) and, in view of Lemma 4.1.3, also in \( M \). Furthermore, the stationarity condition for \( SIP_{aBB}(B^N) \) implies

\[ \|\kappa \nabla f(\bar{x}) + \sum_{k=1}^{n+1} \lambda_k \nabla_x g(\bar{x}, y^k)\| \leq \epsilon_{\text{stat}} \]

with the multipliers \((\kappa, \lambda)\) from \( SIP_{aBB}(B^N) \) and with any \( \epsilon_{\text{stat}} \).

Now choose an arbitrary \( y^k \), \( k = 1, \ldots, n + 1 \). Since \( y^k \) is \( \epsilon_{\text{act}} \)-active we have

\[ g(\bar{x}, y^k) + \frac{\alpha_k}{2} (y^k - b^k, b^k - y^k) = \bar{g}^k(\bar{x}, y^k) \in [-\epsilon_{\text{act}}, 0] \]

and hence

\[ -\frac{\alpha_k}{2} (y^k - b^k, b^k - y^k) \geq g(\bar{x}, y^k) \geq -\epsilon_{\text{act}} - \frac{\alpha_k}{2} (y^k - b^k, b^k - y^k). \]

The first inequality shows \( g(\bar{x}, y^k) \leq 0 \). Due to \( \alpha_k \leq \alpha \) the second inequality implies

\[ g(\bar{x}, y^k) \geq -\epsilon_{\text{act}} - \frac{\alpha}{2} \sum_{j=1}^{m} (y^k_j - b^k_j) (b^k_j - y^k_j). \]
4 The Adaptive-Convexification Algorithm

By assumption the reduced outer approximation is not refined, so that \( P^k(y^k) = \emptyset \) must hold, and in each summand of the above sum one factor can be bounded by \( \epsilon_{\text{split}} \left( \overline{b}_j^k - \underline{b}_j^k \right) \), while the other factor is trivially bounded by \( \left( \overline{b}_j^k - \underline{b}_j^k \right) \). This leads to

\[
g(\overline{x}, y^k) \geq -\epsilon_{\text{act}} - \frac{\alpha}{2} \epsilon_{\text{split}} \| \overline{b}^k - \underline{b}^k \|^2 \geq -\epsilon_{\text{act}} - \frac{\alpha}{2} \epsilon_{\text{split}} \| \overline{b} - \underline{b} \|^2 \geq -2 \epsilon_{\text{act}}
\]

where we have used the upper bound \( 2\epsilon_{\text{act}} / \left( \alpha \| \overline{b} - \underline{b} \|^2 \right) \) on \( \epsilon_{\text{split}} \) from Algorithm 4. We have thus shown \( g(\overline{x}, y^k) \in [-2\epsilon_{\text{act}}, 0] \), that is, \( y^k \) is \( 2\epsilon_{\text{act}} \)-active.

Finally we have to prove that \( y^k \) is \( \epsilon_Y \)-feasible with respect to \( B^N \). In fact, we even have \( y^k \in Y \) as otherwise in Algorithm 3 \( y^k \) would be replaced by the barycenter of \( B^k \), in contradiction to \( P^k = \emptyset \). Consequently \( y^k \) is \( \epsilon_Y \)-feasible with respect to \( B^N \) for any \( \epsilon_Y > 0 \).

The next lemma shows that refining a given reduced outer approximation \( B^N \) of \( Y \) by means of Algorithm 3 enlarges the feasible set of \( \text{SIP}_{aBB}(B^N) \).

**Lemma 4.3.3.** Let a reduced outer approximation \( B^N \) of \( Y \) be given, and for an arbitrary splitting point \( \eta \) let \( \tilde{B}^N \) be the refinement of \( B^N \) by means of Algorithm 3. Then \( M_{aBB}(B^N) \subseteq M_{aBB}(\tilde{B}^N) \) holds.

**Proof.** Let \( \eta \in B^k \) with \( k \in \{1, \ldots, N\} \). In the case \( P^k = \emptyset \) the assertion follows immediately, as \( B^k \) is not split. Otherwise, let \( B^{k,(1)}, B^{k,(2)}, \alpha^{(1)}, \alpha^{(2)} \) be constructed like in Algorithm 3. For \( i = 1, 2 \) and \( y \in B^{k,(i)} \) the inequalities \( \alpha^{(i)} \leq \alpha \) and \( 0 \leq y - b^{k,(i)} \leq y - b^k, 0 \leq b^{k,(i)} - y \leq b^k - y \) hold true. Hence, for all \( x \in M_{aBB}(B^N) \), \( i = 1, 2 \), and all \( y \in B^{k,(i)} \) it follows that

\[
g^{k,(i)}(x, y) = g(x, y) + \frac{\alpha^{(i)}}{2} \langle y - b^{k,(i)}, b^{k,(i)} - y \rangle \\
\leq g(x, y) + \frac{\alpha}{2} \langle y - b^k, b^k - y \rangle = g^k(x, y) \leq 0.
\]

Now we are in the position to state the main convergence result.

**Theorem 4.3.4.** Algorithm 4 terminates after finitely many steps.

**Proof.** As in Theorem 3.3.6 we give the proof by enforcing a contradiction. Assume that Algorithm 4 does not terminate. Then there exist sequences of reduced outer approximations \( \left( \tilde{B}^{N^\nu,\nu}, \nu \right) \) of \( Y \), \( (x^\nu), \nu \), \( (y^{k,\nu})_{k=1,\ldots,n+1}, \nu \) and \( (\kappa^\nu, \lambda^\nu), \nu \), so that for each \( \nu \) the point \( x^\nu \) is stationary for \( \text{SIP}_{aBB}(\tilde{B}^{N^\nu,\nu}) \) with \( \epsilon_{\text{act}} \)-active indices \( (y^{k,\nu})_{k=1,\ldots,n+1} \) and multipliers \( (\kappa^\nu, \lambda^\nu) \), while \( x^\nu \) with all data is not \( \epsilon_{\text{stat}} \)-stationary for \( \text{SIP} \) with \( 2\epsilon_{\text{act}} \)-active, \( \epsilon_Y \)-feasible indices with respect to \( B^{N^\nu,\nu} \).
4.3 Convergence results

To derive a contradiction, in steps 1. to 3. the existence of accumulation points of the sequences $(x^\nu)_\nu$, $(y^{k,\nu})_{k=1,...,n+1}\nu$ and $(\kappa^\nu, \lambda^\nu)_\nu$ is shown. After that it is shown in steps 4. to 6. that there exists a $\nu_0 \in \mathbb{N}$ so that $x^{\nu_0}$ with all data is an $\epsilon_{stat}$-stationary point for SIP with $2\epsilon_{act}$-active, $\epsilon_Y$-feasible indices with respect to $\mathcal{B}^{N^\nu, \nu}$.

1. **Assertion:** The sequence $(x^\nu)_\nu$ has an accumulation point $x^* \in X \cap M$.

   Since $(x^\nu)_\nu$ is contained in the compact set $X$, it possesses an accumulation point $x^* \in X$. Let $M^\nu := M_{\alpha BB}(\mathcal{B}^{N^\nu, \nu})$. By Lemma 4.1.3 and Lemma 4.3.3 we have $M^0 \subset M^1 \subset M^2 \subset \ldots \subset M$, so that $(x^\nu)_\nu$ is also contained in the closed set $M$. From this the assertion immediately follows.

2. **Assertion:** The sequence $(y^{k,\nu})_{k=1,...,n+1}\nu$ has an accumulation point $(y^{k,*})_{k=1,...,n+1} \in Y^{n+1}$.

   To show the assertion it is sufficient to show the existence of an accumulation point in $Y$ for every component of the sequence of the $(n+1)$-tuple. Let $(y^{k,\nu})_\nu$, $k^* \in \{1, \ldots, n+1\}$, be a sequence of such components. We will distinguish two cases. For the first case assume that $y^{k^*,\nu} \in Y$ holds for infinitely many $\nu \in \mathbb{N}$. By the compactness of $Y$, the subsequence of these points possesses an accumulation point in $Y$.

   In the second case there is some $\nu_0 \in \mathbb{N}$ with $y^{k^*,\nu} \notin Y$ for all $\nu \geq \nu_0$. Let $(B^{k^*,\nu})_\nu$ be the sequence of boxes with $y^{k^*,\nu} \in B^{k^*,\nu}$.

   As for each $\nu$ the reduced outer approximation $\mathcal{B}^{N^\nu, \nu}$ only contains finitely many boxes, after possibly switching to a subsequence we may assume that $(B^{k^*,\nu})_\nu$ satisfies $B^{k^*,\nu} \supset B^{k^*,\nu+1}$, $\nu \in \mathbb{N}$. As Algorithm 3 subdivides each box $B^{k^*,\nu}$ with $y^{k^*,\nu} \notin Y$ at its barycenter, Lemma 3.3.4 implies that the maximum edge lengths of the $B^{k^*,\nu}$, $\nu \in \mathbb{N}$, tend to zero. Due to $B^{k^*,\nu} \cap Y \neq \emptyset$, $\nu \in \mathbb{N}$, we arrive at
   \[
   \lim_{\nu \to \infty} \min_{y \in Y} ||y^{k^*,\nu} - y||_2 = 0
   \]
   and, thus, the assertion.

3. **Assertion:** The sequence $(\kappa^\nu, \lambda^\nu)_\nu$ has an accumulation point $(\kappa^*, \lambda^*) \in \sigma^{n+1}$.

   This is clear from the compactness of $\sigma^{n+1}$.

4. **Assertion:** There exists a $\nu_0 \in \mathbb{N}$ so that each $(y^{k,\nu})_\nu$ with $k = 1, \ldots, n+1$ is $\epsilon_Y$-feasible with respect to $\mathcal{B}^{N^\nu, \nu}$ for $Y$ for all $\nu \geq \nu_0$.

   By step 2. each sequence $(y^{k,\nu})_\nu$ has an accumulation point $y^{k,*}$ for each $k = 1, \ldots, n+1$. If $y^{k^*,\nu} \in \text{int}(Y)$ for some $k^* \in \{1, \ldots, n+1\}$, the assertion follows immediately for that part of the $(n+1)$-tuple of the sequence. If $y^{k^*,\nu} \notin \text{int}(Y)$ for some $k^* \in \{1, \ldots, n+1\}$, then the assertion follows from Lemma 3.3.5.

5. **Assertion:** There is some $\nu_0 \in \mathbb{N}$ so that for all $\nu \geq \nu_0$ the point $x^\nu$ is $\epsilon_{stat}$-stationary for SIP with multipliers $(\kappa^\nu, \lambda^\nu)$.

45
Since for all $\nu$ the point $x^\nu$ is stationary for $SIP_{\alpha BB}(B^{N\nu}, \nu)$ with $\epsilon_{\text{act}}$-active indices $(y^{k,\nu})_{k=1, \ldots, n+1}$ and multipliers $(\kappa^{\nu}, \lambda^{\nu})$, parts 1. to 3. of this proof together with a continuity argument entail

$$\kappa^\nu \nabla f(x^*) + \sum_{k=1}^{n+1} \lambda_k^\nu \nabla_x g(x^*, y^{k,*}) = 0.$$ 

Hence there is some $\nu_0 \in \mathbb{N}$ so that for $\nu \geq \nu_0$

$$||\kappa^\nu \nabla f(x^\nu) + \sum_{k=1}^{n+1} \lambda_k^\nu \nabla_x g(x^\nu, y^{k,\nu})|| \leq \epsilon_{\text{stat}}$$

holds.

6. **Assertion:** There is some $\nu_0 \in \mathbb{N}$ so that for all $\nu \geq \nu_0$ and each $k \in \{1, \ldots, n+1\}$ the index $y^{k, \nu}$ is $2\epsilon_{\text{act}}$-active for $g$.

For any $k \in \{1, \ldots, n+1\}$ and all $\nu \in \mathbb{N}$ by assumption we have \( \hat{g}^k(x^\nu, y^{k,\nu}) \in [-\epsilon_{\text{act}}, 0] \). The overestimating property of $\hat{g}^k$ ensures $g(x^\nu, y^{k,\nu}) \leq 0$. Furthermore

$$\hat{g}^k(x^\nu, y^{k,\nu}) = g(x^\nu, y^{k,\nu}) + \alpha \left( y^{k,\nu} - y^{k,\nu} \right) \leq g(x^\nu, y^{k,\nu}) + \alpha \left( b^{k,\nu} - y^{k,\nu} \right)^2$$

holds. In view of Lemma 3.3.5 we have $\lim_{\nu \to \infty} \left( b^{k,\nu} - y^{k,\nu} \right)^2 = 0$, so that there is some $\nu_0 \in \mathbb{N}$ with $g(x^\nu, y^{k,\nu}) \geq -2\epsilon_{\text{act}}$ for all $\nu \geq \nu_0$. Altogether this leads to

$$g(x^\nu, y^{k,\nu}) \in [-2\epsilon_{\text{act}}, 0], \; k = 1, \ldots, n+1$$

for all $\nu \geq \nu_0$, which shows the assertion.

We have shown that there exists a $\nu_0 \in \mathbb{N}$ so that $x^{\nu_0}$ with all data is an $\epsilon_{\text{stat}}$-stationary point of $SIP$ with $2\epsilon_{\text{act}}$-active, $\epsilon_Y$-feasible indices with respect to $B^{N\nu_0 \nu_0}$. Hence, Algorithm 4 would terminate with $x^{\nu_0}$, in contradiction to the assumption.

4.4 A first numerical example

In this section we investigate the problem discussed in Section 3.4. We implemented the presented algorithms in Matlab 7.10.0 (R2010a). The parameter $\alpha$ was computed using the theorem of Gerschgorin and the Matlab toolbox Intlab 5.5, [43]. The nonlinear subproblems were solved using fmincon from the Matlab Optimization Toolbox Version 5.0 with default tolerances. The example was run on a 2.4 GHz AMD Athlon 64 X2 processor with 4 GB RAM under Ubuntu 10.04 (lucid). For more implementation details we refer to Chapter 7.
4.4 A first numerical example

The problem from Section 3.4 is a Chebychev approximation problem (cf. Chapter 1). We want to compute a point $x \in \mathbb{R}^3$ so that the maximal approximation error of the function $a(x, y) = x_3y^2 + x_2y + x_1$ to $F(y) = \sin(\pi y)$ becomes small. Reformulating the problem as a semi-infinite problem we arrive at

\[
SIP_{CA} : \min_{x \in \mathbb{R}^4} x_4 \quad s.t. \quad \sin(\pi y) - (x_3y^2 + x_2y + x_1) - x_4 \leq 0, \, y \in [0, 1] \\
- \sin(\pi y) + x_3y^2 + x_2y + x_1 - x_4 \leq 0, \, y \in [0, 1].
\]

For the set $X$ we choose $X = [-1, 1] \times [3, 5] \times [-5, -3] \times [0, 3]$ and we set $B = Y$. As a termination tolerance on the stationarity and on the activities of the constraints we choose $\epsilon = \epsilon_{act} = 10^{-3}$. For the feasibility of the indices we choose $\epsilon_Y = 0.035$. Furthermore we set $\epsilon_{split} = \min \left( 2\epsilon_{act} \min \left( 1, \frac{1}{\alpha \|b-b^0\|_2} \right), \frac{1}{2} - \epsilon_Y \right)$. We have to mention that we do not need a reduced outer approximation for the set $Y$. It can be handled exactly by the algorithm.

As a starting point we used the same point as in Chapter 3, that is, $x^0 = (1, 5, -3, 2)$. After 1 iteration (0.88 CPU seconds) we obtain the feasible starting point

\[ x^1 = (0.6336, 3.0164, -4.7823, 2.5498)^T. \]

After 7 more iterations (22.9 CPU seconds) the algorithm terminates with the point

\[ x^* = (-0.028005, 4, -4, 0.028005)^T \]

with the objective value 0.028005. The error functions are illustrated in Figure 4.2 and the approximating function in Figure 4.3. As termination criteria we used the norm of the stationarity condition, and, additionally, we required that the change in the value of the objective function is less then $10^{-3}$ and the norm of the change in the iterates is less then $10^{-2}$. At $x^*$ the norm of the stationarity condition is $1.4296 \cdot 10^{-15}$, the change in
the value of the objective function is $6.7847 \cdot 10^{-6}$, and the norm of the change in the iterates is 0.000118.

As desired, the algorithm has a better numerical performance than the algorithm presented in Chapter 3. At first we look at the overall number of boxes generated during the iterations. It is displayed in Figure 4.4.

The number of generated boxes increases only linearly, and in the last iteration the total amount of boxes is 18. Notice here that each generated box corresponds to an increase in the number of variables by 3, in the nonlinear inequality constraints by 1, and in the nonlinear equality constraints by 3. Thus, in the last iteration a problem with 58 variables, 18 nonlinear inequality and 54 nonlinear equality constraints must be solved.

The error made by the convexification procedure on a box $B_k$ is $\frac{\alpha_k}{\delta_k} \| \hat{b}^k - b^k \|_2^2$. Thus, it
4.4 A first numerical example

Figure 4.5: a) The maximum and minimum diameter of the boxes containing active indices generated by Algorithm 4 in each iteration. b) The black line is the maximum of the constants $\alpha$ on the boxes containing active indices and the red line is the error made by the convexification procedure in each iteration.

seems likely that this behavior can be traced back to the changes in the error. Figure 4.5 illustrates the latitude of boxes containing active indices and the maximum of the constant $\alpha$ in each iteration, as well as the error made by the convexification procedure on boxes containing active indices.

As in the adaptive reduction algorithm, the error decreases very fast during the first iterations, while it seems to get stuck in the later iterations. In the last two iterations the error is around $2.3387 \cdot 10^{-4}$ and $9.6327 \cdot 10^{-6}$. Thus, the error is essentially smaller than in the former algorithm.

In this section we do not give an example that illustrates the performance of the algorithms if the index set $Y$ is not box shaped, and, thus, reduced outer approximations are needed. We refer to Chapter 8 for that, since, it will reveal some other problems, especially in a phase 1 algorithm. In the next chapter we will discuss adaptation strategies of the set $X$ for both algorithms, the adaptive reduction and the adaptive convexification algorithm. With the additional adaptation the values for $L, L$ and, respectively, $\alpha$ should be further reduced. Thus, the error by relaxing the constraints should become, additionally, smaller.
5 An $X$-adaptation method

In this chapter we discuss an $X$-adaptation procedure for the Adaptive-Reduction and the Adaptive-Convexification Algorithm.

The solution concept of the algorithms presented in the former chapters is that of stationary points and not that of global optimizers. As we can only expect to find local minimizers for a problem $SIP$, we do not need the global set $X$, but, we can restrict the optimization process to some subset of $X$ that contains a stationary point in its interior. Since the set $X$ affects the size of the constants $L, \bar{L}$ and $\alpha$ the maximum separation distance of the original and the relaxed constraints should be reduced for some subset of $X$. However, as one does not have such a subset for an arbitrary problem in general, one can try to choose some subset of $X$ and move it until a stationary point is in its interior.

In the sequel, in view of the later algorithms, we will denote a subset of $X$ by $X_\nu$ for some $\nu \in \mathbb{N}$. Further, let $Y \subset B \subset \mathbb{R}^m$ and let $B_N = (B_k, k = 1, \ldots, N)$ be a reduced outer approximation of $Y$.

5.1 The Adaptive-Reduction Algorithm with $X$-adaptation

5.1.1 Relaxation and reformulation

We recall that from the formulas (3.1) and (3.2) we have that $L^{\nu,k}, \bar{L}^{\nu,k} : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^m$ are defined by

$\begin{align*}
L(X^{\nu}, B^k) := & \begin{pmatrix}
\min(0, \xi_1(X^{\nu}, B^k)) \\
\vdots \\
\min(0, \xi_m(X^{\nu}, B^k))
\end{pmatrix}
\text{ and } \\
\bar{L}(X^{\nu}, B^k) := & \begin{pmatrix}
\max(0, \zeta_1(X^{\nu}, B^k)) \\
\vdots \\
\max(0, \zeta_m(X^{\nu}, B^k))
\end{pmatrix}
\end{align*}$

with

$\begin{align*}
\xi_i(X^{\nu}, B^k) < & \min_{(x,y) \in X^{\nu} \times B^k} \left( \frac{\partial}{\partial y_i} g(y) \right) \\
\zeta_i(X^{\nu}, B^k) > & \max_{(x,y) \in X^{\nu} \times B^k} \left( \frac{\partial}{\partial y_i} g(y) \right).
\end{align*}$

We will use $L^{\nu,k}$ and $\bar{L}^{\nu,k}$ for $L(X^{\nu}, B^k)$, respectively $\bar{L}(X^{\nu}, B^k)$, to keep the notation short. Let $\nu, k \in \mathbb{R}^m$ with $c^{\nu,k}_j := \frac{L^{\nu,k}_j b_j - \bar{L}^{\nu,k}_j b_j}{L^{\nu,k}_j - \bar{L}^{\nu,k}_j}$ for each $j = 1, \ldots, m$. Define an unimodal
overestimator (w.r.t. \( y \)) of the restriction \( g \) on \( X^\nu \times B^k \) by

\[
\hat{g}^{\nu,k} : X^\nu \times B^k \to \mathbb{R}, \quad (x, y) \mapsto g(x, y) + \phi\left(y; c^{\nu,k}, B^k, L^{\nu,k}, \mu^{\nu,k}\right).
\]

Due to

\[
\min_{(x,y) \in X \times B} \left( \frac{\partial}{\partial y_1} g(y) \right) \leq \min_{(x,y) \in X^\nu \times B^k} \left( \frac{\partial}{\partial y_1} g(y) \right)
\]

and

\[
\max_{(x,y) \in X \times B} \left( \frac{\partial}{\partial y_1} g(y) \right) \geq \max_{(x,y) \in X^\nu \times B^k} \left( \frac{\partial}{\partial y_1} g(y) \right),
\]

both, \( L^{\nu,k} \) and \( \overline{L}^{\nu,k} \) can always be chosen so that \( L(X, B) \leq L^{\nu,k} \) and \( \overline{L}(X, B) \geq \overline{L}^{\nu,k} \) hold for each \( k = 1, \ldots, N \). The next results are immediate consequences of Lemma 3.1.2.

**Lemma 5.1.1.**

(i) For all \( k = 1, \ldots, N \) and \( (x, y) \in X^\nu \times B^k \) the relation \( g(x, y) \leq \hat{g}^{\nu,k}(x, y) \) holds.

(ii) For \( x \in X^\nu \) and \( k = 1, \ldots, N \) the relaxation \( \hat{g}^{\nu,k}(x, y) \) is unimodal in the second argument on \( B^k \).

(iii) For all \( k = 1, \ldots, N \) the maximum separation distance between \( \hat{g}^{\nu,k} \) and \( g \) on \( X^\nu \times B^k \) is \( \langle L^{\nu,k}, b^k - c^k \rangle = \langle L^{\nu,k}, b^k - c^k \rangle \).

(iv) For each \( x \in X^\nu \) the point \( c^{\nu,k} \) is the solution of \( \max_{y \in B^k} \hat{g}^{\nu,k}(x,y) \) for each \( k = 1, \ldots, N \).

(v) If \( x^* \in X^\nu \) and \( \hat{g}^{\nu,k}(x^*, c^{\nu,k}) \leq 0 \) holds for all \( k = 1, \ldots, N \), then we have \( x^* \in M \).

Before we continue our discussion we give an example to illustrate the impact of the \( X \)-adaptation on the constants \( L, \overline{L} \) and the maximum separation distance.

**Example 5.1.2.** Let \( B = [-1, 1] \), \( X = [-2, 2] \) and \( g(x, y) = x^3 y \). As \( \nabla_y g(x, y) = x^3 \) we have \( L(B) = -8, \overline{L}(B) = 8 \) and \( c = 0 \). Thus, we obtain

\[
\max_{y \in B} d_{UF}\left(y; c, B, L(B), \overline{L}(B)\right) = 16.
\]

For \( X^1 = [-1, 1] \) we obtain \( L\left(X^1, B\right) = -1, \overline{L}\left(X^1, B\right) = 1 \) and \( c = 0 \). Thus, we have

\[
\max_{y \in B} d_{UF}\left(y; c, B, L\left(X^1, B\right), \overline{L}\left(X^1, B\right)\right) = 2.
\]

In this example the impact of the set \( X \) on the constants \( L, \overline{L} \) is vast. Thus, by switching to \( X^1 \) the values of these constants and the maximum separation distance of the original and the relaxed function could be reduced enormously.
Define the set
\[ M_{UF}(X^\nu, B^N) := \{ x \in X^\nu | \hat{g}^{\nu,k}(x, y) \leq 0 \text{ for all } y \in B^k, k = 1, \ldots, N \} \]
and the semi-infinite problem
\[ SIP_{UF}(X^\nu, B^N) : \min_{x \in \mathbb{R}^n} f(x) \quad s.t. \quad x \in M_{UF}(X^\nu, B^N). \]

The straightforward proofs of the following results are omitted.

**Lemma 5.1.3.**

(i) We have \( M_{UF}(X^\nu, B^N) \subset M \).

(ii) Let \( SIP_{UF}(X^\nu, B^N) \) be consistent. Then every global, local solution or stationary point of \( SIP_{UF}(X^\nu, B^N) \) is a feasible point of \( SIP \).

From Lemma 5.1.1 (iv) we have that for each \( k = 1, \ldots, N \) it holds \( \hat{g}^{\nu,k}(x, y) \leq 0 \) for all \( y \in B^k \) if and only if \( \hat{g}^{\nu,k}(x, c^{\nu,k}) \leq 0 \) is true. Thus we obtain
\[ M_{UF}(X^\nu, B^N) = \{ x \in X^\nu | \hat{g}^{\nu,k}(x, c^{\nu,k}) \leq 0 \text{ for all } k = 1, \ldots, N \}. \]

Define
\[ G_{UF}(x) := (\hat{g}^{\nu,k}(x, c^{\nu,k}))_{k=1,\ldots,N}. \]

Now \( SIP_{UF}(X^\nu, B^N) \) can be equivalently reformulated as the nonlinear problem
\[ P_{UF}(X^\nu, B^N) : \min_{x} f(x) \quad s.t. \quad G_{UF}(x) \leq 0 \quad x \in X^\nu. \]

This problem may be tackled by different solvers. We briefly discuss that in Chapter 7, Section 7.1.

### 5.1.2 Algorithms and X-Adaptation

Similar to the ideas presented in Chapter 3 the algorithm discussed in this section computes a stationary point of \( SIP_{UF}(X^\nu, B^N) \) with active indices and terminates if it is also a stationary point of \( SIP \) within a given tolerance on stationarity and additional tolerances on the feasibility of the indices. If the computed point is not approximately stationary or the active indices are not approximately feasible, the reduced outer approximation \( B^N \) of \( Y \) is refined, a new set \( X^{\nu+1} \subset X \) is constructed and a respective refined \( SIP_{UF} \) is solved. One has to notice that a stationary point of \( SIP_{UF}(X^\nu, B^N) \) may be an element of the topological boundary of \( X^\nu \), since the entire feasible set
5 An X-adaptation method

\(M_{UF}(X^\nu, B^N)\) is not necessarily in the interior of \(X^\nu\). Thus, the stationary conditions for \(SIP_{UF}(X^\nu, B^N)\) are slightly different.

For a point \(\bar{x} \in M_{UF}(X^\nu, B^N)\) and points \(c^{\nu,k} \in B^k\), \(k = 1, \ldots, N\), define \(K^{\nu,k}_{\bar{x}}(\bar{x}) := \{k \in \{1, \ldots, N\} | \tilde{g}^{\nu,k}(\bar{x}, c^{\nu,k}) = 0\}\). As in Theorem 2.1.1, a point \(\bar{x} \in M_{UF}(X^\nu, B^N)\) is stationary for \(SIP_{UF}(X^\nu, B^N)\) in the sense of John, if there exist an \((n+1)\)-tuple \(J\) with \(\{J\} \subset K^{\nu,k}_{\bar{x}}(\bar{x})\), and \((\kappa, (\lambda_k)_{k \in J}, (\mu_l)_{l \in \{1, \ldots, n\}}, (\xi_r)_{r \in \{1, \ldots, n\}}) \in \sigma^{3n+1}\) with only \(n+1\) non-vanishing entries, so that

\[
\kappa \nabla f(\bar{x}) + \sum_{k \in J} \lambda_k \nabla g(\bar{x}, c^{\nu,k}) - \sum_{l=1}^{n} \mu_l e_l + \sum_{r=1}^{n} \xi_r e_r = 0
\]
\[
\lambda_k \tilde{g}^{\nu,k}(\bar{x}, c^{\nu,k}) = 0, \quad k \in J
\]
\[
\mu_l (\bar{x}^\nu - \bar{x}) = 0, \quad l \in \{1, \ldots, n\}
\]
\[
\xi_r (\bar{x} - \bar{x}^r) = 0, \quad r \in \{1, \ldots, n\},
\]

where \(e_i\) denotes the \(i\)-th unit vector. Notice here that \(\nabla g^{\nu,k}(\bar{x}, c^{\nu,k}) = \nabla g(\bar{x}, c^{\nu,k})\).

If we assume that \(\bar{x} \in \text{int}(X^\nu)\) holds, the former stationary conditions reduce to the condition given in Chapter 3. In the sequel we will call a point \(c^{\nu,k} \in B^k\) with \(k \in K^{\nu,k}_{0,\bar{x}}(\bar{x})\) an active index of \(SIP_{UF}(X^\nu, B^N)\).

Using the splitting function from Chapter 3 and the techniques introduced there, a splitting algorithm is stated in Algorithm 5. It also describes how the set \(M_{UF}(X^\nu, B^N)\) of a refined problem \(SIP_{ UF}(X^\nu, B^N)\) is constructed.

As for Algorithm 1, it is not hard to see that each \(N\)-tuple \((B^k, k = 1, \ldots, N)\) generated by Algorithm 5 is a reduced outer approximation of \(Y\).

Now we discuss the adaptation of the set \(X\). Lemma 3.3.3 implies that for some reduced outer approximation \(B^N\) of \(Y\) we have \(M_{UF}(X, B^N) \subset M_{UF}(X, B^N^*)\) with a refined outer approximation \(B^{N*}\). But for some sets \(X^\nu, X^{\nu+1} \subset X\), as Example 5.1.5 will also show, we can not ensure such a property for \(M_{UF}(X^\nu, B^N)\) and \(M_{UF}(X^{\nu+1}, B^N^*)\).

However, in the later convergence proof we will only need that for a special point of interest \(x^* \in M_{UF}(X^\nu, B^N)\), for example a solution of \(SIP_{UF}(X^\nu, B^N)\), we can ensure that there is some reduced outer approximation \(B^{N*}\) and a set \(X^{\nu+1}\) so that \(x^* \in M_{UF}(X^{\nu+1}, B^{N*})\) holds. The next lemma gives a first impression that finding such a subset \(X^{\nu+1}\) may be a hard task.

**Lemma 5.1.4.** Let \(X^1, X^2 \subset X\) so that \(X^1 \cap X^2 \neq \emptyset\), \(B^N\) be some partition of \(B\), and let \(L_1^k, \bar{L}_1^k, L_2^k, \bar{L}_2^k\) be Lipschitz parameters on \(X^1 \times B^k\), \(X^2 \times B^k\) for some box \(B^k \in B^N\) and a constraint \(g\). Let \(c^{1,k}, c^{2,k}\) be the corresponding centers. If there is some \(x^* \in X^1 \cap X^2\) with \(\bar{g}^{1,k}(x^*, y) \leq 0 \) for each \(y \in B^k\) and \(g(x^*, c^{2,k}) \leq g(x^*, c^{1,k})\), then \(L_1^k \bar{L}_2^k \leq \bar{L}_1^k L_2^k \) and \(\bar{L}_2^k \leq \bar{L}_1^k\), \(j \in \{1, \ldots, m\}\), implies that \(\bar{g}^{2,k}(x^*, y) \leq 0\) holds for each \(y \in B^k\).

