Global Optimization Algorithms for Semi-Infinite and Generalized Semi-Infinite Programs

by

Panayiotis Lemonidis

Submitted to the Department of Chemical Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Chemical Engineering Practice

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2008

Author .....
Department of Chemical Engineering
October 2008

Certified by .....
Paul I. Barton
Professor
Thesis Supervisor

Certified by .....
William H. Green Jr.
Associate Professor
Thesis Supervisor

Accepted by .....
William M. Deen
Chairman, Department Committee on Graduate Students

Global Optimization Algorithms for Semi-Infinite and Generalized Semi-Infinite Programs

by

Panayiotis Lemonidis

Submitted to the Department of Chemical Engineering
on October 2008, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Chemical Engineering Practice

Abstract

The goals of this thesis are the development of global optimization algorithms for semi-infinite and generalized semi-infinite programs and the application of these algorithms to kinetic model reduction.

The outstanding issue with semi-infinite programming (SIP) was a methodology that could provide a certificate of global optimality on finite termination for SIP with nonconvex functions participating. We have developed the first methodology that can generate guaranteed feasible points for SIP and provide ε-global optimality on finite termination. The algorithm has been implemented in a branch-and-bound (B&B) framework and uses discretization coupled with convexification for the lower bounding problem and the interval constrained reformulation for the upper bounding problem. Within the framework of SIP we have also proposed a number of feasible-point methods that all rely on the same basic principle; the relaxation of the lower-level problem causes a restriction of the outer problem and vice versa. All these methodologies were tested using the Watson test set. It was concluded that the concave overestimation of the SIP constraint using McCormcick relaxations and a KKT treatment of the resulting expression is the most computationally expensive method but provides tighter bounds than the interval constrained reformulation or a concave overestimator of the SIP constraint followed by linearization. All methods can work very efficiently for small problems (1-3 parameters) but suffer from the drawback that in order to converge to the global solution value the parameter set needs to subdivided. Therefore, for problems with more than 4 parameters, intractable subproblems arise very high in the B&B tree and render global solution of the whole problem infeasible.

The second contribution of the thesis was the development of the first finite procedure that generates guaranteed feasible points and a certificate of ε-global optimality for generalized semi-infinite programs (GSIP) with nonconvex functions participating. The algorithm employs interval extensions on the lower-level inequality constraints and then uses discretization and the interval constrained reformulation for the lower and upper bounding subproblems, respectively. We have demonstrated that

our method can handle the irregular behavior of GSIP, such as the non-closedness of the feasible set, the existence of re-entrant corner points, the infimum not being attained and above all, problems with nonconvex functions participating. Finally, we have proposed an extensive test set consisting of both literature an original examples. Similar to the case of SIP, to guarantee ε-convergence the parameter set needs to be subdivided and therefore, only small examples (1-3 parameters) can be handled in this framework in reasonable computational times (at present).

The final contribution of the thesis was the development of techniques to provide optimal ranges of valid reduction between full and reduced kinetic models. First of all, we demonstrated that kinetic model reduction is a design centering problem and explored alternative optimization formulations such as SIP, GSIP and bilevel programming. Secondly, we showed that our SIP and GSIP techniques are probably not capable of handling large-scale systems, even if kinetic model reduction has a very special structure, because of the need for subdivision which leads to an explosion in the number of constraints. Finally, we propose alternative ways of estimating feasible regions of valid reduction using interval theory, critical points and line minimization.

Thesis Supervisor: Paul I. Barton
Title: Professor

Thesis Supervisor: William H. Green Jr.
Title: Associate Professor

Acknowledgments

At the verge of completing my PhDCEP thesis I would like to express my gratitude to all the people that have helped me shape this amazing journey.

First and foremost, I would like to thank my academic advisor, Prof. Paul Barton. Despite the short 3-year research period of the PhDCEP program, he provided me with the opportunity to work in cutting-edge optimization. By giving me a lot of help to jump start the thesis and then letting me construct my own path with proper feedback, he has been very influential to my thesis.

I would also really like to thank my academic co-advisor, Prof. Bill Green for his continuous support and specifically for constantly reminding me that at the end of the day I need to deliver something useful and implementable to the engineering community. I would like to thank my thesis committee members Prof. George Stephanopoulos and Prof. Dimitri Bertsimas for their continuous support in thesis committee meetings and in many personal discussions.

I would also like to thank Prof. Alan Hatton for providing with the unique opportunity of being a practice school director in Singapore for two months. That was a wonderful teaching experience and a great opportunity to work at two great companies, Merck and Schering Plough.

I have been extremely fortunate to be part of two very special groups within the chemical engineering community at MIT, namely the Barton and Green groups. I thank them all for their endless support and I would like to thank certain people distinctly. First of all, Chakun and Benoit for their help in coding and academic as well as personal advice and Binita and Luwi for their fruitful discussions on matters concerning various aspects of my thesis. I would also like to thank Ajay for the endless hours we spent in the lab together discussing all issues from research to global warming over salty peanuts. Ajay, *it is what it is*!

I would like to warmfully thank Alexander, a wonderful friend and an academic rising star. We had an excellent research co-operation on semi-infinite and bilevel optimization and I really appreciate his help for the capstone paper as well. Alex, I wish

you all the best with your endeavors and I would love to read the first global optimization algorithm on nonconvex hybrid dynamic mixed-integer trilevel programming problems one day!

I would also like to express my warmest gratitude to the Alexander S. Onassis foundation for their financial assistance during the last year of my thesis work. Their help has been absolutely instrumental for my graduate studies.

Pano, Iasona, Yanni, Vasili, Yanir, Kosta, Wendy, Polina and Hari I thank you so much for your friendship and support over all these years. I do consider you my life-long friends.

I would not be in the position to study at MIT if it weren't for my parents, Kaiti and Petros and my sister Maria. They have always been my engine, my support, my driving force. This thesis is dedicated to you.

To my sister and parents

Maria, Kaiti and Petros

Contents

List of Figures

List of Tables

Chapter 1

Introduction to Semi-Infinite Programs

Semi-infinite programs (SIP) are optimization problems that involve a finite number of decision variables subject to an (potentially) infinite number of constraints. They are encountered in various engineering and economic applications and arise, for example, when a constraint needs to be introduced for every point in a geometric region.

The general formulation of SIP that we will be concerned with is:

$$\begin{aligned} f^* = \min_{\mathbf{x}} \; & f(\mathbf{x}) \\ \text{s.t. } & g(\mathbf{x}, \mathbf{p}) \leq 0, \quad \forall \mathbf{p} \in P \subset \mathbb{R}^{n_p} \\ & \mathbf{x} \in X \subset \mathbb{R}^{n_x}. \end{aligned} \tag{1.1}$$

Similar to finite optimization problems, SIP involve a finite number of decision variables $\mathbf{x}$, an objective function f that depends on these decision variables and a constraint g that determines the feasible set of the problem. In contrast to finite optimization problems, however, SIP involve an infinite number of constraints that are generated from the infinite set P. We begin our analysis with a brief overview of the origin of SIP and the engineering applications that give rise to SIP. We will then describe some well-known methodologies for the numerical treatment of SIP and comment on their limitations.

1.1 Origin and Engineering Applications

The term "Semi-Infinite Programs" first appeared in [31] where Charnes, Cooper and Kortanek introduce the dual of the Haar program [53]. The original Haar program involves the maximization of a linear function of infinitely many variables subject to a finite number of linear inequalities. The corresponding dual program involves the minimization of a linear function of finitely many variables over a convex set defined by an infinite number of linear inequalities. This program is a special case of a semi-infinite program.

One of the most classical engineering applications that gives rise to SIP is the Chebyshev approximation problem (CAP). Let $f : P \to \mathbb{R}$ be a twice continuously differentiable function on $P \subset \mathbb{R}^4$. Let $g : \mathbb{R}^2 \times \mathbb{R}^6 \to \mathbb{R}$ be an approximating function of f that is parameterized in $\mathbf{x}$, e.g., $g(\mathbf{x}, \mathbf{p}) = x_1 p_1^2 + x_2 p_1 p_2 + x_3 p_2^2 + x_4 p_2 + x_5 p_1 + x_6$. The ultimate goal is to minimize the error ε such that the Chebyshev norm (max-norm) of the difference between the original function f and the approximating function $g(\cdot, \mathbf{p})$ on $P \subset \mathbb{R}^2$ is less than ε. This formulation gives rise to the following SIP:

$$\begin{aligned} &\min_{\mathbf{x},\varepsilon} \varepsilon \\ &\text{s.t. } |f(\mathbf{p}) - g(\mathbf{x}, \mathbf{p})| \leq \varepsilon, \forall \mathbf{p} \in P \subset \mathbb{R}^2 \\ &\qquad\qquad \mathbf{x} \in X \subset \mathbb{R}^6, \varepsilon \in \mathbb{R}. \end{aligned} \tag{1.2}$$

We refer the reader to [99] for a review of CAP problems. Some engineering applications that give rise to CAP problems can be found in [37, 40, 86, 89, 123].

A broad class of applications that gives rise to SIP problems originates from reformulating optimization problems with uncertain parameters as worst-case scenario

design problems. Consider the problem:

$$\begin{aligned} &\min_{\mathbf{x}\in X} f(\mathbf{x}) \\ &\text{s.t. } g(\mathbf{x},\mathbf{p}) \leq 0 \\ &\qquad \mathbf{x} \in X, \end{aligned} \tag{1.3}$$

where, for example, f is the total production cost of a pharmaceutical, $\mathbf{x}$ are the decision variables and $\mathbf{p}$ are parameters the values of which are determined upstream in the production. If the parameters $\mathbf{p}$ are certain, then problem (1.3) is an ordinary finite nonlinear program. However, if the parameters are uncertain and allowed to vary within a range, e.g. a n_p - dimensional interval $P = [\mathbf{p}^L, \mathbf{p}^U]$ and, furthermore, we want to ensure that the operational constraint g is satisfied for all $\mathbf{p} \in P$ (worst-case scenario design) then (1.3) is reformulated as SIP (1.1).

Within the context of worst-case scenario design, a very interesting application in the area of building construction is given in [93]. Braced frame buildings are expected to withstand small earthquakes with no damage and large ones with repairable damage such that the survival of occupants is guaranteed at all times. A simple design problem is to minimize the weight of a building subject to the operational constraint that the relative horizontal displacements of the floors does not exceed a threshold for consecutive earthquakes.

SIP are also encountered in optimal control problems, such as robot trajectory planning [54], sterilization of food [69], in the flutter of aircraft wings [107] and in the design of multi-input multi-output (MIMO) control systems [93], in air pollution control [71], in game theory [70] and in kinetic model reduction [84, 90]. The latter application, a very important tool for reacting flow simulations of large-scale combustion mechanisms, will be analyzed in detail in Chapter 5 where the applicability of the algorithms developed in this thesis will be examined.

1.2 Definitions

Definition 1.1. (Decision Variables, Parameters)
The decision variables of the SIP are denoted $\mathbf{x} \in X$ while the auxiliary parameters are denoted $\mathbf{p} \in P$.

Definition 1.2. (Host Sets)
$X \subset \mathbb{R}^{n_x}$ and $P \subset \mathbb{R}^{n_p}$ are the host sets of the decision variables and the parameters, respectively. Typically, X and P are assumed to be compact.

Definition 1.3. (Defining Functions)
$f : X \to \mathbb{R}$ is the objective function of the SIP and $g : X \times P \to \mathbb{R}$ defines the semi-infinite constraint.

Definition 1.4. (Lower-Level Problem)
For a given $\bar{\mathbf{x}} \in X$ the lower-level problem is defined as:

$$O(\bar{\mathbf{x}}, P) = \max_{\mathbf{p} \in P} g(\bar{\mathbf{x}}, \mathbf{p}).$$

To ensure the existence of $O(\mathbf{x}, P)$ for each $\mathbf{x} \in X$ we will assume that $g(\mathbf{x}, \cdot)$ is continuous on P for each $\mathbf{x} \in X$ and P is compact.

Definition 1.5. (Feasible Set of the SIP)
The feasible set of the SIP is defined as:

$$M = \{\mathbf{x} \in X \mid G(\mathbf{x}) \leq 0\}.$$

Definition 1.6. (Active Index Set)
For a given $\bar{\mathbf{x}} \in M$, the index set of the active constraints is defined as:

$$P_0(\bar{\mathbf{x}}) = \{\mathbf{p} \in P \mid g(\bar{\mathbf{x}}, \mathbf{p}) = 0\}.$$

Definition 1.7. (SIP Slater point)
A point $\mathbf{x} \in M$ is a SIP Slater point if $P_0(\mathbf{x}) = \emptyset$. Finally the set of Slater points of the SIP is denoted by $X_s \subset M$.

Definition 1.8. (Interior Point of the Feasible Set)
A point $\bar{\mathbf{x}} \in M$ is an interior point of M if for some $\varepsilon > 0$, the ball $B(\bar{\mathbf{x}}, \varepsilon)$ is a subset of M. The set of interior points of M is denoted $\text{int}(M)$.

Definition 1.9. (Global Solution Value, Points with Minimum Objective Function Value). The global solution value of the SIP is denoted by f^{SIP}. The set of points $\mathbf{x} \in M$ for which $f(\mathbf{x}) = f^{SIP}$ are denoted by the set X_{min}.

Definition 1.10. (Interval Extensions)
$\bar{G} : \mathbb{IR}^{n_x} \times \mathbb{IR}^{n_p} \rightarrow \mathbb{IR}$, is an interval-valued function and refers to the interval extension of the SIP constraint with respect to both the decision variables $\mathbf{x}$ and the parameters $\mathbf{p}$. Specifically, if X and P are intervals, $\bar{G}(X, P) = [\bar{G}^L(X, P), \bar{G}^U(X, P)]$, where $\bar{G}^L$ and $\bar{G}^U$ are real-valued functions and are called the lower-bounding and upper- bounding extensions, respectively, of the SIP constraint
$G : X \times \mathbb{IR}^{n_p} \rightarrow \mathbb{IR}$ is an interval-valued function that refers to an interval extension of $G(\mathbf{x}, \cdot)$ with respect to $\mathbf{p}$. If P is an interval, then $G(\mathbf{x}, P) = [g^L(\mathbf{x}, P)\,,\, g^U(\mathbf{x}, P)]$, where g^L and g^U are real-valued functions.

1.3 Clarifications on SIP

There are two important aspects in the SIP literature that we attempt to clarify:

1. Determining feasibility - Need for global optimization.

2. Slater vs. interior point of the SIP.

1.3.1 Determining Feasibility

The first misconception in the algorithmic treatment of SIP problems arises from the definition of feasibility of a given point $\bar{\mathbf{x}} \in X$. Note that, unlike finite optimization problems where feasibility of $\bar{\mathbf{x}}$ is determined by evaluating a finite number of constraints, in SIP, an infinite number of evaluations need to be carried out corresponding to the infinite number of constraints that arise in the problem. Equivalently,

using Definitions 1.4 & 1.5, feasibility of $\bar{\mathbf{x}}$ is determined by the global solution of the lower-level problem, $O(\bar{\mathbf{x}}, P)$. Price and Coope [32] state that "...It is also worth noting that the global optimization of g is to some extent an inevitable part of any algorithm for SIP if only to establish feasibility."

To illustrate the requirement for global optimization of the lower-level problem in order to guarantee feasibility we provide the following example.

Example 1.11. Consider the following SIP problem:

$$\begin{aligned} &\min_{x\in[0,1]} x \\ &\text{s.t. } g(x,p) = x\exp(0.2p)\sin p - 3x^2 \leq 0, \quad \forall p \in [0,10]. \end{aligned} \tag{1.4}$$

Consider the point $\bar{x} = 1$. Figure 1-1 shows the graph of the SIP constraint on the parameter set $[0, 10]$ at $\bar{x}$.

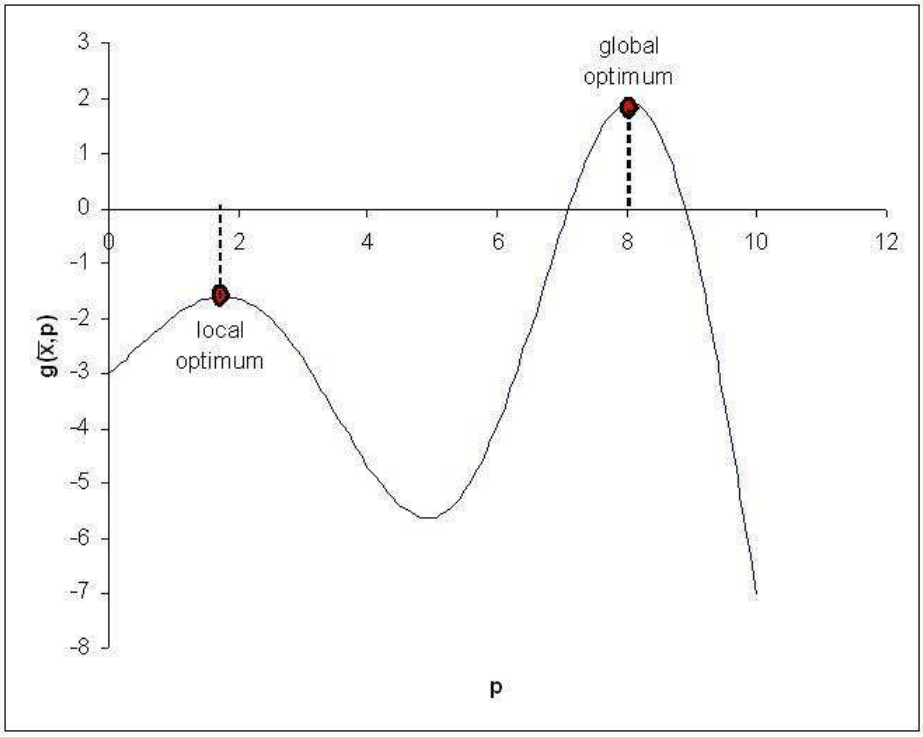

Figure 1-1: Need for Global Optimization

Figure 1-1 shows that if the lower-level problem at $\bar{x}$ is solved to local optimality,

$p^* \approx 1.7$ (local maximum) might be found. Since $g(\bar{x}, p^*) < 0$, $\bar{x}$ would be rendered feasible. This is clearly incorrect, because the global maximum of the lower-level problem is $p^{**} \approx 8.1$ with $g(\bar{x}, p^{**}) > 0$ which renders $\bar{x}$ infeasible.

Oftentimes in the SIP literature, the lower-level problem is either solved to local optimality or the global solution of the lower-level problem is estimated without, however, a certificate of global optimality.

In the former case, the source of error arises from the confusion of local vs. global optimality of the overall problem and the local vs. global optimality of the lower-level problem. There is no flexibility for the lower-level problem, i.e., this problem needs to be solved globally in order to guarantee feasibility of any point $\bar{\mathbf{x}} \in X$.

On the other hand, the overall problem, i.e., optimizing the objective function subject to the feasible set has more flexibility in that global-, local-, KKT- or stationary-based algorithms can be devised.

In the latter case, the source of error arises from the basic principle of reduction-based methods, namely that all local maxima have to be computed for some or all of the intermediate estimates of the algorithm. A multi-local approach, as it is commonly referred to in the SIP literature [32, 97], suffers from the following drawbacks:

1. Even if the cardinality of the set of local maxima has an upper bound (which, in general, cannot be explicitly known), locating a finite number of local maxima does not guarantee that a global maximum has been located. Traditionally, Newton- and stochastic- based methods have been implemented in this context, both of which do not guarantee, on finite termination, that a global maximum has been located.

2. To guarantee feasibility of a candidate $\bar{\mathbf{x}}$ it suffices to find a valid upper bound to the corresponding lower-level problem $O(\bar{\mathbf{x}}, P)$ and it is not required to find all possible local maxima of $O(\bar{\mathbf{x}}, P)$.

1.3.2 Slater vs. Interior Point of the SIP

In order to clarify this matter we start with a quote from [101]: "A feasible point x ... with $g(x, y) < 0$ for all $y \in Y$ is a Slater point or an interior point". This statement is incorrect because the topological definition of a Slater point is not equivalent to the definition of an interior point (see Definitions 1.7 and 1.8).

Specifically, assuming that g is continuous on $X \times P$ and that X and P are compact, it can be easily shown (using continuity arguments) that a SIP Slater point is an interior point of the feasible set. However, an interior point does not have to be a SIP Slater point. This implies that the set of SIP Slater points is a subset of the interior points of the SIP. The following example illustrates the difference between a SIP Slater point and an interior point:

Example 1.12. Consider the following SIP problem:

$$\begin{aligned} &\min_{\mathbf{x}\in\mathbb{R}^2} x_1 x_2 \\ &\text{s.t. } -(x_1-1)^2-(x_2-1)^2-1+p^2 \leq 0, \quad \forall \mathbf{p} \in [0,1]. \end{aligned} \tag{1.5}$$

In this example, the feasible set of the problem is $M = \mathbb{R}^2$. However for $\bar{\mathbf{x}} = (1, 1)$, which is clearly an interior point of the feasible set, the global solution of the lower-level problem is $O(\bar{\mathbf{x}}, P) = 0$ which implies that $\bar{\mathbf{x}}$ is not a Slater point.

1.4 Numerical Methods for SIP

There are two main classes of numerical methods for SIP: discretization- and local reduction-based methods. We provide a brief overview of both methodologies and a comparison in the following sections.

1.4.1 Discretization Approaches

The basic principle of discretization methods is to minimize the objective function of the SIP subject to only a finite subset of the infinite number of constraints and to

enlarge this finite subset in order to obtain a higher precision on the SIP solution. Therefore, the main idea of discretization methods is to solve the following relaxed nonlinear program:

$$\begin{aligned} &\min_{\mathbf{x}\in X} f(\mathbf{x}) \\ &\text{s.t. } g(\mathbf{x},\mathbf{p}) \leq 0, \quad \forall \mathbf{p} \in P_i, \end{aligned} \tag{1.6}$$

where P_i is a finite subset of P and i the iteration number, and increase the solution accuracy by increasing, at successive iterations, the cardinality of P_i such that $\lim_{i\to\infty} \text{dist}(P_i, P) = \lim_{i\to\infty} \sup_{\mathbf{q}\in P_i} \inf_{\mathbf{p}\in P} ||\mathbf{q}-\mathbf{p}|| = 0$. The main characteristics of discretization methods are:

1. They are outer approximation methods. This implies that on finite termination, the incumbent solutions $\mathbf{x}^i$ are, in general, infeasible for the original SIP or equivalently that the feasible set of (1.1) is a subset of the feasible set of (1.6):

$$\{\mathbf{x} \in X : g(\mathbf{x},\mathbf{p}) \leq 0, \forall \mathbf{p} \in P\} \subset \{\mathbf{x} \in X : g(\mathbf{x},\mathbf{p}) \leq 0, \forall \mathbf{p} \in P_i\}. \tag{1.7}$$

   Recall that in order to check feasibility of any of the incumbents $\mathbf{x}^i$ the global solution value, or a valid overestimate on the global solution value, of the lower-level problem at these incumbents must be generated.

2. They provide a valid lower-bounding approach for SIP. Taking into consideration relation (1.7) and assuming that $\mathbf{x}^i$ are global minima of the discretized problems indexed by the set P_i and f^{SIP} is the global solution value of the original SIP, the following relationship holds:

$$f(\mathbf{x}^i) \leq f^{SIP}, \ \forall i \in \mathbb{N}. \tag{1.8}$$

   Of course, it must be possible to guarantee that $\mathbf{x}^i$ is a global solution of (1.6) in order for this inequality to hold.

3. If, on successive iterations, the gridding of the parameter set is exhaustive then the sequence of the optimal solution values of the discretized problems converges to the minimum of the SIP (assuming continuity of the defining functions and compactness of the host sets for the decision variables and parameters):

$$\lim_{i\to\infty} \text{dist}(P_i, P) = \lim_{i\to\infty} \sup_{\mathbf{q}\in P_i} \inf_{\mathbf{p}\in P} ||\mathbf{q}-\mathbf{p}|| = 0 \Rightarrow \lim_{i\to\infty} f(\mathbf{x}^i) = f^{SIP}. \tag{1.9}$$

4. Relation (1.9) implies that in order to converge to the minimum of the SIP, discretization approaches need to introduce an increasing and, potentially, unbounded number of constraints. For SIP that include many parameters this approach could render, very quickly, intractable finite nonlinear programs. Reemtsen and Görner state in [101] that standard nonlinear solvers can handle discretization grids with up to 100,000 points for problems with less than 100 variables.

5. It can be shown [94] that under proper assumptions, local minima and stationary points of the discretized problems converge to local minima and stationary points of the original SIP.

There are two main elements in the numerical implementation of a discretization approach for SIP:

1. The construction of the grid on the parameter set. We have already mentioned that discretization involves the generation of a finite subset of the parameter host set P. There are two main ways of generating this set: a-priori and adaptively. In the former case, or brute-force discretization, an explicit arithmetic rule is used to define the grid, e.g. :

$$P_i = \{\frac{1}{n} : n = 1, \ldots, i\}, i \geq 1. \tag{1.10}$$

when $P = [0, 1]$. To generate the grid adaptively an implicit rule is used. Many such rules have been suggested in the SIP literature [29, 48, 58, 92, 95, 100, 139].

For the sake of completeness, we will describe the discretization algorithm and the grid-defining rule stated in [29]; the rules in the other references are similar in nature.

Blankenship and Falk [29] propose an algorithm consisting of four steps:

(a) *Initialize*: set $i = 0$ and choose $P_i \subset P$.

(b) *Upper-Level Problem*: solve the i-th outer problem: compute $\mathbf{x}^i \in X$ which solves the problem $\min_{\mathbf{x} \in X} f(\mathbf{x})$ s.t. $g(\mathbf{x}, \mathbf{p}) \leq 0$, $\forall\ \mathbf{p} \in P_i$.

(c) *Lower-Level Problem*: solve the i-th lower-level problem: compute $\mathbf{p}^i \in P$ which solves the lower-level problem of $\mathbf{x}^i$: $\max_{\mathbf{p} \in P} g(\mathbf{x}^i, \mathbf{p})$.

(d) *Termination Criteria - Grid-Defining Rule*: If $g(\mathbf{x}^i, \mathbf{p}^i) \leq 0$ then stop (global solution found). Else update the grid on the parameter set by the rule: $P_{i+1} = P_i \cup \mathbf{p}^i$ and go to step 2.