**Proof.** We have \(\bar{g}^{2,k}(x^*, y) \leq 0\) for each \(y \in B^k\) if and only if \(\bar{g}^{2,k}(x^*, c^{2,k}) \leq 0\) is true.
Algorithm 5 Splitting step - \textit{Xrefine}_{UF}(\eta)

Let \( \eta \in B^k \), \( k^* \in \{1, \ldots, N\} \), and let \( S^k \) be the barycenter of \( B^k \) and

\[ Q^k = \min_{l \in B^k} \left( \frac{b_l - g_l^k}{\| b_l - g_l^k \|} \right). \]

if \( \eta \notin Y \) or \( Q^k < \epsilon \) then

Set \( \eta = S^k \).

end if

if \( P^k \neq \emptyset \) then

Compute \((B^{k^*},(1), B^{k^*},(2))\) = \( S(B^k, \eta) \).

Compute \( L_{\nu,k^*},(1), L_{\nu,k^*},(2) \geq \nu, k^* \) and \( \tilde{L}^{\nu,k^*},(1), \tilde{L}^{\nu,k^*},(2) \leq \nu, k^* \) on \( X^\nu \times B^k \), \( X^\nu \times B^k \) and set

\[ c^{\nu,k^*},(1) = \left( \frac{L_{\nu,k^*},(1)}{L_{\nu,k^*},(1)} b_1 - \tilde{L}^{\nu,k^*},(1) b_1}{\tilde{L}^{\nu,k^*},(1)} \right) \]

\[ c^{\nu,k^*},(2) = \left( \frac{L_{\nu,k^*},(2)}{L_{\nu,k^*},(2)} b_2 - \tilde{L}^{\nu,k^*},(2) b_2}{\tilde{L}^{\nu,k^*},(2)} \right) \]

\[ \hat{g}^{\nu,k^*},(1)(x, y) = g(x, y) + \phi \left( y; c^{\nu,k^*},(1), B^{k^*},(1), L_{\nu,k^*},(1), \tilde{L}^{\nu,k^*},(1) \right) \]

\[ \hat{g}^{\nu,k^*},(2)(x, y) = g(x, y) + \phi \left( y; c^{\nu,k^*},(2), B^{k^*},(2), L_{\nu,k^*},(2), \tilde{L}^{\nu,k^*},(2) \right) \]

\[ M^{(1)} = \{ x \in X^\nu | \hat{g}^{\nu,k^*},(1)(x, c^{\nu,k^*},(1)) \leq 0 \} \]

\[ M^{(2)} = \{ x \in X^\nu | \hat{g}^{\nu,k^*},(2)(x, c^{\nu,k^*},(2)) \leq 0 \} \]

\[ \hat{M} = \{ x \in X^\nu | \hat{g}^{\nu,k^*}(x, c^k) \leq 0 \} \text{ for all } k \in \{1, \ldots, N\} \setminus \{k^*\} \]

if \( Y \cap B^k,(1) \neq \emptyset \) and \( Y \cap B^k,(2) \neq \emptyset \) then

Replace \( B^k \) by \( B^k,(1) \) and \( B^k,(2) \) and replace \( L_{\nu,k^*}, \tilde{L}^{\nu,k^*} \) by \( L_{\nu,k^*},(1), L_{\nu,k^*},(2) \)

and \( \tilde{L}^{\nu,k^*},(1), \tilde{L}^{\nu,k^*},(2) \).

Set \( M_{UF}(X^\nu, B^N) = \hat{M} \cap M^{(1)} \cap M^{(2)} \).

Set \( N = N + 1 \).

else if \( Y \cap B^k,(1) \neq \emptyset \) then

Replace \( B^k \) by \( B^k,(1) \) and \( L_{\nu,k^*}, \tilde{L}^{\nu,k^*} \) by \( L_{\nu,k^*},(1) \) and \( \tilde{L}^{\nu,k^*},(1) \).

Set \( M_{UF}(X^\nu, B^N) = \hat{M} \cap M^{(1)} \).

else if \( Y \cap B^k,(2) \neq \emptyset \) then

Replace \( B^k \) by \( B^k,(2) \) and \( L_{\nu,k^*}, \tilde{L}^{\nu,k^*} \) by \( L_{\nu,k^*},(2) \) and \( \tilde{L}^{\nu,k^*},(2) \).

Set \( M_{UF}(X^\nu, B^N) = \hat{M} \cap M^{(2)} \).

else

Delete \( B^k \) and \( L_{\nu,k^*}, \tilde{L}^{\nu,k^*} \).

Set \( M_{UF}(X^\nu, B^N) = \hat{M} \).

Set \( N = N - 1 \).

end if

end if

\[ \frac{\eta}{\nu} \]

55
5 An X-adaptation method

For each \( j \in \{1, \ldots, m\} \) the condition \( L_{ij}^{1,k} T_{ij}^{2,k} \leq L_j^{1,k} L_j^{2,k} \) is equivalent to \( \frac{L_{ij}^{1,k} (b_j^k - b_j^k)}{T_j^{2,k} - L_j^{2,k}} \leq \frac{L_j^{1,k} (b_j^k - b_j^k)}{T_j^{1,k} - L_j^{1,k}} \). As \( L_j^{2,k} \leq L_j^{1,k} \) and \( g(x^*, c_j^{2,k}) \leq g(x^*, c_j^{1,k}) \) we obtain

\[
\hat{g}(x^*, c_j^{2,k}) = g(x^*, c_j^{2,k}) + \left\langle L_j^{2,k}, b_k^j - c_j^{2,k} \right\rangle \\
= g(x^*, c_j^{2,k}) + \left\langle L_j^{2,k}, \left( \frac{L_{ij}^{1,k} (b_j^k - b_j^k)}{T_j^{2,k} - L_j^{2,k}} \right) \right\rangle_{j=1,\ldots,m} \\
\leq g(x^*, c_j^{1,k}) + \left\langle L_j^{1,k}, \left( \frac{L_{ij}^{1,k} (b_j^k - b_j^k)}{T_j^{1,k} - L_j^{1,k}} \right) \right\rangle_{j=1,\ldots,m} \\
\leq g(x^*, c_j^{1,k}) + \left\langle L_j^{1,k}, \left( \frac{L_j^{1,k} (b_j^k - b_j^k)}{T_j^{1,k} - L_j^{1,k}} \right) \right\rangle_{j=1,\ldots,m} \\
\leq 0.
\]

Lemma 5.1.4 says that \( x^* \) stays feasible, while switching from \( X^1 \) to \( X^2 \), if the corresponding values for the constraints in the computed centers \( c_j^{1,k}, c_j^{2,k} \) and the parameters \( L_j^{1,k}, L_j^{2,k}, i = 1, 2 \), are non-increasing. But, as the next example shows, that cannot be ensured in general, even if the diameter if \( X^1 \) is larger then the diameter of \( X^2 \).

**Example 5.1.5.** Let \( X^1 = [-2, 2], X^2 = [0, 3], Y = [0, 2] \) and let \( M = \{ x \in \mathbb{R} | g(x, y) = \langle x^2 - x \rangle y - 4 \leq 0 \text{ for all } y \in Y \} \). It is not hard to see that 0 \( \notin \) M. Since \( \nabla_y \hat{g}(x, y) = x^2 - x \), we obtain \( L = -2, \bar{L} = 6 \) and \( c = 2 \) on \( X^1 \times Y \). On \( X^2 \times Y \) we obtain \( L = -3, \bar{L} = 9 \) and \( c = 2 \). Thus, the relaxed constraints on \( X^1 \times Y \) and \( X^2 \times Y \) result in

\[
\hat{g}^1(x, y) = (x^2 - x) y - 4 + 2y,
\]

respectively

\[
\hat{g}^2(x, y) = (x^2 - x) y - 4 + 3y.
\]

For \( x^* = 0 \) we have

\[
\hat{g}^1(x^*, y) = -4 + 2y,
\]

and

\[
\hat{g}^2(x^*, y) = -4 + 3y.
\]

Thus, we have \( \hat{g}^1(x^*, y) \leq 0 \) for all \( y \in Y \), while \( \hat{g}^2(x^*, y) > 0 \). Even though the diameter of \( X^1 \) is larger then the diameter of \( X^2 \), \( x^* \) is a feasible point of the set \( M_{UP}(X^1, Y) \), while it is not a feasible point of \( M_{UP}(X^2, Y) \). Thus, a given point of interest, feasible for some choice of a subset of \( X \), may not be feasible for another choice of a subset.
Lemma 5.1.6. Let $B_S$ be some box in $B$ that for each $k$ is not in $M$. We show that there is a partition $B_S$ such that each $B_S$ is related to the diameter of the set $S$. By splitting $Y$ into smaller sets one may reduce that problem.

Let $B^1 = [0, 1], B^2 = [1, 2]$ and $B = (B^1, B^2)$. On $X^2 \times B^1$ we have $c = 1$ and, thus, $\hat{g}^2 (x^*, y) = -4 + 3y$. On $X^2 \times B^2$ we have $c = \frac{7}{4}$ and, thus, $\hat{g}^2(x^*, y) = -4 - 3(1 - y)$ for $y < c$, and $\hat{g}^2(x^*, y) = -4 + 9(2 - y)$ for $y \geq c$. It is not hard to see that $B_S^2$ is a tessellation of $Y$ and $\hat{g}^2 (x^*, y) \leq 0$ for each $y \in B^i$.

Thus, $x^*$ is a feasible point of $M_{UF}(X^2, B^2)$.

The next lemma generalizes the observations made in Example 5.1.5. That is, if a point of interest is not in $M_{UF}(X^v, B^N)$ for some set $X^v$ and a tessellation $B^N$, then there is a refined tessellation $B^{N*}$ so that this point is in $M_{UF}(X^v, B^{N*})$. For the proof of this lemma we first extend the splitting operator introduced in Section 3.2. Let $S(B)$ denote the barycenter of a box $B$. For $\rho \geq 1$ define the $\rho$-times splitting operator $S^\rho$ of a box $B$ by

$$S^\rho (B, S(B)) := (S^{\rho - 1} (B^1, S(B^1)), S^{\rho - 1} (B^2, S(B^2)))$$

with $S^1 (B, S(B)) = S(B, S(B))$. Thus, $S^\rho (B, S(B))$ generates a tessellation of $B$ with $2^\rho$ boxes.

Lemma 5.1.6. Let $X^1, X^2 \subset X$, $x^* \in X^1 \cap X^2$ and $B^N$ be some partition of $B$. Let there be some box $B^k \in B^N$ and Lipschitz parameters $L_{1,k}^1, L_{1,k}^2 \neq 0$ and $L_{2,k}^1, L_{2,k}^2 \neq 0$ for a constraint $g$ on $X^1 \times B^k, X^2 \times B^k$ so that $\hat{g}^{1,k}(x^*, y) \leq 0$ for each $y \in B^k$, and let there be some $y^* \in B^k$ so that $\hat{g}^{1,k}(x^*, y^*) > 0$ is satisfied. There is a partition $B^{N*}$ of $B$ so that for each $k^*$ with $B^{k^*} \in B^{N*}$ and $B^{k^*} \subset B^k$, and corresponding Lipschitz parameters $L_{2,k^*}^1, L_{2,k^*}^2$ on $X^2 \times B^{k^*}$, the relaxation $\hat{g}^{2,k^*}$ is unimodal in the second argument and $\hat{g}^{2,k^*}(x^*, y) \leq 0$ for each $y \in B^{k^*}$.

Proof. By construction it is clear that $\hat{g}^{2,k^*}$ is unimodal in the second argument. If we show that there is a partition $B^{N*}$ of $B$ so that for each $k^*$ with $B^{k^*} \in B^{N*}$ and $B^{k^*} \subset B^k$ the inequality $\hat{g}^{2,k^*}(x^*, y) \leq \hat{g}^{1,k}(x^*, y)$ holds for each $y \in B^{k^*}$, the assertion of the lemma follows immediately. Let $L$ be the Lipschitz parameter for $g$ on $X \times B$ and let $L = (L, \ldots, L) \in \mathbb{R}^m$. The parameters $L_{2,k^*}^2$ and $L_{2,k^*}^2$ can always be chosen so that $-L < L_{2,k^*}^2$ and $L > L_{2,k^*}^2$ for each $k^*$ and $X^2 \subset X$. Thus, it is sufficient to show that there is a partition $B^{N*}$ of $B$ so that for each $k^*$ with $B^{k^*} \in B^{N*}$ and $B^{k^*} \subset B^k$ the inequality

$$\phi \left( y; e^{2,k^*}, B^{k^*}, -L, L \right) \leq \phi \left( y; e^{1,k}, B^k, L_{1,k}, L_{1,k}^1 \right)$$

holds for each $y \in B^{k^*}$ and each $k^*$.

Let $\rho \geq 1$ and let $B^{N*}$ arise from $B^N$ by applying $S^\rho$ to the box $B^k$ in $B^N$. Let $B^{k^*} \subset B^k$ be a box in $B^{N*}$ and $y \in B^{k^*}$. Let $\phi \left( y; e^{1,k^*}, B^{k^*}, L_{1,k}, L_{1,k} \right) = 0$ hold for that $y$. As
Let \( c > 0 \) with 
\[
\phi \left( y; c, B^k, L_1^k, T_1^k \right) \geq \| L \|_2 c.
\]
It is not hard to see that the relation 
\[
\phi \left( y; c, B^k, -L, L \right) \leq \| L \|_2 \| b^k - b^k \|_2 \quad \text{holds.}
\]
From Lemma 3.3.4 we obtain that the inequality 
\[
\| b^k - b^k \|_2 \leq \left( 1 - \frac{3}{4m} \right)^\frac{1}{2} \| b^k - b^k \|_2 \quad \text{holds.}
\]
Since \( 0 < 1 - \frac{3}{4m} < 1 \) for each \( m \), we have that for each \( c > 0 \) there is some \( \rho \) so that 
\[
\left( 1 - \frac{3}{4m} \right)^\frac{1}{2} \| b^k - b^k \|_2 \leq c.
\]
Altogether we have that there is some \( \rho \) so that 
\[
\phi \left( y; c, B^k, -L, L \right) \leq \| L \|_2 \| b^k - b^k \|_2 \leq \| L \|_2 \left( 1 - \frac{3}{4m} \right)^\frac{1}{2} \| b^k - b^k \|_2 \leq \phi \left( y; c, B^k, L_1^k, T_1^k \right)
\]
holds. \( \square \)

The way a subset of \( X \) is generated will first be characterized by abstract properties needed within the later convergence proof of our algorithm. Later we will give two examples for adaptation strategies possessing these features.

**Condition 5.1.7.** Given a point \( x \) and a set \( X^\nu \subset X \), with \( x \in X^\nu \), the new \( X^\nu+1 \) is generated so that the following properties hold.

(i) \( x \in \text{int} \left( X^{\nu+1} \right) \).

(ii) If \( x \in \partial X^\nu \) then there is some constant \( c_1 > 1 \) so that for each \( \nu \) the relation 
\[
\| x^{\nu+1} - x^{\nu+1} \|_2 \geq c_1 \| x^\nu - x^\nu \|_2 \quad \text{holds.}
\]

(iii) There is some \( c_2 > 0 \) so that \( \| x^{\nu+1} - x^{\nu+1} \|_2 > c_2 \).

(iv) \( X^{\nu+1} \subset X \).

Using Lemma 5.1.6 and Condition 5.1.7 the \( X \)-adaptation algorithm is stated in Algorithm 6.

With Lemma 5.1.6 we have that Algorithm 6 is well defined. That is, after finitely many splitting steps the set \( \mathcal{J} \) is empty. Using the splitting and the \( X \)-adaptation algorithm, the adaptive reduction algorithm with \( X \)-adaptation is stated in Algorithm 7.

If for a given reduced outer approximation \( B^N \), the problem \( SIP_{UF} \left( X^0, B^N \right) \) is not consistent in Algorithm 7, a phase 1 algorithm can be performed.

**5.1.3 Convergence results**

At first we show that Algorithm 7 is well defined. In fact, if for some iterate \( x^\nu \in \text{int} \left( X^\nu \right) \), the set \( P^k \) is empty at each active index \( e^k \) of an approximating problem
5.1 The Adaptive-Reduction Algorithm with X-adaptation

Algorithm 6 X-Adaptation - Xadapt$_{UF}(x, \Delta x)$

Let $x \in X^\nu$, and $B^N$ a reduced outer approximation of $Y$.

Generate a set $X^{\nu+1} \subset X$ so that Condition 5.1.7 is satisfied.

Generate new parameters $L^{\nu+1,k}, L^{\nu+1,k}$ and $c^{\nu+1,k}$ on $X^{\nu+1} \times B^k$ for each $k \in \{1, \ldots, N\}$ and replace $L^{\nu,k}, L^{\nu,k}$ by $L^{\nu+1,k}, L^{\nu+1,k}$ as well as $c^{\nu,k}$ by $c^{\nu+1,k}$. For each $k \in \{1, \ldots, N\}$ set

$$\hat{g}^{\nu+1,k}(x,y) = g(x,y) + \phi(y; c^{\nu+1,k}, B^k, L^{\nu+1,k}, L^{\nu+1,k})$$

Set $J = \{k \in \{1, \ldots, N\} | \hat{g}^{\nu+1,k}(x,y) > 0 \text{ for some } y \in B^k\}$.

while $J \neq \emptyset$ do

for $k \in J$ do

Let $S^k$ be the barycenter of $B^k$.

$Xrefine_{UF}(S^k)$

end for

Set $K = \{k \in \{1, \ldots, N\} | \hat{g}^{\nu+1,k}(x,y) > 0 \text{ for some } y \in B^k\}$.

end while

$SIP_{UF}(X^\nu, B^N)$, the reduced outer approximation of $Y$ is not refined any further. Thus, we are in the same situation as in Chapter 3. That is, Algorithm 7 might loop. However, the following lemma is a direct consequence of Lemma 3.3.2 and shows that the algorithm terminates in this case.

Lemma 5.1.8. Let $X^\nu \subset X$ and $\bar{x} \in \text{int}(X^\nu)$ be a stationary point of $SIP_{UF}(X^\nu, B^N)$ with $\epsilon_{act}$-active indices $c^k$, $k \in J^\nu$, for an $(n+1)$-tuple $J^\nu$, $\{J^\nu\} \subset K^{1,F}_{0,F}(\bar{x})$. Let $(\kappa, (\lambda_i)_{i \in \{1, \ldots, n\}}, (\zeta_r)_{r \in \{1, \ldots, n\}})$ be the corresponding multipliers, and let the refined reduced outer approximation arising from applying Algorithm 5 to each $c^k$, $k \in J^\nu$, coincide with $B^N$. Then Algorithm 7 terminates at $\bar{x}$.

The next lemma is an implication of Lemma 3.3.3. It means that refining a given reduced outer approximation $B^N$ of $Y$ by means of Algorithm 5, using the splitting function, enlarges the feasible set of $SIP_{UF}(X^\nu, B^N)$.

Lemma 5.1.9. Let a reduced outer approximation $B^N$ of $Y$ be given, and for a splitting point $c^k \in B^k$ let $B^N$ be the refinement of $B^N$ by means of Algorithm 5. Then $M_{UF}(X^\nu, B^N) \subset M_{UF}(X^\nu, B^N)$ holds.

Since Algorithm 5 is used in Algorithm 6, Lemma 5.1.9 together with Condition 5.1.7 (i) implies that for a point $x \in M_{UF}(X^\nu, B^N)$ we also have $x \in M_{UF}(X^{\nu+1}, B^N)$. Here, $B^N$ denotes a refinement of a reduced outer approximation $B^N$ by means of Algorithm 5.

Now we state the main convergence result.
Algorithm 7 Adaptive reduction algorithm - Xara

Choose $X \subset \mathbb{R}^n$ with $M \subset X$, $B = \{b, \bar{b}\} \subset \mathbb{R}^m$ with $Y \subset B$ and compute $L, \overline{L}$ on $X \times B$. Set $\nu = 0$, $X^0 = X$ and $L = \max(\|L\|_{\infty}, \|\overline{L}\|_{\infty}) \in \mathbb{R}$.

Determine a reduced outer approximation $B^N$ of $Y$ with some $N \in \mathbb{N}$ as well as $L^{0,k} \leq \overline{L}, \overline{L}^{0,k} \geq \overline{L}$ and $c^{0,k}$ on $X^0 \times B^k$, $k = 1, \ldots, N$, so that $SIP_{UF}(X^0, B^N)$ is consistent.

Choose $\epsilon_{act}, \epsilon_{stat}, \epsilon_Y > 0$ and $\epsilon_{split} \in (0, \frac{1}{2})$ with $\epsilon_{split} \leq \epsilon_{act}/(L\|\bar{b} - b\|_1)$.

Compute a stationary point $x^0$ of $SIP_{UF}(X^0, B^N)$ with $\epsilon_{act}$-active indices $c^{0,k}$, $k \in J^0$ for some $(n+1)$-tuple $J^0$ with $\{J^0\} \subset K_{0,0}^{UF}(x^0)$, and $3n+1$ multipliers $(\kappa, (\lambda_k)_{k \in \nu}, (\mu_l)_{l \in \{1, \ldots, m\}}, (\xi_r)_{r \in \{1, \ldots, m\}})$ by solving $P_{UF}(X^0, B^N)$ so that $n+1$ multipliers are non-vanishing.

**while** $x^\nu$ is not a stationary point of $SIP$ with $2\epsilon_{act}$-active, $\epsilon_Y$-feasible indices $c^{\nu,k}$, $k \in J^\nu$, with respect to $B^N$, and multipliers $(\kappa, \lambda_k \in J^\nu)$ do

**for** $k \in J^\nu$ do

$x_{\text{refine}_{UF}}(c^k)$

end **for**

**if** $\nu > 1$ then

Set $\Delta x^\nu = \|x^\nu - x^{\nu-1}\|_{\infty}$.

$x_{\text{adapt}_{UF}}(x^\nu, \Delta x^\nu)$

end **if**

**else**

Set $X^1 = X$.

end **if**

Set $\nu = \nu + 1$.

Compute a stationary point $x^\nu$ of $SIP_{UF}(X^\nu, B^N)$ with $\epsilon_{act}$-active indices $c^{\nu,k}$, $k \in J^\nu$ for some $(n+1)$-tuple $J^\nu$ with $\{J^\nu\} \subset K_{0,0}^{UF}(x^\nu)$, and $3n+1$ multipliers $(\kappa, (\lambda_k)_{k \in \nu}, (\mu_l)_{l \in \{1, \ldots, m\}}, (\xi_r)_{r \in \{1, \ldots, m\}})$ by solving $P_{UF}(X^\nu, B^N)$ so that $n+1$ multipliers are non-vanishing.

end while

**Theorem 5.1.10.** Algorithm 7 terminates after finitely many steps.

**Proof.** We give the proof by enforcing a contradiction. Assume that Algorithm 7 does not terminate. Then there exist sequences of reduced outer approximations $(B^{N_{\nu}})^\nu_Y$, $(X^\nu)^\nu_Y$, $(\lambda_k)^\nu_Y$, $(\mu_l)^\nu_Y$, $(\xi_r)^\nu_Y$, $(K_{0,0}^{UF}(X^\nu))^\nu_Y$, $(J^\nu)^\nu_Y$ with $(J^\nu) \subset K_{0,0}^{UF}(X^\nu)$, $(\{(c^{\nu,k})_{k \in J^\nu}\})^\nu_Y$ and multipliers $(\kappa^\nu, (\lambda_k^\nu)_{k \in J^\nu}, (\mu_l^\nu)_{l \in \{1, \ldots, n\}}, (\xi_r^\nu)_{r \in \{1, \ldots, n\}})^\nu_Y$ with only $n+1$ non-vanishing entries, so that for each $\nu$ the point $x^\nu$ is stationary for $SIP_{UF}(X^\nu, B^{N_{\nu}})$ with $\epsilon_{act}$-active indices $(c^{\nu,k})_{k \in J^\nu}$ and multipliers $(\kappa^\nu, (\lambda_k^\nu)_{k \in J^\nu}, (\mu_l^\nu)_{l \in \{1, \ldots, n\}}, (\xi_r^\nu)_{r \in \{1, \ldots, n\}})^\nu_Y$, while $x^\nu$ with all data is not $\epsilon_{stat}$-stationary for $SIP$ with $2\epsilon_{act}$-active, $\epsilon_Y$-feasible indices with respect to $B^{N_{\nu}}$. 

60
To derive a contradiction, in step 1. the existence of an accumulation point of the sequence \((x^\nu)\nu\) is shown. In step 2. and 3. it is shown that for infinitely many \(\nu\) the point \(x^\nu\) is an interior point of the box \(X^\nu\) and, thus, the multipliers \((\mu^\nu_l)_{l \in \{1,\ldots,n\}}\) and \((\xi^\nu_r)_{r \in \{1,\ldots,n\}}\) vanish for infinitely many \(\nu\). Then, in the steps 4. and 5. the existence of accumulation points of the sequences \(((c^k\nu^l)_{k \in J^\nu})\nu\) and \(((\kappa^\nu, \lambda^\nu_k, (\mu^\nu_l)_{l \in \{1,\ldots,n\}}, (\xi^\nu_r)_{r \in \{1,\ldots,n\}})\nu\) is shown. After that it is shown in steps 6. to 8. that there exists a \(\nu_0 \in \mathbb{N}\) so that \(x^\nu\) with all data is an \(\epsilon_{\text{stat}}\)-stationary point for SIP with \(2\epsilon_{\text{act}}\)-active, \(\epsilon_Y\)-feasible indices with respect to \(B^{N^\nu,\nu}\) for infinitely many \(\nu \geq \nu_0\).

1. **Assertion:** The sequence \((x^\nu)\nu\) has an accumulation point \(x^* \in M\).

Since \((x^\nu)\nu\) is contained in the compact set \(X\), it possesses an accumulation point \(x^* \in X\). For \(i \geq 0\) let \(M^i := \bigcup_{\nu = 1}^i M_{U^F}(X^\nu, B^{N^\nu,\nu})\). By Lemma 5.1.9 and Lemma 5.1.3 we have \(M^0 \subset M^1 \subset M^2 \subset \ldots \subset M\). Thus, we have that \((x^\nu)\nu\) is also contained in the closed set \(M\). From this the assertion immediately follows.

2. **Assertion:** There is some accumulation point \(x^* \in M\) of the sequence \((x^\nu)\nu\) so that for each \(\epsilon > 0\) there is some \(\nu_0 \in \mathbb{N}\) so that it holds \(\|x^* - x^\nu\|_\infty \leq \epsilon\) as well as \(x^\nu \in \text{int}(X^\nu)\) for infinitely many \(\nu \geq \nu_0\).

From step 1 we have that the sequence \((x^\nu)\nu\) has an accumulation point \(x^* \in M\). That means that for each \(\epsilon > 0\) there is some \(\nu_0 \in \mathbb{N}\) so that it holds \(\|x^* - x^\nu\|_\infty \leq \epsilon\).

By Condition 5.1.7 (i),(iii) we have that the interior of \(X^\nu\) is nonempty. Thus, we only have to distinguish two cases to show the assertion. The first case is that for infinitely many \(\nu \geq \nu_0\) we have \(x^\nu \in \text{int}(X^\nu)\). That is exactly the statement of the assertion. The second case is that there are not infinitely many \(\nu \geq \nu_0\) with \(x^\nu \in \text{int}(X^\nu)\). In this case we have that there is some \(\nu_1 \geq \nu_0\) so that \(x^\nu \in \partial X^\nu\) for each \(\nu \geq \nu_1\). Condition 5.1.7 (ii),(iv) and Assumption 1 imply that there is some \(\nu_2 \geq \nu_1\) so that \(\partial X^\nu \cap M = \emptyset\) for each \(\nu \geq \nu_2\). As \(x^\nu \in M_{U^F}(X^\nu, B^{N^\nu,\nu}) \subset M\) holds for each \(\nu \in \mathbb{N}\), we obtain a contradiction.

3. **Assertion:** There is some \(\nu_0 \in \mathbb{N}\) so that \((\mu^\nu_l)_{l \in \{1,\ldots,n\}} = 0\) and \((\xi^\nu_r)_{r \in \{1,\ldots,n\}} = 0\) for infinitely many \(\nu \geq \nu_0\).

For all \(\nu\) the point \(x^\nu\) with all data is stationary for \(SIP_{U^F}(X^\nu, B^N)\). Thus, we have that

\[
\begin{align*}
\mu_l (x^\nu_l - \bar{x}) &= 0, & l \in \{1,\ldots,n\} \\
\xi_r (\bar{x} - x^\nu_r) &= 0, & r \in \{1,\ldots,n\}
\end{align*}
\]

holds. By step 2. there is some \(\nu_0 \in \mathbb{N}\) so that \(x^\nu \in \text{int}(X^\nu)\) holds for infinitely many \(\nu \geq \nu_0\). Thus, we obtain the assertion.

4. **Assertion:** The sequence \(((c^k\nu^l)_{k \in J^\nu})\nu\) has an accumulation point \((c^k)_{k \in J^*} \in Y^{n+1}\).

Step 3. implies that there is some \(\nu_0 \in \mathbb{N}\) so that \((c^j\nu^l)_{k \in J^\nu} \in \mathbb{R}^{n+1}\) with pairwise different entries for infinitely many \(\nu \geq \nu_0\). Thus, by possibly switching to a

61
subsequence, it is sufficient to show the existence of an accumulation point in 
Y for every component of the sequence of the \((n + 1)\)-tuple. Let \((c^{k^\nu})_\nu\), \(k^\nu \in \{1, \ldots, n+1\}\), be a sequence of such components. As in the proof of Theorem 4.3.4 we will distinguish two cases. For the first case assume that for \(\nu \geq \nu_0\) we have \(c^{k^\nu} \in Y\) for infinitely many \(\nu \in \mathbb{N}\). By the compactness of \(Y\), the subsequence of these points possesses an accumulation point in \(Y\).

In the second case there is some \(\nu_1 \in \mathbb{N}\) with \(\nu_1 \geq \nu_0\) so that \(c^{k^\nu} \notin Y\) for all \(\nu \geq \nu_1\). Let \((B^{k^\nu})_\nu\) be the sequence of boxes with \(c^{k^\nu} \in B^{k^\nu}\).

After possibly switching to a subsequence we may assume that \((B^{k^\nu})_\nu\) satisfies \(B^{k^\nu} \supset B^{k^\nu+1}\) for \(\nu \in \mathbb{N}\), since for each \(\nu\) the reduced outer approximation \(B^{N^\nu}\) of \(Y\) only contains finitely many boxes. As Algorithm 5 subdivides each box \(B^{k^\nu}\) with \(c^{k^\nu} \notin Y\) at its barycenter, Lemma 3.3.4 implies that the maximum edge lengths of the \(B^{k^\nu}\), \(\nu \in \mathbb{N}\), tend to zero. Due to \(B^{N^\nu} \cap Y \neq \emptyset\), \(\nu \in \mathbb{N}\), we arrive at

\[
\lim_{\nu \to \infty} \min_{y \in Y} \|c^{k^\nu} - y\|_2 = 0
\]

and, thus, the assertion.

5. **Assertion:** The sequence \(\left(\kappa^\nu, (\lambda^\nu_k)_{k \in J^\nu}, (\mu^\nu_l)_{l \in \{1, \ldots, n\}}, (\xi^\nu_r)_{r \in \{1, \ldots, n\}}\right)_\nu\) has an accumulation point \((\kappa^*, \lambda^*, 0, 0) \in \sigma^{3n+1}\).

This is clear from the compactness of \(\sigma^{3n+1}\) and step 3.

6. **Assertion:** There exists a \(\nu_0 \in \mathbb{N}\) so that each \((c^{k^\nu})_\nu\) with \(k \in J^\nu\) is \(\epsilon_Y\)-feasible with respect to \(B^{N^\nu}\) for \(Y\) for all \(\nu \geq \nu_0\).

With Lemma 3.3.5 the assertion results as in the proof of Theorem 3.3.6, step 4.

7. **Assertion:** There is some \(\nu_0 \in \mathbb{N}\) so that for infinitely many \(\nu \geq \nu_0\) the point \(x^\nu\) is \(\epsilon_{\text{stat}}\)-stationary for SIP with multipliers \((\kappa^\nu, \lambda^\nu) \in \sigma^{n+1}\).