This implicit rule takes advantage of the maximum violation of the semi-infinite constraint at the intermediate incumbents of the procedure. It has been observed that discretization algorithms that do take advantage of the global maxima of the lower-level problems at the intermediate incumbents converge faster than brute-force approaches.

2. The choice of the NLP solver that is used in solving the subproblems generated at each iteration of a SIP algorithm. For any SIP algorithm, the vast majority of computational effort is spent on solving the approximating NLP subproblems. Many local SQP and interior-point algorithms have been used for this purpose [52, 132]. Furthermore, there has been a specific attempt to develop both a user interface with AMPL [133] and a SQP solver [73] that can handle, more effectively, the large number of similar constraints that arise from the discretization of the semi-infinite constraint.

1.4.2 Local Reduction

In finite nonlinear optimization, around a feasible point $\mathbf{x} \in \mathbb{R}^{n_x}$ there exists a neighborhood $B(\mathbf{x}, \varepsilon)$ for which the feasible set can be precisely described by the active constraints at $\mathbf{x}$. However, it is well known [61, 101] that this result does not hold for SIP in general.

Nevertheless, under proper regularity assumptions [61, 101] it can be shown that around a point $\bar{\mathbf{x}} \in X$ there exists a neighborhood $B(\bar{\mathbf{x}}, \varepsilon)$ and also a finite number of implicitly-defined inequality constraints such that the feasible set defined by these constraints on $B(\bar{\mathbf{x}}, \varepsilon) \cap X$ coincides with the intersection of the feasible set of the SIP with $B(\bar{\mathbf{x}}, \varepsilon)$. Therefore, on $B(\bar{\mathbf{x}}, \varepsilon)$, problem (1.1) is equivalent to the following finite problem:

$$\begin{aligned} \min_{\mathbf{x} \in B(\bar{\mathbf{x}}, \varepsilon) \cap X} \; & f(\mathbf{x}) \\ \text{s.t. } & g(\mathbf{x}, \mathbf{v}^j(\mathbf{x})) \leq 0, \; \forall j \in J, \end{aligned} \tag{1.11}$$

where $\mathbf{v}^j : X \to \mathbb{R}^{n_p}$, $j \in J$, are implicitly defined vector-valued functions and J is a finite index set.

Essentially, local-reduction methods assume that for a point $\bar{\mathbf{x}}$ there exists a neighborhood $B(\bar{\mathbf{x}}, \varepsilon)$ such that for each point $\mathbf{x} \in B(\bar{\mathbf{x}}, \varepsilon)$ the number of local maxima for the lower-level problem is finite and constant. The vector-valued mappings $\mathbf{v}_j$, $j \in J$, map the point $\mathbf{x}$ to exactly one of these isolated maxima of the lower-level problem $O(\mathbf{x}, P)$.

The main characteristics of local reduction methods are:

1. They are also outer approximation methods. Similar to discretization methods, local reduction approaches only consider a finite subset of the parameter set P and thus, on finite termination, feasibility of the incumbents is not guaranteed.

2. Since the number of constraints that are necessary to describe, locally, the feasible set of the SIP is finite, local reduction methods usually do not explode in the number of constraints required in order to achieve convergence.

3. The number and the explicit functional form of these constraints are not known in general.

4. They are local in nature. This means that they offer an attractive reduction of the SIP to a finite program locally in the decision-variable space. In [101] it was shown that under regularity assumptions, a strict local minimum of the reduced problem (1.11) is also a strict local minimum of the original SIP (1.1).

There are two types of algorithms within the framework of reduction based methods. On the one hand, locally convergent methods require that the starting point $\mathbf{x}^0$ is located in the reduction neighborhood $B(\bar{\mathbf{x}}, \varepsilon)$ of a SIP KKT point $\bar{\mathbf{x}}$. On the other hand, globally convergent reduction-based methods [33, 124, 125, 135] can be carried out when the intermediate estimates do not belong to a neighborhood of a SIP KKT point. Similar to the analysis of discretization-based methods, we are going to describe a prototypical reduction-based method. As described in [101], typically, a globalized reduction-based method consists of four steps:

1. *Initialization*: Pick $\mathbf{x}^0$ and set $k = 0$.

2. *Global Solution of the Lower-Level Problem*: Compute all the local maxima of $O(\mathbf{x}^k, P)$.

3. *Inner Iterations - Adapt Local Maxima*: Apply a finite nonlinear approach to the reduced problem (1.11). Recalculate the local maxima of the intermediate iterates using local adaptation at the local maxima of $O(\mathbf{x}^k, P)$. Let $\mathbf{x}^*$ be the final estimate of the inner iterations.

4. *Termination*: If $\left|\left|\mathbf{x}^k - \mathbf{x}^*\right|\right| \leq \varepsilon$, terminate. Else set $k = k + 1$, $\mathbf{x}^{k+1} = \mathbf{x}^*$ and go to step 2.

With respect to the numerical implementation of any reduction based algorithm there are a lot of open questions:

1. What is the number, or at least an upper bound on the number, of local maxima of the lower-level problem? This is essential for Step 2 of the aforementioned

algorithm which involves the calculation, at least approximately, of all local maxima of the lower-level problem. This is bypassed in the literature by employing a fixed number of iterations to locate the local maxima.

2. How are the implicitly defined inequality constraints evaluated? This is essential in Step 3 where a nonlinear approach is used to provide an update of the local solution of the SIP. This difficulty is bypassed by evaluating the functions, first and second derivatives only at the major iteration points $\mathbf{x}^k$ [32].

3. How is the neighborhood of valid reduction $B(\bar{\mathbf{x}}, \varepsilon)$ computed? In [101] the following statement is made for globalized reduction based methods: "It has to be respected, however, that the inner iterates (i.e. the iterates in step 3) may move away from $\mathbf{x}^k$ and hence may leave $U(\mathbf{x}^k)$ (the valid region of reduction) when $\mathbf{x}^k$ is not a KKT point of the SIP problem". This neighborhood is crucial in solving the reduced problem in step 3. This problem is bypassed in the literature by ignoring the condition $\mathbf{x} \in B(\bar{\mathbf{x}}, \varepsilon)$ in problem (1.11).

It should be noted that if there is a unique global maximum for the lower-level problem throughout the neighborhood in question and it can be computed reliably, it is really only necessary to include this point in (1.11).

1.4.3 Discretization vs. Local Reduction

The major similarities between these two classes of methods are:

1. They are both outer approximation methods. Therefore, on finite termination, feasibility of the incumbent solutions is not guaranteed. To check for feasibility of an incumbent, the lower-level problem at the incumbent must be solved to global optimality.

2. They consider a finite subset of the constraints.

3. The state-of-the-art algorithms in both methodologies. calculate and use the global maxima of the lower-level problem.

The major differences between discretization- and local reduction- based methods are:

1. The necessary assumptions for convergence are much milder for discretization methods than for local reduction approaches.

2. The number of constraints that needs to be generated to ensure convergence is bounded for local reduction methods while it is unbounded, in general, for discretization methods. In other words, discretization methods explode in the number of constraints while local reduction methods remain finite.

3. At the expense of exploding, discretization methods can be used to provide valid lower bounds on the SIP global solution value and furthermore converge to the solution value when the discretization grid gets finer. Therefore they can be used for local, global, stationary and KKT approaches for SIP. On the other hand, local reduction methods provide convergence to a local solution of the SIP and are, therefore, more limited in their scope in comparison to discretization methods.

Within the context of numerical algorithms for nonconvex SIP, the classification proposed in [57] includes the so-called exchange methods as a generic class of methods that add some new constraints (cuts) at the intermediate steps of the algorithm and, with heuristics, remove some old constraints from the problem. To our understanding, exchange methods are discretization methods in nature in that they explode, in general, in the number of constraints while offering the advantage of a global, local, stationary or KKT approach to SIP.

1.5 Limitations of SIP algorithms

To conclude this introductory material, there are currently two main classes of numerical procedures for SIP: discretization and local-reduction methods. Both methodologies use a finite subset of the infinite number of constraints and provide, on finite

termination, an approximation to a global or local solution of the SIP. However, these two methodologies do not provide:

1. Guaranteed feasibility of the incumbent solutions.

2. A certificate of global optimality upon termination.

In light of these limitations, in the following sections, we are going to describe a general methodology to provide guaranteed-feasible points for SIP (and thus rigorous upper bounds to the global solution value of the SIP). This will be the basis to propose global optimization algorithms in Chapters 2 and 3.

1.6 A Bilevel Reformulation

An exact reformulation of (1.1) is the following nonsmooth program:

$$\begin{aligned} &\min_{\mathbf{x}\in X} f(\mathbf{x}) \\ &\text{s.t. } O(\mathbf{x}, P) \equiv \max_{\mathbf{p}\in P} g(\mathbf{x}, \mathbf{p}) \leq 0, \end{aligned} \tag{1.12}$$

where $O : X \to \mathbb{R}$ is well defined for all $\mathbf{x} \in X$ provided that $g(\mathbf{x}, \cdot)$ is continuous on P and P is a compact set. The following example illustrates that $O(\cdot, P)$ can be a nonsmooth function.

Example 1.13. Consider the following SIP:

$$\begin{aligned} &\min_{x\in[-1,1]} x \\ &\text{s.t. } g(x, p) = xp \leq 0, \quad \forall p \in [-1, 1]\,. \end{aligned}$$

Then $O(\cdot, P)$ is defined as:

$$O(x, P) = \begin{cases} x, & 0 \leq x \leq 1 \\ -x, & -1 \leq x < 0. \end{cases}$$

Clearly, $O(\cdot, P)$ is nonsmooth at $x = 0$. For the numerical difficulties that standard NLP solvers face with nonsmooth functions, we refer to [74].

Although (1.12) is a trivial reformulation of (1.1) it provides the basis for constructing an upper-bounding (feasible-point) approach for SIP. Specifically, if a real-valued function $h : X \to \mathbb{R}$ can be constructed such that $O(\mathbf{x}, P) \leq h(\mathbf{x})$, $\forall \mathbf{x} \in X$, then the following program:

$$\begin{aligned} &\min_{\mathbf{x} \in X} f(\mathbf{x}) \\ &\text{s.t. } h(\mathbf{x}) \leq 0, \end{aligned} \tag{1.13}$$

is a valid restriction of (1.1). Therefore any feasible point of (1.13) is a feasible point for the original SIP (1.1). Furthermore, if h is continuously differentiable on X, then (1.13) can be tackled by a local NLP solver, a feature that would facilitate the convergence of global optimization algorithms. In the following section we are going to describe how such a function can be constructed with the help of interval analysis.

1.7 Interval Methods for SIP

In order to aid the understanding of the interval approach for SIP we provide a brief introduction to interval analysis and then discuss the interval-constrained reformulation [26]. For a more detailed analysis of interval theory and for the original source of sections 1.7.1 and 1.7.2, we refer the reader to [27].

1.7.1 Interval Analysis

An n_p-dimensional interval, P, bounded by the n_p-dimensional vectors $\mathbf{p}^L$ and $\mathbf{p}^U$, is defined by:

$$P = P_1 \times \ldots \times P_{n_p} = [\mathbf{p}^L, \mathbf{p}^U]$$
$$P_j = [p_j^L, p_j^U], j = 1, \ldots, n_p.$$

The width of each dimension P_j and the overall width of P are defined to be:

$$\begin{aligned} w(P_j) &= p_j^L - p_j^U, \quad j = 1, \ldots, n_p, \\ w(P) &= \max_j w(P_j). \end{aligned}$$

The range of values assumed by a real-valued, continuous function $g : P \to \mathbb{R}$ on the domain P is denoted by the scalar interval $g_R(P)$:

$$g_R(P) = [g_R^L(P), g_R^U(P)] = \{g(\mathbf{p}) : \mathbf{p} \in P\}.$$

An interval-valued function $G : P \to \mathbb{IR}$ which satisfies the following relation is referred to as an inclusion function for g on P:

$$g_R(Q) \subset G(Q) = [g^L(Q), g^U(Q)], \; \forall Q \in \mathbb{IR}^{n_p}, Q \subset P. \tag{1.14}$$

The natural interval extension, G_{int}, is an example of such an inclusion function. For a given function g, G_{int} is derived by replacing each real variable p_j with the corresponding interval variable, P_j, and evaluating the resulting expression using the rules of interval arithmetic [85]. G_{int} can be expressed using only the bounds $\mathbf{p}^L$, $\mathbf{p}^U$ and selected constants (e.g., 1, -1, in the case of sine and cosine). For functions with special structure, e.g., rational functions in which each variable p_j appears only once, the natural interval extension yields an exact inclusion such that equality holds in (1.14). In general, the natural interval extension is inexact and overestimates the true range of a real-valued function. The tightness of any inclusion function may be quantified using the Hausdorff metric $q(g_R(P), G(P))$, which is defined as follows for the scalar intervals $g_R(P)$ and $G(P)$:

$$q(g_R(P), G(P)) = \max(\left|g_R^L(P) - g^L(P)\right|, \left|g_R^U(P) - g^U(P)\right|).$$

The width of the natural interval extension calculated for a given function g depends on the underlying expression used. In certain cases, tighter inclusions can be cal-

culated by representing the function using a different underlying expression, i.e., by rearranging the expression before evaluating its natural interval extension. One such rearrangement is the small Horner scheme [98].