Since for all \(\nu\) the point \(x^\nu\) is stationary for SIP_{UF} \(\left(X^\nu, B^N\right)\) with \(\epsilon_{\text{act}}\)-active indices \((c^{k^\nu})_{k \in J^\nu}\) and multipliers \((\kappa^\nu, (\lambda^\nu_k)_{k \in J^\nu}, (\mu^\nu_l)_{l \in \{1, \ldots, n\}}, (\xi^\nu_r)_{r \in \{1, \ldots, n\}})\), parts 1.,3.,4., and 5. of this proof together with a continuity argument entail

\[
k^\nu \nabla f(x^\nu) + \sum_{k \in J^\nu} \lambda^\nu_k \nabla_x g(x^\nu, c^{k^\nu}) = 0.
\]

Hence there is some \(\nu_0 \in \mathbb{N}\) so that for infinitely many \(\nu \geq \nu_0\)

\[
\|k^\nu \nabla f(x^\nu) + \sum_{k \in J^\nu} \lambda^\nu_k \nabla_x g(x^\nu, c^{k^\nu})\| \leq \epsilon_{\text{stat}}
\]

holds.

8. **Assertion:** There is some \(\nu_0 \in \mathbb{N}\) so that for infinitely many \(\nu \geq \nu_0\) and each \(k \in J^\nu\) the index \(c^{k^\nu}\) is \(2\epsilon_{\text{act}}\)-active for \(g\).
5.2 The Adaptive-Convexification Algorithm with X-adaptation

With step 2. of this proof the assertion results as in the proof of Theorem 4.3.4, step 6.

With these assertion we have that there exists a \( \nu_0 \in \mathbb{N} \) so that for infinitely \( \nu \geq \nu_0 \) the point \( x^\nu \) with all data is an \( \epsilon_{stat} \)-stationary point of SIP with \( 2\epsilon_{act} \)-active, \( \epsilon_Y \)-feasible indices with respect to \( B^{N_0,\nu_0} \). Hence, Algorithm 7 would terminate with a \( x^\nu, \nu \geq \nu_0 \), in contradiction to the assumption.

\[ \square \]

5.2 The Adaptive-Convexification Algorithm with X-adaptation

5.2.1 Relaxation and reformulation

Let \( \lambda_{max}(x,y) \) be the maximal eigenvalue of \( D_2 g(x,y) \). Define a concave overestimator (w.r.t. \( y \)) of the restriction of \( g \) to \( X^\nu \times B^k \) by

\[ \tilde{g}^{\nu,k} : X^\nu \times B^k \rightarrow \mathbb{R}, \quad (x,y) \mapsto g(x,y) + \psi(y; \alpha_k^{\nu}, B^k) \]

with \( \alpha_k^{\nu} > \max \{0, \max_{(x,y) \in X^\nu \times B^k} \lambda_{max}(x,y)\} \). Due to

\[ \alpha \geq \max \{0, \max_{(x,y) \in X \times B} \lambda_{max}(x,y)\} \geq \max \{0, \max_{(x,y) \in X \times B} \lambda_{max}(x,y)\} \]

\( \alpha_k^{\nu} \) can always be chosen so that \( \alpha_k^{\nu} \leq \alpha \) holds for each \( k = 1, \ldots, N \). The next results immediately follow from Lemma 4.1.2.

**Lemma 5.2.1.**

(i) For all \( k = 1, \ldots, N \) and \( (x,y) \in X^\nu \times B^k \) the relation \( g(x,y) \leq \tilde{g}^{\nu,k}(x,y) \) holds.

(ii) For \( x \in X^\nu \) and \( k = 1, \ldots, N \) the relaxation \( \tilde{g}^{\nu,k}(x,y) \) is concave in the second argument on \( B^k \).

(iii) For all \( k = 1, \ldots, N \) the maximum separation distance between \( \tilde{g}^{\nu,k} \) and \( g \) on \( X^\nu \times B^k \) is

\[ \frac{\alpha_k^{\nu}}{8} \| b^k - \bar{b}^k \|_2^2. \]

(iv) If \( \bar{x} \in X^\nu \) and \( \tilde{g}^{\nu,k}(\bar{x},y) \leq 0 \) holds for all \( y \in B^k, k = 1, \ldots, N \), then we have \( \bar{x} \in M \).

The X-adaptation can have an huge impact on the constant \( \alpha \), and, thus, on the maximum separation distance of \( \tilde{g} \) and \( g \). Before we continue our discussion we give an example to illustrate the impact of the X-adaptation.

**Example 5.2.2.** Let \( B = [-1,1] \), \( X = [-2,2] \) and \( g(x,y) = x^2 y^3 \). As \( D_2^2 g(x,y) = 6x^2 y \) we have \( \alpha = 24 \). Thus, we obtain

\[ \max_{y \in B} d_{\alpha BB}(y; \alpha, B) = 6. \]
5 An $X$-adaptation method

For $X^1 = [-1, 1]$ we obtain $\alpha_1 = 6$. Thus, we have

$$\max_{y \in B} d_{\alpha BB}(y; \alpha_1, B) = \frac{3}{2}.$$ 

In this example the impact of the set $X$ on the constant $\alpha$ is very huge. Thus, by switching to $X^1$ the values of this constant and the maximum separation distance of the original and the relaxed function could be reduced enormously.

Define the set

$$M_{\alpha BB} (X^\nu, B^N) := \{ x \in X^\nu | \bar{g}^{\nu,k}(x, y) \leq 0 \text{ for all } y \in B^k, k = 1, \ldots, N \}$$

and the semi-infinite problem

$$SIP_{\alpha BB} (X^\nu, B^N) : \min_{x \in \mathbb{R}^n} f(x) \text{ s.t. } x \in M_{\alpha BB} (X^\nu, B^N)$$

as well as the lower level problems

$$Q^{\nu,k}(x) : \max_{y \in \mathbb{R}^m} \bar{g}^{\nu,k}(x, y) \text{ s.t. } y \in B^k$$

for $k = 1, \ldots, N$. The straightforward proofs of the following results are omitted.

**Lemma 5.2.3.**

(i) We have $M_{\alpha BB} (X^\nu, B^N) \subset M$.

(ii) Let $SIP_{\alpha BB} (X^\nu, B^N)$ be consistent. Then every global, local solution or stationary point of $SIP_{\alpha BB} (X^\nu, B^N)$ is a feasible point of $SIP$.

(iii) For $\alpha^\nu_k > \max \{ 0, \max_{(x,y) \in X^\nu \times B^k} \lambda_{\max} (x, y) \}$ the solution of $Q^{\nu,k}(x)$ is unique for each $k = 1, \ldots, N$.

The problem $SIP_{\alpha BB} (X^\nu, B^N)$ is equivalent to the Stackelberg game

$$SG_{\alpha BB} (X^\nu, B^N) : \min_{x, y^1, \ldots, y^N} f(x) \text{ s.t. } \bar{g}^{\nu,k}(x, y^k) \leq 0,$$

$$x \in X^\nu,$$

$$y^k \text{ solves } Q^{\nu,k}(x), \ k = 1, \ldots, N.$$ 

For every $k \in \{1, \ldots, N\}$ the function $\bar{g}^{\nu,k}$ is concave in the second argument, and $B^k$ is convex and possesses a Slater point. For this reason the Karush-Kuhn-Tucker conditions are necessary and sufficient for $y^k$ to be a solution of $Q^{\nu,k}(x)$. Thus for each $k \in \{1, \ldots, N\}$ the point $y^k$ is a solution of $Q^{\nu,k}(x)$ if and only if there exist multipliers $\gamma^k, \tau^k \in \mathbb{R}^m$ with

$$\nabla_y \bar{g}^{\nu,k}(x, y^k) + \gamma^k - \tau^k = 0$$

$$0 \leq y^k - b^k \perp \gamma^k \geq 0$$

$$0 \leq b^k - y^k \perp \tau^k \geq 0.$$
5.2 The Adaptive-Convexification Algorithm with $X$-adaptation

Define

\[
\omega := (x, (y^k)_{k=1},\ldots,N^\prime, (\gamma^k)_{k=1},\ldots,N^\prime, (\tilde{\gamma}^k)_{k=1},\ldots,N^\prime)
\]

\[
F(\omega) := f(x)
\]

\[
G(\omega) := \left( \left( \tilde{\gamma}^\nu, (x, y^k) \right)_{k=1},\ldots,N, x^\nu - x, x - x^\nu \right)
\]

\[
C^1(\omega) := \left( \gamma^k, \tilde{\gamma}^k \right)_{k=1},\ldots,N
\]

\[
C^2(\omega) := \left( (y^k - y^k), \left( \bar{b}^k - y^k \right) \right)_{k=1},\ldots,N
\]

\[
H(\omega) := \nabla y \tilde{\gamma}^\nu, (x, y^k) + \gamma^k - \bar{\gamma}^k_{k=1},\ldots,N.
\]

Now $SG_{\alpha BB}(X^\nu, B^N)$ can be equivalently reformulated as the MPCC

\[
P_{\alpha BB}(X^\nu, B^N) : \min_{\omega} F(\omega) \quad \text{s.t.} \quad G(\omega) \leq 0
\]

\[
H(\omega) = 0
\]

\[
0 \leq C^1(\omega) \perp C^2(\omega) \geq 0.
\]

For details on solving this problem we refer to Section 7.2.

5.2.2 Algorithms and $X$-Adaptation

On the basis of the discussion in the former chapter the algorithm presented in this section computes a stationary point of $SIP_{\alpha BB}(X^\nu, B^N)$ with active indices and terminates if it is also a stationary point of $SIP$ within a given tolerance on stationarity and additional tolerances on the feasibility of the indices. As in former section a stationary point of $SIP_{\alpha BB}(X^\nu, B^N)$ may be located on the topological boundary of $X^\nu$. But, in contrast to the discussion in Section 5.1 the indices are not predetermined. Thus, the stationary conditions from Theorem 2.1.1 must, again, be modified.

For a point $x \in M_{\alpha BB}(X^\nu, B^N)$ and solutions $y^k(\bar{x})$ of $Q_{\nu,k}(x)$, $k = 1,\ldots,N$, define $K_{\alpha BB}^0(\bar{x}) := \{ k \in \{1,\ldots,N \} | \tilde{y}^\nu, (\bar{x}, y^k(\bar{x})) = 0 \}$. As in Theorem 2.1.1, a point $x \in M_{\alpha BB}(X^\nu, B^N)$ is stationary in the sense of John for $SIP_{\alpha BB}(X^\nu, B^N)$, if there exist an $(n+1)$-tuple $J$ with $\{J\} \subset K_0(\bar{x})$ and $y^k \in B_{\alpha BB}^0(\bar{x})$, $k \in J$, and $\left( \kappa, (\lambda_k)_{k \in J}, (\mu_l)_{l \in \{1,\ldots,n \}}, (\xi_r)_{r \in \{1,\ldots,n \}} \right) \in \sigma_{3n+1}^n$ with only
n + 1 non-vanishing entries, so that
\[\kappa \nabla f(\bar{x}) + \sum_{k \in J} \lambda_k \nabla_x g(\bar{x}, y^k) - \sum_{l=1}^n \mu_l e_l + \sum_{r=1}^n \xi_r e_r = 0\]
\[\lambda_k \tilde{g}^{\nu,k}(\bar{x}, y^k) = 0, \quad k \in J\]
\[\mu_l (\bar{x}_l' - \bar{x}_l) = 0, \quad l \in \{1, \ldots, n\}\]
\[\xi_r (\bar{x}_r - \bar{x}_r') = 0, \quad r \in \{1, \ldots, n\},\]
where \(e_i\) denotes the \(i\)-th unit vector. Notice here that \(\nabla_x \tilde{g}^{\nu,k}(\bar{x}, y^k) = \nabla_x g(\bar{x}, y^k)\).

If we assume that \(\bar{x} \in \text{int}(X')\) holds, the former stationary conditions reduce to a more common condition. That is, there exist \(y^k \in B_{0}^{\alpha_{BB}}(\bar{x}), k = 1, \ldots, n + 1, \) and \((\kappa, \lambda) \in \sigma^{n+1}\), so that
\[\kappa \nabla f(\bar{x}) + \sum_{k=1}^{n+1} \lambda_k \nabla_x g(\bar{x}, y^k) = 0\]
\[\lambda_k \tilde{g}^{\nu,k}(\bar{x}, y^k) = 0, \quad k = 1, \ldots, n + 1.\]

Using the splitting function from Chapter 3 and the techniques introduced there, a splitting algorithm is stated in Algorithm 8. It also describes how the set \(M_{\alpha_{BB}}(X', B^N)\) of a refined problem \(SIP_{\alpha_{BB}}(X', B^N)\) is constructed.

It is not hard to see that each \(N\)-tuple \((B^k, k = 1, \ldots, N)\) generated by Algorithm 8 is a reduced outer approximation of \(Y\).

As in the former section the basic idea of the adaptation of the set \(X\) is to choose some subsets of \(X\) around some point of interest, for example a solution of a subproblem \(SIP_{\alpha_{BB}}\), so that the point stays feasible for the new problem. Corresponding to Lemma 5.1.4 the next result gives a first impression that choosing arbitrary sets may not lead to the desired result.

**Lemma 5.2.4.** Let \(X^1, X^2 \subset X\) so that \(X^1 \cap X^2 \neq \emptyset\), \(B^N\) be some partition of \(B\), and let \(\alpha^1_k\) and \(\alpha^2_k\) be concavification parameters on \(X^1 \times B^k, X^2 \times B^k\) for some box \(B^k \in B^N\) and a constraint \(g\). If there is some \(x^* \in X^1 \cap X^2\) with \(\tilde{g}^{1,k}(x^*, y) \leq 0\) for each \(y \in B^k\), it holds:

1. \(\alpha^1_k \geq \alpha^2_k\) implies that \(\tilde{g}^{2,k}(x^*, y) \leq 0\) holds for each \(y \in B^k\).
2. \(\alpha^1_k \geq \alpha^2_k\) is true if there is some \(y^* \in \text{int}(B^k)\) so that \(\tilde{g}^{1,k}(x^*, y^*) = 0\) and \(\tilde{g}^{2,k}(x^*, y^*) \leq 0\) hold.

**Proof.** To see part (i) notice that \(\tilde{g}^{1,k}(x^*, y) \leq 0\) for each \(y \in B^k\) if \(g(x^*, y) + \alpha^1_k \langle y - \tilde{y}^k, \tilde{b}^k - y \rangle \leq 0\) for each \(y \in B^k\). Since \(\alpha^1_k \geq \alpha^2_k\), and \(\langle y - \tilde{y}^k, b^k - y \rangle \geq 0\) holds for each \(y \in B^k\), part (i) follows immediately. Part (ii) can easily be seen by subtracting \(\tilde{g}^{1,k}\) and \(\tilde{g}^{2,k}\). ☐
Lemma 5.2.4 (i) says that $x^*$ stays feasible, while switching from $X^1$ to $X^2$, if the corresponding concavification parameter is non-increasing. But, as the next example shows, that can not be ensured in general.
5 An X-adaptation method

**Example 5.2.5.** Let \( X^1 = [-2, 2], X^2 = [0, 3], Y = [0, 2] \) and let \( M = \{ x \in \mathbb{R} | g(x,y) = (x^4 - x^2) y^2 - 12 \leq 0 \text{ for all } y \in Y \}. \) Since \( D_2^2 g(x,y) = 2(x^4 - x^2), \) we obtain \( \alpha^1 = 24 \) on \( X^1 \times Y, \) and \( \alpha^2 = 144 \) on \( X^2 \times Y. \) Thus, the relaxed constraints on \( X^1 \times Y \) and \( X^2 \times Y \) result in

\[
\bar{g}^1(x,y) = (x^4 - x^2) y^2 - 12 + 12y(2-y),
\]

respectively

\[
\bar{g}^2(x,y) = (x^4 - x^2) y^2 - 12 + 72y(2-y).
\]

For \( x^* = \frac{1}{2} \) we have

\[
\bar{g}^1(x^*, y) = -\frac{3}{16} y^2 - 12 + 12y(2-y),
\]

and

\[
\bar{g}^2(x^*, y) = -\frac{3}{16} y^2 - 12 + 72y(2-y).
\]

It is not hard to see that \( \bar{g}^1(x^*, y) \leq 0 \) for all \( y \in Y, \) while \( \bar{g}^2(x^*, 1) > 0. \) Even though the diameter of \( X^2 \) is smaller then the diameter of \( X^1, \) \( x^* \) is a feasible point of the set \( M_{\alpha^BB}(X^1, Y), \) while it is not a feasible point of \( M_{\alpha^BB}(X^2, Y). \) Thus, given some point of interest, feasible for some choice of a subset of \( X, \) may not be feasible for another choice of a subset.

Since the first summands in \( \bar{g}^1 \) and \( \bar{g}^2 \) coincide, a reduction of the maximum value of the second summand in \( \bar{g}^2 \) below the maximum value of the second summand of \( \bar{g}^1 \) may avoid that problem. Keeping the set \( X^2 \) unchanged, the maximum value of the second summand is related to the diameter of the set \( Y. \) By splitting \( Y \) into smaller sets one may reduce that value as we show next. For \( i = 1, \ldots, 4 \) let \( B^i = \left[ \frac{-1}{2}, \frac{1}{2} \right], B^i = (B^1, \ldots, B^4), \) and \( \bar{g}^{2,i}(x^*, y) = -\frac{3}{16} y^2 - 12 + 72(y - b^i) \left( \frac{b^i}{y} \right). \) It is not hard to see that \( B^4 \) is a tessellation of \( Y \) and \( \bar{g}^{2,i}(x^*, y) \leq 0 \) for each \( y \in B^i \) and each \( i = 1, \ldots, 4. \) Thus, \( x^* \) is a feasible point of \( M_{\alpha^BB}(X^2, B^4). \)

The next lemma generalizes the observations made in Example 5.2.5, that is, if a point of interest is not in \( M_{\alpha^BB}(X^\nu, B^N) \) for some set \( X^\nu \) and a tessellation \( B^N, \) then there is a refined tessellation \( B'^N \) so that this point is in \( M_{\alpha^BB}(X^\nu, B'^N). \)

**Lemma 5.2.6.** Let \( X^1, X^2 \subset X, x^* \in X^1 \cap X^2 \) and \( B^N \) be some partition of \( B. \) Let there be some box \( B^k \in B^N \) and concavification parameters \( \alpha^1_k \neq 0 \) and \( \alpha^2_k \neq 0 \) for a constraint \( g \) on \( X^1 \times B^k, X^2 \times B^k \) so that \( \bar{g}^{1,k}(x^*, y) \leq 0 \) for each \( y \in B^k, \) and let there be some \( y^* \in B^k \) so that \( \bar{g}^{2,k}(x^*, y^*) > 0 \) is satisfied. Then there is a partition \( B'^N \) of \( B \) so that for each \( k^* \in B'^N \) and \( B^{k^*} \subset B^k, \) and corresponding concavification parameters \( \alpha^2_{k^*}, \) on \( X^2 \times B^{k^*}, \) the relaxation \( \bar{g}^{2,k^*} \) is concave in the second argument and \( \bar{g}^{2,k*}(x^*, y) \leq 0 \) for each \( y \in B^{k^*}. \)

**Proof.** Since \( \alpha^2_{k^*} \) is a concavification parameter it follows immediately that \( \bar{g}^{2,k^*} \) is concave in the second argument. Now we show that there is a partition \( B'^N \) of \( B \) so that for each \( k^* \) with \( B^{k^*} \in B'^N \) and \( B^{k^*} \subset B^k \) the inequality \( \bar{g}^{2,k^*}(x^*, y) \leq g^{1,k}(x^*, y) \) holds.
for each \( y \in B^{k^*} \). Let \( \alpha \) be the concavification parameter for \( g \) on \( X \times B \). Since \( \alpha_{2^*} \), can always be chosen so that \( \alpha_{2^*} < \alpha \) for each \( k^* \) and \( X^2 \subset X \), it is sufficient to show that there is a partition \( B^{N^*} \) of \( B \) so that for each \( k^* \) with \( B^{k^*} \in B^{N^*} \) and \( B^{k^*} \subset B^k \) we have \( \langle y - \bar{b}^{k^*}, \bar{b}^{k^*} - y \rangle \leq \frac{\alpha_{2^*}}{\alpha} \langle y - \bar{b}^k, \bar{b}^k - y \rangle \) for each \( y \in B^{k^*} \) and each \( k^* \). For \( \rho \geq 1 \) let \( S^\rho \) be the \( \rho \)-times splitting operator defined in Subsection 5.1.2. Let \( B^{N^*} \) arise from \( B^N \) by applying \( S^\rho \) to the box \( B^k \) in \( B^N \), and let \( B^{k^*} \subset B^k \) be a box in \( B^{N^*} \) with \( y \in B^{k^*} \). If \( \langle y - \bar{b}^k, \bar{b}^k - y \rangle = 0 \) holds, we also have that \( \langle y - \bar{b}^{k^*}, \bar{b}^{k^*} - y \rangle = 0 \) is true. Let \( \langle y - \bar{b}^k, \bar{b}^k - y \rangle > 0 \). Since \( \alpha_k \neq 0 \) can also always be chosen so that \( \alpha_k < \alpha \) for each \( k \) and \( X^1 \subset X \), we have that there is a constant \( c > 0 \) so that \( \frac{\alpha_k}{\alpha} \langle y - \bar{b}^k, \bar{b}^k - y \rangle \geq c \). Furthermore we have that \( \langle y - \bar{b}^{k^*}, \bar{b}^{k^*} - y \rangle \leq \frac{1}{4} ||\bar{b}^{k^*} - \bar{b}^k||^2 \) holds. By Lemma 3.3.4 we obtain that \( ||\bar{b}^{k^*} - \bar{b}^k||^2 \leq \left( 1 - \frac{3}{4m} \right)^{\frac{1}{2}} ||\bar{b}^k - \bar{b}^k||^2 \) holds. Since \( 0 < 1 - \frac{3}{4m} < 1 \) for each \( m \), we have that for each \( c > 0 \) there is some some \( \rho \) so that \( \left( 1 - \frac{3}{4m} \right)^{\frac{1}{2}} ||\bar{b}^k - \bar{b}^k||^2 \leq c \). Altogether we have that there is some \( \rho \) so that

\[
\langle y - \bar{b}^{k^*}, \bar{b}^{k^*} - y \rangle \leq \frac{1}{4} ||\bar{b}^{k^*} - \bar{b}^k||^2 \\
\leq \frac{1}{4} \left( 1 - \frac{3}{4m} \right)^{\frac{1}{2}} ||\bar{b}^k - \bar{b}^k||^2 \\
\leq \frac{\alpha_k}{\alpha} \langle y - \bar{b}^k, \bar{b}^k - y \rangle
\]

holds.

Using Lemma 5.2.6 and Properties 5.1.7 the \( X \)-adaptation algorithm is stated in Algorithm 9.

By Lemma 5.2.6 we have that Algorithm 9 is well defined. That is, after finitely many splitting steps the set \( K \) is empty. Using the splitting and the \( X \)-adaptation algorithm, the adaptive convexification algorithm with \( X \)-adaptation is stated in Algorithm 10.

If for a given reduced outer approximation \( B^N \) the problem \( SIP_{\alpha BB}(X^0, B^N) \) is not consistent in Algorithm 10, a phase 1 algorithm like that described in [16] can be performed.

### 5.2.3 Convergence results

At first we will show that Algorithm 10 is well defined. If \( P^k = \emptyset \) holds at each active index \( y^k \) of an approximating problem \( SIP_{\alpha BB}(X^\nu, B^N) \) for some iterate \( x^\nu \in int(X^\nu) \), we are in the same situation as in the former chapters. That is, the reduced outer approximation of \( Y \) is not refined any further and Algorithm 10 might loop. The following lemma is a direct consequence of Lemma 4.3.2 and shows that the algorithm terminates in this case.
Algorithm 9 X-Adaptation - Xadapt\(_{aBB}(x, \Delta x)\)

Let \(x \in X^\nu\), and \(B^N\) a reduced outer approximation of \(Y\).

Generate a set \(X^{\nu+1} \subset X\) so that Condition 5.1.7 is satisfied.

Compute new parameters \(\alpha_k^{\nu+1}\) on \(X^{\nu+1} \times B^k\) for each \(k \in \{1, \ldots, N\}\) and replace \(\alpha^\nu\) by \(\alpha^{\nu+1}\). For each \(k \in \{1, \ldots, N\}\) set

\[
\bar{g}^{\nu+1,k}(x,y) = g(x,y) + \frac{\alpha_k^{\nu+1}}{2} \langle y - b_k, b_k \rangle - y
\]

Set \(K = \{k \in \{1, \ldots, N\} | \bar{g}^{\nu+1,k}(x,y) > 0 \text{ for some } y \in B^k\}\).

while \(K \neq \emptyset\) do
  for \(k \in K\) do
    Let \(S^k\) be the barycenter of \(B^k\).
    \(X\text{refine}_{aBB}(S^k)\)
  end for
  Set \(K = \{k \in \{1, \ldots, N\} | \bar{g}^{\nu+1,k}(x,y) > 0 \text{ for some } y \in B^k\}\).
end while

**Lemma 5.2.7.** Let \(X^\nu \subset X\) and \(\bar{\pi} \in \text{int}(X^\nu)\) be a stationary point of \(SIP_{aBB}(X^\nu, B^N)\) with \(\epsilon_{\text{act}}\)-active indices \(y^k, k \in J^\nu\), for an \((n + 1)\)-tuple \(J^\nu\) with \(\{J^\nu\} \in K^0_{aBB}(x^\nu)\).

Let \((\kappa, (\lambda_k)_{k \in J^\nu}, (\mu_l)_{l \in \{1, \ldots, n\}}, (\xi_r)_{r \in \{1, \ldots, n\}})\) be the corresponding multipliers, and let the refined reduced outer approximation arising from applying Algorithm 8 to each \(y^k, k \in J^\nu\), coincide with \(B^N\). Then Algorithm 10 terminates at \(\bar{\pi}\).

An implication of Lemma 4.3.3 is, that refining a given reduced outer approximation \(B^N\) of \(Y\) by means of Algorithm 8, using the splitting function, enlarges the feasible set of \(SIP_{aBB}(X^\nu, B^N)\).

**Lemma 5.2.8.** Let a reduced outer approximation \(B^N\) of \(Y\) be given, and for an arbitrary splitting point \(\eta\) let \(B^\tilde{N}\) be the refinement of \(B^N\) by means of Algorithm 8. Then \(M_{aBB}(X^\nu, B^N) \subset M_{aBB}(X^\nu, B^\tilde{N})\) holds.

As Algorithm 8 is used in Algorithm 9, Lemma 5.2.8 together with Condition 5.1.7 (i) implies that for a point \(x \in M_{aBB}(X^\nu, B^N)\) we also have \(x \in M_{aBB}(X^{\nu+1}, B^\tilde{N})\). Here, \(B^\tilde{N}\) denotes a refinement of a reduced outer approximation \(B^N\) by means of Algorithm 8.

Now we are in the position to state the main convergence result.
Algorithm 10: Adaptive convexification algorithm - Xaca

Choose $X \subset \mathbb{R}^n$ with $M \subset X$, $B = [\underline{b}, \overline{b}] \subset \mathbb{R}^m$ with $Y \subset B$ and compute $\alpha$ on $X \times B$. Set $\nu = 0$ and $X^0 = X$.

Determine a reduced outer approximation $B^N$ of $Y$ with some $N \in \mathbb{N}$ as well as $\alpha_0^N \leq \alpha$ on $X^0 \times B^k$, $k = 1, \ldots, N$, so that $SIP_{\alpha_{BB}}(X^0, B^N)$ is consistent.

Choose $\epsilon_{act}, \epsilon_{stat}, \epsilon_Y > 0$ and $\epsilon_{split} \in (0, \frac{1}{2})$ with $\epsilon_{split} \leq 2\epsilon_{act}/(\alpha\|\overline{b} - \underline{b}\|^2)$.

Compute a stationary point $x^0$ of $SIP_{\alpha_{BB}}(X^0, B^N)$ with $\epsilon_{act}$-active indices $y^k$, $k \in J^0$, and $n + 1$ non vanishing multipliers $(\kappa, (\lambda_k)_{k \in J^0}, (\mu_l)_{l \in \{1, \ldots, n\}}, (\xi_r)_{r \in \{1, \ldots, n\}})$ with an $(n + 1)$-tuple $J^0$, $(J^0) K_{0,0}^{\alpha_{BB}}(x^0)$ by solving $P_{\alpha_{BB}}(X^0, B^N)$.

while $x^\nu$ is not a stationary point of SIP with $2\epsilon_{act}$-active, $\epsilon_Y$-feasible indices $y^k$, $k \in J^\nu$, with respect to $B^N$, and multipliers $(\kappa, (\lambda_k)_{k \in J^\nu})$ do

for $k = 1$ to $n + 1$ do

$X refine_{\alpha_{BB}}(y^k)$

end for

if $\nu > 1$ then

Set $\Delta x^\nu = \|x^\nu - x^{\nu-1}\|_{\infty}$.

$X adapt_{\alpha_{BB}}(x^\nu, \Delta x^\nu)$

else

Set $X^1 = X$.

end if

Set $\nu = \nu + 1$.

Compute a stationary point $x^\nu$ of $SIP_{\alpha_{BB}}(X^\nu, B^N)$ with $\epsilon_{act}$-active indices $y^k$, $k \in J^\nu$, and $n + 1$ non vanishing multipliers $(\kappa, (\lambda_k)_{k \in J^\nu}, (\mu_l)_{l \in \{1, \ldots, n\}}, (\xi_r)_{r \in \{1, \ldots, n\}})$ with an $(n + 1)$-tuple $J^\nu$, $(J^\nu) K_{0,0}^{\alpha_{BB}}(x^\nu)$ by solving $P_{\alpha_{BB}}(X^\nu, B^N)$.

end while

Theorem 5.2.9. Algorithm 10 terminates after finitely many steps.

Proof. We give the proof by enforcing a contradiction. Assume that Algorithm 10 does not terminate. Then there exist sequences of reduced outer approximations $(B^{N,\nu})_\nu$ of $Y$, $(X^{\nu})_\nu$, $(x^{\nu})_\nu$, $(K_{0,0}^{\alpha_{BB}}(x^{\nu}))_\nu$, $(J^{\nu})_\nu$ with $(J^{\nu}) \subset K_{0,0}^{\alpha_{BB}}(x^{\nu})$, $(\{y^{k,\nu}\}^{\nu}_{k \in J^{\nu}})_\nu$ and multipliers $(\kappa^\nu, (\lambda^\nu_k)_{k \in J^{\nu}}, (\mu^\nu_l)_{l \in \{1, \ldots, n\}}, (\xi^\nu_r)_{r \in \{1, \ldots, n\}})_\nu$ with only $n + 1$ non-vanishing entries, so that for each $\nu$ the point $x^{\nu}$ is stationary for $SIP_{\alpha_{BB}}(X^{\nu}, B^{N,\nu})$ with $\epsilon_{act}$-active indices $(y^{k,\nu})_{k \in J^{\nu}}$ and multipliers $(\kappa^\nu, (\lambda^\nu_k)_{k \in J^{\nu}}, (\mu^\nu_l)_{l \in \{1, \ldots, n\}}, (\xi^\nu_r)_{r \in \{1, \ldots, n\}})_\nu$, while $x^\nu$ with all data is not $\epsilon_{stat}$-stationary for SIP with $2\epsilon_{act}$-active, $\epsilon_Y$-feasible indices with respect to $B^{N,\nu}$.

To derive a contradiction, in step 1. the existence of accumulation point of the sequence $(x^{\nu})_\nu$ is shown. In step 2. and 3. it is shown that for infinitely many $\nu$ the point $x^{\nu}$ is an interior point of the box $X^{\nu}$ and, thus, the multipliers $(\mu^\nu_l)_{l \in \{1, \ldots, n\}}$ and $(\xi^\nu_r)_{r \in \{1, \ldots, n\}}$
vanish for infinitely many $\nu$. Then, in the steps 4. and 5. the existence of accumulation points of the sequences $\left((y^{k,\nu})_{k\in J^\nu}\right)_\nu$ and $\left((\nu^r, (\lambda^r_k)_{k\in J^\nu}, (\mu^r_l)_{l\in \{1, \ldots, n\}}, (\xi^r_r)_{r\in \{1, \ldots, n\}}\right)_\nu$ is shown. After that it is shown in steps 6. to 8. that there exists a $\nu_0 \in \mathbb{N}$ so that $x^\nu$ with all data is an $\epsilon_{\text{stat}}$-stationary point for $SIP$ with $2\epsilon_{\text{act}}$-active, $\epsilon_Y$-feasible indices with respect to $B^{N^\nu,\nu}$ for infinitely many $\nu \geq \nu_0$.