For general nonlinear functions, an exact inclusion over a nondegenerate domain ($\mathbf{p}^L \neq \mathbf{p}^U$) cannot be computed with finite computational effort. However, convergent inclusions of continuous functions can provide a bound on the degree of overestimation incorporated by the inclusion:

$$q(g_R(P), G(P)) \leq \gamma w(P)^\beta \tag{1.15}$$

$$w(G(P)) \leq \delta w(P)^\beta, \tag{1.16}$$

where $\beta \geq 1$ is the convergence order of the inclusion function, e.g., $\beta = 1$ for natural interval extensions, and $\gamma \geq 0$ and $\delta \geq 0$ are constants that depend on the form of the function g, and the interval P. This property suggests that progressively tighter inclusions may be calculated at the cost of more expensive function evaluations. Each dimension of the n_p-dimensional interval $P = [\mathbf{p}^b, \mathbf{p}^u]$ may be subdivided into n_k subintervals of equal width such that:

$$P_j^\kappa = [p_j^L + \frac{(\kappa - 1)w(P_j)}{n_k}, \quad p_j^L + \frac{\kappa \cdot w(P_j)}{n_k}], \quad \kappa = 1, \ldots, n_k.$$

Denoting $I_k = \{1, 2, \ldots, n_k\}^{n_p}$, it follows that

$$P = \bigcup_{\tau \in I_{n_k}} P_\tau,$$

where $P_\tau = P_1^{\kappa_1} \times \ldots \times P_{n_p}^{\kappa_{n_p}}$ for $\tau = (\kappa_1, \kappa_2, \ldots, \kappa_{n_p}) \in I_k$. The range of the function over the interval P is then the union of the range of values assumed over each subinterval, i.e.:

$$g_R(P) = \bigcup_{\tau \in I_{n_k}} g_R(P_\tau).$$

Similarly, from inclusion monotonicity [98], the union of the $n_k^{n_p}$ interval extensions

$G(P_\tau)$ yields a valid inclusion function for $g(\mathbf{p})$ on P, i.e.,

$$\bigcup_{\tau \in I_k} g_R(P_\tau) \subset \bigcup_{\tau \in I_k} G(P_\tau) \subset G(P).$$

The inclusion $G_{n_k}(P) = \bigcup_{\tau \in I_k} G(P_\tau)$ is referred to as the n_k^{th} (uniform) refinement of G. Applying (1.15), we arrive at the following relations between G_{n_k} and g_R:

$$\begin{array}{rcccl} q(g_R(P), G_{n_k}(P)) & \leq & \gamma(w(P_\tau))^\beta & = & \gamma \left(\frac{\omega(P)}{n_k}\right)^\beta \\ w(G_{n_k}(P)) & \leq & \delta((P_\tau))^\beta & = & \delta \left(\frac{\omega(P)}{n_k}\right)^\beta. \end{array}$$

Thus the subdivision approach generates inclusion functions of arbitrary accuracy, at the cost of performing $n_k^{n_p}$ interval function evaluations. Considerable effort has been directed towards developing higher-order inclusions which converge more rapidly as the number of subdivisions increases. In general, up to second-order convergence can be achieved for inclusions of real-valued functions [7] by using the centered form. Using the following underlying expression for a given function g:

$$g(\mathbf{p}) = g(\mathbf{c}) + g_s(\mathbf{p} - \mathbf{c}), \quad \mathbf{c} \in P, \tag{1.17}$$

the centered form is calculated as the natural interval extension of the right-hand side of (1.17), and provides a quadratically-convergent inclusion for g on P. In particular, the n_T^{th} - order Taylor model of a function of $n_T + 1^{th}$order differentiability can be shown to satisfy the definition in (1.17) [7].

For a thorough review on the theory and applications of interval analysis we refer the reader to [7, 15, 55, 56, 85].

1.7.2 The Interval Constrained Reformulation

From now on we will assume that $P \subset \mathbb{IR}^{n_p}$ is an interval. A rigorous upper-bounding methodology for SIP, i.e., a methodology to generate guaranteed feasible points for SIP, can be constructed by replacing the nondifferentiable constraint in (1.12) with

one that has a smaller feasible set. Any valid inclusion function for $g(\mathbf{x}, \cdot)$ on P for each $\mathbf{x} \in X$, may be used for this purpose.

The interval-valued function $G : X \times P \to \mathbb{IR}$ which refers to an interval extension of $g(\mathbf{x}, \cdot)$ with respect to $\mathbf{p}$, is an example of such an inclusion function. Denoting $G(\mathbf{x}, P) = [g^L(\mathbf{x}, P), g^U(\mathbf{x}, P)]$ we obtain that:

$$g^U(\mathbf{x}, P) \geq \max_{\mathbf{p} \in P} g(\mathbf{x}, \mathbf{p}), \ \forall \mathbf{x} \in X. \tag{1.18}$$

Relation (1.18) implies that:

$$\{\mathbf{x} \in X : g^U(\mathbf{x}, P) \leq 0\} \subset \{\mathbf{x} \in X : \max_{\mathbf{p} \in P} g(\mathbf{x}, \mathbf{p}) \leq 0\}. \tag{1.19}$$

Therefore, the following finitely-constrained nonlinear program:

$$\begin{aligned} &\min_{\mathbf{x} \in X} f(\mathbf{x}) \\ &\text{s.t. } g^U(\mathbf{x}, P) \leq \ 0, \end{aligned} \tag{1.20}$$

provides a guaranteed feasible point for (1.1). This is the interval-constrained reformulation. It should be noted that the global solution of (1.20) is not required to furnish a feasible point for (1.1): a local, global, stationary or KKT approach would suffice for this purpose.

In order to construct the inclusion function G, the $\mathbf{p}$- and $\mathbf{x}$-dependent terms are first isolated from the remaining mixed terms in g such that:

$$\begin{aligned} g(\mathbf{x}, \mathbf{p}) &= g_x(\mathbf{x}) + g_p(\mathbf{p}) + g_{xp}(\mathbf{x}, \mathbf{p}) \\ g^U(\mathbf{x}, P) &= g_x(\mathbf{x}) + G_p^u(P) + g_{xp}^U(\mathbf{x}, P). \end{aligned} \tag{1.21}$$

The term g_x involves degenerate intervals $\mathbf{x}$ and does not require any further treatment. The inclusion term g_p^U is calculated based on interval arithmetic [7, 26]. The inclusion term g_{xp}^U is calculated similarly with the exception that the degenerate intervals $\mathbf{x}$ add an extra degree of complexity. In both inclusion bounds g_p^U and g_{xp}^U,

the resulting expression may contain min and max terms and as a consequence, the interval extension $g^U(\mathbf{x}, P)$ might be a nonsmooth function. The following example illustrates the construction of the interval extension and the nonsmoothness that this construction might introduce.

Example 1.14. Consider the following SIP problem:

$$\begin{aligned} &\min_{x\in[x^L,x^U]} x \\ &\text{s.t. } g(x,p) = x + \exp(p) + xp \leq\ 0,\ \forall p \in [p^L, p^u]. \end{aligned} \tag{1.22}$$

Clearly, $g_x(x) = x$, $g_p(p) = \exp(p)$ and $g_{xp}(x,p) = xp$. Using the rules of interval arithmetic we obtain $g_p^U(P) = \exp(p^U)$ and $g_{xp}^U(\mathbf{x}, P) = \max\{p^L x, p^U x\}$. Therefore, a finitely-constrained upper bounding program for this SIP would be:

$$\begin{aligned} &\min_{x\in[x^L,x^U]} x \\ &\text{s.t. } g^U(\mathbf{x}, P) = x + \exp(p^U) + \max\{p^L x, p^U x\} \leq\ 0. \end{aligned} \tag{1.23}$$

Despite the fact that (1.23) furnishes feasible points for the original SIP it also involves a nonsmooth constraint and standard NLP solvers are not expected to solve this problem robustly. Bhattacharjee et al. [26] propose two ways to alleviate this nonsmoothness. The first involves the introduction of auxiliary binary variables and the resulting problem is a MINLP:

$$\begin{aligned} &\min_{xin[x^L,x^U]} x \\ &\text{s.t. } x + \exp(p^U) + y(p^L x - p^U x) + (1-y)(p^U x - p^L x) \leq 0 \\ &\qquad\qquad y(p^L x - p^U x) + (y-1)(p^L x - p^U x) \leq 0, \end{aligned} \tag{1.24}$$

while the second involves explicit enumeration of all the constraints that arise from the min/max terms in the inclusion bounds and the resulting upper bounding problem

is a NLP:

$$\begin{aligned} &\min_{\mathbf{x}\in[x^L,x^U]} x \\ &\text{s.t. } x + \exp(p^U) + p^L x \leq 0 \\ &\qquad x + \exp(p^U) + p^U x \leq 0. \end{aligned} \tag{1.25}$$

The numerical application of both these methodologies for the SIP problems of the Watson test set can be found in [26].

Tighter inclusions can be generated by applying the subdivision principle that was discussed in Section 1.7.1. Therefore, if I_k denotes the set of indices of the subdivision at iteration k, the following relationship holds:

$$\max_{\mathbf{p}\in P} g(\mathbf{x},\mathbf{p}) \leq g_k^U(\mathbf{x},P) = \max_{\tau\in I_k} g^U(\mathbf{x},P_\tau) \leq g^U(\mathbf{x},P), \forall \mathbf{x}\in X. \tag{1.26}$$

Thus, a tighter upper-bounding problem than (1.20) is the following program:

$$\begin{aligned} &\min_{\mathbf{x}\in X} f(\mathbf{x}) \\ &\text{s.t. } g^U(\mathbf{x},P_\tau) \leq\ 0, \forall \tau \in I_k. \end{aligned} \tag{1.27}$$

The following Theorem [27] shows that when the subdivision of the parameter host set P is exhaustive, the global solution values of the interval-constrained reformulations (1.27) f_k^{ICR} converge to the global solution value of the SIP f^{SIP}.

Theorem 1.15. *Assume that f and g are continuous functions on X and $X \times P$, respectively. When the parameter host set P is subdivided uniformly at each iteration such that $\omega(P_\tau) = \frac{\omega(P)}{n_k}$, problem (1.27) yields an exact reformulation of the SIP in the limit $k \to \infty$, such that $\lim_{k\to\infty} f_k^{ICR} = f^{SIP}$, provided $n_{k+1} > n_k$ for all $k \geq 1$, and that there exists a minimizer $\mathbf{x}^*$ of the SIP for which a Slater point can be found arbitrarily close to $\mathbf{x}^*$.*

Proof. g is a continuous mapping from a compact metric space $X \times P$ into the metric

space $\mathbb{R}$. Thus g is uniformly continuous. Define: $\bar{g}^U(\mathbf{x}, P) = \max_{\mathbf{p}\in P} g(\mathbf{x}, \mathbf{p})$, $\mathbf{x} \in X$. It is well known that $\bar{g}^U(\cdot\, P)$ is a continuous function on X.

Since there exists a minimizer $\mathbf{x}^*$ of the SIP for which a Slater point can be found arbitrarily close to $\mathbf{x}^*$, this implies that there exists a feasible SIP point $\mathbf{x}^{**}$ for which $\bar{g}^U(\mathbf{x}^{**}, P) < 0$. Pointwise convergence of $g_k^U(\cdot, P)$ to $\bar{g}^U(\cdot, P)$ implies that $\exists\ k^*$ such that $g_k^U(\mathbf{x}^{**}, P) \leq 0, \forall k \geq k^*$, i.e., $\{\mathbf{x} \in X : g_k^U(\mathbf{x}, P) \leq 0\}$ is a non-empty, compact set for all $k \geq k^*$. It follows that the sequence of values obtained by solving interval-constrained reformulations is a bounded, monotonic (non-increasing) sequence such that $f_k^{ICR} \geq f_{k+1}^{ICR} \geq f^{SIP}$ for all $k \geq k^*$. This establishes the existence of the limit $\lim_{k\to\infty} f_k^{ICR}$. The sequence of minimizers $\{\mathbf{x}_k^{ICR}\}$, $k \geq k^*$ identified by solving the interval-constrained reformulations, is a sequence in the compact set X. Thus $\{\mathbf{x}_k^{ICR}\}$ has a convergent subsequence. The accumulation point of this subsequence belongs to the set $\{\mathbf{x} \in X : g^U(\mathbf{x}, P) \leq 0\}$, i.e., $\lim_{k'\to\infty} \mathbf{x}_{k'}^{ICR} = \bar{\mathbf{x}} \in \{\mathbf{x} \in X : g^U(\mathbf{x}, P) \leq 0\}$ for some subset of indices $\{k'\} \subset \{k\}$.

To prove the last statement, assume the contrary. Thus assume that $\bar{\mathbf{x}} \notin \{\mathbf{x} \in X : g^U(\mathbf{x}, P) \leq 0\}$. This implies that $g^U(\bar{\mathbf{x}}, P) > 0$. Since $\bar{x}$ is an accumulation point of the subsequence, and $g^U(\cdot, P)$ is continuous, $\exists\ k'_* \in \{k'\}$, such that $\forall\ k' > k'_*$, $g^u(\mathbf{x}_{n_{k'}}, P) > 0$. This is in clear contradiction with the fact that each member of the subsequence $\mathbf{x}_{n_{k'}}$ is a feasible point for the original SIP.