1. **Assertion:** The sequence $(x^\nu)_\nu$ has an accumulation point $x^* \in M$.

Since $(x^\nu)_\nu$ is contained in the compact set $X$, it possesses an accumulation point $x^* \in X$. For $i \geq 0$ let $M^i := \bigcup_{\nu \geq 1} M_{i\nu}(X^\nu, B^{N^\nu,\nu})$. By Lemma 5.2.8 and Lemma 5.2.3 we have $M^0 \subset M^1 \subset M^2 \subset \ldots \subset M$. Thus, we have that $(x^\nu)_\nu$ is also contained in the closed set $M$. From this the assertion immediately follows.

2. **Assertion:** There is some accumulation point $x^* \in X \cap M$ of the sequence $(x^\nu)_\nu$ so that for each $\epsilon > 0$ there is some $\nu_0 \in \mathbb{N}$ so that it holds $\|x^* - x^\nu\|_\infty \leq \epsilon$ as well as $x^\nu \in \text{int}(X^\nu)$ for infinitely many $\nu \geq \nu_0$.

From step 1 we have that the sequence $(x^\nu)_\nu$ has an accumulation point $x^* \in M$. That means that for each $\epsilon > 0$ there is some $\nu_0 \in \mathbb{N}$ so that it holds $\|x^* - x^\nu\|_\infty \leq \epsilon$. By Condition 5.1.7 (i),(iii) we have that the interior of $X^\nu$ is nonempty. Thus, we only have to distinguish two cases to show the assertion. The first case is that for infinitely many $\nu \geq \nu_0$ we have $x^\nu \in \text{int}(X^\nu)$. That is exactly the statement of the assertion. The second case is that there are not infinitely many $\nu \geq \nu_0$ with $x^\nu \in \text{int}(X^\nu)$. In this case we have that there is some $\nu_1 \geq \nu_0$ so that $x^\nu \in \partial X^\nu$ for each $\nu \geq \nu_1$. Condition 5.1.7 (ii),(iv) and Assumption 1 imply that there is some $\nu_2 \geq \nu_1$ so that $\partial X^\nu \cap M = \emptyset$ for each $\nu \geq \nu_2$. As $x^\nu \in M_{\alpha_{BB}}(X^\nu, B^{N^\nu,\nu}) \subset M$ holds for each $\nu \in \mathbb{N}$, we obtain a contradiction.

3. **Assertion:** There is some $\nu_0 \in \mathbb{N}$ so that $(\mu^\nu_l)_{l\in \{1, \ldots, n\}} = 0$ and $(\xi^\nu_r)_{r\in \{1, \ldots, n\}} = 0$ for infinitely many $\nu \geq \nu_0$.

For all $\nu$ the point $x^\nu$ with all data is stationary for $SIP_{\alpha_{BB}}(X^\nu, B^N)$. Thus, we have that

\begin{align*}
\mu_l (x^\nu_l - x_l) &= 0, \quad l \in \{1, \ldots, n\} \\
\xi_r (x_r - x^\nu_r) &= 0, \quad r \in \{1, \ldots, n\}
\end{align*}

holds. By step 2. there is some $\nu_0 \in \mathbb{N}$ so that $x^\nu \in \text{int}(X^\nu)$ holds for infinitely many $\nu \geq \nu_0$. Thus, we obtain the assertion.

4. **Assertion:** The sequence $\left((y^{k,\nu})_{k\in J^\nu}\right)_\nu$ has an accumulation point $(y^{k^*})_{k\in J^*} \in Y^{n+1}$.

Step 3. implies that there is some $\nu_0 \in \mathbb{N}$ so that $(y^{k,\nu})_{k\in J^\nu} \in \mathbb{R}^{n+1}$ with pairwise different entries for infinitely many $\nu \geq \nu_0$. Thus, by possibly switching to a subsequence, it is sufficient to show the existence of an accumulation point in $Y$ for every component of the sequence of the $(n + 1)$-tuple. Let $(y^{k^*,\nu})_\nu$, $k^* \in
5.2 The Adaptive-Convexification Algorithm with X-adaptation

\{1, \ldots, n+1\}, be a sequence of such components. As in the proof of Theorem 4.3.4 we will distinguish two cases. For the first case assume that for \( \nu \geq \nu_0 \) we have \( y^{k, \nu} \in Y \) for infinitely many \( \nu \in \mathbb{N} \). By the compactness of \( Y \), the subsequence of these points possesses an accumulation point in \( Y \).

In the second case there is some \( \nu_1 \in \mathbb{N} \) with \( \nu_1 \geq \nu_0 \) so that \( y^{k, \nu} \notin Y \) for all \( \nu \geq \nu_1 \). Let \( (B^{k, \nu})_\nu \) be the sequence of boxes with \( y^{k, \nu} \in B^{k, \nu} \).

After possibly switching to a subsequence we may assume that \( (B^{k, \nu})_\nu \) satisfies \( B^{k, \nu} \supset B^{k, \nu+1} \) for \( \nu \in \mathbb{N} \), since for each \( \nu \) the reduced outer approximation \( B^{N, \nu} \) of \( Y \) only contains finitely many boxes. As Algorithm 8 subdivides each box \( B^{\nu} \) with \( y^{k, \nu} \notin Y \) at its barycenter, Lemma 3.3.4 implies that the maximum edge lengths of the \( B^{k, \nu} \), \( \nu \in \mathbb{N} \), tend to zero. Due to \( B^{k, \nu} \cap Y \neq \emptyset, \nu \in \mathbb{N} \), we arrive at

\[ \lim_{\nu \to \infty} \min_{y \in Y} ||y^{k, \nu} - y||_2 = 0 \]

and, thus, the assertion.

5. Assertion: The sequence \( \left( \kappa^\nu, (\lambda_k^\nu)_{k \in J^\nu}, (\mu_t^\nu)_{t \in \{1, \ldots, n\}}, (\xi_r^\nu)_{r \in \{1, \ldots, n\}} \right)_\nu \) has an accumulation point \( (\kappa^*, \lambda^*, 0, 0) \in \sigma^{3n+1} \).

This is clear from the compactness of \( \sigma^{3n+1} \) and Step 3.

6. Assertion: There exists a \( \nu_0 \in \mathbb{N} \) so that each \( (y^{k, \nu})_\nu \) with \( k \in J^\nu \) is \( \epsilon_\text{Y} \)-feasible with respect to \( B^{N, \nu} \) for \( Y \) for all \( \nu \geq \nu_0 \).

With Lemma 3.3.5 the assertion results as in the proof of Theorem 4.3.4, step 4.

7. Assertion: There is some \( \nu_0 \in \mathbb{N} \) so that for infinitely many \( \nu \geq \nu_0 \) the point \( x^\nu \) is \( \epsilon_\text{stat} \)-stationary for SIP with multipliers \( (\kappa^\nu, \lambda^\nu) \in \sigma^{n+1} \).

Since for all \( \nu \) the point \( x^\nu \) is stationary for SIP\(_{\alpha BB} \left( X^\nu, B^N \right) \) with \( \epsilon_\text{act} \)-active indices \( (y^{k, \nu})_{k \in J^\nu} \) and multipliers \( \left( \kappa^\nu, (\lambda_k^\nu)_{k \in J^\nu}, (\mu_t^\nu)_{t \in \{1, \ldots, n\}}, (\xi_r^\nu)_{r \in \{1, \ldots, n\}} \right) \), parts 1, 3, 4, and 5. of this proof together with a continuity argument entail

\[ \kappa^* \nabla f(x^*) + \sum_{k \in J^*} \lambda_k^* \nabla_x g(x^*, y^{k, *}) = 0. \]

Hence there is some \( \nu_0 \in \mathbb{N} \) so that for infinitely many \( \nu \geq \nu_0 \)

\[ ||\kappa^\nu \nabla f(x^\nu) + \sum_{k \in J^\nu} \lambda_k^\nu \nabla_x g(x^\nu, y^{k, \nu})|| \leq \epsilon_\text{stat} \]

holds.

8. Assertion: There is some \( \nu_0 \in \mathbb{N} \) so that for infinitely many \( \nu \geq \nu_0 \) and each \( k \in J^\nu \) the index \( y^{k, \nu} \) is \( 2 \epsilon_\text{act} \)-active for \( g \).

With step 2. of this proof the assertion is an implication of the proof of Theorem 4.3.4, step 6.
With these assertion we have that there exists a $\nu_0 \in \mathbb{N}$ so that for infinitely $\nu \geq \nu_0$ the point $x^{\nu}$ with all data is an $\epsilon_{\text{stat}}$-stationary point of $SIP$ with $2\epsilon_{\text{act}}$-active, $\epsilon_Y$-feasible indices with respect to $B^{N_{\nu_0},\nu_0}$. Hence, Algorithm 10 would terminate with a $x^{\nu}$, $\nu \geq \nu_0$, in contradiction to the assumption.

☐
6 The hybrid method

In this chapter we give a brief discussion on a hybrid method constructed with the algorithms introduced in the former chapters. Here we only give the algorithms with an additional $X$-adaptation. The construction of the algorithms without the $X$-adaptation is, then, straightforward.

6.1 Motivation and reformulation

Let $B^k$ be a box in a reduced outer approximation $\mathcal{B}^N = (B^k, \ k = 1, \ldots, N)$ of a set $Y$. A measure for the quality of the overestimators used in the adaptive reduction algorithm and the adaptive convexification algorithm is the maximum separation distance of the original constraints and the relaxations on the set $X^\nu \times B^k$. In the algorithms using the unimodal relaxations the distance results in $\langle L^\nu_k, b^k - c^\nu_k \rangle$, and in the algorithms using the concave overestimators we have $\alpha^\nu_k \| b^k - b^\nu_k \|^2_2$. The first numerical example already pointed out that the numerical behavior of both algorithms is closely related to the maximum separation distances. Since we can compute the separation distance on each set $X \times B^k$, $X^\nu \times B^k$ it is possible to construct an algorithm using the ‘better’ relaxation technique, that is, the overestimator with the smaller separation distance.

First of all we recall that we defined an unimodal overestimator of the restriction $g$ on $X^\nu \times B^k$ by

$$\hat{g}^\nu_k : X^\nu \times B^k \rightarrow \mathbb{R}, \ (x, y) \mapsto g(x, y) + \phi \left(y; c^\nu_k, B^k; L^\nu_k, \mathcal{L}^\nu_k \right),$$

and a concave overestimator of the restriction of $g$ to $X^\nu \times B^k$ by

$$\check{g}^\nu_k : X^\nu \times B^k \rightarrow \mathbb{R}, \ (x, y) \mapsto g(x, y) + \psi(y; \alpha^\nu_k, B^k),$$

with the parameters $L^\nu_k, \mathcal{L}^\nu_k, \alpha^\nu_k$ chosen so that

$$L^\nu_k = \begin{pmatrix} \min(0, \mathcal{L}_1 (X^\nu, B^k)) \\ \vdots \\ \min(0, \mathcal{L}_m (X^\nu, B^k)) \end{pmatrix}$$

and

$$\mathcal{L}^\nu_k = \begin{pmatrix} \max(0, \mathcal{L}_1 (X^\nu, B^k)) \\ \vdots \\ \max(0, \mathcal{L}_m (X^\nu, B^k)) \end{pmatrix},$$

with

$$\mathcal{L}_i (X^\nu, B^k) < \min_{(x,y)\in X^\nu \times B^k} (\frac{\partial}{\partial y_i} g(y)) \quad \text{and} \quad \overline{\mathcal{L}}_i (X^\nu, B^k) > \max_{(x,y)\in X^\nu \times B^k} (\frac{\partial}{\partial y_i} g(y)),$$
The hybrid method

\[ \alpha^\nu_k > \max\{0, \max_{(x,y) \in X^\nu \times B^k} \lambda_{\max}(x,y)\}. \]

Let \( \hat{K}, \bar{K} \subset \{1, \ldots, N\} \) with

\[ \hat{K} = \{k \in \{1, \ldots, N\} | \langle L^\nu_k, b^k - c^\nu_k \rangle \leq \alpha_k^\nu \|b^k - b^k\|^2 \| \} \]

and \( \bar{K} = \{1, \ldots, N\} \setminus \hat{K} \). Define the set

\[ M_{hyb}(X^\nu, B^N) := \{ x \in X^\nu | \hat{g}^\nu_k(x, c^\nu_k) \leq 0 \text{ for all } k \in \hat{K}, \bar{g}^\nu_k(x, y) \leq 0 \text{ for all } y \in B^k, k \in \bar{K} \}. \]

and the semi-infinite problem

\[ SIP_{hyb}(X^\nu, B^N) : \min_x f(x) \quad s.t. \quad x \in M_{hyb}(X^\nu, B^N). \]

We have to mention that we can define a lower level problem \( Q^\nu_k(x) \) as in Chapter 5.2 if \( \bar{K} \) is not empty. It is

\[ Q^\nu_k(x) : \max_{y \in \mathbb{R}^m} \bar{g}^\nu_k(x, y) \quad s.t. \quad y \in B^k \]

for \( k \in \bar{K} \). Define

\[
\begin{align*}
\omega & := (x, (y^k)_{k \in \bar{K}}, (z^k)_{k \in \bar{K}}, (\tau^k)_{k \in \bar{K}}) \\
F(\omega) & := f(x) \\
\bar{G}(\omega) & := \left(\left(\bar{g}^\nu_k(x, y^k)\right)_{k \in \bar{K}}, \left(\tau^k - x - \bar{z}^\nu - x\right)\right) \\
\bar{G}(x) & := \left(\bar{g}^\nu_k(x, c^\nu_k)\right)_{k \in \bar{K}} \\
C^1(\omega) & := \left(\left(y^k - b^k\right), \left(b^k - y^k\right)\right)_{k \in \bar{K}} \\
C^2(\omega) & := \left(\nabla_y \bar{g}^\nu_k(x, y^k) + \frac{1}{2} - \tau^k\right)_{k \in \bar{K}}. \\
H(\omega) & := \left(\nabla_y \bar{g}^\nu_k(x, y^k) + \frac{1}{2} - \tau^k\right)_{k \in \bar{K}}.
\end{align*}
\]

Following the ideas presented in the former chapters, \( SIP_{hyb}(X^\nu, B^N) \) can be equivalently rewritten as

\[ P_{UF}(X^\nu, B^N) : \min_\omega F(\omega) \quad s.t. \quad \bar{G}(\omega) \leq 0 \\
\bar{G}(\omega) \leq 0 \\
H(\omega) = 0 \\
0 \leq C^1(\omega) \perp C^2(\omega) \geq 0. \]

We have to notice that \( P_{UF}(X^\nu, B^N) \) is a standard nonlinear optimization problem if \( \bar{K} = \emptyset \) and an MPCC otherwise.
6.2 Algorithms and convergence results

In this section we present the algorithms for the hybrid method. As a combination of the algorithms introduced in the former chapter, the algorithm presented in this section computes a stationary point of $SIP_{hyb}(X^\nu, B^N)$ with active indices and terminates if it is also a stationary point of $SIP$ within a given tolerance on stationarity and additional tolerances on the feasibility of the indices. First of all notice that for $SIP_{hyb}(X^\nu, B^N)$ we have to rewrite the stationarity conditions from Theorem 2.1.1 again.

For a point $\pi \in M_{hyb}(X^\nu, B^N)$ and solutions $y^k(\pi)$ of $Q^{\nu,k}(\pi)$, $k = \hat{K}$, as well as points $c^{\nu,k} \in B^k$, $k = \hat{K}$, define

$$
K^{UF}_{0,\nu}(\pi) := \{k \in \hat{K} | g^{\nu,k}(\pi, c^{\nu,k}) = 0\},
$$

$$
K^{BB}_{0,\nu}(\pi) := \{k \in \hat{K} | g^{\nu,k}(\pi, y^k(\pi)) = 0\},
$$

$$
B^{BB}_{0,\nu}(\pi) := \{y^k(\pi) | k \in K^{BB}_{0,\nu}(\pi)\}.
$$

A point $\pi \in M_{hyb}(X^\nu, B^N)$ is stationary for $SIP_{hyb}(X^\nu, B^N)$ in the sense of John, if there exist an $(n + 1)$-tuple $(\hat{J}, J)$ with $\{\hat{J}\} \subset K^{UF}_{0,\nu}(\pi)$, $\{\hat{J}\} \subset K^{BB}_{0,\nu}(\pi)$, and $y^k \in B^{BB}_{0,\nu}(\pi)$, $k \in \hat{J}$, as well as $(\kappa, (\hat{\lambda}_k)_{k \in J, j}, (\hat{\lambda}_k)_{k \in J, j}, (\mu_l)_{l \in \{1, \ldots, n\}}, (\xi_r)_{r \in \{1, \ldots, n\}}) \in \sigma^{n+1}$ with only $n + 1$ non-vanishing entries, so that

$$
\kappa \nabla f(\pi) + \sum_{k \in J} \hat{\lambda}_k \nabla x_g(\pi, c^{\nu,k}) + \sum_{k \in J} \hat{\lambda}_k \nabla x_g(\pi, y^k) - \sum_{l=1}^n \mu_l e_l + \sum_{r=1}^n \xi_r e_r = 0
$$

$$
\hat{\lambda}_k g^{\nu,k}(\pi, c^{\nu,k}) = 0, \quad k \in \hat{J}
$$

$$
\hat{\lambda}_k g^{\nu,k}(\pi, y^k) = 0, \quad k \in \hat{J}
$$

$$
\mu_l (x^\nu_l - \pi_l) = 0, \quad l \in \{1, \ldots, n\}
$$

$$
\xi_r (\pi_r - x^\nu_r) = 0, \quad r \in \{1, \ldots, n\},
$$

where $e_i$ denotes the $i$-th unit vector. In Algorithm 11 we state the error computation step. The algorithm also describes how all parameters and restrictions for the feasible set of a refined problem $SIP_{hyb}$ are computed. As a combination of the Algorithms 6 and 9 the splitting step for $SIP_{hyb}$ is stated in Algorithm 12. After that, in Algorithm 13, we will give the $X$-adaptation algorithm for the hybrid method. The sets $\hat{K}$ and $\hat{K}$ may change when a new subset of $X$ is determined in Algorithm 13. Thus, the algorithm also describes how to construct new sets $\hat{K}$ and $\hat{K}$.

As a direct consequence from Lemma 5.1.6 and 5.2.6 we have that Algorithm 13 is well defined. That is, after finitely many steps the set $\hat{K}$ is empty.

With the latter presented algorithms we can give the adaptive hybrid algorithm in Algorithm 14. It is a combination of the Algorithms 7 and 10.
In this subsection we briefly discuss some convergence results for the adaptive hybrid method. All results are consequences from the discussions in Chapter 5. As a direct result from Lemma 5.1.8 and 5.2.7 we have that Algorithm 14 is well defined.

The next lemma is a consequence from Lemma 5.1.9 and 5.2.8. It means that refining a given reduced outer approximation $B^N$ of $Y$ by means of Algorithm 12, using the splitting function, enlarges the feasible set of $SIP_{hyb}$.

**Lemma 6.2.2.** Let a reduced outer approximation $B^N$ of $Y$ be given, and for an arbitrary splitting point $\eta$ let $B^N$ be the refinement of $B^N$ by means of Algorithm 12. Then...
6.2 Algorithms and convergence results

Algorithm 12 Splitting step - refine\(_{hyb}(\nu, \eta)\)

Let \(\eta \in B^k, k^* \in \{1, \ldots, N\}, S^{k^*}\) be the barycenter of \(B^{k^*}\) and
\[
Q^{k^*} = \frac{\min_{i \in P^{k^*}} (b_i^* - t_i^*)}{\|B^{k^*} - \bar{b}^{k^*}\|_\infty}.
\]
if \(\eta \notin Y\) or \(Q^{k^*} < \epsilon\) then
Set \(\eta = S^{k^*}\).
end if
if \(P^{k^*} \neq \emptyset\) then
Compute \((B^{k^*,(1)}, B^{k^*,(2)}) = S(B^{k^*}, \eta)\) and set \(\hat{K}^\nu = \hat{K}^\nu \setminus \{k^*\}, \bar{K}^\nu = \bar{K}^\nu \setminus \{k^*\}\).
Set \(\tilde{M} = \{x \in X^\nu \mid \tilde{g}^k(x, y) \leq 0 \quad \text{for all } y \in B^k, k \in \hat{K}^\nu, \}
\tilde{g}^k(x, e^k) \leq 0 \quad \text{for all } k \in \bar{K}^\nu\}.

\(err_{hyb}(X^\nu, B^{k^*,(1)}, B^{k^*,(2)}, \alpha, L, \bar{L}, \hat{K}^\nu, \bar{K}^\nu)\)
if \(Y \cap B^{k^*,(1)} \neq \emptyset\) and \(Y \cap B^{k^*,(2)} \neq \emptyset\) then
Replace \(B^{k^*}\) by \(B^{k^*,(1)}\) and \(B^{k^*,(2)}\) and replace \(\alpha^{k^*}_{\nu}\) by \(\alpha^{\nu,(1)}_{k^*}\) and \(\alpha^{\nu,(2)}_{k^*}\), as well as \(L_{\nu,k^*}, \bar{L}_{\nu,k^*}\) by \(L_{\nu,k^*,(1)}\) and \(L_{\nu,k^*,(2)}\) and \(\bar{L}_{\nu,k^*,(1)}\) and \(\bar{L}_{\nu,k^*,(2)}\).
Set \(M_{hyb}(X^\nu, B^N) = \tilde{M} \cap M(1) \cap M(2)\).
Set \(N = N + 1\).
else if \(Y \cap B^{k^*,(1)} \neq \emptyset\) then
Replace \(B^{k^*}\) by \(B^{k^*,(1)}\) and \(\alpha^{\nu,(1)}_{k^*}\), as well as \(L_{\nu,k^*}, \bar{L}_{\nu,k^*}\) by \(L_{\nu,k^*,(1)}\), \(\bar{L}_{\nu,k^*,(1)}\).
Set \(M_{hyb}(X^\nu, B^N) = \tilde{M} \cap M(1)\).
else if \(Y \cap B^{k^*,(2)} \neq \emptyset\) then
Replace \(B^{k^*}\) by \(B^{k^*,(2)}\) and \(\alpha^{\nu,(2)}_{k^*}\), as well as \(L_{\nu,k^*}, \bar{L}_{\nu,k^*}\) by \(L_{\nu,k^*,(2)}\), \(\bar{L}_{\nu,k^*,(2)}\).
Set \(M_{hyb}(X^\nu, B^N) = \tilde{M} \cap M(2)\).
else
Delete \(B^{k^*}, \alpha^{\nu}_{k^*}\) and \(L_{\nu,k^*}, \bar{L}_{\nu,k^*}\).
Set \(M_{hyb}(X^\nu, B^N) = \tilde{M}\).
Set \(N = N - 1\).
end if
end if

\(M_{hyb}(X^\nu, B^N) \subset M_{hyb}(X^\nu, B^\bar{N})\) holds.

As an implication from Lemma 6.2.2 and Condition 5.1.7 we have that for a point \(x \in M_{hyb}(X^\nu, B^N)\) we also have \(x \in M_{hyb}(X^{\nu+1}, B^N)\), with a refined reduced outer approximation \(B^\bar{N}\). Now, as a result from Theorem 5.1.10 and 5.2.9 we can state the main convergence result for Algorithm 14.
6 The hybrid method

Algorithm 13 X-Adaptation - \textit{Xadapt}_{hyb}(x, \Delta x)

Let $x \in X^\nu$, and $B^N$ a reduced outer approximation of $Y$.
Generate a set $X^{\nu+1} \subset X$ so that Condition 5.1.7 is satisfied.
Set $K^{\nu+1} = K^\nu = \emptyset$.
Compute new parameters $\alpha_k^{\nu+1}, L_k^{\nu+1,k}, T_k^{\nu+1,k}$ and $c_k^{\nu+1,k}$ on $X^{\nu+1} \times B^k$ for each $k \in \{1, \ldots, N\}$.
Replace $\alpha$ by $\alpha^{\nu+1}$, $L$ by $L^{\nu+1}$, $L^*$ by $L^{\nu+1}$ and $c$ by $c^{\nu+1}$.

for $k = 1$ to $N$ do
Compute $\text{ERR}^{k}_{\alpha BB} = \frac{\alpha^{\nu+1}}{8} \| b^k - b_k^k \|_2^2$ and $\text{ERR}^{k}_{UF} = \langle T^{\nu+1,k}, b^k - c^{\nu+1,k} \rangle$.
if $\text{ERR}^{k}_{\alpha BB} < \text{ERR}^{k}_{UF}$ then
Set $\hat{K}^{\nu+1} = \hat{K}^{\nu+1} \cup \{k\}$,
\begin{equation*}
\hat{g}^{(i)}(x, y) = g(x, y) + \psi(y; \alpha_k^{\nu+1}, B_k)
\end{equation*}
else
Set $\hat{K}^{\nu+1} = \hat{K}^{\nu+1} \cup \{k\}$,
\begin{equation*}
\hat{g}^{(i)}(x, y) = g(x, y) + \phi(y; c_k^{\nu+1,k}, B^k, L_k^{\nu+1,k}, T_k^{\nu+1,k})
\end{equation*}
end if
end for
Set
\begin{equation*}
K = \{ k \in \hat{K}^{\nu+1} | \hat{g}^{\nu+1,k} (x, c^{\nu+1,k}) > 0 \}
\end{equation*}
\begin{equation*}
\cup \{ k \in \tilde{K}^{\nu+1} | \tilde{g}^{\nu+1,k} (x, y) > 0 \text{ for some } y \in B^k \}.
\end{equation*}
while $K \neq \emptyset$ do
for $k \in K$ do
Let $S^k$ be the barycenter of $B^k$.
\text{refine}_{hyb}(\nu + 1, S^k)
end for
Set
\begin{equation*}
K = \{ k \in \hat{K}^{\nu+1} | \hat{g}^{\nu+1,k} (x, c^{\nu+1,k}) > 0 \}
\end{equation*}
\begin{equation*}
\cup \{ k \in \tilde{K}^{\nu+1} | \tilde{g}^{\nu+1,k} (x, y) > 0 \text{ for some } y \in B^k \}.
\end{equation*}
end while

\textbf{Theorem 6.2.3.} Algorithm 14 terminates after finitely many steps.
Algorithm 14: Adaptive hybrid algorithm - aha

Choose $X \subset \mathbb{R}^n$ with $M \subset X$, $B = [\underline{b}, \overline{b}] \subset \mathbb{R}^m$ with $Y \subset B$ and compute $\alpha, L, \overline{L}$ on $B$.

Set $\nu = 0$, $X^0 = X$ and $L = \max(\|L\|_\infty, \|\overline{L}\|_\infty) \in \mathbb{R}$.

Determine a reduced outer approximation $\mathcal{B}_N^0$ of $Y$ with some $N \in \mathbb{N}$, $\alpha^0 \leq \alpha$ as well as $\underline{L}_k^0 \geq \underline{L}, \overline{L}^0_k \leq \overline{L}$ and $\epsilon_k^0$ on $B^k$, $k = 1, \ldots, N$, with sets

$$\hat{K}_0^0 = \{k \in \{1, \ldots, N\} | \frac{\alpha_k^0}{2} \|\hat{b} - \hat{y}_k\|_2 \geq (\overline{L}^0_k, \overline{b}^k - \epsilon_k^0)\},$$

$$\hat{K}_0 = \{1, \ldots, N\} \setminus \hat{K}_0^0,$$

so that $SIP_{hyb}(X^0, \mathcal{B}_N^0)$ is consistent.

Choose $\epsilon_{stat}, \epsilon_{act} > 0$ and $\epsilon_{act}, \epsilon_{act}^F, \epsilon_{act}^{UB} \in (0, \frac{1}{2})$ with $\epsilon_{act} \leq 2\epsilon_{act}/(\alpha \|\hat{b} - \hat{b}\|_2^2)$, $\epsilon_{act}^F \leq \epsilon_{act}/(L \|\hat{b} - \hat{b}\|_1)$.

Compute a stationary point $x^0$ of $SIP_{hyb}(X^0, \mathcal{B}_N^0)$ with $\epsilon_{act}$-active indices $\epsilon_k^0$, $k \in \hat{J}^0$, $y_k^0$, $k \in \hat{J}^0$ for an $(n+1)$-tuple $(\hat{J}^0, \tilde{J}^0)$ with \{J\} $\in K_{0,0}^{ac} x^0$ ,{J\} $\in K_{0,0}^{ac} x^0$, and $n+1$ non vanishing multipliers $(\kappa, (\hat{\lambda}_k)_{k \in \hat{J}^0}, (\hat{\lambda}_k)_{k \in \tilde{J}^0}, (\mu_t)_{t \in \{1, \ldots, n\}}, (\xi_r)_{r \in \{1, \ldots, n\}})$ by solving $P_{hyb}(X^0, \mathcal{B}_N)$.

while $x^\nu$ is not a stationary point of $SIP$ with $2\epsilon_{act}$-active, $\epsilon_Y$-feasible indices $\epsilon_k$, $k \in \tilde{J}^\nu, y_k^\nu$, $k \in \tilde{J}^\nu$, with respect to $\mathcal{B}_N^0$, and multipliers $(\kappa, (\hat{\lambda}_k)_{k \in \tilde{J}^\nu}, (\hat{\lambda}_k)_{k \in \tilde{J}^\nu}, (\mu_t)_{t \in \{1, \ldots, n\}}, (\xi_r)_{r \in \{1, \ldots, n\}})$ do

for $k = 1$ to $n+1$ do

$\text{refine}_{hyb}(\nu, y_k^\nu)$

end for

if $\nu > 1$ then

Set $\Delta x^\nu = \|x^\nu - x^{\nu-1}\|_\infty$.

$X_{\text{adapt}}_{hyb}(x^\nu, \Delta x^\nu)$

else

Set $X^1 = X$.

end if

Set $\nu = \nu + 1$.

Compute a stationary point $x^\nu$ of $SIP_{hyb}(X^\nu, \mathcal{B}_N^0)$ with $\epsilon_{act}$-active indices $\epsilon_k^\nu$, $k \in \tilde{J}^\nu, y_k^\nu$, $k \in \tilde{J}^\nu$ for an $(n+1)$-tuple $(\tilde{J}^\nu, \tilde{J}^\nu)$ with \{J\} $\in K_{0,0}^{ac} x^\nu$ ,{J\} $\in K_{0,0}^{ac} x^\nu$, and $n+1$ non vanishing multipliers $(\kappa, (\hat{\lambda}_k)_{k \in \tilde{J}^\nu}, (\hat{\lambda}_k)_{k \in \tilde{J}^\nu}, (\mu_t)_{t \in \{1, \ldots, n\}}, (\xi_r)_{r \in \{1, \ldots, n\}})$ by solving $P_{hyb}(X^\nu, \mathcal{B}_N)$.

end while
7 Implementation Details

In this chapter we give implementation details on the presented algorithms. First, in Section 7.1, we present some general information of the implementation and the system the algorithms were implemented on. After that, in Section 7.2, we give some details of the solving technique for the subproblems arising in the adaptive convexification algorithm and, possibly, in the hybrid method. In Section 7.3 we discuss the basic idea of a phase 1 algorithm. Two X-adaptation strategies are presented in Section 7.4. At last, in Section 7.5 we give some ideas how the complexity of the subproblems can be reduced. The techniques and algorithms described in this chapter are part of the implementation of the former algorithms. In the sequel let $B^N$ be some reduced outer approximation of $Y$, and $X^\nu \subset X$, $B^k \in B^N$ be some boxes.

7.1 General information

We implemented all presented algorithms in Matlab 7.10.0 (R2010a). The examples, especially those from Chapter 8, were run on a 2.4 GHz AMD Athlon 64 X2 processor with 4 GB RAM under Ubuntu 10.04 (lucid). As termination criteria for all algorithms we used the norm of the stationarity condition, and, additionally, we required that the change in the value of the objective function is less then some small constant and the norm of the change in the iterates is less then another small constant. The choice of these and all other constants for the numerical examples are specified there.