Assume that $\bar{\mathbf{x}}$ is not a minimizer of the SIP. Thus, assume that there exists a $\mathbf{x}^*$ such that $\mathbf{x}^* \in \{\mathbf{x} \in X : \bar{g}^U(\mathbf{x}, P) \leq 0\}$ and $f(\mathbf{x}^*) < f(\bar{\mathbf{x}})$. By hypothesis, there exists at least one minimizer to which Slater points are arbitrarily close. Without loss of generality, assume that $\mathbf{x}^*$ is such a minimizer. Thus, from continuity of f there exists a point $\mathbf{x}^{**}$ such that $\bar{g}^U(\mathbf{x}^{**}, P) < 0$ and $f(\mathbf{x}^*) < f(\mathbf{x}^{**}) < f(\bar{\mathbf{x}})$. From pointwise convergence of $g_k^U(\cdot, P)$ to $g^U(\cdot, P)$, $\exists\ k^*$ such that $g_{k^*}^U(\mathbf{x}^{**}, P) < 0$. Thus, the feasibility of $\mathbf{x}^{**}$ will be detected with at most k^* applications of the ICR procedure. Hence, for all $k > k^*$ the current optimal value can't be larger than $f(\mathbf{x}^{**})$ and thus, from continuity of f, $\bar{\mathbf{x}}$ cannot be an accumulation point. This is a clear contradiction and thus $\bar{\mathbf{x}}$ is a minimizer of the SIP.

Since $\bar{\mathbf{x}}$ is a minimizer for $f(\mathbf{x})$ on the SIP-feasible set, we have $f(\bar{\mathbf{x}}) = f^{SIP}$. By

the continuity of f, we also have $\lim_{k' \to \infty} f(\mathbf{x}_{k'}^{ICR}) \equiv \lim_{k' \to \infty} f_{k'}^{ICR} = f^{SIP}$. Since we have already shown that the sequence of minimum values is convergent, it follows that $\lim_{k \to \infty} f_k^{ICR} = f^{SIP}$. □

Remark 1. There are three main results from this theorem:

1. If a Slater point exists for the SIP, a SIP feasible point will be located in finite iterations by the ICR procedure.

2. If there exists a minimizer of the SIP for which a Slater point can be found arbitrarily close, then the ICR procedure guarantees convergence to the minimum of the SIP.

3. If none of the minimizers of the SIP have Slater points arbitrarily close, then the ICR procedure doesn't guarantee convergence. For special problems for which the interval extensions are exact then the ICR procedure is guaranteed to converge in one iteration.

Remark 2. There is a special class of problems for which the hypothesis of Theorem 2 will always be satisfied, and thus convergence to the global minimum will be achieved. If one of the minimizers of f, $\mathbf{x}_o$, is a Slater point then, by continuity of $\bar{g}^u(\cdot, P)$ on X, there exists a neighborhood around this minimizer, $B(\mathbf{x}_o, \delta)$, for which every $\mathbf{x} \in B(\mathbf{x}_o, \delta)$ is also a Slater point. This implies that $\mathbf{x}_o$ is a minimizer for which Slater points can be found arbitrarily close.

1.8 Robust Optimization

There is a very interesting category of semi-definite problems that is handled in the framework of robust optimization. Within the framework of robust optimization, uncertainty sets which are described by a system of linear inequality constraints or by a system of conic quadratic inequalities or by a system of linear matrix inequalities can be efficiently handled. The numerical algorithms that have been proposed in the literature have focused on optimization problems constrained by linear, quadratic, conic

quadratic, second-order cone or semi-definite constraints the robust counterparts of which are formulated as either linear, conic quadratic or semi-definite problems. We refer the reader to the work of Ben-Tal and Nemirovski [16, 17, 18, 19], Bertsimas & co-workers [21, 22, 23, 24], Vandenberghe & co-workers [103, 130, 131], Correa & Ramirez [34] and Jung & Lee [66].

Unfortunately, the application of kinetic model reduction that we are trying to address within this thesis, involves highly nonconvex constraints. It is clear that the robust counterpart of this problem is NP-hard. The paper by Kostina et al. [38] targets robust nonlinear optimization by linearizing the constraints around a nominal point followed by a convex approximation of the resulting problem. This method, however, would not generate guaranteed feasible points for the nonconvex KMR problem. It appears that while kinetic model reduction has a simple box uncertainty that can be targeted very efficiently by robust optimization (robust optimization can handle uncertainty sets that are much more complex), it also involves highly nonconvex constraints, and thus it is not yet clear how robust optimization can be used to tackle this problem and generate guaranteed feasible points. Specifically, and in contrast to [17], it does not seem possible to restrict this problem to a semi-definite one and thus guarantee the feasibility of the incumbent solutions.

Chapter 2

Global Optimization of SIP using Interval Methods

The goal of this chapter is to describe a global optimization algorithm for SIP based on interval methods and implemented in a branch-and-bound (B&B framework). We refer to [27, 75] for the original contribution and to [62] for background material on B&B algorithms for continuous global optimization. First of all, we provide definitions and necessary assumptions for the global optimization algorithm in Section 2.1. In Sections 2.2 and 2.3 we describe the upper and lower bounding methodologies respectively. Then, in Section 2.4 we propose the global optimization algorithm and prove its convergence to an ε-optimal solution value in Section 2.5. Finally, we provide a heuristic way to alleviate part of the computational cost in Section 2.6 and also provide some numerical results and comment on the efficiency of the algorithm in Section 2.7.

2.1 Assumptions and Definitions

To the definitions mentioned in Section 1.2 we add the following:

Definition 2.1 (Diameter of a Set)**.** Let $Z \subset \mathbb{R}^{n_z}$. The diameter of Z, denoted $w(Z)$,

is the maximal distance between two points in Z

$$w(Z) = \sup_{\mathbf{z}_1, \mathbf{z}_2 \in Z} ||\mathbf{z}_1 - \mathbf{z}_2||.$$

Definition 2.2 (Subdivision)**.** A *subdivision* of the set P is a finite collection of subsets $P_\tau \subset P$, with index set I such that

$$P = \bigcup_{\tau \in I} P_\tau \qquad \text{and} \qquad \text{int}\,(P_{\tau_1}) \cap \text{int}\,(P_{\tau_2}) = \emptyset, \quad \forall \tau_1, \tau_2 \in I : \tau_1 \neq \tau_2.$$

A subdivision of P with index set I_2 is a *refinement* of the subdivision with index set I_1, if for all $\tau_2 \in I_2$ there exists $\tau_1 \in I_1$ such that $P_{\tau_2} \subset P_{\tau_1}$ and for some $\tau_2 \in I_2$ there exists $\tau_1 \in I_1$ such that $P_{\tau_2} \subset P_{\tau_1}$ and $P_{\tau_2} \neq P_{\tau_1}$. A sequence of refined partitions with index sets $I_1, I_2, \ldots, I_k$ is called *exhaustive* if for $k \to \infty$ for all $\tau_k \in I_k$ the diameter of the corresponding set vanishes $w(P_{\tau_k}) \to 0$.

Assumption 2.3 (Host Sets)**.** The host sets $X \subset \mathbb{R}^{n_x}$, $P \subset \mathbb{R}^{n_p}$ are Cartesian products of (compact) intervals, i.e., for all variables and parameters explicit bounds are known ($X = [\mathbf{x}^L, \mathbf{x}^U]$ and $P = [\mathbf{p}^L, \mathbf{p}^U]$).

The set of vertices of P is denoted P_e. Based on Assumption 2.3,

$$P_e = \{\mathbf{p} \in P : p_j \in \{p_j^L, p_j^U\},\ \forall j = 1, \ldots, n_p\}$$

and the cardinality of P_e is given by $|P_e| = 2^{n_p}$.

Assumption 2.4 (Basic Properties of Functions)**.** The functions $f : X \to \mathbb{R}$ and $g : X \times P \to \mathbb{R}$ are twice continuously differentiable on some open set containing X and $X \times P$, respectively. Moreover, the constraint g is a factorable composite function [81] of functions with known convex underestimating and concave overestimating functions.

2.2 Upper-Bounding Problem

As discussed in Section 1.7.2, a finitely-constrained upper-bounding problem for a SIP may be constructed using an inclusion for the constraint function g on P. A convergent sequence of upper bounds may be generated using increasingly tighter inclusion functions derived using the subdivision notion. A partition of the interval P is used to formulate the upper-bounding problem solved at each node of the B&B tree. This partition, $P = \bigcup_{\tau \in T_q} P_\tau$, is determined solely by the depth, q, at which the node occurs in the B&B tree, and is independent of the iteration number, k, and the node in question. For an infinite sequence of nested nodes $\{M_{k_q}\}$, the partition elements P_τ are required to be monotonically decreasing in width such that degeneracy is approached in the limit $q \to \infty$, i.e.,

$$\begin{aligned} \max_{\tau \in T_q} w(P_\tau) &> \max_{\tau \in T_{q+1}} w(P_\tau) \\ \lim_{q \to \infty} \max_{\tau \in T_q} w(P_\tau) &= 0. \end{aligned} \tag{2.1}$$

The collection of sets $\{P_\tau\}_{\tau \in T_q}$ is used to define the feasible region for the following upper-bounding problem:

$$\begin{aligned} &\min_{\mathbf{x} \in M_{k_q}} f(\mathbf{x}) \\ &\text{s.t. } g^U(\mathbf{x}, P_\tau) \leq 0, \forall \tau \in T_q. \end{aligned} \tag{2.2}$$

The objective function value at any feasible point of (2.2) provides an upper bound on the minimum solution value of the SIP on M_{k_q}.

$$f_{k_q}^{ICR} \in \{f(\mathbf{x}) : \mathbf{x} \in M_{k_q},\ g^U(\mathbf{x}, P_\tau) \leq 0,\ \forall \tau \in T_q\}$$

If no feasible point can be found for (2.2), $f_{k_q}^{ICR} = +\infty$ is assigned. Once (2.2) has been determined for each node $M_i \in I_k$, the overall best available solution and upper

bound are updated by setting

$$\alpha_k = \min_{M_i \in I_k} f_i^{ICR}$$
$$\mathbf{x}^k \in \{\mathbf{x} : f(\mathbf{x}) = \alpha_k,\ \mathbf{x} \in M_i,\ g^U(\mathbf{x}, P_\tau) \leq 0,\ \forall \tau \in T_q\}.$$

2.3 Lower-Bounding Problem

As discussed in Section 1.4.1, discretization methods can be used to generate convergent outer approximations for a semi-infinite program. In the context of the SIP B&B algorithm, a discretized approximation may be used to generate a valid relaxation for the SIP on a given node. The grid, or index set, $S_q \subset P$, is determined only by the depth, q, at which the corresponding node occurs in the B&B tree. To preserve the convergence of the SIP B&B procedure, the grid sequence $\{S_q\}$ associated with an infinite sequence of nested nodes $\{M_{k_q}\}$ is required to satisfy the following properties:

$$\begin{aligned} S_q &\subset S_{q+1} \subset P \\ \lim_{q \to \infty} \text{dist}(S_q, P) &= 0, \end{aligned} \tag{2.3}$$

where the grid density $\text{dist}(S_q, P)$ is defined in (1.9). If $f_{k_q}^{SIP}$ is defined to be the solution value of the SIP on a (feasible) node $M_{k_q} \subset X$ such that

$$\begin{aligned} f_{k_q}^{SIP} = \min_{\mathbf{x} \in M_{k_q}}\ & f(\mathbf{x}) \\ \text{s.t. } & g(\mathbf{x}, \mathbf{p}) \leq 0, \quad \forall \mathbf{p} \in P, \end{aligned} \tag{2.4}$$

then solving the following finite relaxation yields a lower bound $f_{k_q}^D \leq f_{k_q}^{SIP}$:

$$\begin{aligned} f_{k_q}^D = \min_{\mathbf{x} \in M_{k_q}}\ & f(\mathbf{x}) \\ \text{s.t. } & g(\mathbf{x}, \mathbf{p}) \leq 0, \quad \forall \mathbf{p} \in S_q. \end{aligned} \tag{2.5}$$

The feasible sets defined by the constraints in (2.4) and (2.5) are referred to hereafter as $\{\mathbf{x} : g^s(\mathbf{x}) \leq 0\}$ and $\{\mathbf{x} : g_q^D(\mathbf{x}) \leq 0\}$ respectively. When both the inclusion-

constrained reformulation and the discretized approximation are convex in the optimization variables, $\mathbf{x}$, the upper and lower-bounding problems formulated at each node can be solved globally without excessive computational effort. In such cases, it is not necessary to branch on the set X; a single upper and lower-bounding problem is solved at each iteration using $I_k = \{X\}$ such that the depth, q, used to define S_q and T_q, is set by $q = k$. To solve the convex SIP to $\epsilon-$optimality, the upper and lower-bounding problems so defined are solved over a finite number of iterations k^* such that $f_k^{ICR} - f_k^D \leq \epsilon, \quad k \geq k^*$. When the SIP is convex in the optimization variables but the ICR introduces nonconvexities in (2.2), the discretized problem in (2.5) is solved (globally) for a lower-bounding solution, and the inclusion-constrained reformulation is solved (locally) for an upper-bounding solution defined by (2.2), and in general it will be necessary to branch on X in order to converge the upper bound.