With the routines provided by the Matlab toolbox Intlab5.5, [43], we have a practical method to identify boxes $B^k$ in a reduced outer approximation whose intersection with the original index set $Y$ of SIP is empty. As $Y = \{ y \in \mathbb{R}^m \mid v_l(y) \leq 0, l \in \mathcal{L} \}$, one can evaluate $v_l(B^k)$ for each $l \in \mathcal{L}$ with the methods of Intlab. Then, if there is some $l^* \in \mathcal{L}$ with $\min v_l(B^k) > 0$ the intersection of $Y$ and $B^k$ is empty, and $B^k$ can be deleted. Later we will briefly discuss a similar method to identify boxes which are not needed during the optimization process.

Now we discuss the computation of the parameters $L$, $\overline{L}$ and $\alpha$ on the set $X^\nu \times B^k$. Note that the computation of the parameters on $X \times B^k$ or $X \times B$ is similar. Given a restriction $g(x, y)$ on a set $X^\nu \times B^k$ and the gradient with respect to the second argument, that is, $\nabla_y g(x, y)$, we evaluate $\nabla_y g(X^\nu, B^k) \in \mathbb{R}^m$ with the routines provided by the Matlab toolbox Intlab 5.5. Thus, we obtain lower and upper bounds for $\min_{(x, y) \in X^\nu \times B^k} \left( \frac{\partial}{\partial y_i} g(x, y) \right)$.
and $\min_{(x,y)\in X^\nu \times B^k} \left( \frac{\partial}{\partial g_{ij}} g(x, y) \right)$ for each $i \in \{1, \ldots, m\}$. Now, using the constant $\text{EPS}$ form Matlab which is the distance from 1.0 to the next larger double precision number, we set

$$L(X^\nu, B^k) = \left( \min_{i=1}^m \left( 0, -\sqrt{\text{EPS}} + \min_{j=1}^n \frac{\partial}{\partial y_{ij}} g(X^\nu, B^k) \right) \right)$$

and

$$L(X^\nu, B^k) = \left( \max_{i=1}^m \left( 0, \sqrt{\text{EPS}} + \max_{j=1}^n \frac{\partial}{\partial y_{ij}} g(X^\nu, B^k) \right) \right).$$

With this approach we can ensure the required conditions given by the relations (3.2).

As, for example, also explained in [14] the parameters $\alpha$ on $X^\nu \times B^k$ are computed using the Hessian with respect to the second argument of $g(x, y)$, that is, $D^2_y g(x, y)$, and Gerschgorin circles, as well as the Theorem of Gerschgorin. Let $A \in \mathbb{C}^{m \times m}$ be a matrix with $A = (a_{ij})_{ij}$ and $r_i = \sum_{j=1}^m |a_{ij}|-|a_{ii}|$. For $i \in \{1, \ldots, m\}$ the sets

$$B_{r_i}(a_{ii}) = \{a \in \mathbb{C} \mid |a - a_{ii}| \leq r_i\}$$

are called Gerschgorin circles. With the aid of these circles one can enclose the eigenvalues of $A$.

**Theorem 7.1.1** (Theorem of Gerschgorin, [19]). All eigenvalues of $A$ are contained in the set $\bigcup_{i=1}^m B_{r_i}(a_{ii})$.

Notice here that the Hessian matrix $D^2_y g(x, y)$ is symmetric, thus, all eigenvalues are real. To compute $\alpha$ on $X^\nu \times B^k$ we first evaluate $D^2_y g(X^\nu, B^k) \in \mathbb{R}^{m \times m}$ with the routines provided by the Matlab toolbox Intlab 5.5. Let $D^2_y G_{ij} = (D^2_y g(X^\nu, B^k))_{ij}$ denote the interval in the $i$-th row and $j$-th column of the interval matrix. By applying Theorem 7.1.1 to $D^2_y g(X^\nu, B^k)$ it is not hard to see that for the choice

$$\alpha = \max \left( 0, \sqrt{\text{EPS}} + \max_{i \in \{1, \ldots, m\}} \left( \max_{i \neq j} \left( D^2_y G_{ii} \right) + \sum_{i \neq j} \max \left( |D^2_y G_{ij}| \right) \right) \right)$$

we have $\alpha > \max \left( 0, \max_{(x,y)\in X^\nu \times B^k} \lambda_{\max} (x, y) \right)$.

The nonlinear subproblems arising in the adaptive reduction algorithms were solved using fmincon from the Matlab Optimization Toolbox Version 5.0 with default tolerances. If fmincon was not able to compute a Karush-Kuhn-Tucker point of a subproblem we used a filter SQP method described in [50, 51]. The subproblems arising in the adaptive convexification or the adaptive hybrid method are more complicated. The solution method is briefly discussed in the next section.
7.2 Regularization of MPCC

In this section let \( I, I^c, J \subset \mathbb{N} \) and \( f : \mathbb{R}^n \to \mathbb{R}, g_i : \mathbb{R}^n \to \mathbb{R} \) for \( i \in I \), \( c_1^i, c_2^i : \mathbb{R}^n \to \mathbb{R} \) for \( i \in I^c \), as well as \( h_j : \mathbb{R}^n \to \mathbb{R} \) for \( j \in J \) be some twice continuously differentiable functions. Further, let \( g = (g_i)_{i \in I}, c_1 = (c_1^i)_{i \in I^c}, c_2 = (c_2^i)_{i \in I^c} \) and \( h = (h_j)_{j \in J} \). In general the subproblems in the adaptive convexification algorithms have the form

\[
\text{MPCC} : \quad \min_x f(x) \quad \text{s.t.} \quad g(x) \leq 0 \\
\quad \quad \quad \quad h(x) = 0 \\
\quad \quad \quad \quad 0 \leq c_1^i(x) \perp c_2^i(x) \geq 0.
\]

That type of problem is called mathematical problem with complementarity constraints. It is well known, cf. [32], that this problem is non-smooth as a results of the constraints \( 0 \leq c_1^i(x) \perp c_2^i(x) \geq 0 \). We will illustrate that on the next example.

**Example 7.2.1.** Let \( I^c = \{1\} \), \( n = 2 \) and \( c_1^1(x) = x_1, c_2^1(x) = x_2 \). The constraints \( 0 \leq c_1^i(x) \perp c_2^i(x) \geq 0 \) can be equivalently reformulated as \( x_1 \geq 0, x_2 \geq 0, x_1x_2 = 0 \). The set given by these constraints are the nonnegative coordinate axis \( x_1, x_2 \), as Figure 7.1 illustrates. One can see that the set is almost everywhere smooth except for the point \( x_1 = x_2 = 0 \). Even though the non-smoothness of the feasible set of MPCC is limited to only finitely many points, we can not, in general, expect that solvers like SQP methods are able to solve them, since they require at least twice continuously differentiable functions.

For more details on the structure of the problem MPCC we refer to [27, 28, 32] and the references there in.

Solution methods for MPCC are discussed in several works. Some of the methods, like those introduced in [9, 44], use a smoothing or relaxing approach. Other methods use techniques from non-smooth optimization, cf. [10, 37]. A new method, first introduced in [46], uses a lifting approach. The method lifts the feasible set of MPCC into a higher
dimensional space so that it gets smooth. The solution method we used is a smoothing approach discussed in [44]. As a start we pick up Example 7.2.1 to illustrate the main ideas of the method. After that we give a brief outline of the used techniques.

**Example 7.2.2.** As seen in the latter example the set \( C = \{ x \in \mathbb{R}^2 \mid x_1 \geq 0, x_2 \geq 0, x_1 x_2 = 0 \} \) is almost everywhere smooth except for the point \( x_1 = x_2 = 0 \). Let \( C^t = \{ x \in \mathbb{R}^2 \mid x_1 \geq 0, x_2 \geq 0, x_1 x_2 = t^2 \} \) for \( t \geq 0 \). It is not hard to see that \( C^t \) is smooth for \( t \neq 0 \) and that \( C^0 = C \) holds. Thus, for small \( t > 0 \) the set \( C^t \) can be interpreted as an smooth approximation of \( C \). The sets \( C \) and \( C^t \) for some \( t > 0 \) are illustrated in Figure 7.2.

The main idea of the method presented in [44] is to compute Karush-Kuhn-Tucker points of a sequence of smooth approximations of MPCC with a sequence of smoothing parameters \( t \) tending to zeros. There it is shown that these sequence of stationary points tends to \( a \), in some kind, stationary point of the original problem.

Let \( c^1 (x) \ast c^2 (x) = (c^1_i (x) c^2_i (x))_{i \in I} \). With the idea introduced in Example 7.2.1 we obtain

\[
\text{MPCC}^t : \quad \min_x f(x) \quad \text{s.t.} \quad \begin{align*}
g(x) &\leq 0 \\
h(x) &= 0 \\
-c^1 &\leq 0 \\
-c^2 &\leq 0 \\
c^1 (x) \ast c^2 (x) &= t^2.
\end{align*}
\]

as a smooth approximation of MPCC for some \( t \neq 0 \). For numerical reasons, as discussed for example in [13], it is reasonable to replace the constraints \( -c^1 \leq 0, c^2 \leq 0, c^1 (x) \ast c^2 (x) = 0 \) in MPCC by a nonlinear complementarity function, NCP-function. A function \( \Phi : \mathbb{R}^2 \rightarrow \mathbb{R} \) that satisfies

\[
a, b \geq 0, ab = 0 \quad \text{if and only if} \quad \Phi (a, b) = 0
\]
7.2 Regularization of MPCC

is called NCP-function. An example for an NCP-function is the so called Fischer-Burmeister function, cf. [12],

$$\tilde{\Phi}_{FB}(a, b) = a + b - \sqrt{a^2 + b^2}.$$  

Naturally $\tilde{\Phi}_{FB}$ is non-differentiable for $a = b = 0$, but one can regularize it. The smoothed Fischer-Burmeister function is given by

$$\tilde{\Phi}_{FB}(t, a, b) = a + b - \sqrt{a^2 + b^2 + 2t^2}.$$  

Obviously, $\tilde{\Phi}_{FB}(t, a, b)$ is differentiable for each $t \neq 0$ and we have $\tilde{\Phi}_{FB}(0, a, b) = \tilde{\Phi}_{FB}(a, b)$. In [30] the following property was observed.

Lemma 7.2.3. For $t \neq 0$ we have

$$a, b \geq 0, \quad ab = t^2 \quad \text{if and only if} \quad \tilde{\Phi}_{FB}(t, a, b) = 0.$$  

Taking all that into account, and with the notations used in Chapter 4 and Section 5.2, we are now in the position to state the used solution method. We solved the problem $P_{\alpha BB}(B^N)$, respectively, $P_{\alpha BB}(X^\nu, B^N)$ by applying fmincon, or the former mentioned SQP method to the problems

$$P_{\alpha BB}(B^N) : \min_{\omega, t} F(\omega) \quad \text{s.t.} \quad G(\omega) \leq 0$$  

$$H(\omega) = 0$$  

$$\tilde{\Phi}_{FB}(t, C^1(\omega), C^2(\omega)) = 0$$  

$$e^t = 1$$

and

$$P_{\alpha BB}(X^\nu, B^N) : \min_{\omega, t} F(\omega) \quad \text{s.t.} \quad G(\omega) \leq 0$$  

$$H(\omega) = 0$$  

$$\tilde{\Phi}_{FB}(t, C^1(\omega), C^2(\omega)) = 0$$  

$$e^t = 1.$$  

Here we used

$$\tilde{\Phi}_{FB}(t, C^1(\omega), C^2(\omega)) = \left( \frac{\phi_{FB}(y^k - b^k)}{\phi_{FB}(\gamma^k, b^k - y^k)} \right)_{k=1,\ldots,N}.$$  

Note that we did not use a predefined sequence of smoothing parameters $t$. Instead we use $e^t = 1$ as an additional constraint, and, thus, let the nonlinear solver determine a value for $t$ in each iteration. It is not hard to see that in a feasible point for $P_{\alpha BB}(B^N)$ and $P_{\alpha BB}(X^\nu, B^N)$ we have $t = 0.$
7 Implementation Details

Figure 7.3: Illustration of the function \( g(x, y) = x^4 - 6x^2 + 8x - 6y^2 \) on \( X = [-2, 2], Y = [0, 1] \). The dots mark stationary points of \( EPI_{SIP} \).

7.3 A phase 1 algorithm

In this section, we discuss a phase 1 for the presented algorithms. That is, a method to find a feasible point of the problem as starting point for the main algorithms. Let us recall that a point \( \mathbf{x}^* \) is feasible for \( SIP \) if the relation \( g(\mathbf{x}^*, y) \leq 0 \) holds for each \( y \in Y \). By an epigraph reformulation, we obtain that this is equivalent to \( g(\mathbf{x}^*, y) \leq z^* \) holds for each \( y \in Y \), for some \( z^* \leq 0 \). Thus, we have that \( \mathbf{x}^* \) is feasible for \( SIP \) if for a solution \( (\mathbf{x}^*, z^*) \) of the problem

\[
EPI_{SIP} : \quad \min_{(x, z) \in X \times \mathbb{R}} z \quad s.t. \quad g(x, y) \leq z \quad \text{for all } y \in Y
\]

\[ \quad z \geq -\tilde{\varepsilon}_{epi}. \]

the relation \( z^* \leq 0 \) holds for some \( \tilde{\varepsilon}_{epi} > 0 \). We add the additional constraint \( z \geq -\tilde{\varepsilon}_{epi} \) to the problem to ensure that \( z \) is bounded below. Thus, we may find a feasible starting point by applying the algorithms, as they are, at first to a problem that is reformulated in that way. However, there may occur a problem. As the solution methods for the subproblems, no matter which algorithm is used, compute stationary points and not global optima of a problem, the algorithms may get stuck in a local minimum of \( EPI_{SIP} \) with a positive optimal value. That means the algorithms may not find a feasible point and break, even though there exist feasible points. We illustrate that in the next example.

Example 7.3.1. Let \( Y = [0, 1], X = [-2, 2] \) and \( g(x, y) = x^4 - 6x^2 + 8x - 6y^2 \). It is not hard to see that the point \( (\mathbf{x}^*, z^*) = (1, 3) \) with \( \kappa = \lambda = 1 \) is stationary in the sense of John but not feasible, cf. Figure 7.3. Thus, our algorithms may get stuck in that point, even though there are feasible points.

In fact, the discussed problem is a basic phenomenon in nonlinear optimization and can be avoided with techniques, and computationally expensive algorithms from global optimization. In spite of that problem, we apply the algorithms to reformulated problems.
In this section we introduce two techniques for the adaptation of the set $X$, that is, constructing the sequence $(X^\nu)_\nu$ of subsets of $X$ satisfying Condition 5.1.7. Let $X^\nu \subset X \subset \mathbb{R}^n$ and $x^{\nu-1}, x^\nu \in X^\nu$ be two iterates of one of the presented algorithms so that $x^{\nu-1}$ is the barycenter of $X^\nu$.

We obtain a possible strategy to construct $X^{\nu+1}$ by choosing the box with barycenter $x^\nu$ and side lengths $2\delta\|x^\nu - x^{\nu-1}\|_{\infty}$ for some $\delta > \sqrt{n}$. As $x^\nu \in \text{int} \ (X^{\nu+1})$ is true Condition 5.1.7 (i) is satisfied. To see that Condition 5.1.7 (ii) is satisfied, notice first that for $x^\nu \in \partial X^\nu$ and $\delta > \sqrt{n}$ the demanded side lengths imply that there is some constant $\tilde{c} > 0$ so that the relation $(\sqrt{n} + \tilde{c}) \|x^\nu - x^\nu\|_{\infty} \leq \|x^{\nu+1} - x^{\nu+1}\|_{\infty}$ holds for each $\nu$. Thus, Condition 5.1.7 (ii) is a consequence of the relation

$$\frac{\|x^\nu - x^\nu\|_2 + \tilde{c}\|x^\nu - x^\nu\|_{\infty}}{\sqrt{n} + \tilde{c}} \leq \frac{\|x^{\nu+1} - x^{\nu+1}\|_\infty}{\|x^{\nu+1} - x^{\nu+1}\|_\infty} \leq \frac{\|x^{\nu+1} - x^{\nu+1}\|_\infty}{\|x^{\nu+1} - x^{\nu+1}\|_2}.$$ 

Since $\Delta x^\nu = \|x^\nu - x^{\nu-1}\|_{\infty}$ may become small we have to bound the side lengths below by some constant to ensure Condition 5.1.7 (iii). More precisely we set

$$X^{\nu+1}_i = \left[ \max \{x^\nu_i - 2\delta \max (\Delta x^\nu_i, \epsilon_X), x^\nu_i\} \right] \left[ \min \{x^\nu_i + 2\delta \max (\Delta x^\nu_i, \epsilon_X), x^\nu_i\} \right] \quad (7.1)$$

for $i = 1, \ldots, n$ and some $\epsilon_X > 0$. With that choice it is not hard to see that Condition 5.1.7 (iv) is also satisfied.

Another strategy inspired by the first one is to move the box, constructed like in (7.1), into the direction of steepest descent for the objective function $f$. That is,

$$X^{\nu+1}_i = \left[ \max \{x^\nu_i - 2\delta \max (\Delta x^\nu_i, \epsilon_X), x^\nu_i\} \right] \left[ \min \{x^\nu_i + 2\delta \max (\Delta, \epsilon_X), x^\nu_i\} \right] \quad (7.2)$$
for \( i = 1, \ldots, n, \epsilon_x > 0 \), and

\[
\Delta x^\nu = \max \left( 1 - \frac{\nabla f(x)}{\|\nabla f(x)\|_\infty}, \epsilon_X \right) \|x^\nu - x^{\nu-1}\|_\infty
\]

\[
\overline{\Delta} x^\nu = \max \left( 1 + \frac{\nabla f(x)}{\|\nabla f(x)\|_\infty}, \epsilon_X \right) \|x^\nu - x^{\nu-1}\|_\infty.
\]

As for the former strategy it is not hard to see that this choice also satisfies Condition 5.1.7.

As mentioned in the latter section, we can neglect the condition \( \delta > \sqrt{n} \) for a phase 1 algorithm with \( X \)-adaptation.

### 7.5 Reduction of complexity

This section starts, with an implementable method to perform monotonicity tests of the constraints in the second argument. After that, we give a practical method to identify boxes not needed during the iterations. Notice that we only look at constraints \( g \) on sets \( X^\nu \times B^k \). A generalization to other box shaped sets is straightforward.

We start our discussion with the overestimator used for the adaptive convexification algorithm, cf. Chapter 4 and Section 5.2. Let us recall that for a given constraint \( g \) of \( \text{SIP} \) the algorithms introduced in Chapter 4 and Section 5.2 constructed concave overestimators for \( g \) with respect to the second argument of the form

\[
\tilde{g}^{\nu,k}: X^\nu \times B^k \longrightarrow \mathbb{R}, \quad (x, y) \mapsto g(x, y) + \psi(y; \alpha^\nu_k, B^k).
\]

The constant \( \alpha^\nu_k \) was chosen so that \( \alpha^\nu_k > \max \{0, \max_{(x,y) \in X^\nu \times B^k} \lambda_{\max}(x, y)\} \), where \( \lambda_{\max}(x, y) \) denotes the maximal eigenvalue of \( D^2_{yy} g(x, y) \) and \( \psi(y; \alpha^\nu_k, B^k) = \frac{\alpha^\nu_k}{2}(y - \frac{b^k}{2}, b^k - y) \). Let us assume for the moment that \( \frac{\partial}{\partial y} g(x, y) > 0 \) on \( X^\nu \times B^k \) for some \( i \in \{1, \ldots, m\} \), and that \( g \) is not concave in the second argument. Then it is clear that the coordinate \( y_i \) of the maximum of \( g \) with respect to the second argument on \( B^k \) is \( \overline{y}_i \).

Though we overestimate \( g \) on the whole set by our adaptive convexification algorithm, and, thus, also in coordinate direction \( y_i \) and make an unnecessary error. Before we continue we give an illustrative example.

**Example 7.5.1.** Let \( X^\nu = [0, 1] \), \( B^k = [-1, 1] \) and \( g(x, y) = x^2 + y^2 \sin (y) + y \). It is not hard to see that \( g \) is monotonically increasing and not concave in the second argument.

With \( \alpha = 4 \) we obtain \( \tilde{g}^{\nu,k}(x, y) = x^2 + y^2 \sin (y) + y + 2(y + 1)(1 - y) \) as a concave overestimator for \( g \). As one can see in Figure 7.4 the maximum of \( \tilde{g}^{\nu,k} \) in \( y \) does not coincide with the one of \( g \) for arbitrary \( x \in X^\nu \). Moreover, we make an unnecessary error by overestimating \( g \). However, it is not hard to see that the function \( g(x, 1) \) is also an overestimator for \( g \) on the whole set and coincides with the maximum of \( g \) with respect to \( y \) for each \( x \).
7.5 Reduction of complexity

Figure 7.4: Illustration of the function $g(x, y) = x^2 + y^2 \sin(y) + y$ and its overestimator $\tilde{g}^{\nu,k}$. The blue line are the maxima of $g$ with respect to the second argument and the yellow the maxima of $\tilde{g}^{\nu,k}$.

A generalization of the latter observation is straightforward. It is not hard to see that $\tilde{g}^{\nu,k}$ is still a concave overestimator for $g$ on $X^\nu \times B^k$ if we fix those $y_i$ to $b_i^k$ for which $\min_{(x,y) \in X^\nu \times B^k} \nabla_y g(x, y) \geq 0$ is satisfied and to $\bar{b}_i^k$ for which $\max_{(x,y) \in X^\nu \times B^k} \nabla_y g(x, y) \leq 0$ is true. Now, we can modify the Algorithms 3 and 8 for $SIP_{\alpha BB}$ to include a monotonicity test. In Algorithm 15 we only state the new splitting procedure for the case a X-adaptation is used. It also describes how the feasible set of a refined problem of $SIP_{\alpha BB}$ is constructed. The monotonicity test is stated in Algorithm 16. The computation of the bounds for $\min_{(x,y) \in X^\nu \times B^k} \nabla_y g(x, y)$ and $\max_{(x,y) \in X^\nu \times B^k} \nabla_y g(x, y)$ is performed with the routines provided by the Matlab toolbox Intlab.

For the overestimator used for the adaptive reduction algorithm, cf. Chapter 3 and Section 5.1, we do not need to introduce a monotonicity test. From the calculation of the $i$-th component of the center $c_i$ and the choice of $\underline{L}_i$, $\overline{L}_i$ on the corresponding sets for each $i \in \{1, \ldots, m\}$, we have that $c_i = \overline{b}_i^k$ if $\min_{(x,y) \in X^\nu \times B^k} \nabla_y g(x, y) \geq 0$ and $c_i = \underline{b}_i^k$ if $\max_{(x,y) \in X^\nu \times B^k} \nabla_y g(x, y) \leq 0$ is true.

With another useful observation and the routines provided by Intlab one can try to reduce the number of boxes in a reduced outer approximation of $Y$ in a problem $SIP_{UF}$ and $SIP_{\alpha BB}$, even if the intersection of a box and the original index set is not empty. Here we have to distinguish the cases for $X$ and $X^\nu$.

Let there be some $B^k$ in the reduced outer approximation of $Y$ so that the relation $g(x, y) < 0$ holds for each $(x, y) \in X \times B^k$. That means $g$ does not have any active index on $B^k$ for each $x \in X$. As $Y \cap B^k$ is a subset of $B^k$, we also have that $g(x, y) < 0$ holds for each $(x, y) \in X \times (Y \cap B^k)$. Since the feasible set $M$ of $SIP$ is a subset of $X$, $g$ does not constrain $M$ for indices in $B^k$. Thus, $B^k$ can be deleted from the reduced...
Algorithm 15 Splitting step with monotonicity test - $X_{\text{refine}_{\alpha BB}}(\eta)$

Let $x \in X^\nu$, $\eta \in B^{k^*}$, $k^* \in \{1, \ldots, N\}$, and let $S^{k^*}$ be the barycenter of $B^{k^*}$ and $Q^{k^*} = \min_{i \in P^{k^*}} \left( \frac{b_i^{k^*} - b_j^{k^*}}{\|b_i^{k^*} - b_j^{k^*}\|_\infty} \right)$.

if $\eta \not\in Y$ or $Q^{k^*} < \epsilon$ then
  Set $\eta = S^{k^*}$.
end if

if $P^{k^*} \neq \emptyset$ then
  Compute $(B^{k^*,(1)}, B^{k^*,(2)}) = S(B^{k^*}, \eta)$.
  Compute $\alpha^\nu_{k^*,(1)}$, $\alpha^\nu_{k^*,(2)} \leq \alpha_{k^*}$ on $X^\nu \times B^{k^*,(1)}$, $X^\nu \times B^{k^*,(2)}$.
  \(X_{\text{testmono}_{\alpha BB}}(X^\nu, B^{k^*,(1)}, B^{k^*,(2)})\).
  for $j = 1$ to $2$ do
    Set
    \[
    g^\nu_{k^*,(j)}(x, y) = g(x, y) + \frac{\alpha^\nu_{k^*,(j)}}{2} (y - b^{k^*,(j)} - b^{k^*,(j)}) - y
    \]
    \[
    M^{(j)} = \{ x \in X^\nu \mid g^\nu_{k^*,(1)}(x, y) \leq 0 \text{ for all } y \in B^{k^*,(1)} \}
    \]
    with fixed $y_i = \tilde{b}^{k^*,(j)}$ for $i \in P^{(j)}$ and $y_i = b^{k^*,(j)}$ for $i \in \mathcal{P}^{(j)}$.
  end for
  Set $M = \{ x \in X^\nu \mid g^\nu_{k^*,(j)}(x, y) \leq 0 \text{ for all } y \in B^k, k \in \{1, \ldots, N\} \setminus \{k^*\} \}$
  if $Y \cap B^{k^*,(1)} \neq \emptyset$ and $Y \cap B^{k^*,(2)} \neq \emptyset$ then
    Replace $B^{k^*}$ by $B^{k^*,(1)}$ and $B^{k^*,(2)}$ and replace $\alpha_{k^*}^\nu$ by $\alpha_{k^*}^\nu_{(1)}$ and $\alpha_{k^*}^\nu_{(2)}$.
    Set $M_{\alpha BB}(X^\nu, B^N) = M \cap M^{(1)} \cap M^{(2)}$.
    Set $N = N + 1$.
  else if $Y \cap B^{k^*,(1)} \neq \emptyset$ then
    Replace $B^{k^*}$ by $B^{k^*,(1)}$ and $\alpha_{k^*}^\nu$ by $\alpha_{k^*}^\nu_{(1)}$.
    Set $M_{\alpha BB}(X^\nu, B^N) = M \cap M^{(1)}$.
  else if $Y \cap B^{k^*,(2)} \neq \emptyset$ then
    Replace $B^{k^*}$ by $B^{k^*,(2)}$ and $\alpha_{k^*}^\nu$ by $\alpha_{k^*}^\nu_{(2)}$.
    Set $M_{\alpha BB}(X^\nu, B^N) = M \cap M^{(2)}$.
  else
    Delete $B^{k^*}$ and $\alpha_{k^*}^\nu$.
    Set $M_{\alpha BB}(X^\nu, B^N) = M$.
    Set $N = N - 1$.
end if
7.5 Reduction of complexity

Algorithm 16 Monotonicity test - $X_{test\ mono_{\alpha BB}} (X, B^{(1)}, B^{(2)})$

Set $P^{(1)} = P^{(2)} = P^{(1)} = P^{(2)} = \emptyset$.

for $i = 1$ to $m$ do

for $j = 1$ to 2 do

Compute lower bounds for $L_i^{(j)}$ for $\min_{(x,y) \in X \times B^{(j)}} \frac{\partial}{\partial y} g(x, y)$.

Compute upper bounds for $U_i^{(j)}$ for $\max_{(x,y) \in X \times B^{(j)}} \frac{\partial}{\partial y} g(x, y)$.

if $L_i^{(j)} \geq 0$ then

Set $P_i^{(j)} = P^{(j)} \cup \{i\}$.

else if $L_i^{(j)} \leq 0$ then

Set $P_i^{(j)} = P^{(j)} \cup \{i\}$.

end if

end for

end for

Though we are in a slightly different situation if the relation $g(x, y) < 0$ holds only for each $(x, y) \in X^\nu \times B^k$. In this case we can only ensure that $g$ does not constrain $X^\nu \cap M$ for indices in $B^k$, but we can not ensure that for the whole set $M$. In any case, we can ensure that $B^k$ is temporarily not needed in the problems $SIP_{UF} (X^\nu, B^N)$ and $SIP_{\alpha BB} (X^\nu, B^N)$. Thus, we can ignore this box temporarily in our optimization processes. However, if the set $X^\nu$ changes during the iterations one has to check again whether $g$ is negative, or not, for each point $(x, y)$ on the new set.
8 Numerical examples

In this chapter we present some numerical examples to give an impression of the performance of the algorithms and the interaction of the used concepts. The first example is the Chebyshev approximation problem $CA$ given in Section 3.4 and 4.4. On that problem we consider the behavior of the algorithms and the impact of the additional $X$-adaptation strategies. After that we investigate two illustrative examples from design centering with $Y \subset \mathbb{R}^2$. On the first problem we also consider the behavior of the algorithms and the impact of the additional $X$-adaptation strategies. Then, on the second problem, we will briefly discuss the numerical behavior of the reduced outer approximation of the index set. At last we give another design centering problem with $Y \subset \mathbb{R}^3$.

In the sequel we will denote the maximum separation distance of a constraint and the corresponding relaxation, cf. Lemma 3.1.2 and 4.1.2, shortly as error of a relaxation. As termination criteria for all algorithms on each problem we used the norm of the stationarity condition, the feasibility of the indices, and, additionally, we required that the change in the value of the objective function is less than $10^{-3}$ and the norm of the change in the iterates is less then $10^{-2}$.

We recall that the semi-infinite reformulation of the Chebyshev approximation problem introduced in Section 3.4 was

$$
SIP_{CA} : \min_{x \in \mathbb{R}^4} x_4 \quad s.t. \quad \sin(\pi y) - \left(x_3 y^2 + x_2 y + x_1\right) - x_4 \leq 0, \; y \in [0, 1]$

$$
- \sin(\pi y) + x_3 y^2 + x_2 y + x_1 - x_4 \leq 0, \; y \in [0, 1].$

with $X = [-1, 1] \times [3, 5] \times [-5, -3] \times [0, 3]$ and $B = Y = [-1, 1]$. As termination tolerances on the stationarity and on the activities of the constraints we choose $\epsilon = \epsilon_{act} = 10^{-3}$. As the index set can be handled exactly by the algorithms we do not need some tolerance $\epsilon_Y$ on the feasibility of the indices.

For Algorithm 7 we set $\epsilon_{split} = \min \left( \frac{\epsilon_{act}}{\max(\|L\|_{\infty}, \|L\|_{\infty})\|b-b_i\|_1}, \frac{1}{2} - \epsilon_Y \right)$, and for Algorithm 10 we set $\epsilon_{split} = \min \left( 2\epsilon_{act} \min \left( 1, \frac{1}{\alpha}\|b-b_i\|_2 \right), \frac{1}{2} - \epsilon_Y \right)$. As a starting point for both algorithms and each $X$-adaptation strategy we used $x^0 = (1, 5, -3, 2)$. In the phase 1 procedure we choose $\delta = 1.1$, and after the algorithms found a feasible starting point we used $\delta = 2 + \sqrt{\text{EPS}}$ for the $X$-adaptations. The constant $\text{EPS}$ form Matlab is the distance from 1.0 to the next larger double precision number.

With Algorithm 7 and the adaptation strategy (7.1) we obtained the feasible starting
8 Numerical examples

Figure 8.1: The overall number of boxes and the overall number of boxes which may contain active indices generated by Algorithm 7 for the problem SIPCA. In a) with the X-adaptation strategy given by (7.1), and in b) with the X-adaptation strategy given by (7.2).

point

\[ x^{13} = (0.73025, 4.7303, -3.2697, 2.2697)^T \]

after 13 iterations (4.76 CPU seconds). After 41 more iterations (717.75 CPU seconds) the algorithm terminates with the point

\[ x^* = (-0.028054, 4.0006, -4.0006, 0.028054)^T \]

and the objective value 0.028054. At \( x^* \) the norm of the stationarity condition is less then \( 10^{-16} \), the change in the value of the objective function is \( 9.5774 \cdot 10^{-5} \), and the norm of the change in the iterates is 0.0018846.

And with Algorithm 7 and the adaptation strategy (7.2) we obtained the feasible starting point

\[ x^{14} = (0.69228, 4.6923, -3.3077, 2.154)^T \]

after 14 iterations (4.27 CPU seconds). After 29 more iterations (1557.21 CPU seconds) the algorithm terminates with the point

\[ x^* = (-0.028136, 4.002, -4.002, 0.028136)^T \]

and the objective value 0.028136. At \( x^* \) the norm of the stationarity condition is less then \( 10^{-16} \), the change in the value of the objective function is \( 2.8488 \cdot 10^{-4} \), and the norm of the change in the iterates is 0.0062716.

As a start one can see that the results from Algorithm 2 and Algorithm 7 with both adaptation strategies differ only marginally, while the latter algorithms perform better then Algorithm 2 after a feasible starting point is found, but worse in the phase 1. For that example, the adaptation strategy given by (7.1) performs better than the one given by (7.2).
Figure 8.2: Results for Algorithm 7 for the problem $SIP_{\mathcal{X}_A}$. a) The maximum and minimum diameter of the boxes containing active indices with the $X$-adaptation strategy given by (7.1). b) The green line is the maximum of the constants $L, \overline{L}$ on the boxes containing active indices and the black line is the maximal error of the relaxations with the $X$-adaptation strategy given by (7.1). c) The maximum and minimum diameter of the boxes containing active indices with the $X$-adaptation strategy given by (7.2). d) The green line is the maximum of the constants $L, \overline{L}$ on the boxes containing active indices and the black line is the maximal error of the relaxations with the $X$-adaptation strategy given by (7.2).

A comparison of the overall number of boxes generated during the iteration by Algorithm 7 which may contain active indices, cf. Figure 8.1, and the overall number of boxes generated during the iteration by Algorithm 2, cf. Figure 3.5, shows a reason for the difference in the runtime. We have to mention that in Algorithm 2 none of the generated boxes could be deleted during the iterations, since, by our test, each box may contain an active index. For the algorithm with the additional $X$-adaptation the number of boxes which have to be taken into account is smaller than for the other algorithm, and, thus, the subproblems become smaller. Figure 8.2 displays the diameter of boxes generated during the iterations, which may contain active indices, and the maximal error made by the relaxations in Algorithm 7, as well as the maximum values for the constants $L, \overline{L}$. It can be seen that the diameter of boxes generated during the iterations, which may contain active indices, and the error made by the relaxations in Algorithm 7, as well as the values for $L, \overline{L}$ are also smaller than for Algorithm 2. With the $X$-adaptation strategies one can reduce the maximum separation distance of the relaxation and the original constraint, and, thus, the number of boxes needed in the iterations is reduced, and the subproblems become smaller.

Furthermore one can see that there are jumps in the number of generated boxes. The cause of these jumps can, again, be found in the $X$-adaptation procedure. If an iterate is not feasible for a relaxed problem with a constructed subset of $X$, then the reduced
Numerical examples

outer approximation is refined until the point is feasible for the new problem, and, thus, a large number of boxes might be generated.

Next we consider Algorithm 10. With the adaptation strategy (7.1) the algorithm finds the feasible starting point

\[ x^{13} = (0.73025, 4.7303, -3.2697, 2.2697)^T \]

after 13 iterations (4.57 CPU seconds). After 14 iterations (7.87 CPU seconds) the algorithm terminates with the point

\[ x^* = (-0.028005, 4, -4, 0.028005)^T \]

and the objective value 0.028005. At \( x^* \) the norm of the stationarity condition is \( 2.6122 \cdot 10^{-7} \), the change in the value of the objective function is \( 1.5959 \cdot 10^{-16} \), and the norm of the change in the iterates is \( 5.0351 \cdot 10^{-15} \).

Algorithm 10 with the adaptation strategy (7.2) identified the feasible starting point

\[ x^{14} = (0.69228, 4.6923, -3.3077, 2.154)^T \]

after 14 iterations (4.78 CPU seconds). After 9 iterations (7.52 CPU seconds) the algorithm terminates with the point

\[ x^* = (-0.028005, 4, -4, 0.028005)^T \]

and the objective value 0.028005. At \( x^* \) the norm of the stationarity condition is \( 3.5525 \cdot 10^{-9} \), the change in the value of the objective function is less than \( 10^{-16} \), and the norm of the change in the iterates is \( 1.9898 \cdot 10^{-15} \).

For Algorithm 10 we can observe the same effect as for Algorithm 7, that is, in comparison to Algorithm 4 both adaptation strategies lead to a worsening of the runtime in the phase 1 of the algorithm. An indication for the cause can be found in the overall number of boxes which may contain active indices, cf. Figure 8.3.

In the phase 1 none of the generated boxes contain an active index. That means none of the relaxed nonlinear constraints become active. A too small value for \( \delta \) can be the only reason for that outcome, since the feasible set of the relaxed problems is given by these nonlinear constraints and the subset of \( X \) constructed by the additional adaptation strategy. That is additionally supported by the results displayed in Figure 4.5. As none of the boxes contains an active index the diameters stay unchanged during the first iterations, and the error made by the relaxations stays nearly unchanged. However, with a large value for \( \delta \) one can enlarge the subset of \( X \) so that the feasible set of the relaxed problem is in its interior, and, thus, the subproblems in Algorithm 10 and Algorithm 4 coincide. Nevertheless, after a feasible point is found Algorithm 10 with both adaptation strategies performs better than Algorithm 4.

Before we investigate Algorithm 14 on the problem \( SIP_{CA} \), we illustrate the latter remark. That is, another choice for \( \delta \) in the algorithms should intend to improve the
Figure 8.3: The overall number of boxes and the overall number of boxes which may contain active indices generated by Algorithm 10 for the problem $SIP_{CA}$. In a) with the $X$-adaptation strategy given by (7.1), and in b) with the $X$-adaptation strategy given by (7.2).

Figure 8.4: Results for Algorithm 10 for the problem $SIP_{CA}$. a) The maximum and minimum diameter of the boxes containing active indices with the $X$-adaptation strategy given by (7.1). b) The green line is the maximum of the constants $\alpha$ on the boxes containing active indices and the black line is the maximal error of the relaxations with the $X$-adaptation strategy given by (7.1). c) The maximum and minimum diameter of the boxes containing active indices with the $X$-adaptation strategy given by (7.2). d) The green line is the maximum of the constants $\alpha$ on the boxes containing active indices and the black line is the maximal error of the relaxations with the $X$-adaptation strategy given by (7.2).
8 Numerical examples

runtime. We consider, again, Algorithm 10 with the $X$-adaptation (7.1) on the problem $SIP_{CA}$. We leave all termination tolerances and constants, except for $\delta$, unchanged. In the phase 1 we choose $\delta = 100$ and after a feasible starting point is identified by the Algorithm we set $\delta = 2 + \sqrt{\text{EPS}}$.

The algorithms identified the feasible starting point

$$x^1 = (0.90748, 4, -3.9935, 2.0925)^T$$

after 1 iterations (1.02 CPU seconds). After 6 more iterations (8.62 CPU seconds) the algorithm terminates with the point

$$x^* = (-0.028005, 4, -4, 0.028005)^T$$

and the objective value 0.028005. At $x^*$ the norm of the stationarity condition is $5.6842 \cdot 10^{-13}$, the change in the value of the objective function is $3.6312 \cdot 10^{-13}$, and the norm of the change in the iterates is $5.1353 \cdot 10^{-13}$.

One can see that these results coincide with latter remark, and that for this example both adaptation strategies for $X$ perform nearly the same in Algorithm 10. In a first summary we have, from the latter results, that there occur two opposing effects in the $X$-adaptation. If, on the one hand, $\delta$ for the $X$-adaptation is chosen to small the size of the computed step may become small and the algorithm needs more iterations to compute a stationary point. On the other hand, with a smaller value for $\delta$ the feasible set of a subproblem can be reduced, and, thus the number of boxes in a reduced outer approximation, as well as the values for $L$, $\overline{L}$ and $\alpha$ may become smaller. As discussed in Section 7.4, we have a lower bound for $\delta$ to ensure that stationary points for the original problem can be computed, but we do not have an upper bound, and in the phase 1 algorithm the lower bound can also be dropped.

At last we investigate Algorithm 14 on the problem $SIP_{CA}$. Here we only test the algorithm with the additional $X$-adaptation.

With the adaptation strategy (7.1) Algorithm 14 computed the feasible starting point

$$x^{13} = (0.73025, 4.7303, -3.2697, 2.2697)^T$$

after 13 iterations (5.65 CPU seconds). After 11 more iterations (12.95 CPU seconds) the algorithm terminates with the point

$$x^* = (-0.028005, 4, -4, 0.028005)^T$$

and the objective value 0.028005. At $x^*$ the norm of the stationarity condition is $2.1104 \cdot 10^{-16}$, the change in the value of the objective function is $3.335 \cdot 10^{-4}$, and the norm of the change in the iterates is 0.007383.

And with the adaptation strategy (7.2) the algorithm identified the feasible starting point

$$x^{14} = (0.69228, 4.6923, -3.3077, 2.154)^T$$
after 14 iterations (5.48 CPU seconds). After 12 iterations (10.81 CPU seconds) the algorithm terminates with the point

\[ x^* = (-0.028005, 4, -4, 0.028005)^T \]

and the objective value 0.028005. At \( x^* \) the norm of the stationarity condition and the change in the value of the objective function are less than \( 10^{-16} \). The norm of the change in the iterates is \( 1.9159 \cdot 10^{-15} \). The results from Algorithm 14 with both \( X \)-adaptation strategies differ only slightly from the results obtained by the former algorithms.

One can see that the needed computational time is lower than for Algorithm 2 and Algorithm 7, while it is slightly worse than for Algorithm 4 and Algorithm 10. As in each iteration the constants \( L, L \) and \( \alpha \) must be computed for each box in the tessellation, the difference in the computational time of Algorithm 4, 10 and Algorithm 14 might be a result of the computational overhead.

That is supported by the results displayed in Figure 8.5. It can be seen that for both adaptation strategies the overall number of generated boxes and the number of boxes which may contain active indices differs only slightly from that one of Algorithm 4 and Algorithm 10. Moreover, one can observe the same effect as in the latter algorithms for the number of boxes which may contain active indices. That is, at the beginning the number these boxes is zero. Almost on all boxes the algorithm chooses the concave relaxation for the constraints, except for the last iterations. In the last iterations the diameter of some boxes is that small, cf. Figure 8.6, that the corresponding constraints are monotonic in the second argument, and, thus, the algorithm does not relax these constraints.

The next examples we investigate are taken from the class of design centering problems, cf. Chapter 1.
8 Numerical examples

Figure 8.6: Results for Algorithm 14 for the problem $SIP_{CA}$. a) The maximum and minimum diameter of the boxes containing active indices with the $X$-adaptation strategy given by (7.1). b) The green line is the maximum of the constants $\alpha$, the yellow line the maximum of $L$, $L$ on the boxes containing active indices and the black line is the maximal error of the relaxation with the $X$-adaptation strategy given by (7.1). c) The maximum and minimum diameter of the boxes containing active indices with the $X$-adaptation strategy given by (7.2). d) The green line is the maximum of the constants $\alpha$, the yellow line the maximum of $L$, $L$ on the boxes containing active indices and the black line is the maximal error of the relaxation with the $X$-adaptation strategy given by (7.2).

Figure 8.7: Illustration of the container $C$ for $SIP_{DC1}$. 
For the first example we consider the container
\[ C = \{ x \in \mathbb{R}^2 | c_1(x) = (x_1 - \frac{1}{2})^2 - x_2 - 1 \leq 0, c_2(x) = x_1^2 + x_2 \leq 0 \}, \]
cf. Figure 8.7, and the design
\[ D(x) = \{ z \in \mathbb{R}^2 | (\cos(x_3)\frac{z_1 - x_1}{x_3} - \sin(x_3)\frac{z_2 - x_2}{x_3}) \]
\[ + (\sin(x_3)\frac{z_1 - x_1}{x_4} + \cos(x_3)\frac{z_2 - x_2}{x_4})^2 \leq 1 \}, \]
that is, an ellipse with free center \((x_1, x_2)^T\) and axes \(x_3, x_4\) in arbitrary position. To maximize the area of \(D(x)\) we put \(f(x) = x_3x_4\). As a parametrization for \(D(x)\) we use
\[ D(x) = \{ z(x, y) | y_1^2 + y_2^2 \leq 1 \} \]
with
\[ z(x, y) = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} \cos(x_3) \\ \sin(x_3) \end{pmatrix} y_1 \begin{pmatrix} x_3 \\ x_4 \end{pmatrix} + \begin{pmatrix} -\sin(x_3) \\ \cos(x_3) \end{pmatrix} y_2 \begin{pmatrix} x_3 \\ x_4 \end{pmatrix}. \]
Thus, the constraint \(D(x) \subset C\) is equivalent to
\[ g_i(x, y) = c_i(z(x, y)) \leq 0 \text{ for all } y \in Y = \{ y \in \mathbb{R}^2 | y_1^2 + y_2^2 \leq 1 \}, i \in \{1, 2\} \]
and we arrive at the semi-infinite problem
\[ \text{SIP}_{DC^1} : \min_{x \in \mathbb{R}^5} -x_3x_4 \quad \text{s.t.} \quad g_i(x, y) \leq 0 \quad \text{for all } y \in Y, i \in \{1, 2\}. \]

For this example we show how a box \(X\) can be constructed that contains the feasible set of \(\text{SIP}_{DC^1}\). First note that the rotation variable \(x_5\) takes values in the interval \([-\pi, \pi]\). Since \(C\) and, thus, \(D(x)\) are contained in the box \([-\frac{1}{2}, 1] \times [-\frac{3}{2}, \frac{1}{2}]\), it is not hard to see that the radii \(x_3, x_4\) take values in \([0, 1]\). Thus, each feasible point of \(\text{SIP}_{DC^1}\) is contained in the box \(X = [x, \pi]\) with \(x = (-\frac{1}{2}, -\frac{3}{2}, 0, 0, -\pi)^T\) and \(\pi = (1, 1, 1, 1, \pi)^T\). As an initial approximation \(B\) for the index set \(Y\) we choose \(B = [\overline{b}, \overline{b}]\) with \(\overline{b} = (-1, -1)^T\) and \(\overline{b} = (1, 1)^T\). For all algorithms we choose the midpoint of \(X\) as the initial point \(x^0\), and we use the tolerances \(\epsilon = 10^{-2}, \epsilon_{act} = 10^{-3}, \epsilon_{Y} = 0.035\). Since the starting point \(x^0\) is not feasible, a phase 1 of the algorithms is performed to find a feasible point.

At first we test Algorithm 2. After 52 iterations (336.85 CPU seconds) we obtain the feasible starting point
\[ x^{52} = (0.25, -0.5, 8.791 \cdot 10^{-18}, 2.8225 \cdot 10^{-18}, -0.089828)^T. \]

After 2 more iterations (29.78 CPU seconds) the algorithm terminates with the point
\[ x^* = (0.25, -0.5, 8.791 \cdot 10^{-18}, 2.8225 \cdot 10^{-18}, -0.089828)^T \]
and the objective value \(-2.4812 \cdot 10^{-35}\). At \(x^*\) the norm of the stationarity condition, the change in the value of the objective function, the norm of the change in the iterates are less then \(10^{-16}\), and the approximate feasibility of the indices is 0.
8 Numerical examples

Even though that result seem to be curious, it is not hard to check that $x^*$ is a stationary point of $SIP_{DC1}$. It is the ellipse with center $(0.25, -0.5)$ whose semi axes have the lengths $8.791 \cdot 10^{-18}$ and $2.8225 \cdot 10^{-18}$, so, the area of the ellipse is approximately 0. That stationary point is not a minimum of the area. It is a maximum. However, that result is not very surprising, since the solution concept of our algorithm is that of stationary points, and the feasible starting point, determined in the phase 1 procedure, is $x^*$. There we can also find the reason for that unwelcome result. In the phase 1 of our algorithm we try to find a feasible starting point for the main algorithm. The feasible set of the relaxed problem is a subset of the feasible set of $SIP_{DC1}$. Our starting point $x^0$ is not feasible for the original problem, and, thus, not feasible for the relaxed problem. On the one hand, the error in the feasibility and the error made by relaxing the constraints decreases with the diameter of the generated sequence of boxes and from Lemma 3.3.1 we have that there is some tessellation of $B$, respectively some reduced outer approximation of $Y$ so that the relaxed problem has a feasible point. On the other hand, by reducing the lengths of the semi axes of the ellipse we also have a reduction of the infeasibility. If the error in the feasibility of the generated sequence of points decreases slower with the diameter of boxes than with the lengths of the semi axes, we must expect that result.

An indication that we are in this situation for example $SIP_{DC1}$ is the change in the overall number of boxes generated during the iterations, cf. Figure 8.8, the change in the diameter of boxes and in the maximum separation distance, cf. Figure 8.9, and the reduced outer approximation of the set $Y$, cf. Figure 8.10. It can be seen that the reduced outer approximation of $Y$ is very precise and approximates the whole boundary of the index set well. Moreover, the number of boxes which may contain active indices becomes large during the iterations. But the error, made by the unimodal relaxations, does not decrease fast, even though the maximal diameter of the boxes decreases. From the discussions in Chapter 2 we know that the maximum separation distance of the
Figure 8.9: Results for Algorithm 2 for the problem \( \text{SIP}_{DC1} \). a) The maximum and minimum diameter of the boxes containing active indices. b) The green line is the maximum of the Euclidean norm of the constants \( L, \bar{L} \) on the boxes containing active indices and the black line is the maximal error of the relaxations.

Figure 8.10: Illustration of the reduced outer approximation of \( Y \) for the constraints \( g_1, g_2 \) at \( x^* \) computed by Algorithm 2 for the problem \( \text{SIP}_{DC1} \).

relaxation and the original constraint on a set \( X \times B \) is \( \langle L, b - c \rangle \). As the last term tends at least linearly to zeros with the diameter of the boxes the constants \( L, \bar{L} \) must be still large, cf. Figure 8.9. For our problem we have

\[
\nabla_y g_1 (x, y) = \begin{pmatrix}
2 (x_1 + \cos (x_5) y_1 x_2 - \sin (x_5) y_2 x_4 - \frac{1}{2}) \cos (x_5) x_3 - \sin (x_5) x_3 \\
2 (x_1 + \cos (x_5) y_1 x_2 - \sin (x_5) y_2 x_4 - \frac{1}{2}) \sin (x_5) x_4 - \cos (x_5) x_4
\end{pmatrix},
\]

\[
\nabla_y g_2 (x, y) = \begin{pmatrix}
2 (x_1 + \cos (x_5) y_1 x_2 - \sin (x_5) y_2 x_4 - \frac{1}{2}) \cos (x_5) x_3 + \sin (x_5) x_3 \\
2 (x_1 + \cos (x_5) y_1 x_2 - \sin (x_5) y_2 x_4 - \frac{1}{2}) \sin (x_5) x_4 + \cos (x_5) x_4
\end{pmatrix}.
\]

\( L, \bar{L} \) are computed by evaluating \( \nabla_y g_1 \) and \( \nabla_y g_2 \) on \( X \) and a box \( B \) of the reduced outer approximation of \( Y \). As a result of the structure of \( \nabla_y g_i, i = 1, 2, \) and, due to \( X \) is

105
unchanged during the iterations, one can see that the values of these constants are, in some kind, bounded by the latitude of $X$.

We have to mention that the zigzagging in the separation distance is a result of the relaxation technique. The set of active indices, and, thus, the corresponding boxes in the reduced outer approximation which constitute the error in the relaxation change in each iteration.

After the unfruitful result for Algorithm 2, we test Algorithm 4 on $SIP_{DC1}$. After 5 iterations (30.65 CPU seconds) the feasible point

$$x^5 = (0.20954, -0.47452, 0.18668, 6.263 \cdot 10^{-19}, 0.1038)$$

was identified. After 65 more iterations (195415.36 CPU seconds) the algorithm terminates with the point

$$x^* = (0.25, -0.5, 0.64025, 0.34474, 2.4359)^T$$

and the objective value $-0.22072$. The solution is illustrated in Figure 8.11. At $x^*$ the computed norm of the stationarity condition is less then $10^{-16}$, the approximate feasibility of the indices is 0.03125, the change in the value of the objective function is 0.00041, and the norm of the change in the iterates is 0.00402. The reduced outer approximation of $Y$ in $x^*$ for $g_1$ and $g_2$ is illustrated in Figure 8.12. It can be seen that the approximations of $Y$ are not uniform and only refined locally.

A major drawback of the Algorithm 4 is the high numerical effort, although an acceptable solution was computed. It may be explained by the fact that, as a result of the reformulation, the number of variables in the subproblems increases with the number of boxes arising from splitting $B$ into smaller boxes. The overall number of boxes and the number of boxes which may contain active indices in the reduced outer approximation
generated during the iterations is displayed in Figure 8.13. As both numbers differ only marginally in each iteration we also have a low reduction of the number of variables and constraints in the subproblems.

Each generated box corresponds to an increase in the number of variables by 6. In the last iteration the total amount of boxes is 481. Thus, a subproblem with 2891 variables and 8673 constraints must be solved.

The number of boxes, however, is not the only criterion to benchmark the numerical effort. Another point, intimately related to the size of the generated boxes, is the error made by the convexification procedure. This error depends on the convexification parameter $\alpha$ and the diameter of the box. In the present example, on each box in each
8 Numerical examples

Figure 8.14: Maximum and minimum diameter of boxes containing active indices for \( g_1 \) and \( g_2 \) of the problem \( SIP_{DC1} \) generated by Algorithm 4.

iteration the computed value of \( \alpha \) is 4. To understand this, first notice that

\[
D_y^2 g_i(x, y) = \begin{pmatrix}
2(\cos(x_5)x_3)^2 & -2 \cos(x_5) \sin(x_5)x_3x_4 \\
-2 \cos(x_5) \sin(x_5)x_3x_4 & 2(\sin(x_5)x_4)^2
\end{pmatrix}
\]

holds for \( i = 1, 2 \). Thus, the choice of \( \alpha \) is independent of the size of the boxes in the reduced outer approximation and only depends on the size of \( X \). Since \( X \) does not change, it is not hard to see that the computed value is correct. For this example an additional adaptation of the set \( X \) may have an enormous impact on the performance of the algorithm, since the values for \( \alpha \) and for this reason also the error of the relaxation might be reduced. Since \( \alpha \) does not become small, the diameter of the boxes must become small to decrease the approximation error. In Figure 8.14 the maximum and minimum diameter of boxes in the reduced outer approximation of \( Y \) containing active indices for \( g_1 \) and \( g_2 \) is displayed.

It can be seen that the maximum diameter of boxes containing active indices is not monotonically decreasing. Since the boxes containing active indices change in every iteration we can only ensure that there is a subsequence of boxes whose diameter decreases.

Now we test Algorithms 7 and 10 on \( SIP_{DC1} \) with both adaptation strategies. In the phase 1 procedure we choose \( \delta = 1.1 \), and after the algorithms found a feasible starting point we used \( \delta = \sqrt{5} + \sqrt{\text{EPS}} \) for the \( X \)-adaptations.

With the adaptation strategy (7.1) Algorithm 7 finds the feasible starting point

\[
x^{11} = (0.25, -0.5, 0.29616, 0.29616, 0.20384)^T
\]

after 11 iterations (8.65 CPU seconds). After 24 more iterations (340.12 CPU seconds) the algorithm terminates with the point

\[
x^* = (0.25106, -0.50142, 0.34698, 0.60703, 0.81899)^T
\]
and the objective value $-0.21063$. At $x^*$ the norm of the stationarity condition is less then $10^{-16}$, the change in the value of the objective function is $5.2767 \cdot 10^{-5}$, and the norm of the change in the iterates is $0.002694$. The error in the feasibility of the indices is $0.015625$.

And with the adaptation strategy (7.2) the algorithm finds the feasible starting point

$$x^{16} = (0.25, -0.5, 0.324, 0.324, 0.042395)^T$$

after 16 iterations (15.27 CPU seconds). After 30 more iterations (501.87 CPU seconds) the algorithm terminates with the point

$$x^* = (0.25056, -0.49986, 0.34466, 0.62222, 0.8347)^T$$

and the objective value $-0.21446$. At $x^*$ the norm of the stationarity condition, the change in the value of the objective function, the norm of the change in the iterates, and the error in the feasibility of the indices is less then $10^{-16}$.

As a first result we obtain that Algorithm 7 with both $X$-adaptation strategies performs better than Algorithm 2. It can be seen, cf. Figure 8.15 and 8.16, that the reduced outer approximations of $Y$ in $x^*$ for both adaptation strategies are only good locally. The uncolored boxes are those in the reduced outer approximation which do not contain any active indices. Moreover, with the additional $X$-adaptation we are, in some sense, able to control the result from the phase 1 of the algorithm. We will return to that observation later. For $SIP_{DC}^{1}$ the adaptation strategy (7.1) performs better than (7.2) for Algorithm 7.

For Algorithm 10 with the adaptation strategy (7.1) the feasible starting point

$$x^{11} = (0.25, -0.5, 0.29616, 0.29616, 0.15197)^T$$
is found after 11 iterations (12.92 CPU seconds). After 35 more iterations (385.97 CPU seconds) the algorithm terminates with the point

\[ x^* = (0.24941, -0.50012, 0.33347, 0.62194, 0.87069)^T \]

and the objective value \(-0.2074\). At \(x^*\) the norm of the stationarity condition is \(1.0581 \cdot 10^{-5}\), the change in the value of the objective function is \(1.0066 \cdot 10^{-6}\), and the norm of the change in the iterates is \(0.0027758\). The error in the feasibility of the indices is 0.

With the adaptation strategy (7.2) Algorithm 10 finds the feasible starting point

\[ x^{18} = (0.25, -0.5, 0.33361, 0.302, 0.7242)^T \]

is found after 18 iterations (41.27 CPU seconds). And after 49 more iterations (2221.47 CPU seconds) the algorithm terminates with the point

\[ x^* = (0.25, -0.5, 0.33799, 0.62653, 0.86492)^T \]

and the objective value \(-0.21176\). At \(x^*\) the norm of the stationarity condition is \(6.3206 \cdot 10^{-16}\), the change in the value of the objective function is \(1.193 \cdot 10^{-10}\), the norm of the change in the iterates is \(1.5208 \cdot 10^{-8}\), and the error in the feasibility of the indices is 0.

As for the former algorithm, we obtain that Algorithm 10 with both \(X\)-adaptation strategies performs better than Algorithm 4. It can be seen, cf. Figure 8.17 and 8.18, that the reduced outer approximations of \(Y\) in \(x^*\) for both adaptation strategies are only good locally. Here, again, the uncolored boxes are boxes in the reduced outer approximation which do not contain any active indices. For Algorithm 10 the \(X\)-adaptation strategy given by (7.1) performs also better than (7.2) on example \(SIP_{DC1}\).
The resulting approximately stationary points all algorithms computed differ only slightly, while in contrast to the computational behavior for the problem $SIP_{CA}$, the computational time to compute an approximately stationary point, and the number of iterations of Algorithm 7 is smaller than for Algorithm 10 on the problem $SIP_{DC1}$ and both $X$-adaptation strategies. However, the overall number of generated boxes containing active indices, cf. Figure 8.19, is smaller for Algorithm 10 than for Algorithm 7.

If we look at the error made by the different relaxation strategies, cf. Figure 8.20, then one can see that the error made by the unimodal relaxations is smaller than the error of the concave relaxations. An error of 0 means that the functions are monotonically increasing or decreasing, or respectively are concave. The computational behaviors must be based on the structure of the problem, the properties of the relaxations, and the harder to solve subproblems arising in Algorithm 10.
8 Numerical examples

Figure 8.19: Overall number of boxes and number of boxes which may contain active indices for $SIP_{DC}$. In a) for Algorithm 7 with adaptation strategy given by (7.1). In b) for Algorithm 7 with adaptation strategy given by (7.2). In c) for Algorithm 10 with adaptation strategy given by (7.1). In d) for Algorithm 10 with adaptation strategy given by (7.2).

Figure 8.20: The black lines are the maximal errors of the relaxations in each iteration for the problem $SIP_{DC}$. The red lines are the maximal and the blue lines the minimal diameter of generated boxes which may contain active indices. In a) for Algorithm 7 with adaptation strategy given by (7.1). In b) for Algorithm 7 with adaptation strategy given by (7.2). In c) for Algorithm 10 with adaptation strategy given by (7.1). In d) for Algorithm 10 with adaptation strategy given by (7.2).

Since the hybrid method always chooses that relaxation technique with the smaller
separation distance, one may expect, from the latter results, that Algorithm 14 may perform better than Algorithm 7 and Algorithm 10. Now we investigate the performance of Algorithm 14 on problem \( SIP_{DC1} \).

With the adaptation strategy (7.1) Algorithm 14 identifies the feasible starting point
\[
x^{11} = (0.25, -0.5, 0.29616, 0.29616, 0.20384)^T
\]

after 11 iterations (8.56 CPU seconds). After 34 more iterations (573.18 CPU seconds) the algorithm terminates with the point
\[
x^* = (0.2473, -0.49886, 0.33853, 0.61846, 0.86004)^T
\]

and the objective value \(-0.20937\). At \( x^* \) the norm of the stationarity condition is \( 5.6049 \cdot 10^{-11} \), the change in the value of the objective function is \( 6.8952 \cdot 10^{-11} \), and the norm of the change in the iterates is \( 2.2443 \cdot 10^{-9} \). The error in the feasibility of the indices is 0.

With the adaptation strategy (7.2) Algorithm 10 finds the feasible starting point
\[
x^{16} = (0.25, -0.5, 0.324, 0.324, 0.042395)^T
\]

after 16 iterations (14.77 CPU seconds). And after 32 more iterations (642.75 CPU seconds) the algorithm terminates with the point
\[
x^* = (0.24849, -0.50027, 0.3369, 0.61806, 0.85359)^T
\]

and the objective value \(-0.20822\). At \( x^* \) the norm of the stationarity condition is less than \( 10^{-16} \), the change in the value of the objective function is \( 7.5333 \cdot 10^{-12} \), the norm of the change in the iterates is \( 3.4728 \cdot 10^{-11} \), and the error in the feasibility of the indices is 0.

It can be seen that for both \( X \)-adaptation strategies Algorithm 14 performs worse than Algorithm 7, but better then Algorithm 7 with the \( X \)-adaptation (7.2). A similar setting can be observed for the number of iterations.

The overall number of boxes in the reduced outer approximation which may contain active indices, cf. Figure 8.21, is comparable to that from Algorithm 10. But the made error, cf. Figure 8.22, is slightly worse than the ones from Algorithm 7 and 10. Thus, it is nearly that, this time, we can not explain the poor performance by the computational overhead of the hybrid algorithm.

If we again investigate the maximum separation distance of the relaxation and the original constraint, that is, the error made by relaxing the constraints, cf. Figure 8.22, and the number of boxes where an unimodal and a convex relaxation is used, cf. Figure 8.21, we can observe the following.

Even though we choose that relaxation technique in each iteration so that the maximal error made by relaxing the constraints is the smallest, the reduction of that error after
8 Numerical examples

Figure 8.21: Overall number of boxes and those which may contain active indices generated by Algorithm 14 for $SIP_{DC1}$ which may contain active indices. The blue line is the number of boxes on which a concave relaxation is used. In a) for Algorithm 14 with the $X$-adaptation (7.1). In b) for Algorithm 14 with the $X$-adaptation (7.2).

Figure 8.22: The black lines are the maximal errors of the relaxations in each iteration for the problem $SIP_{DC1}$. The red lines are the maximal and the blue lines the minimal diameter of generated boxes which may contain active indices. In a) for Algorithm 14 with adaptation strategy given by (7.1). In b) for Algorithm 14 with adaptation strategy given by (7.2).

splitting up boxes in the reduced outer approximation might decrease not as fast as one might wish. The reasons for that are not quiet obvious and can be explained by the special error we consider and the refinement step. We give an example to illustrate that.