In the general case, the functions $f(\mathbf{x})$ and/or $g(\mathbf{x}, \mathbf{p})$ are nonconvex in $\mathbf{x}$ and, therefore, (2.5) must be solved globally in order to provide a valid lower bound on $f_{k_q}^{SIP}$. Such an approach requires multiple nonconvex NLPs to be solved to global optimality at each iteration, and quickly becomes computationally prohibitive with increasing q. Instead, a convex relaxation of (2.5) may be solved for a lower bound on $f_{k_q}^{SIP}$. This approach entails significantly lower computational cost per node, but yields a potentially looser lower bound than (2.5). Consequently, a larger number of iterations may be required for the lower-bounding sequence $\{\beta_k\}$ to converge. In order to derive a valid convex relaxation, the McCormick factorization scheme [81] is applied to (2.5) to generate an equivalent reformulation in the following form:

$$\begin{aligned} f_{k_q}^D = \min_{\mathbf{y}} \; & y_N(\mathbf{x}) \\ \text{s.t. } & x_{n,k_q}^l \leq y_n(\mathbf{x}) \leq x_{n,k_q}^u, \; \forall n = 1, \ldots, n_x \\ & y_n(\mathbf{x}) \leq 0, \; \forall n = (N - |S_q|), \ldots, (N-1) \; , \end{aligned} \tag{2.6}$$

where N is total number of McCormick factors needed to reformulate (2.5) exactly. The factors $y_n(\mathbf{x}), \; n = 1, \ldots, n_x$ correspond to the elements of the decision vector $\mathbf{x}$. The remaining factors $y_n, \; n = (n_x + 1), \ldots, N$ are defined recursively as univariate

compositions, sums or bi-products of previously-defined factors. Wherever necessary, the bounds for the intermediate factors $y_n(\mathbf{x}),\ n = (n_x + 1), \ldots, (N - |S_q| - 1)$ are estimated using interval analysis methods. The (unconstrained) terminal factor y_N is defined to evaluate to the objective function value of (2.5). The constrained factors $y_n,\ n = (N - |S_q|), \ldots, (N - 1)$ are defined to evaluate to the constraint values $g(\mathbf{x}, \mathbf{p}_m),\ m = 1, \ldots, |S_q|$.

The convex underestimating program derived from (2.6) is:

$$\begin{aligned} f_{k_q}^{DC} = \min_{\mathbf{y}^c} & \ y_N^c(\mathbf{x}) \\ \text{s.t. } & x_{n,k_q}^l \leq y_n^c(\mathbf{x}) \leq x_{n,k_q}^u,\ \forall n = 1, \ldots, n_x \\ & y_n^c(\mathbf{x}) \leq 0,\ \forall n = (N - |S_q|), \ldots, (N - 1), \end{aligned} \tag{2.7}$$

where the factors $y_n^c(\mathbf{x}),\ n = n_x + 1, \ldots, N$ are defined recursively using convex and concave inequality and equality constraints. It can be easily shown that since all the convex and concave envelopes of the elementary functions of the problems under investigation are continuous, then the McCormick underestimators are also continuous. For notational convenience, the feasible set of (2.7) is denoted as $\{\mathbf{x} \in M_{k_q} : g_q^{DC}(\mathbf{x}) \leq 0\}$. A lower-bounding solution for each node $M_i \in I_k$ is assigned by solving (2.7) on each of the active sets in the current partition. Each node for which f_i^{DC} exceeds the best available upper bound α_k is fathomed. The overall lower bound is then updated by setting:

$$\beta_k = \min_{M_i \in I_k} f_i^{DC}.$$

The bound-improving property is established by selecting the node (or one of the nodes) at which a lower bound of β_k is attained, for further refinement. Two new nodes are subsequently generated by bisecting the selected node along the dimension (or one of the dimensions) which maximizes $x_{n,i}^u - x_{n,i}^l$. This refinement procedure results in an exhaustive subdivision scheme such that any infinite sequence of nested nodes generated by the B&B procedure approaches degeneracy in the limit $q \to \infty$, i.e., $\lim_{q \to \infty} w(M_{k_q}) = 0$ [63].

2.4 Global Optimization Algorithm

The B&B procedure for solving general nonconvex SIPs to $\epsilon-$optimality is outlined below:

1. Define the grid sequence $\{S_q\}$ and the partition sequence $\{T_q\}$.
2. Set $k := 0$, $I_k = \{X\}$.
3. Set $\alpha_0 := +\infty$.
4. Locate a feasible point for the inclusion-constrained reformulation on X if possible. Solve the convexified discretized approximation (2.7) on X. Set α_0 and β_0 to the respective solution values.
5. If $\alpha_0 - \beta_0 \leq \epsilon$ then stop and assign $f^{SIP} = \alpha_0$ and $\mathbf{x}^{SIP} = \mathbf{x}^k \in \{\mathbf{x} : f(\mathbf{x}) = \alpha_0, \mathbf{x} \in X,\ g^U(\mathbf{x}, P_\tau) \leq 0, \tau \in T_1\}$.
6. Delete (fathom) from I_k all nodes $M_i \in I_k$ for which $f_i^{DC} \geq \alpha_k$.
7. Select a $M_i \in I_k$ such that $\beta_k = f_i^{DC}$.
8. Generate two new nodes by bisecting M_i along the dimension (or one of the dimensions) which maximizes $x_{n,i}^u - x_{n,i}^l$. Delete M_i from I_k.
9. Set $k := k + 1$.
10. Add the two newly-created nodes to the set I_k. Copy all surviving nodes from I_{k-1} to I_k.
11. Solve (2.7) for f_i^{DC} on each of the newly-created nodes $i \in I_k$. Assign $f_i^{DC} = +\infty$ for each node at which (2.7) is infeasible. For each node attempt to identify a feasible point $\mathbf{x}_i$ in the ICR-feasible set. If a feasible point is found, assign the corresponding objective function value to f_i^{ICR}. Otherwise assign $f_i^{ICR} = +\infty$.
12. Set $\alpha_k = \min\limits_{M_i \in I_k} f_i^{ICR}$.
13. Set $\beta_k = \min\limits_{M_i \in I_k} f_i^{DC}$.

14. If $\alpha_k - \beta_k \leq \epsilon$ then stop and assign $f^{SIP} = \alpha_k$ and $\mathbf{x}^{SIP} \in \{\mathbf{x} : f(\mathbf{x}) = \alpha_k,\ \mathbf{x} \in M_i,\ g^U(\mathbf{x}, P_\tau) \leq 0, \tau \in T_q\}$. Else repeat steps 6 - 13.

2.5 Finite ε-convergence of the SIP B&B algorithm

In this section the finite $\varepsilon-$convergence of the SIP B&B algorithm is proved. Specifically, the generated sequences of lower and upper bounds are guaranteed to converge to the true SIP solution value in the limit $k \to \infty$ whenever for every minimizer $\mathbf{x}^*$ there exists a sequence of Slater points $\mathbf{x}_n$ for which $\lim_{n\to\infty} \mathbf{x}_n = \mathbf{x}^*$ and $q^1_{\mathbf{x}_n} < q^2_{\mathbf{x}_n}$, $\forall$ n. First of all, we provide a basic result on the McCormick underestimators. Then, we show that the lower bounding operation is strongly consistent. Furthermore, we prove that the B&B scheme cannot generate an infinite sequence of nested nodes, $\{M_{k_q}\}$, which converges to an infeasible point in X, i.e., fathoming nodes for which the convex underestimating program is infeasible is a deletion-by-infeasibility rule [62] which is certain in the limit. Finally, the convergence of the lower and upper bounding procedures is shown which leads to the conclusion of finite ε-convergence of the B&B algorithm.

Lemma 2.5. *Assume an infinite sequence of successively refined partitions converging to a point $\bar{\mathbf{x}}$, and a function f that is factorable in the sense of McCormick. Denote $C(M_{k_q})$ the value of the McCormick lower bound of f at node M_{k_q}. Then, $\lim_{q\to\infty} C(M_{k_q}) = f(\bar{\mathbf{x}})$.*

Proof. The rectangular partitioning of the X-space guarantees that every undeleted partition element can be further refined. Assume an infinite decreasing sequence $\{M_{k_q}\}$ of refined partition elements.

The construction of the McCormick relaxation for f ensures that

$$C(M_{k_q}) \leq f(\bar{\mathbf{x}}).$$

$C(M_{k_q})$ is a non-decreasing (from the McCormick construction) sequence of rigorous lower bounds that is bounded above by $f(\bar{\mathbf{x}})$ and therefore it converges.

From continuity of f, $\forall\ \varepsilon > 0$, $\exists\ \delta_1 > 0$: $||\mathbf{x} - \bar{\mathbf{x}}|| < \delta_1$ implies that $|f(\mathbf{x}) - f(\bar{\mathbf{x}})| < \varepsilon$. From the properties of the McCormick relaxation: $\forall\ \varepsilon > 0$, $\exists\ q' > 0$: $q \geq q'$ implies $\left|f_{k_q}(\bar{\mathbf{x}}) - f(\bar{\mathbf{x}})\right| < \varepsilon$, where f_{k_q} is the McCormick underestimator for the partition M_{k_q}. The McCormick underestimators employed are continuous by construction (see Section 2.3): $\forall\ \varepsilon > 0$, $\exists\ \delta_2 > 0$ for which: $||\mathbf{x} - \bar{\mathbf{x}}|| < \delta_2$ implies $\left|f_{k_{q'}}(\bar{\mathbf{x}}) - f_{k_{q'}}(\mathbf{x})\right| < \varepsilon$. Thus, $\forall\ \varepsilon > 0$, $\exists\ \delta^* < \min\{\delta_1, \delta_2\}$, for which:

1. $|f(\mathbf{x}) - f(\bar{\mathbf{x}})| < \varepsilon$, $\forall\ \mathbf{x} \in X_{\delta^*}$, $X_{\delta^*} = \{\mathbf{x} \in X : ||\mathbf{x} - \bar{\mathbf{x}}|| \leq \delta^*\}$.

2. $\left|f_{k_{q'}}(\mathbf{x}) - f_{k_{q'}}(\bar{\mathbf{x}})\right| < \varepsilon$, $\forall\ \mathbf{x} \in X_{\delta^*}$.

3. $\left|f_{k_q}(\bar{\mathbf{x}}) - f(\bar{\mathbf{x}})\right| < \varepsilon$, $\forall\ q \geq q'$.

For any two points $\mathbf{x}_1$, $\mathbf{x}_2 \in M_{k_{q'}} \cap X_{\delta^*}$:

$$\left|f_{k_{q'}}(\mathbf{x}_1) - f(\mathbf{x})\right| \leq \left|f_{k_{q'}}(\mathbf{x}_1) - f_{k_{q'}}(\bar{\mathbf{x}})\right| + \left|f_{k_{q'}}(\bar{\mathbf{x}}) - f(\bar{\mathbf{x}})\right| + \left|f(\bar{\mathbf{x}}) - f(\mathbf{x}_2)\right|. \quad (2.8)$$

From the analysis above each term on the right hand side of inequality (2.8) is less than ε and thus:

$$\left|f_{k_{q'}}(\mathbf{x}_1) - f(\mathbf{x}_2)\right| \leq 3\varepsilon. \quad (2.9)$$

For all $q > q'$ that satisfy $M_{k_q} \subset M_{k_{q'}} \cap X_{\delta^*}$, and for any two points $\mathbf{x}_1$, $\mathbf{x}_2 \in M_{k_q}$:

$$\left|f_{k_q}(\mathbf{x}_1) - f(\mathbf{x}_2)\right| \leq \left|f_{k_q}(\mathbf{x}_1) - f_{k_{q'}}(\mathbf{x}_1)\right| + \left|f_{k_{q'}}(\mathbf{x}_1) - f(\mathbf{x}_2)\right|. \quad (2.10)$$

Considering (2.9), (2.10) becomes:

$$\left|f_{k_q}(\mathbf{x}_1) - f(\mathbf{x}_2)\right| \leq \left|f_{k_q}(\mathbf{x}_1) - f_{k_{q'}}(\mathbf{x}_1)\right| + 3\varepsilon. \quad (2.11)$$

Analyzing the 1st term on the right hand side of (2.11) we obtain:

$$\left|f_{k_q}(\mathbf{x}_1) - f_{k_{q'}}(\mathbf{x}_1)\right| \leq \left|f_{k_q}(\mathbf{x}_1) - f(\mathbf{x}_1)\right| + \left|f(\mathbf{x}_1) - f_{k_{q'}}(\mathbf{x}_1)\right|. \quad (2.12)$$

From the properties of the McCormick underestimation we know that:

$$f_{k_{q'}}(\mathbf{x}_1) \leq f_{k_q}(\mathbf{x}_1) \leq f(\mathbf{x}_1), \forall q > q'. \tag{2.13}$$

From (2.9), for $\mathbf{x}_1 \equiv \mathbf{x}_2$ we obtain:

$$\left| f_{k_{q'}}(\mathbf{x}_1) - f(\mathbf{x}_1) \right| \leq 3\varepsilon. \tag{2.14}$$

From (2.13) and (2.14) we obtain:

$$\left| f_{k_q}(\mathbf{x}_1) - f(\mathbf{x}_1) \right| \leq 3\varepsilon. \tag{2.15}$$