Example 8.1.2. Let $g(y) = y^3 + y^2$ and $Y = [-2, 1]$. It is not hard to see that we have $\alpha = 8$ and $L = -4$, $U = 14$, $c = \frac{1}{3}$ for $g$ on $Y$. The function $g$ and the relaxations $\hat{g}$, $\tilde{g}$ are illustrated in Figure 8.23. $g$ attains its maximum at $y^* = 1$, while $\hat{g}$ attains its
maximum at $c$ and $\bar{g}$ at $\bar{c} = 1 - \sqrt{\frac{2}{3}}$. Obviously, the maximum separation distance of $\bar{g}$ and $g$ on $Y$ is smaller then the one of $\bar{g}$ and $g$. Thus, our algorithm would choose $\bar{g}$ as a relaxation. But the distance of the points $c$ and $y^*$ is smaller then the distance of $\bar{g}$ and $y^*$. Even though we choose the overestimator with the smallest maximum separation distance we do not have that the distance of the optimal points is also small. Moreover, in the splitting step we would split $Y$ into two boxes through the point where the relaxation attains its maximum. Here we would obtain a better result for the next iteration by splitting $Y$ at $c$, since $g$ is monotone on $[c, 1]$, but neither monotone nor concave on $[\bar{y}, 1]$.

The observation made in the latter example is a general problem in nonlinear and global optimization. Mostly the distance of two function values is bounded by the distance of the corresponding arguments, whereas the converse is not true in general. Thus, we also have that an active index of a relaxed constraint might not be close to an active index of the original constraint, and for this reason it might be a bad splitting point. For more details on error bounds we refer to [38]. For an analysis of splitting procedures in a branch and bound framework we refer to [8] and the references there in.

Now we briefly discuss the choice of the parameter $\delta$ in the $X$-adaptation strategies. In general it is not hard to see that with the additional $X$-adaptation strategy one might reduce the values of the parameters used to relax the problems with the unimodalization or the concavification techniques. The factor of that reduction is not clear, since, it is strongly related to the structure of the function. In the sequel we will point out some other effects which are a result of the $X$-adaptation. We start with the choice of $\delta$ in the phase 1 algorithm.

From the results for the problem $SIPC_A$ one can see that with the $X$-adaptation the algorithms need more iterations to compute a feasible starting point. The algorithms without the additional adaptation, Algorithm 2 and 4, found a feasible point after 1 iteration, while Algorithm 7 and 10 needed 13 iterations with the adaptation strategy (7.1), and 14 iterations with the adaptation strategy (7.2). A reason for that outcome
is, that the step size is constricted by the $X$-adaptation. Especially from Figure 8.4 we see that none of the generated boxes do contain an active index, thus no box is splitted.

In contrast to the results for the problem $SIP_{CA}$, we have another effect for the problem $SIP_{DC1}$. The results for the algorithms without the $X$-adaptation were poor. Actually, in the phase 1 procedure Algorithm 2 computed a maximum of the problem. Indeed, Algorithm 10 with both adaptation strategies needed more iterations to compute a feasible point than Algorithm 4, but the CPU time was less. Moreover, with Algorithm 7 and the $X$-adaptation strategies we could avoid that the algorithm computes a maximum in the phase 1 procedure by restricting the feasible set of the subproblems, additionally, to subsets of $X$.

We have to mention that it is possible to choose $\delta = 0$ in the phase 1 algorithm, if the starting point is feasible for the original problem. If one chooses $\delta = 0$ for a starting point, feasible for the original problem, the algorithms will compute a reduced outer approximation of the corresponding index set so that the point is also feasible for the relaxed problem. That such a reduced outer approximation exists is a consequence from Lemma 3.3.1 and 4.3.1.

After the algorithms computed a stationary point the role of $\delta$ changes. In contrast to the phase 1 procedure, the choice of $\delta$ is limited for Algorithm 7 and 10. From Section 7.4 we know that the relation $\delta > \sqrt{n}$ must hold to ensure Condition 5.1.7, (ii). As a result from both examples, $SIP_{CA}$ and $SIP_{DC1}$, we have that the $X$-adaptation may increase the performance of the algorithms, since with the $X$-adaptation we can reduce the feasible set of a subproblem. Moreover, as also observed in the phase 1 of the algorithms on the previous examples, with that additional restriction we can identify more boxes in the reduced outer approximation of the index set which do not contain active indices for a subproblem, cf. Figures 8.1, 8.3 and 8.19. One reason for that behavior is that more restrictions may become redundant by the additional restriction to a subset of $X$. To reduce the number of boxes in the subproblems, which lead to a reduction in the number of the constraints and variables, one might choose $\delta$ as small as possible. But, as recalled earlier, $\delta$ is bounded below by $\sqrt{n}$, and choosing smaller subsets of $X$ may lead to an inefficient shortening of the step size, and, thus, the algorithms need more iterations, which may again lead to a larger amount of generated boxes. An optimal choice for $\delta$ can not be given in general, since, it differs with the structure of the problem.

On the next examples we illustrate the latter remarks on the choice of $\delta$, and we investigate the reduced outer approximation of an index set $Y$. It is a modification of a design-centering problem taken from [16]. For the example we consider the container

$$ C = \{ x \in \mathbb{R}^2 | c_1(x) = \frac{3}{10} \sin(\pi x_1) - x_2 \leq 0, c_2(x) = x_1^2 + \frac{3}{10} x_2^2 - 1 \leq 0 \}, $$

cf. Figure 8.24, and the design

$$ D(x) = \{ z \in \mathbb{R}^2 | \left( \frac{z_1 - x_1}{x_3} \right)^2 + \left( \frac{z_2 - x_2}{x_4} \right)^2 \leq 1 \}. $$
that is, an ellipse with free center \((x_1, x_2)^T\) and axes \(x_3, x_4\) parallel to the coordinate axes.

To maximize the area of \(D(x)\) we put \(f(x) = x_3 x_4\). As a parametrization for \(D(x)\) we use \(D(x) = \{z(x,y) | y_1^2 + y_2^2 \leq 1\}\) with

\[
z(x, y) = \begin{pmatrix} x_1 \\ x_2 \\ y_1 x_3 \\ y_2 x_4 \end{pmatrix}.
\]

As in the latter example, we obtain

\[
g_i(x,y) = c_i (z(x,y)) \leq 0 \text{ for all } y \in Y = \{y \in \mathbb{R}^2 | y_1^2 + y_2^2 \leq 1\}, i \in \{1,2\}
\]

and we arrive at the semi-infinite problem

\[
SIP_{DC2} : \min_{x \in \mathbb{R}^5} -x_3 x_4 \quad \text{s.t.} \quad g_i(x,y) \leq 0 \text{ for all } y \in Y, i \in \{1,2\}.
\]

By switching to polar coordinates on the set \(Y\), the problem \(SIP_{DC2}\) is equivalent to

\[
SIP_{DC3} : \min_{x \in \mathbb{R}^5} -x_3 x_4 \quad \text{s.t.} \quad g_i(x,y) \leq 0 \text{ for all } y \in Y, i \in \{1,2\},
\]

with \(Y = [0, 1] \times [0, 2\pi]\), \(g_i(x,y) = c_i (\tilde{z}(x,y))\), \(i \in \{1,2\}\), and

\[
\tilde{z}(x,y) = \begin{pmatrix} x_1 \\ x_2 \\ y_1 \sin (y_2) x_3 \\ y_1 \cos (y_2) x_4 \end{pmatrix}.
\]

For the problems \(SIP_{DC2}\) and \(SIP_{DC3}\) we use the same termination tolerances as in the latter example. It is not hard to see that the feasible sets of both problems is contained in the set \(X = [-\frac{11}{10}, \frac{11}{10}] \times [-\frac{1}{2}, 2] \times [0, 1] \times [0, 1]\). For \(SIP_{DC2}\) we choose \(B = [-1, 1] \times [-1, 1]\) as an initial approximation of \(Y\). For \(SIP_{DC3}\) we choose \(B = Y\). As a starting point we
choose the infeasible point $x^0 = (0, \frac{3}{4}, \frac{1}{2})$ which is the barycenter of $X$, and for the first results we choose $\delta = 1.1$ in the phase 1 algorithm and $\delta = \sqrt{5} + \sqrt{\text{EPS}}$ after a feasible point was found. We have to notice that the stationary points of the problems $SIP_{DC2}$ and $SIP_{DC3}$ are the same. But one may expect that the algorithms perform better for the problem $SIP_{DC3}$ than for $SIP_{DC2}$, since the index set $Y$ can be handled exactly.

For Algorithm 7 on the problem $SIP_{DC2}$ we obtain the following results. For the $X$-adaptation strategy (7.1) the algorithm found the feasible starting point

$$x^3 = (-0.03641, 0.78641, 0.46359, 0.46359)^T$$

after 3 iterations (2.62 CPU seconds). After 54 more iterations (72.16 CPU seconds) the algorithm terminates with the point

$$x^* = (-0.0026466, 0.96028, 0.74917, 0.75139)^T$$

and the objective value $-0.56292$. The solution is illustrated in Figure 8.25. At $x^*$ the norm of the stationarity condition is less than $10^{-16}$, and the change in the value of the objective function, the norm of the change in the iterates, as well as the error in the feasibility of the indices is 0.

With the adaptation strategy (7.2) Algorithm 7 finds the feasible starting point

$$x^3 = (-0.03641, 0.78641, 0.467, 0.467)^T$$

after 3 iterations (1.39 CPU seconds). And after 39 more iterations (57.56 CPU seconds) the algorithm terminates with the point

$$x^* = (3.1094 \cdot 10^{-17}, 0.96622, 0.7379, 0.76083)^T$$

and the objective value $-0.56141$. At $x^*$ the norm of the stationarity condition is $9.4919 \cdot 10^{-8}$, and the change in the value of the objective function, the norm of the
change in the iterates, as well as the error in the feasibility of the indices is 0. The reduced outer approximation of $Y$ at $x^*$ for both strategies are illustrated in Figure 8.26 and Figure 8.27.

For the problem $SIP_{DC3}$ we obtain the following results with Algorithms 7. For the $X$-adaptation strategy (7.1) the algorithm found the feasible starting point

$$x^3 = (-0.0231, 0.7731, 0.4819, 0.4819)^T$$

after 3 iterations (2.24 CPU seconds). After 21 more iterations (42751, 69 CPU seconds) the algorithm was terminated by Matlab at the point

$$x^* = (-0.0256, 0.9333, 0.7566, 0.7643)^T$$

with the message out of memory. And indeed, for that problem the capacity of 4GB RAM and the 2GB swap memory was used. When the algorithm stopped at $x^*$ the
norm of the stationarity condition was 0.9048, the change in the value of the objective function was 0.0179, and the norm of the change in the iterates was 0.0275.

In Figure 8.28 the overall number of boxes and those that may contain active indices is displayed. It can be seen that in the last step the amount of generated boxes is very large. The cause for that can be found in the X-adaptation. As the next to last iterate is not feasible for the new generated subset of X, the algorithm splits boxes in the reduced outer approximation of Y, which explains the huge number of boxes and requirement of memory. With the adaptation strategy (7.2) and Algorithm 7 we obtain similar results as for the strategy (7.1). Later we will survey that example again, and we will trace back that result on the choice of δ.

Now we give the results for SIP_{DC2} obtained with Algorithms 10. For the X-adaptation strategy (7.1) the algorithm found the feasible starting point

\[x^3 = (-0.03641, 0.78641, 0.46359, 0.46359)^T\]

after 3 iterations (1.58 CPU seconds). After 44 more iterations (66.29 CPU seconds) the algorithm terminates with the point

\[x^* = (-4.7995 \cdot 10^{-17}, 0.97076, 0.73416, 0.76169)^T\]

and the objective value −0.5592. At \(x^*\) the norm of the stationarity condition is 3.0868 \cdot 10^{-6}, the change in the value of the objective function is 8.5587 \cdot 10^{-6}, and the norm of the change in the iterates is 0.0036774. The error in the feasibility of the indices is 0.

With the adaptation strategy (7.2) Algorithm 10 finds the feasible starting point

\[x^3 = (-0.03641, 0.78641, 0.467, 0.467)^T\]

after 3 iterations (1.68 CPU seconds). And after 44 more iterations (78.08 CPU seconds) the algorithm terminates with the point

\[x^* = (2.6017 \cdot 10^{-17}, 0.97076, 0.73416, 0.76169)^T\]
and the objective value $-0.5592$. At $x^*$ the norm of the stationarity condition is $4.4119 \cdot 10^{-10}$, the change in the value of the objective function is $8.0228 \cdot 10^{-8}$, the norm of the change in the iterates is $3.561 \cdot 10^{-4}$, and the error in the feasibility of the indices is 0. The reduced outer approximation of $Y$ at $x^*$ for both strategies are displayed in Figure 8.29 and Figure 8.30.

At last we give the results for $SIP_{DC3}$ obtained with Algorithms 10. For the $X$-adaptation strategy (7.1) the algorithm found the feasible starting point

$$x^3 = (-0.0231, 0.7731, 0.4819, 0.4819)^T$$

after 3 iterations (3.78 CPU seconds). After 24 more iterations (913.72 CPU seconds) the algorithm terminates with the point

$$x^* = (-1.1335 \cdot 10^{-07}, 0.96612, 0.77095, 0.78329)^T$$
8 Numerical examples

Figure 8.31: Illustration of the tessellation of $Y$ for the constraints $g_1$, $g_2$ of the problem $\text{SIP}_{DC3}$ at $x^*$ computed by Algorithm 10 with the $X$-adaptation strategy given by (7.1).

and the objective value $-0.60387$. At $x^*$ the norm of the stationarity condition is $5.7073 \cdot 10^{-7}$, the change in the value of the objective function is $1.1535 \cdot 10^{-7}$, and the norm of the change in the iterates is $4.5935 \cdot 10^{-5}$. The error in the feasibility of the indices is 0.

With the adaptation strategy (7.2) Algorithm 10 finds the feasible starting point

$$x^3 = (-0.03641, 0.78641, 0.472, 0.472, 0)^T$$

after 3 iterations (3.21 CPU seconds). And after 9 more iterations (442.86 CPU seconds) the algorithm terminates with the point

$$x^* = (6.8888 \cdot 10^{-7}, 0.96627, 0.77074, 0.7835)^T$$

and the objective value $-0.60387$. At $x^*$ the norm of the stationarity condition is $1.4418 \cdot 10^{-7}$, the change in the value of the objective function is $1.153 \cdot 10^{-4}$, the norm of the change in the iterates is $0.0068836$, and the error in the feasibility of the indices is 0. The tessellation of $Y$ at $x^*$ for both strategies are displayed in Figure 8.31 and Figure 8.32.

Before we investigate the reduced outer approximation, we reconsider Algorithm 7 on the problems $\text{SIP}_{DC3}$ and $\text{SIP}_{DC2}$. The bad results we obtain for Algorithm 7 with both adaptation strategies may be a consequence of an inappropriate choice of $\delta$. That is, $\delta$ was chosen to small, and, thus, the step size in each iteration was restricted too keen. Motivated by that consideration we test again the algorithms on the problems $\text{SIP}_{DC2}$ and $\text{SIP}_{DC3}$, but, with the parameters $\delta = 1.1$ in the phase 1 and $\delta = \sqrt{5} + 1$ after a feasible starting point was found.

For Algorithms 7 on the problem $\text{SIP}_{DC2}$ we obtain the following results. For the $X$-adaptation strategy (7.1) the algorithm found the feasible starting point

$$x^3 = (-0.03641, 0.78641, 0.46359, 0.46359)^T$$
Figure 8.32: Illustration of the tessellation of $Y$ for the constraints $g_1, g_2$ of the problem $SIP_{DC3}$ at $x^*$ computed by Algorithm 10 with the X-adaptation strategy given by (7.2).

after 3 iterations (1.86 CPU seconds). After 37 more iterations (56.5 CPU seconds) the algorithm terminates with the point

$$x^* = (-7.2195 \cdot 10^{-17}, 0.96957, 0.73435, 0.76334)^T$$

and the objective value $-0.56056$. At $x^*$ the norm of the stationarity condition is $3.0842 \cdot 10^{-9}$, and the change in the value of the objective function, the norm of the change in the iterates and the error in the feasibility of the indices is 0.

With the adaptation strategy (7.2) Algorithm 7 finds the feasible starting point

$$x^3 = (-0.03641, 0.78641, 0.467, 0.467, 0)^T$$

after 3 iterations (1.37 CPU seconds). And after 34 more iterations (61.72 CPU seconds) the algorithm terminates with the point

$$x^* = (-9.7662 \cdot 10^{-17}, 0.96622, 0.7379, 0.76083)^T$$

and the objective value $-0.56141$. At $x^*$ the norm of the stationarity condition is $9.4919 \cdot 10^{-9}$, and the change in the value of the objective function, the norm of the change in the iterates and the error in the feasibility of the indices is 0. The reduced outer approximation of $Y$ at $x^*$ for both strategies are illustrated in Figure 8.33 and Figure 8.34.

For the problem $SIP_{DC3}$ we obtain the following results with Algorithms 7. For the X-adaptation strategy (7.1) the algorithm found the feasible starting point

$$x^3 = (-0.0231, 0.7731, 0.4819, 0.4819)^T$$

after 3 iterations (1.78 CPU seconds). After 16 more iterations (247.85 CPU seconds) the algorithm terminates with the point

$$x^* = (-0.00013083, 0.96506, 0.77213, 0.7819)^T$$
and the objective value $-0.60373$. At $x^*$ the norm of the stationarity condition is $9.5124 \cdot 10^{-8}$, the change in the value of the objective function is $2.1454 \cdot 10^{-4}$, and the norm of the change in the iterates is $4.0336 \cdot 10^{-4}$. The error in the feasibility of the indices is 0.

With the adaptation strategy (7.2) Algorithm 7 finds the feasible starting point

$$x^3 = (-0.0231, 0.7731, 0.483, 0.483)^T$$

is found after 3 iterations (1.6 CPU seconds). And after 10 more iterations (90.51 CPU seconds) the algorithm terminates with the point

$$x^* = (-1.9341 \cdot 10^{-6}, 0.96631, 0.77064, 0.78291)^T$$
Figure 8.35: Illustration of the tessellation of $Y$ for the constraints $g_1$, $g_2$ of the problem $SIP_{DC3}$ at $x^*$ computed by Algorithm 7 with the $X$-adaptation strategy given by (7.1) and $\delta = \sqrt{5} + 1$.

Figure 8.36: Illustration of the tessellation of $Y$ for the constraints $g_1$, $g_2$ of the problem $SIP_{DC3}$ at $x^*$ computed by Algorithm 7 with the $X$-adaptation strategy given by (7.2) and $\delta = \sqrt{5} + 1$.

and the objective value $-0.60334$. At $x^*$ the norm of the stationarity condition is $7.6658 \cdot 10^{-8}$, the change in the value of the objective function is $6.0282 \cdot 10^{-4}$, the norm of the change in the iterates is $0.0010633$, and the error in the feasibility of the indices is $0$. The tessellation of $Y$ at $x^*$ for both strategies are illustrated in Figure 8.35 and Figure 8.36.

Next we also give the results for Algorithm 10 on the problems $SIP_{DC2}$ and $SIP_{DC3}$ with the changed values for $\delta$. For the $X$-adaptation strategy (7.1) the algorithm found the feasible starting point

$$x^3 = \begin{pmatrix} -0.03641, 0.78641, 0.46359, 0.46359 \end{pmatrix}^T$$

after 3 iterations (1.69 CPU seconds). After 51 more iterations (110.25 CPU seconds) the algorithm terminates with the point

$$x^* = \begin{pmatrix} 3.7098 \cdot 10^{-18}, 0.96981, 0.7411, 0.75908 \end{pmatrix}^T$$
8 Numerical examples

![Figure 8.37: Illustration of the reduced outer approximation of Y for the constraints $g_1, g_2$ of the problem SIPDC2 at $x^*$ computed by Algorithm 10 with the X-adaptation strategy given by (7.1) and $\delta = \sqrt{5} + 1$.]

and the objective value $-0.56255$. At $x^*$ the norm of the stationarity condition is $3.7341 \cdot 10^{-16}$, the change in the value of the objective function is $1.7158 \cdot 10^{-9}$, and the norm of the change in the iterates is $2.4269 \cdot 10^{-9}$. The error in the feasibility of the indices is 0.

With the adaptation strategy (7.2) Algorithm 10 finds the feasible starting point

$$x^3 = (-0.03641, 0.78641, 0.467, 0.467)^T$$

after 3 iterations (1.53 CPU seconds). And after 46 more iterations (96.59 CPU seconds) the algorithm terminates with the point

$$x^* = (-4.3448e - 20, 0.97076, 0.73416, 0.76169)^T$$

and the objective value $-0.5592$. At $x^*$ the norm of the stationarity condition is $1.2745 \cdot 10^{-8}$, the change in the value of the objective function is $8.5579 \cdot 10^{-6}$, the norm of the change in the iterates is 0.0036795, and the error in the feasibility of the indices is 0. The reduced outer approximation of $Y$ at $x^*$ for both strategies are illustrated in Figure 8.37 and Figure 8.38.

For the problem $SIP_{DC3}$ we obtain the following results with Algorithms 10. For the X-adaptation strategy (7.1) the algorithm found the feasible starting point

$$x^3 = (-0.0231, 0.7731, 0.4819, 0.4819)^T$$

after 3 iterations (2.24 CPU seconds). After 19 more iterations (391.23 CPU seconds) the algorithm terminates with the point

$$x^* = (2.7974 \cdot 10^{-5}, 0.96558, 0.77167, 0.78251)^T$$
and the objective value $-0.60384$. At $x^*$ the norm of the stationarity condition is $3.2655 \cdot 10^{-10}$, the change in the value of the objective function is $2.9851 \cdot 10^{-4}$, and the norm of the change in the iterates is $0.0047031$. The error in the feasibility of the indices is $0$.

With the adaptation strategy (7.2) Algorithm 10 finds the feasible starting point

$$x^3 = (-0.03641, 0.78641, 0.472, 0.472)^T$$

is found after 3 iterations (2.21 CPU seconds). And after 26 more iterations (1696.54 CPU seconds) the algorithm terminates with the point

$$x^* = (-1.2537 \cdot 10^{-7}, 0.96613, 0.77095, 0.78329)^T$$

and the objective value $-0.60388$. At $x^*$ the norm of the stationarity condition is $1.289 \cdot 10^{-8}$, the change in the value of the objective function is $7.9448 \cdot 10^{-7}$, the norm of the change in the iterates is $2.27 \cdot 10^{-5}$, and the error in the feasibility of the indices is $0$. The tessellation of $Y$ at $x^*$ for both strategies are illustrated in Figure 8.39 and Figure 8.40.

These results for the algorithms with the changed value for $\delta$ support the former assertion, that is, a too small value for $\delta$ has a bad effect on the performance. The best choice for these parameters in the algorithms can not be answered in general, as it depends on the structure of the considered optimization problem. However, it can be seen that the reduced outer approximation and the tessellation of $Y$ in each algorithm is only refined and good locally whatever the choice of $\delta$ is. In this example the adaptation strategy for the set $X$ given by (7.1) performs mostly better than the one given by (7.2).

Now we focus on the effect of the reduced outer approximation. As the problems $SIP_{DC2}$ and $SIP_{DC3}$ are equivalent, one might expect that the algorithms have a better perfor-
Figure 8.39: Illustration of the tessellation of $Y$ for the constraints $g_1, g_2$ of the problem $SIP_{DC3}$ at $x^*$ computed by Algorithm 10 with the $X$-adaptation strategy given by (7.1) and $\delta = \sqrt{5} + 1$.

Figure 8.40: Illustration of the tessellation of $Y$ for the constraints $g_1, g_2$ of the problem $SIP_{DC3}$ at $x^*$ computed by Algorithm 10 with the $X$-adaptation strategy given by (7.2) and $\delta = \sqrt{5} + 1$.

Performance for the latter problem, as the index set $Y$ can be handled exactly and the algorithms do not need a reduced outer approximation. Though the results differ from that expectation. All algorithms perform better on the problem $SIP_{DC2}$ than on $SIP_{DC3}$. The reason for that outcome can be found in the dependency effect of interval arithmetic. The dependency effect denotes a simple, but major problem in interval arithmetic, that is, dependencies between different and even the same variables can not be handled. Before we discuss the impact of this effect on the algorithms, we illustrate the effect in an example.

**Example 8.1.3.** Let $g(x) = \sin (\pi x) x$. On $X = [0, 1]$ we have that the minimum of $g$ is 0 and the maximum of $g$ is around 0.5792. If we evaluate $g$ on $X$ with the techniques
From interval arithmetic we obtain the following result.

\[ g(X) = \sin(\pi X) X \]
\[ = [0,1] \cdot [0,1] \]
\[ = [0,1]. \]

Thus, we have 0 as lower bound and 1 as an upper bound for \( g \) on \( X \). The upper bound is larger than the original upper bound, since with the techniques from interval arithmetic one evaluates \( \sin \) on \( X \), and, after that, the multiplication of the resulting interval with \( X \) is executed without regarding the dependency.

Here we only discuss the impact of the dependency effect on Algorithm 10 with the \( X \)-adaptation strategy (7.1), since for the other strategy and the other algorithm the impact is similar. Figure 8.41 displays the overall number of boxes, and those that contain active indices, generated by Algorithm 10 for the problems \( SIP_{DC^2} \) and \( SIP_{DC^3} \). It can be seen that the total amount of generated boxes is larger for the problem \( SIP_{DC^3} \) than for \( SIP_{DC^2} \), while the number of those boxes which may contain active indices is comparable. Figure 8.42 illustrates the maximal values for \( \alpha \) and the error of the relaxations on boxes which may contain active indices for \( SIP_{DC^2} \) and \( SIP_{DC^3} \). We have that in most of the iterations the values for the constants \( \alpha \) and the error of the unimodal relaxation for \( SIP_{DC^3} \) are larger than for \( SIP_{DC^2} \), while in the last iterations both get stuck for the problem \( SIP_{DC^2} \). As for both problems the error tends quadratically to zero with the diameter of the boxes, and as the diameters show a similar behavior like the error in the relaxations for both problems, the difference in the generated number of boxes can only be a result of the parameters \( \alpha \).

To compute the parameters \( \alpha \) we evaluate the Hessian of the constraints with respect to \( y \) on the generated subsets of \( X \) and the tessellation, respectively the reduced outer approximation of \( Y \). After that we apply the Theorem of Gerschgorin, cf. Theorem 7.1.1,
8 Numerical examples

Figure 8.42: a) The maximum and minimum diameter of the boxes containing active indices with for the problem $SIP_{DC^2}$ generated by Algorithm 10 and $X$-adaptation strategy (7.1). b) The green line is the maximum of the constants $\alpha$ on the boxes containing active indices and the black line is the error of the relaxation for the problem $SIP_{DC^2}$ computed with Algorithm 10 and $X$-adaptation strategy (7.1). c) The maximum and minimum diameter of the boxes containing active indices with for the problem $SIP_{DC^3}$ generated by Algorithm 10 and $X$-adaptation strategy (7.1). d) The green line is the maximum of the constants $\alpha$ on the boxes containing active indices and the black line is the error of the relaxation for the problem $SIP_{DC^2}$ computed with Algorithm 10 and $X$-adaptation strategy (7.1).
to these matrices to obtain some bound for the eigenvalues. For the constraint $g_2$ in $SIP_{DC^2}$ we obtain

$$D^2_y g_2(x, y) = \begin{pmatrix} 2x_3^2 & 0 \\ 0 & \frac{6}{10} x_4^2 \end{pmatrix}.$$ 

And for the constraint $g_2$ in $SIP_{DC^2}$ we obtain

$$D^2_y g_2(x, y) = \begin{pmatrix} \left(D^2_y g_2(x, y)\right)_{1,1} & \left(D^2_y g_2(x, y)\right)_{1,2} \\ \left(D^2_y g_2(x, y)\right)_{2,1} & \left(D^2_y g_2(x, y)\right)_{2,2} \end{pmatrix},$$

with

$$\left(D^2_y g_2(x, y)\right)_{1,1} = 2 \sin^2(y_2) x_3^2 + \frac{6}{10} \cos^2(y_2) x_4^2$$

$$\left(D^2_y g_2(x, y)\right)_{1,2} = 2 \cos(y_2) x_3 (x_1 + y_1 \sin(y_2) x_3) + 2 y_1 \sin(y_2) \cos(y_2) x_3^2$$

$$- \frac{6}{10} \sin(y_2) x_4 (x_2 + y_1 \cos(y_2) x_4) - \frac{6}{10} y_1 \cos(y_2) \sin(y_2) x_4^2$$

$$\left(D^2_y g_2(x, y)\right)_{2,2} = -2 y_1 \sin(y_2) x_3 (x_1 + y_1 \sin(y_2) x_3) + 2 y_1^2 \cos^2(y_2) x_3^2$$

$$- \frac{6}{10} y_1 \cos(y_2) x_4 (x_2 + y_1 \cos(y_2) x_4) + \frac{6}{10} y_1^2 \sin^2(y_2) x_4^2.$$ 

Thus, it is not hard to check that the bounds $\alpha$ for the eigenvalues should be worse for the problem $SIP_{DC^2}$ than for $SIP_{DC}$, especially if one keeps the dependency effect in mind. In principle the derivatives of the constraints become in some kind easier for $SIP_{DC^2}$, as a part of the nonlinearity of the constraints has moved to the index set $Y$. And, in the latter example, the impact of the error in the feasibility of the indices can be handled better by the algorithms than the error in the constants which are used to relax the problem.

In the latter example we chose an ellipse with axes parallel to the coordinate axes as design. This choice was only made for practical reasons as we will briefly show in the next example.

For the example we consider the container from the latter example, that is,

$$C = \{ x \in \mathbb{R}^2 | c_1(x) = \frac{3}{10} \sin(\pi x_1) - x_2 \leq 0, c_2(x) = x_1^2 + \frac{3}{10} x_2^2 - 1 \leq 0 \},$$

and the design

$$D(x) = \{ z \in \mathbb{R}^2 | (\cos(x_5)\frac{z_1 - x_1}{x_3} - \sin(x_5)\frac{z_2 - x_2}{x_3})^2$$

$$+ (\sin(x_5)\frac{z_1 - x_1}{x_4} + \cos(x_5)\frac{z_2 - x_2}{x_4})^2 \leq 1 \}.$$ 

that is, an ellipse with free center $(x_1, x_2)^T$ and axes $x_3, x_4$ in arbitrary position.
To maximize the area of $D(x)$ we again put $f(x) = x_3 x_4$. As a parametrization for $D(x)$ we use $D(x) = \{ z(x,y) | y_1^2 + y_2^2 \leq 1 \}$ with

$$z(x,y) = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} \cos(x_5) \\ \sin(x_5) \end{pmatrix} \begin{pmatrix} y_1 x_3 \\ y_2 x_4 \end{pmatrix}.$$

As in the latter example, we obtain

$$g_i(x,y) = c_i(z(x,y)) \leq 0 \text{ for all } y \in Y = \{ y \in \mathbb{R}^2 | y_1^2 + y_2^2 \leq 1 \}, i \in \{1,2\}$$

and we arrive at the semi-infinite problem

$$SIP_{DC4} : \min_{x \in \mathbb{R}^5} -x_3 x_4 \quad s.t. \quad g_i(x,y) \leq 0 \quad \text{for all } y \in Y, i \in \{1,2\}.$$

With the adaptation strategy (7.1) Algorithm 7 finds the feasible starting point

$$x^{10} = (-0.0056486, 0.84321, 0.63059, 0.57469, 0.1431)^T$$

after 10 iterations (10.07 CPU seconds). And after 23 more iterations (204.76 CPU seconds) the algorithm terminates with the point

$$x^* = (-0.00096438, 0.94506, 0.76448, 0.74082, 0.14138)^T$$

and the objective value $-0.56634$. The solution is illustrated in Figure 8.43 At $x^*$ the norm of the stationarity condition is $2.238 \cdot 10^{-16}$, and the change in the value of the objective function, the norm of the change in the iterates, as well as the error in the feasibility of the indices is 0.

Algorithm 10 with the adaptation strategy (7.1) identifies the feasible starting point

$$x^9 = (-0.030499, 0.87226, 0.60063, 0.60063, 0.13567)^T$$
after 9 iterations (11 CPU seconds). And after 54 more iterations (1308.63 CPU seconds) the algorithm terminates with the point

\[ x^* = (-0.19841, 0.78378, 1.0688, 0.62099, 1.0819)^T \]

and the objective value \(-0.6637\). The solution is illustrated in Figure 8.44. At \( x^* \) the norm of the stationarity condition is \( 5.7651 \times 10^{-8} \), the change in the value of the objective function is \( 1.4688 \times 10^{-12} \), the norm of the change in the iterates is \( 3.5723 \times 10^{-7} \), and the error in the feasibility of the indices is 0.03125.

In contrast to the former examples the results from the algorithms for \( SIP_{DC^4} \) are not comparable. That is not an error in the algorithms, it is a result of our solution concept. The algorithms try to compute a stationary point of a semi-infinite problem and not a global solution. Thus, if more than one stationary point exists we can not expect to find the same point with the different algorithms from an arbitrary starting point.

The last example we investigate is also a design centering problem, but with an index set \( Y \subset \mathbb{R}^3 \). We consider the container

\[
C = \{ x \in \mathbb{R}^2 | c_1(x) = -x_1 \leq 0, c_2(x) = -x_2 \leq 0, c_3(x) = -x_3 \leq 0, \\
c_4(x) = x_1 + x_2 + x_3 - 1 \leq 0 \},
\]

cf. Figure 8.45, and as design \( D(x) \) an ellipsoid with free center \( (x_1, x_2, x_3)^T \) and axes \( x_4, x_5, x_6 \) in arbitrary position.

To maximize the volume of \( D(x) \) we put \( f(x) = x_4x_5x_6 \). As a parametrization for \( D(x) \) we use \( D(x) = \{ z(x, y) | y_1^2 + y_2^2 + y_3^2 \leq 1 \} \) with

\[
z(x, y) = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + R(x_7, x_8, x_9) \begin{pmatrix} y_1x_4 \\ y_2x_5 \\ y_3x_6 \end{pmatrix},
\]
and a rotation matrix

\[
R(x_7, x_8, x_9) = \begin{pmatrix}
\cos(x_7) & -\sin(x_7) & 0 \\
\sin(x_7) & \cos(x_7) & 0 \\
0 & 0 & 1
\end{pmatrix} \cdot \begin{pmatrix}
\cos(x_8) & 0 & -\sin(x_8) \\
0 & 1 & 0 \\
\sin(x_8) & 0 & \cos(x_8)
\end{pmatrix}
\]

As in the latter example, we obtain

\[
g_i(x, y) = c_i(z(x, y)) \leq 0 \text{ for all } y \in Y, i \in \{1, 2, 3, 4\},
\]

with \(Y = \{y \in \mathbb{R}^2 | y_1^2 + y_2^2 + y_3^2 \leq 1\}\). Finally we arrive at the semi-infinite problem

\[
SIP_{DC5} : \min_{x \in \mathbb{R}^5} -x_4 x_5 x_6 \quad \text{s.t.} \quad g_i(x, y) \leq 0 \text{ for all } y \in Y, i \in \{1, 2, 3, 4\}.
\]

Here we only give the results for Algorithm 7 and 10 with the X-adaptation strategy (7.1). For \(SIP_{DC5}\) we use the same termination tolerances for the algorithms as in the latter design centering problems. It is not hard to see that the feasible set of this problem is contained in the set \(X = [-\frac{1}{10^{11}} \times [-\frac{1}{10^{11}} \times [-\frac{1}{10^{11}} \times [0, \frac{11}{10}] \times [0, \frac{11}{10}] \times [-\pi, \pi] \times [-\pi, \pi] \times [-\pi, \pi].\) We choose \(B = [-1, 1] \times [-1, 1] \times [-1, 1] \times [0, 0] \times [0, 0]\) as an initial approximation of \(Y\) for each constraint. As a starting point we choose the infeasible point \(x_0 = (\frac{1}{2}, \frac{1}{2}, \frac{11}{22}, \frac{11}{22}, 0, 0, 0)\) which is the barycenter of \(X\), and we choose \(\delta = 0.5\) in the phase 1 algorithm and \(\delta = 4\) after a feasible point was found.

For Algorithm 10 with the X-adaptation strategy (7.1) a feasible starting point

\[
x_{69} = (0.21026, 0.20698, 0.20696, 0.2083, 0.20502, 0.205, 6.2161 \cdot 10^{-17}, 1.6447 \cdot 10^{-12}, -1.6445 \cdot 10^{-12})^T
\]
was identified after 69 iterations (2918.41 CPU seconds). After 4 more iterations that took around 6 days we aborted the optimization process at the point

\[ x^{73} = (0.20779, 0.20779, 0.20779, 0.20779, 0.20779, 0, 0, 0)^T. \]

and the objective value \(-0.00897\). At \( x^{73} \) the norm of the stationarity condition is \(2.2933 \cdot 10^{-6}\), the change in the value of the objective function is \(1.1609 \cdot 10^{-14}\), and the norm of the change in the iterates is \(1.6762 \cdot 10^{-12}\). And the error in the feasibility of the indices is 0.125. Thus, the only reason the algorithm did not terminate was the feasibility of the active indices in the reduced outer approximation of the index set.

For Algorithm 7 with the X-adaptation strategy (7.1) a feasible starting point

\[ x^{69} = (0.20911, 0.20901, 0.21117, 0.205, 0.205, 0.2072, \]
\[ 7.7993 \cdot 10^{-17}, 9.8709 \cdot 10^{-17}, 3.0343 \cdot 10^{-17})^T \]

was identified after 69 iterations (2695.96 CPU seconds). After 2 more iterations (17642.05 CPU seconds) the algorithm terminates with the point

\[ x^* = (0.20991, 0.20985, 0.20983, 0.20979, 0.20979, 0.20979, \]
\[ 8.0081 \cdot 10^{-17}, 1.0135 \cdot 10^{-16}, 2.72 \cdot 10^{-17})^T \]

and the objective value \(-0.00923\). The solution is illustrated in Figure 8.46, and the reduced outer approximation of \( Y \) for each constraint in \( x^* \) is illustrated in Figure 8.47. At \( x^* \) the norm of the stationarity condition is less than \(10^{-16}\), the change in the value of the objective function is \(9.1314 \cdot 10^{-5}\), and the norm of the change in the iterates is \(0.0016836\). The error in the feasibility of the indices is 0.015625.
Figure 8.47: Illustration of the reduced outer approximation of $Y$ for the constraints $g_1$ on the upper left, to the constraint $g_4$ on the lower right of the problem $SIP_{DC^5}$ at $x^*$ generated by Algorithm 7.
First notice that we have $D_y^2 g_i(x, y) = 0$ for each $i \in \{1, 2, 3, 4\}$, $x \in \mathbb{R}^9$ and $y \in \mathbb{R}^3$. Thus, $g$ is concave in the second argument and we can always choose $\alpha = 0$ in Algorithm 10. In contrast to that we have $D_y g_i(x, y) = D_y z(x, y) (Dc_i)(z(x, y))$ for each $i \in \{1, 2, 3, 4\}$, where $(Dc_i)(z(x, y))$ is a constant vector for each $i \in \{1, 2, 3, 4\}$, $x \in \mathbb{R}^9$ and $y \in \mathbb{R}^3$, but $D_y z(x, y)$ is not constant. Thus, the unimodal relaxations of the constraints do not coincide with the original constraints in general. That implies that the bad performance of Algorithm 10 can not be a result of the maximum separation distance of the constraints and their relaxations. Indeed we have observed that for Algorithm 10 the termination criterion on the feasibility of the indices in the reduced outer approximation of the index set $Y$ was not satisfied for $x^{73}$, while all other termination criteria were fulfilled. Figure 8.48 displays the diameter of boxes that might contain active indices in the reduced outer approximation of $Y$ generated by Algorithm 10 and Algorithm 7. The minimum of the diameter of boxes generated by Algorithm 10 is larger than the one generated by Algorithm 7, while the maximum is similar. In Figure 8.49 the overall number of boxes generated by Algorithm 10 and Algorithm 7 is illustrated. It can be seen that the number of boxes generated by Algorithm 7 is much larger than the number of boxes generated by Algorithm 10. Though, cf. Figure 8.47, the approximation of $Y$ for each constraint generated by Algorithm 7 is only good locally. In the last iteration the overall number of boxes generated by Algorithm 10 is 780, and the overall number of boxes generated by Algorithm 7 is 15546. Since the index sets $Y$ for each constraint is a ball, certainly, both algorithms must generate a very large number of boxes to approximate $Y$. However, the number of generated boxes in the last iteration corresponds to a subproblem $P_{\alpha BB}$ with an overall number of 7030 variables and 7801 constraints in Algorithm 10, and to a subproblem $P_{UF}$ with an overall number of 9 variables and 15546 constraints in Algorithm 7. We have to notice here that 4680 of the constraints in the last subproblem in Algorithm 10 are complementarity constraints. Since the subproblems arising in Algorithm 10 are mathematical problems with complementarity constraints.
and the subproblems arising in Algorithm 7 are standard nonlinear problems, we have that the latter subproblems are in some kind easier to solve, cf. Section 7.2. Indeed, we could observe that in Algorithm 10 more than 80% of the computational time was spent for solving the subproblems.

Now, at the end of this chapter, we give a brief summary of the results. With both additional adaptation strategies of the set $X$ the performance of the basic algorithms, Algorithm 2 and Algorithm 4, could be increased. The strategy given by (7.1) performs mostly better than the one given by (7.2) on the given examples. It turned out that the choice of the parameter $\delta$ in these adaptation strategies has an enormous impact on the numerical behavior of Algorithm 7 and Algorithm 10, but, an optimal choice for this parameter depends on the structure of the problem.

Moreover, on the given examples the number of boxes in the reduced outer approximation of an index set generated by Algorithm 4 and 10 is smaller than the one generated by Algorithm 2 and 7, whereas the subproblems $P_{\alpha BB}$ arising in Algorithm 4 and 10 are in some kind more complex than the subproblems $P_{UF}$ arising in Algorithm 2 and 7. Thus, the performance of the algorithms results not only from the structure of the constraints, but also from an interaction of the latter mentioned aspects. If we disregard, for the moment the structure of the constraints, then one could expect that, both, Algorithm 4 and 10 perform better than Algorithm 2 and, respectively, Algorithm 7, if the needed number of boxes in the reduced outer approximation to approximate an index set is small enough. However, as also seen for the problem $SIP_{DC5}$, if the number of boxes in the reduced outer approximation exceeds some number, then Algorithm 2 and 7 might perform better then the other algorithms, since the numerical effort to tackle the subproblems $P_{UF}$ is lower than for $P_{\alpha BB}$. 

Figure 8.49: Overall number of boxes and those which may contain active indices. In a) are those generated by Algorithm 10 and in b) are those generated by Algorithm 7.
9 Final remarks

In this thesis we presented two basic algorithms for solving semi-infinite optimization problems with arbitrary index sets whose iterates are feasible for the original problem. Furthermore we showed that they identify an approximately stationary point of semi-infinite problem within finitely many steps. Additionally we discussed two $X$-adaptation strategies, a hybrid method and showed their convergence. The first numerical tests showed that the basic algorithms are implementable, but computationally inefficient. The reason for this misbehavior could be traced back to the large number of boxes in the reduced outer approximation of the index set. To overcome the inefficiency we introduced an additional adaptation of the set $X$. In contrast to the results for the basic algorithms, the algorithms with an additional $X$-adaptation strategy perform much better. However, several new questions arise, not only from the numerical examples.

It turned out that the size of the parameters $L$, $\bar{L}$ and $\alpha$ play an important role in the behavior and performance of the algorithms, since the influence on the error made by relaxing the constraints can not be neglected, even though this error decreases with the diameter of the boxes in the reduced outer approximation. Our approach to compute these parameters is based on the techniques of interval arithmetic, although we have to accept drawbacks like the dependency effect. However, there are other, more complex, techniques which can be used to obtain values for the needed parameters. One might, for example, use affine arithmetic, cf. [11], to reduce the influence of the dependency effect in the computation, and, thus, obtain more accurate values. Another possibility to compute values for the parameter $\alpha$ for a given constraint $g$ on a set $X \times Y$ is given by semidefinite optimization. One can obtain that parameter as the solution of

$$\min_{\alpha} \alpha \quad \text{s.t.} \quad D^2_y g(x, y) + \alpha E \preceq 0 \quad \text{for all } (x, y) \in X \times Y,$$

where $E$ denotes the unit matrix. Of course, the solution of this problem may be a hard task. For details on semidefinite optimization we refer to the review [7] and the references there in. Apart from that it would also be possible to use another reformulation technique, and, thus, construct a different solver. It is not hard to see that SIP is equivalent to the problem

$$\min_{x \in X} f(x) \quad \text{s.t.} \quad \max_{y \in Y} g(x, y) \leq 0.$$

If we assume now that $g$ is concave in the second argument and $Y$ is a convex, compact set and possesses a Slater point, then we have from duality theory that this problem is
9 Final remarks

equivalent to

\[
\min_{x \in X} f(x) \quad \text{s.t.} \quad L(x, y, \gamma) \leq 0 \\
\nabla_y L(x, y, \gamma) = 0 \\
\gamma \geq 0.
\]

Here \( L \) denotes the Lagrangian function of the problem \( \max_{y \in Y} g(x, y) \) and \( \gamma \) is the corresponding multiplier. A solution of the latter problem is, under the made assumptions, also a solution of SIP. If \( g \) is not concave in \( y \) or the index set \( Y \) is not convex, one might use the techniques presented in this thesis to construct approximations of SIP possessing the needed properties, and then try to approximate a solution of SIP with this reformulation technique. However, if \( g \) is not concave in \( y \) the presented concavification technique ensures that \( g \) is only 'piecewise concave' on \( Y \), thus, one has to check if one can apply duality theory to this problem, and if this reformulation yields an equivalent problem.

Other crucial points in the presented algorithms are the reduced outer approximation \( B^N \) used to approximate an index set \( Y \), and the refinement step for the boxes in \( B^N \). As the given numerical examples actually showed, naturally, one might need a large amount of boxes to approximate the boundary, or a part of the boundary, for an arbitrary index set \( Y \). The examples \( SIP_{DC^2} \) and \( SIP_{DC^3} \) also showed that a transformation of \( Y \) to a box shaped set, so that it can be handled exactly by the algorithms, might solve this problem, but moves the approximation error of the reduced outer approximation to the error made in the relaxations. Nevertheless it is possible to use other sets, for example simplices instead of boxes, to build a tessellation, and, thus, another type of a reduced outer approximation which might be more adequate to approximate some index set. Of course, one would also need appropriate relaxation techniques. Another possibility to reduce the amount of boxes could be a re-gluing of boxes in a special sense, and, thus, reduce the number of variables and constraints in the subproblems of the algorithms. As discussed in [15] for the \( \alpha BB \) relaxation, it is possible to construct a smooth relaxation on the entire tessellation covering the index set \( Y \). Even though none of the boxes in the tessellation are glued together, one could reduce the necessary number of relaxations by using the ideas of these technique to construct a unimodal or convex relaxation on the union of some boxes in the reduced outer approximation. Attention should be paid to the fact that these boxes must be neighboring. In addition the choice of the boxes on which such a relaxation technique should be made is not elementary. In the refinement step for a box in \( B^N \) we used an active index as splitting point for the corresponding box. As we discussed for the hybrid method on example \( SIP_{DC^1} \), this choice might be poor, since there might be more suitable points. Locating such a point is a hard task, as we pointed out there. However, the refinement step splits the longest among some special edges of a box, and it is possible to modify the choice of the edge which is split, as long as one ensures that the generated sets or non degenerated. Thus, it might be possible to split up the sets in a way that the relaxation error becomes smaller than with the presented choice.
We have to mention that the presented examples of the adaptation strategies for the set $X$ in Section 7.4 are not the only possibilities to choose subsets of $X$ satisfying Condition 5.1.7. Another way might be to replace the gradient of the function $f$ in the strategy (7.2) by the gradient with respect to $x$ of the Lagrangian function of the problem $SIP$. For the multipliers and the needed indices one could use the ones determined in the subproblems $P_{UF}$ or $P_{aBB}$. One might also use ideas from trust-region methods to construct some subsets of $X$, as long as one can ensure that Condition 5.1.7 or equivalent conditions are satisfied.
Bibliography


Bibliography


[44] S. Scholtes, Convergence Properties of a Regularization Scheme for Mathematical Programs with Complementarity Constraints, SIAM J. Optim., 11 (2001), pp. 918-


# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Illustration of $\hat{G}(B, c)$ and $\hat{G}(B, c^+)$ for $g(y) = \sin(x)$ on $B = [0, 2\pi]$</td>
<td>10</td>
</tr>
<tr>
<td>2.2</td>
<td>$\tilde{G}(y; c)$ for $g(y) = \sin(y)$ on $B = [0, 2\pi]$ with $c = \frac{\pi}{2}$ and $\tilde{L}(B) = -1.1$, $\tilde{U}(B) = 1.1$</td>
<td>11</td>
</tr>
<tr>
<td>2.3</td>
<td>$\tilde{G}(y; c)$ for $g(y) = \sin(y) + (y_2 - 1)^2$ on $B = [0, 2\pi] \times [0, 3]$ with $c = (\frac{\pi}{2}, \frac{3}{2})^T$ and $\tilde{L}(B) = (-1.1, -2.1)^T$, $\tilde{U}(B) = (1.1, 4.1)^T$</td>
<td>12</td>
</tr>
<tr>
<td>2.4</td>
<td>$\tilde{G}(y; c)$ for $g(y) = \sin(y)$ on $B = [0, 2\pi]$ and for $g(y) = \sin(y_1) + (y_2 - 1)^2$ on $B = [0, 2\pi] \times [0, 3]$ with optimal centers $c^+$</td>
<td>13</td>
</tr>
<tr>
<td>2.5</td>
<td>$\tilde{g}(y; \alpha, B)$ for $g(y) = \sin(y)$ on $B = [0, 2\pi]$ with $\alpha = \frac{1}{4}$ and with $\alpha = 1$</td>
<td>15</td>
</tr>
<tr>
<td>3.1</td>
<td>The overestimator $\tilde{g}$ for $g(x, y) = \sin(xy)$ on $X \times B$ with $X = B = [0, 2\pi]$, $\tilde{L} = -\frac{63}{19}$, $\tilde{U} = \frac{63}{19}$ and $c = \pi$</td>
<td>18</td>
</tr>
<tr>
<td>3.2</td>
<td>Sequence of splitting points $(\eta^\nu)<em>{\nu}$ with $\eta^\nu = (\epsilon</em>{\text{split}}, \frac{1}{\nu+1})$ and the corresponding sequence of boxes tending to a degenerated box</td>
<td>22</td>
</tr>
<tr>
<td>3.3</td>
<td>Illustration of the error functions at $x^+$ for $SIP_{CA}$ computed with Algorithm 2</td>
<td>33</td>
</tr>
<tr>
<td>3.4</td>
<td>Illustration of the final result $x^+$ for $SIP_{CA}$ computed by Algorithm 2</td>
<td>34</td>
</tr>
<tr>
<td>3.5</td>
<td>Overall number of boxes in each iteration for $SIP_{CA}$ generated by Algorithm 2</td>
<td>34</td>
</tr>
<tr>
<td>3.6</td>
<td>Diameter of boxes, $\tilde{U}$, $\tilde{L}$ and maximum separation distance of the constraints for the problem $SIP_{CA}$ computed by Algorithm 2</td>
<td>35</td>
</tr>
<tr>
<td>4.1</td>
<td>The overestimator $\tilde{g}$ for $g(x, y) = \sin(xy)$ on $X \times B$ with $X = B = [0, 2\pi]$ and $\alpha = \frac{7\pi}{19}$</td>
<td>38</td>
</tr>
<tr>
<td>4.2</td>
<td>Illustration of the error functions at $x^+$ for $SIP_{CA}$ computed with Algorithm 4</td>
<td>47</td>
</tr>
<tr>
<td>4.3</td>
<td>Illustration of the final result $x^+$ for $SIP_{CA}$ computed by Algorithm 4</td>
<td>48</td>
</tr>
<tr>
<td>4.4</td>
<td>Overall number of boxes in each iteration for $SIP_{CA}$ generated by Algorithm 4</td>
<td>48</td>
</tr>
<tr>
<td>4.5</td>
<td>Diameter of boxes, $\alpha$ and maximum separation distance of the constraints for the problem $SIP_{CA}$ computed by Algorithm 4</td>
<td>49</td>
</tr>
<tr>
<td>7.1</td>
<td>Illustration of the set given by $x_1 \geq 0$, $x_2 \geq 0$, $x_1x_2 = 0$.</td>
<td>85</td>
</tr>
<tr>
<td>7.2</td>
<td>Illustration of the sets $C$ and $C^t$ for $t \in {0.3, 0.15, 0.05}$.</td>
<td>86</td>
</tr>
<tr>
<td>7.3</td>
<td>Illustration of the function $g(x, y) = x^4 - 6x^2 + 8x - 6y^2$ on $X = [-2, 2]$, $Y = [0, 1]$.</td>
<td>88</td>
</tr>
</tbody>
</table>

147
List of Figures

7.4 Illustration of the function $g(x, y) = x^2 + y^2 \sin(y) + y$ and its overestimator $\hat{g}^{\mu,k}$. .......................................................... 91

8.1 Overall number of boxes for $SIP_{CA}$ generated by Algorithm 7. ............. 96
8.2 Diameter of boxes, $\overline{L}$, $\underline{L}$ and maximum separation distance of the constraints for the problem $SIP_{CA}$ computed by Algorithm 7. ................. 97
8.3 Overall number of boxes for $SIP_{CA}$ generated by Algorithm 10. .............. 99
8.4 Diameter of boxes, $\alpha$ and maximum separation distance of the constraints for the problem $SIP_{CA}$ computed by Algorithm 10. ....................... 99
8.5 Overall number of boxes for $SIP_{CA}$ generated by Algorithm 14. ............ 101
8.6 Diameter of boxes, $\alpha$, $\overline{L}$, $\underline{L}$ and maximum separation distance of the constraints for the problem $SIP_{CA}$ computed by Algorithm 14. .............. 102
8.7 Illustration of the container $C$ for $SIP_{DC1}$. .................................. 102
8.8 Overall number of boxes for $SIP_{DC1}$ generated by Algorithm 2. ............ 104
8.9 Diameter of boxes, $\overline{L}$, $\underline{L}$ and maximum separation distance of the constraints for the problem $SIP_{DC1}$ computed by Algorithm 2. .......... 105
8.10 Illustration of the reduced outer approximation of $Y$ for $SIP_{DC1}$ generated by Algorithm 2. .................................................... 105
8.11 Illustration of the final result $x^*$ for $SIP_{DC1}$ computed by Algorithm 4. .... 106
8.12 Illustration of the reduced outer approximation of $Y$ for $SIP_{DC1}$ generated by Algorithm 4. .................................................... 107
8.13 Overall number of boxes for $SIP_{DC1}$ generated by Algorithm 4. ............ 107
8.14 Maximum and minimum diameter of boxes containing active indices for $g_1$ and $g_2$ of the problem $SIP_{DC1}$ generated by Algorithm 4. .......... 108
8.15 Illustration of the reduced outer approximation of $Y$ for $SIP_{DC1}$ generated by Algorithm 7 with the adaptation strategy (7.1). ......................... 109
8.16 Illustration of the reduced outer approximation of $Y$ for $SIP_{DC1}$ generated by Algorithm 7 with the adaptation strategy (7.2). ......................... 110
8.17 Illustration of the reduced outer approximation of $Y$ for $SIP_{DC1}$ generated by Algorithm 10 with the adaptation strategy (7.1). ...................... 111
8.18 Illustration of the reduced outer approximation of $Y$ for $SIP_{DC1}$ generated by Algorithm 10 with the adaptation strategy (7.2). ...................... 111
8.19 Overall number of boxes for $SIP_{DC1}$ generated by Algorithm 7 and 10. .... 112
8.20 Diameter of boxes and maximum separation distance of the constraints for the problem $SIP_{DC1}$ computed by Algorithm 7 and 10. ................. 112
8.21 Overall number of boxes for $SIP_{DC1}$ generated by Algorithm 14. ........... 114
8.22 Diameter of boxes and maximum separation distance of the constraints for the problem $SIP_{DC1}$ computed by Algorithm 14. ...................... 114
8.23 The function $g(y) = y^3 + y^2$ on $Y = [-2,1]$ and the relaxations $\hat{g}$ and $\hat{g}_1$. 115
8.24 Illustration of the container $C$ for $SIP_{DC2}$ and $SIP_{DC3}$. .................... 117
8.25 Illustration of the final result $x^*$ for $SIP_{DC2}$ computed by Algorithm 7. .... 118
8.26 Illustration of the reduced outer approximation of $Y$ for $SIP_{DC2}$ generated by Algorithm 7 with the adaptation strategy (7.1). ....................... 119
List of Figures

8.27 Illustration of the reduced outer approximation of $Y$ for $SIP_{DC^2}$ generated by Algorithm 7 with the adaptation strategy (7.2). .......................... 119
8.28 Overall number of boxes for $SIP_{DC^3}$ generated by Algorithm 7. .................. 120
8.29 Illustration of the reduced outer approximation of $Y$ for $SIP_{DC^2}$ generated by Algorithm 10 with the adaptation strategy (7.2). ................. 121
8.30 Illustration of the reduced outer approximation of $Y$ for $SIP_{DC^2}$ generated by Algorithm 10 with the adaptation strategy (7.1). ................. 121
8.31 Illustration of the reduced outer approximation of $Y$ for $SIP_{DC^2}$ generated by Algorithm 10 with the adaptation strategy (7.2). ................. 121
8.32 Illustration of the reduced outer approximation of $Y$ for $SIP_{DC^3}$ generated by Algorithm 10 with the adaptation strategy (7.2). ................. 122
8.33 Illustration of the reduced outer approximation of $Y$ for $SIP_{DC^2}$ generated by Algorithm 7 with the adaptation strategy (7.1). ................. 124
8.34 Illustration of the reduced outer approximation of $Y$ for $SIP_{DC^2}$ generated by Algorithm 7 with the adaptation strategy (7.2). ................. 124
8.35 Illustration of the tessellation of $Y$ for $SIP_{DC^3}$ generated by Algorithm 7 with the adaptation strategy (7.1). .......................... 125
8.36 Illustration of the tessellation of $Y$ for $SIP_{DC^3}$ generated by Algorithm 7 with the adaptation strategy (7.2). .......................... 125
8.37 Illustration of the reduced outer approximation of $Y$ for $SIP_{DC^2}$ generated by Algorithm 7 with the adaptation strategy (7.2). ................. 126
8.38 Illustration of the reduced outer approximation of $Y$ for $SIP_{DC^2}$ generated by Algorithm 10 with the adaptation strategy (7.1). .......................... 127
8.39 Illustration of the tessellation of $Y$ for $SIP_{DC^3}$ generated by Algorithm 10 with the adaptation strategy (7.1). .......................... 128
8.40 Illustration of the tessellation of $Y$ for $SIP_{DC^3}$ generated by Algorithm 10 with the adaptation strategy (7.2). .......................... 128
8.41 Overall number of boxes for $SIP_{DC^2}$ and $SIP_{DC^3}$ generated by Algorithm 7 and X-adaptation (7.1). .......................... 129
8.42 Diameter of boxes, $\alpha$ and maximum separation distance of the constraints for the problems $SIP_{DC^2}$ and $SIP_{DC^3}$ computed by Algorithm 10. .......................... 130
8.43 Illustration of the final result $x^*$ for $SIP_{DC^4}$ computed by Algorithm 7 and the X-adaptation strategy (7.1). .......................... 132
8.44 Illustration of the final result $x^*$ for $SIP_{DC^4}$ computed by Algorithm 10 and the X-adaptation strategy (7.1). .......................... 133
8.45 Illustration of the container $C$ for $SIP_{DC^5}$. .......................... 134
8.46 Illustration of the final result $x^*$ for $SIP_{DC^5}$ computed by Algorithm 7. .......................... 135
8.47 Illustration of the reduced outer approximation of $Y$ for $SIP_{DC^2}$ generated by Algorithm 7. .......................... 136
8.48 Diameter of boxes in the reduced outer approximation of $Y$ for $SIP_{DC^5}$ generated by Algorithm 7 and 10. .......................... 137
8.49 Overall number of boxes for $SIP_{DC^5}$ generated by Algorithm 7 and 10. .......................... 138

149
# List of Algorithms

<table>
<thead>
<tr>
<th></th>
<th>Algorithm Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Splitting step - $\text{refine}_{UF}(\eta)$</td>
<td>23</td>
</tr>
<tr>
<td>2</td>
<td>Adaptive reduction algorithm - $\text{ara}$</td>
<td>24</td>
</tr>
<tr>
<td>3</td>
<td>Splitting step - $\text{refine}_{\alpha BB}(\eta)$</td>
<td>41</td>
</tr>
<tr>
<td>4</td>
<td>Adaptive convexification algorithm - $\text{aca}$</td>
<td>42</td>
</tr>
<tr>
<td>5</td>
<td>Splitting step - $X\text{refine}_{UF}(\eta)$</td>
<td>55</td>
</tr>
<tr>
<td>6</td>
<td>$X$-Adaptation - $X\text{adapt}_{UF}(x, \Delta x)$</td>
<td>59</td>
</tr>
<tr>
<td>7</td>
<td>Adaptive reduction algorithm - $X\text{ara}$</td>
<td>60</td>
</tr>
<tr>
<td>8</td>
<td>Splitting step - $X\text{refine}_{\alpha BB}(\eta)$</td>
<td>67</td>
</tr>
<tr>
<td>9</td>
<td>$X$-Adaptation - $X\text{adapt}_{\alpha BB}(x, \Delta x)$</td>
<td>70</td>
</tr>
<tr>
<td>10</td>
<td>Adaptive convexification algorithm - $X\text{aca}$</td>
<td>71</td>
</tr>
<tr>
<td>11</td>
<td>Error computation step - $\text{err}_{hyb}(X^\nu, B^{(1)}, B^{(2)}, \alpha, L, \hat{K}^\nu, \hat{K}^\nu)$</td>
<td>78</td>
</tr>
<tr>
<td>12</td>
<td>Splitting step - $\text{refine}_{hyb}(\nu, \eta)$</td>
<td>79</td>
</tr>
<tr>
<td>13</td>
<td>$X$-Adaptation - $X\text{adapt}_{hyb}(x, \Delta x)$</td>
<td>80</td>
</tr>
<tr>
<td>14</td>
<td>Adaptive hybrid algorithm - $\text{aha}$</td>
<td>81</td>
</tr>
<tr>
<td>15</td>
<td>Splitting step with monotonicity test - $X\text{refine}_{\alpha BB}(\eta)$</td>
<td>92</td>
</tr>
<tr>
<td>16</td>
<td>Monotonicity test - $X\text{testmono}_{\alpha BB}(X, B^{(1)}, B^{(2)})$</td>
<td>93</td>
</tr>
</tbody>
</table>
Curriculum Vitae

Heinz-Paul Steuermann

geboren am 17. September 1979 in Mönchengladbach

1986 - 1990  Grundschule, Gierath
1990 - 1999  Franz-Meyers-Gymnasium, Mönchengladbach
2000 - 2001  Studium der Mathematik an der RWTH-Aachen
2001 - 2007  Studium der Mathematik an der Heinrich-Heine Universität Düsseldorf
Seit 2007  Wissenschaftlicher Mitarbeiter am Institut für Operations Research des Karlsruher Institut für Technologie
Erklärung

(gemäß §4, Abs. 4 der Promotionsordnung vom 27.12.2006)

Ich versichere wahrheitsgemäß, die Dissertation bis auf die in der Abhandlung angegebene Hilfe selbständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und genau kenntlich gemacht zu haben, was aus Arbeiten anderer und aus eigenen Veröffentlichungen unverändert oder mit Abänderungen entnommen wurde.

Karlsruhe, den 11.05.2011