From (2.14) and (2.15), (2.12) becomes:

$$\left| f_{k_q}(\mathbf{x}_1) - f_{k_{q'}}(\mathbf{x}_1) \right| \leq 6\varepsilon. \tag{2.16}$$

Taking into consideration (2.16), (2.11) becomes:

$$\left| f_{k_q}(\mathbf{x}_1) - f(\mathbf{x}_2) \right| \leq 9\varepsilon. \tag{2.17}$$

Since M_{k_q} is compact and f_{k_q} is continuous, $\min\limits_{\mathbf{x} \in M_{k_q}} f_{k_q}$ is attained. Thus $\exists\ \mathbf{x}_1$ such that: $f_{k_q}(\mathbf{x}_1) = \min\limits_{\mathbf{x} \in M_{k_q}} f_{k_q}$. Furthermore, taking $\mathbf{x}_2 \equiv \bar{\mathbf{x}}$, (2.17) becomes:

$$\forall \varepsilon > 0,\ \exists\, q^* > q' : \forall q > q^*, \left| \min_{\mathbf{x} \in M_{k_q}} f_{k_q} - f(\bar{\mathbf{x}}) \right| \leq 9\varepsilon. \tag{2.18}$$

Since $C(M_{k_q}) = \min\limits_{\mathbf{x} \in M_{k_q}} f_{k_q}$, relationship (2.18) becomes:

$$\forall \varepsilon > 0,\ \exists\, q^* > q' : \forall q > q^*, \left| C(M_{k_q}) - f(\bar{\mathbf{x}}) \right| \leq 9\varepsilon. \tag{2.19}$$

(2.19) implies that:

$$\lim_{q \to \infty} C(M_{k_q}) = f(\bar{\mathbf{x}}). \tag{2.20}$$

□

Lemma 2.6. *The fathoming of nodes which are infeasible for the convex lower-bounding problem (2.7) is a deletion-by-infeasibility rule which is certain in the limit.*

Proof. As noted in the previous section, the exhaustiveness of the partitioning procedure guarantees that any infinite sequence of nested nodes converges to a point $\bar{\mathbf{x}}$. Assume that $\bar{\mathbf{x}}$ is infeasible, thus there exists $\bar{\mathbf{p}}$ for which $g(\bar{\mathbf{x}}, \bar{\mathbf{p}}) > 0$. By continuity of $g(\bar{\mathbf{x}}, \cdot)$ it follows that there exists an open ball around $\bar{\mathbf{p}}$ of some radius δ, namely P_δ, for which $\forall\ \mathbf{p} \in P_\delta$, $g(\bar{\mathbf{x}}, \mathbf{p}) > 0$.

The exhaustiveness of the gridding of the P-space ensures that there exists a level of the branch-and-bound tree, for which a point $\mathbf{p}' \in P_\delta$ will be found. From (2.3) one constraint for the function g at $\mathbf{p}'$ will be present at every subsequent level. Let the function $g'(\mathbf{x})$ correspond to the constraint at $\mathbf{p}'$. From Lemma 2.5, $\lim\limits_{q_n \to \infty} \min\limits_{\mathbf{x} \in M_{k_q}} g'^{DC}_{k_q}(\mathbf{x}) = g'(\bar{\mathbf{x}})$, where $g'^{DC}_{k_q}$ is the McCormick relaxation for the constraint function g' at node k_q.

The statement above along with the fact that $g'(\bar{\mathbf{x}}) > 0$ imply that there exists some finite q^* for which $q > q^*$ implies that $\min\limits_{\mathbf{x} \in M_{k_q}} g'^{DC}_{k_q}(\mathbf{x}) > 0$. This finally implies that the lower bounding problem is clearly infeasible for $q > q^*$ and a node containing $\bar{\mathbf{x}}$ will be fathomed no later than at node $q^* + 1$.

Thus, the branch-and-bound tree can't generate an infinite decreasing sequence of nested nodes converging to an infeasible point. □

Lemma 2.7. *The lower bounding operation is strongly consistent.*

Proof. Assume an infinite sequence $\{M_{k_q}\}$ of successively refined partitions. From Lemma 2.6 we already know that this sequence will converge to a feasible point $\bar{\mathbf{x}}$. This implies that the lower bounding problem will always be feasible for this sequence of nodes and specifically:

$$C(M_{k_q}) \leq \beta(M_{k_q}) \leq f(\bar{\mathbf{x}}). \tag{2.21}$$

where $\beta(M_{k_q})$ is the minimum of the convex (McCormick) lower-bounding problem

on the partition M_{k_q} for the constrained problem. From (2.20) and (2.21) and the sandwich theorem we finally obtain:

$$\lim_{q\to\infty} \beta(M_{k_q}) = f(\bar{\mathbf{x}}).$$

□

Lemma 2.8. *The lower bounding operation converges to the global solution of the SIP, i.e.,* $\lim_{k\to\infty} \beta_k = f^{SIP}$.

Proof. From the analysis that has preceded we already know that:

1. The rectangular partitioning guarantees that the partitioning is exhaustive.
2. The selection of the partition sets to be refined is bound improving. This was shown in the description of the algorithm.
3. The lower bounding operation is strongly consistent (Lemma 2.7).
4. The deletion by infeasibility rule is certain in the limit (Lemma 2.6).

From [62] we obtain that: $\lim_{k\to\infty} \beta_k = f^{SIP}$ □

Let X_s be the set of the Slater points within the feasible set of the SIP. From the assumption that for every global minimizer $\mathbf{x}^*$ there exists a sequence of Slater points converging to $\mathbf{x}^*$, the set X_s is non-empty.

For every $\mathbf{x} \in X_s$ assign two numbers. Call $q^1_{\mathbf{x}}$ the earliest level of the branch-and-bound tree for which $\mathbf{x}$ is found feasible to the upper bound operation. From (2.1) and (2.3), the fact that the subdivision of the parameter space is uniform and finally from Theorem 1, the existence of such a level is guaranteed for every $\mathbf{x}$.

Call $q^2_{\mathbf{x}}$ the earliest level of the branch-and-bound tree for which $\mathbf{x}$ and the global minimizer $\mathbf{x}^*$ do not belong to the same node. The root node ensures that these two points are together initially, and the exhaustiveness of the partitioning of the X-space ensures that these two points will finally belong to different nodes (if $\mathbf{x}^* \neq \mathbf{x}$).

Lemma 2.9. *Assume that for every minimizer* $\mathbf{x}^*$*, there exists a sequence of Slater points* $\mathbf{x}_n$ *satisfying:*

1. $\lim_{n\to\infty} \mathbf{x}_n = \mathbf{x}^*$
2. $q^1_{\mathbf{x}_n} < q^2_{\mathbf{x}_n}, \forall\ n.$

Then, the upper bounding operation converges to the global solution of the SIP, i.e., $\lim_{k\to\infty} \alpha_k = f^{SIP}$.

Proof. Consider, initially, only the lower bounding operation of the branch-and-bound tree. From Lemma 2.8, $\lim_{k\to\infty} \beta_k = f^{SIP}$. The fact that $k \to \infty$ implies that there exists an infinite sequence of nested nodes $\{M_{k_q}\}$. This infinite sequence of nested nodes converges to a point, $\bar{\mathbf{x}}$. From Lemma 2.7, $\lim_{q\to\infty} \min_{\mathbf{x}\in M_{k_q}} \beta(M_{k_q}) = f(\bar{\mathbf{x}})$. This implies that $\bar{\mathbf{x}}$ must be a global minimizer, else $f(\bar{\mathbf{x}}) > f^{SIP}$ and there would exist q' for which $\beta(M_{k_q}) > f^{SIP}$. This suggests that the best bound heuristic would never have selected that node again for branching (that node could never provide the best lower bound). Therefore, we have shown that if the lower bounding procedure creates a infinite sequence of nested nodes, this sequence converges to a global minimizer.

Assume $\varepsilon > 0$. Since f is continuous, $\exists\ \delta > 0$, such that, $||\mathbf{x} - \bar{\mathbf{x}}|| < \delta$, namely X_δ, implies $|f(\mathbf{x}) - f(\bar{\mathbf{x}})| < \varepsilon$.

Consider the sequence of nested nodes $\{M_{k_q}\}$ containing $\bar{\mathbf{x}}$. There exists q' for which $q > q'$ implies $M_{k_q} \subset X_\delta$. Now consider $q^* > q'$ for which there exists a member of the sequence $\mathbf{x}_n$, namely $\mathbf{x}_{n^*}$ for which $\mathbf{x}_{n^*} \in M_{k_{q^*}}$ but $\mathbf{x}_{n^*} \notin M_{k_{q^*+1}}$. Every member of the sequence $\mathbf{x}_n$ will finally not belong to the same node as $\bar{\mathbf{x}}$, thus the existence of q^* is justified.

Thus, at some finite iteration either a solution to the SIP will be found or $M_{k_{q^*}}$ will be branched on. It is evident that $\tau^2_{\mathbf{x}_{n^*}} = q^* + 1$ and, from the 2nd assumption, that $\tau^1_{\mathbf{x}_{n^*}} \leq q^*$.

Thus, when the node $M_{k_{q^*}}$ is examined at some finite iteration k_{q^*}, $\mathbf{x}_{n^*}$ is feasible to the upper bounding problem. Therefore, an incumbent at that node will be found, and any such incumbent, say $\alpha(M_{k_{q^*}})$ would satisfy $\left|\alpha(M_{k_{q^*}}) - f(\bar{\mathbf{x}})\right| < \varepsilon$. Since

$\alpha(M_{k_{q^*}}) > f(\bar{\mathbf{x}})$, this implies that $\alpha(M_{k_{q^*}}) < f(\bar{\mathbf{x}}) + \varepsilon$. Thus for all $k > k_{q^*}$, $\alpha_k \leq \alpha(M_{k_{q^*}}) < f(\bar{\mathbf{x}}) + \varepsilon$.

Finally, $\forall\ \varepsilon > 0$, $\exists\ k_{q^*}$ for which $k > k_{q^*}$ implies $f(\bar{\mathbf{x}}) \leq \alpha_k \leq f(\bar{\mathbf{x}}) + \varepsilon$. This implies that $\lim_{k\to\infty} \alpha_k = f(\bar{\mathbf{x}}) = f^{SIP}$. □

Remark 3. In this remark, the assumption on the convergence of the upper bounding operation will be commented on.

1. In general, the hypothesis of the Lemma is hard to verify a priori. However, the difficulty eliminated by this hypothesis was not encountered for any of the test problems (see Appendix A) as evidenced by finite convergence for all these problems. It is conceivable that the basic algorithm can be modified to incorporate an adaptive subdivision strategy that would enable this assumption to be relaxed to one in which Slater points exist arbitrarily close to all minimizers. Such adaptive subdivision strategies will be the topic of future research.

2. It can be easily shown that the breadth first search converges with much milder assumptions than the best bound heuristic that was actually employed. Specifically, the breadth first search requires only the existence of a minimizer $\mathbf{x}^*$ for which a Slater point can be found arbitrarily close to $\mathbf{x}^*$.

Theorem 2.10. *The SIP B&B procedure is finitely-convergent to ϵ-optimality, such that $\alpha_k - \beta_k \leq \varepsilon$ at termination and $\mathbf{x}^k$ approximately solves the SIP.*

Proof. Follows immediately from Lemmas (2.8) and (2.9). □

2.6 The Exclusion Heuristic

One major drawback of the SIP B&B implementation described thus far is that the number of constraints used to formulate the upper-bounding problem grows exponentially with the depth of a node, q. Clearly, the maximum width $\max_{\tau\in T_q} w(P_\tau)$, must be decreased at each successive level of the B&B tree in order to preserve the finite

$\epsilon-$convergence of the procedure. However, repeated (uniform) subdivision of the interval P may generate certain upper-bounding constraints in (2.2) which are inactive at all feasible points of a given node. Furthermore, if the constraint associated with a particular subinterval P_τ is inactive at all feasible points of a given node M_{k_q}, any constraints arising from further division of this subinterval will clearly be inactive at all of the descendant nodes, i.e.,

$$\exists \tau_1 \in T_q :\ g^U(\mathbf{x}, P_{\tau_1}) < 0, \forall \mathbf{x} \in M_{k_q} \Rightarrow \exists \tau_2 \in T_{q+1} :\ g^U(\mathbf{x}, P_{\tau_2}) < 0, \forall \mathbf{x} \in M_{k_{q+1}}.$$

Such constraints can safely be eliminated without altering the feasible set of the upper-bounding problem. We refer to this elimination procedure as the *exclusion heuristic*. This modification to the original SIP B&B algorithm may generate considerable computational savings by enabling smaller NLPs to be solved at each node. Interval analysis may be used to identify the redundant constraints by evaluating an inclusion for $g(\mathbf{x}, \mathbf{p})$ over each of the domains $M_{k_q} \times P_\tau, \tau \in T_q$. An inclusion bound $\bar{G}^U(M_{k_q}, P_\tau) < 0$ indicates that the upper-bounding constraint corresponding to P_τ is inactive for all $\mathbf{x} \in M_{k_q}$. Consequently, this constraint may be dropped from the upper-bounding problem solved at the current node, and all constraints arising from the subdivision of P_τ may be dropped from the upper-bounding problems solved at its descendant nodes. Similar to the formulation of the upper-bounding problem, the exclusion heuristic benefits from the use of tighter inclusions for $g(\mathbf{x}, \mathbf{p})$. As before, some trade-off must be made between the quality of the inclusion bound, and the computational cost associated with calculating it.

Analogous to the inclusion-constrained reformulation, the lower-bounding problem also suffers from an explosion in the number of constraints if the discretization mesh is subjected to uniform refinement at each level of the B&B tree. Once again, the exclusion heuristic may be helpful in identifying redundant constraints, thereby reducing the size of the convex relaxation which must be solved at each node. It

follows from the properties of inclusion functions that

$$\begin{aligned} \bar{G}^U(M_{k_q}, P_\tau) < 0 \quad &\Rightarrow \quad g(\mathbf{x}, \mathbf{p}) < 0, \quad \forall (\mathbf{x}, \mathbf{p}) \in M_{k_q} \times P_\tau \\ &\Rightarrow \quad g^c(\mathbf{x}, \mathbf{p}) < 0, \quad \forall (\mathbf{x}, \mathbf{p}) \in M_{k_q} \times P_\tau \end{aligned} \tag{2.22}$$

i.e., the endpoints of the 'inactive' subintervals will yield inactive constraints in the corresponding lower-bounding problem. Moreover, these points will remain 'inactive' at all descendant nodes.

2.7 Numerical Implementation and Results

The SIP B&B algorithm was implemented using an in-house B&B code. The upper and lower bounding problems were solved using the SQP solver SNOPT 6.1-1 [46]. Natural interval extensions were used to calculate the inclusion bounds $g^U_{n_k}(\mathbf{x}, P)$. The set of partition elements used to define the upper-bounding problem at a node occurring at level q was defined to be $\{P_\tau\},\ \tau \in T_q$, where $T_q = \{1, 2, \ldots, 2^q\}^{n_p}$. The index set S_q associated with each node occurring at level q was defined to be the set of upper right endpoints of the subintervals P_τ, i.e., $S_q = \{P^u_\tau\}, \tau \in T_q$. These sets can be shown to satisfy (2.1) and (2.3) respectively.

The SIP B&B procedure was applied to examples 1,2,3 and H (see Appendix A) from the Watson test set. The relative and absolute tolerances for retaining a node within the B&B tree were set to 0.01, apart from Problem 3 where the relative tolerance was relaxed to 0.02. The optimality tolerances of the NLP solver were set to 10^{-8}. When solving certain convex lower bounding problems SNOPT provided the message that *the current point cannot be improved*. Examining those nodes it was found that in each case the incumbent was infeasible and thus infeasibility was assumed for these nodes. All numerical experiments were carried out in a 3,4 GHz PIV running Linux.

Tables 1-4 summarize the numerical results from the implementation of the exclusion heuristic and/or the upper bounding methodology using different methodologies. Table 1 refers to the implementation of the basic B&B procedure, while Table 2 refers

to the implementation of the exclusion heuristic using natural interval extensions. Table 3 refers to the implementation of the exclusion heuristic using optimally-centered Taylor forms while Table 4 refers to the implementation of both the upper-bounding methodology and the exclusion heuristic using optimally-centered Taylor forms.

For Tables 1-4, column 1 refers to the index of the problem, column 2 refers to the total number of nodes that are required for convergence and column 3 refers to the maximum depth within the B&B tree that the procedure visited in order to converge. For Table 1, columns 4,5,6,7 and 8 refer to the total CPU time, the incumbent solution value of the proposed algorithm, the incumbent solution value of the reduction algorithm applied in [33], the incumbent upper bound on the root node and the incumbent solution of the proposed algorithm, respectively. For Tables 2-4, columns 4,5,6 and 7 refer to the depth that a constraint was first dropped using the exclusion heuristic, the total number of constraints dropped, the total number of constraints visited by the lower-bounding procedure in all the nodes and the CPU time, respectively.

For examples 1 and 4, the interval-constrained reformulation produced the optimal solution value of the SIP on the root node. For all examples, even for 2 and 3 where the interval-constrained reformulation was inexact at the root node, the upper bounding methodology had essentially terminated quite early and the bulk of the CPU time was spent on converging the lower bounds by making the discretization grids finer and solving the corresponding lower-bounding problems.

For all of the examples, the exclusion heuristic did not alter the number of nodes that were visited by the algorithm. This was expected because the heuristic was only employed to remove redundant constraints. However, both implementations of the heuristic, i.e. the natural interval extensions and the optimally-centered Taylor models did accelerate the convergence of the algorithm by making each subproblem cheaper to solve. Especially for example 3, the most computationally demanding problem, the reduction of CPU time was almost 45%.

The implementation of the upper bounding methodology using optimally-centered Taylor forms accelerated the convergence with respect to the basic implementation

Table 2.1: Convergence Results for Basic SIP B&B

Problem	Total Nodes	Max. Depth	CPU	f	f^{RED}	f^{ICR}	Solution
1	27	13	16.05	-0.250	-0.250	-0.250	(0.000,0.500)
2	19	6	0.08	0.195	0.195	0.381	(-0.751,0.618)
3	75	15	415	5.334	5.334	27.41	(-0.214,-1.362,1.854)
H	43	13	24.36	0	0	0	(0.0,0.0)

Table 2.2: Results for Exclusion Heuristic using Natural Interval Extensions

Problem	Total Nodes	Max. Depth	Depth of First Drop	Total Dropped	Total Constraints	CPU
1	27	13	9	58610	65530	7.2
2	19	6	6	126	421	0.07
3	75	15	4	277134	991210	306
H	43	13	1	20622	112637	20.80

but did not offer any computational gain over the other implementations; on the contrary it decelerated the convergence and that can be accounted to an increasing cost of evaluating the upper bounding functions.

In conclusion, the overall solution time represents some trade-off between the cost of evaluating more complicated upper-bounding constraints and the benefits derived from using higher-order inclusion functions to formulate the upper-bounding problem and/or the exclusion heuristic.

Table 2.3: Results for Exclusion Heuristic Implemented using Optimally-Centered Taylor Models

Problem	Total Nodes	Max. Depth	Depth of First Drop	Total Dropped	Total Constraints	CPU
1	27	13	4	59082	65530	7.04
2	19	6	6	128	421	0.07
3	75	15	3	597420	991210	230
H	43	13	11	20620	112637	20.84

Table 2.4: Results for Exclusion Heuristic and Upper-Bounding Problem Formulated using Optimally-Centered Taylor Models

Problem	Total Nodes	Max. Depth	Depth of First Drop	Total Dropped	Total Constraints	CPU
1	27	13	4	59082	65530	19.45
2	19	6	6	128	421	0.08
3	75	15	3	597420	991210	307
H	43	13	11	20620	112637	28.56

Chapter 3

Relaxation-Based Bounds for SIP

The goal of this chapter is to introduce and analyze a new methodology to provide guaranteed feasible points for SIP. We refer the reader to [84] for the original contribution. First of all, we introduce the notion of relaxing and restricting the lower-level problem for SIP in Section 3.1. In Section 3.2, we provide some definitions and assumptions. Then, we provide a brief overview of the construction of convex relaxations for finite programs in Section 3.3. Furthermore, we describe various alternatives ways of generating guaranteed feasible points for SIP in Sections 3.4, 3.5 and 3.6 and we provide a brief convergence proof in Section 3.7. Finally, we provide numerical results from the Watson test set [135] in Section 3.8.

3.1 Bounding the Lower-Level Problem

In Section 1.6 we mentioned that any feasible point of the following program:

$$\begin{aligned} &\min_{\mathbf{x}\in X} f(\mathbf{x}) \\ &\text{s.t. } h(\mathbf{x}) \leq 0, \end{aligned} \tag{3.1}$$

where $h(\mathbf{x}) \geq O(\mathbf{x}, P) \equiv \max_{\mathbf{p}\in P} g(\mathbf{x}, \mathbf{p})$, is a feasible point of the original SIP (1.1). For each $\mathbf{x} \in X$, $h(\mathbf{x})$ is an upper bound to the global solution value of the lower-level problem at $\mathbf{x}$. Since the lower-level problem involves maximization, h can be

considered as a relaxation of $g(\cdot, P)$ on X. Therefore, any upper bounding approach for SIP, whether the ICR or a method employing convex relaxations, constructs a relaxation of the lower-level problem for each $\mathbf{x} \in X$ which results in a restriction of the overall problem.

Similarly, lower-bounding methodologies for SIP, such as discretization approaches, construct a restriction of the lower-level program for each $\mathbf{x} \in X$. To illustrate this, recall that in discretization methods the following generic subproblems are generated:

$$\begin{aligned} &\min_{\mathbf{x}\in X} f(\mathbf{x}) \\ &\text{s.t. } g(\mathbf{x}, \mathbf{p}) \leq 0, \quad \forall \mathbf{p} \in P_i, \end{aligned} \tag{3.2}$$

where P_i is a finite subset of P indexed by the iteration number i. Denote $k_i(\mathbf{x}) = \max_{\mathbf{p}\in P_i} g(\mathbf{x}, \mathbf{p})$. Then, problem (3.2) is equivalent to

$$\begin{aligned} &\min_{\mathbf{x}\in X} f(\mathbf{x}) \\ &\text{s.t. } k_i(\mathbf{x}) \leq 0\,. \end{aligned} \tag{3.3}$$

Clearly, $k_i(\mathbf{x}) \leq O(\mathbf{x}, P)$, for all $\mathbf{x} \in X$ and for all $i \in \mathbb{N}$. Therefore, the lower-level problem is restricted for all $\mathbf{x} \in X$ (recall that the lower-level problem involves maximization) and this leads to a relaxation of the overall problem (3.3).

3.2 Definitions and Assumptions

To the assumptions and definitions that were stated in Sections 1.2 and 2.1 we add the following definitions:

Definition 3.1 (Relaxation of Functions)**.** Given a convex set $C \subset \mathbb{R}^{n_z}$ and a function $h : C \to \mathbb{R}$, a convex function $h^u : C \to \mathbb{R}$ is a convex relaxation of h on C if

$$h^u(\mathbf{z}) \leq h(\mathbf{z}), \quad \forall \mathbf{z} \in C$$

and a concave function $h^o : C \to \mathbb{R}$ is a concave relaxation of h on C if

$$h^o(\mathbf{z}) \geq h(\mathbf{z}), \quad \forall \mathbf{z} \in C.$$

The convex envelope $\bar{h}^u : C \to \mathbb{R}$ of h on C is a convex relaxation of h on C such that for any convex relaxation h^u of h on C

$$h^u(\mathbf{z}) \leq \bar{h}^u(\mathbf{z}), \quad \forall \mathbf{z} \in C.$$

Similarly, the concave envelope $\bar{h}^o : C \to \mathbb{R}$ of h on C is a concave relaxation of h on C such that for any concave relaxation h^o of h on C

$$\bar{h}^o(\mathbf{z}) \leq h^o(\mathbf{z}), \quad \forall \mathbf{z} \in C.$$

Definition 3.2 (Relaxation of Programs)**.** Let $Z^D, Z^E \subset \mathbb{R}^{n_z}$ and consider the optimization problems

$$\inf_{\mathbf{z} \in Z^D} f^D(\mathbf{z}) \quad \text{and} \quad \inf_{\mathbf{z} \in Z^E} f^E(\mathbf{z})$$

. If $Z^D \subset Z^E$ and $f^E(\mathbf{z}) \leq f^D(\mathbf{z})$, $\forall \mathbf{x} \in Z^D$, the optimization problem $\inf_{\mathbf{z} \in Z^E} f^E(\mathbf{z})$ is said to be a *relaxation* of $\inf_{\mathbf{z} \in Z^D} f^D(\mathbf{z})$. Similarly the optimization problem $\inf_{\mathbf{z} \in Z^D} f^D(\mathbf{z})$ is said to be a *restriction* of $\inf_{\mathbf{z} \in Z^E} f^E(\mathbf{z})$.

A direct consequence of relaxations/restrictions is that $\inf_{\mathbf{z} \in Z^E} f^E(\mathbf{z}) \leq \inf_{\mathbf{z} \in Z^D} f^D(\mathbf{x})$. For maximization problems the above inequalities have to be reversed.

Definition 3.3 (Convex Program)**.** The minimization problem $\inf_{\mathbf{z} \in Z} f(\mathbf{z})$ is called convex, if $Z \subset \mathbb{R}^{n_z}$ is convex and f is convex on Z. Similarly the maximization problem $\sup_{\mathbf{z} \in Z} f(\mathbf{z})$ is called convex, if $Z \subset \mathbb{R}^{n_z}$ is convex and f is concave on Z.

Definition 3.4 (Convex Relaxation of Programs)**.** Let $Z^D, Z^E \subset \mathbb{R}^{n_z}$. The optimization problem $\inf_{\mathbf{z} \in Z^D} f^D(\mathbf{z})$ is a *convex relaxation* of $\inf_{\mathbf{z} \in Z^E} f^E(\mathbf{z})$ if it is a convex program and a relaxation of $\inf_{\mathbf{z} \in Z^E} f^E(\mathbf{z})$.

3.3 Convex Relaxation

Most global optimization algorithms, such as spatial branch-and-bound and outer approximation, rely on the construction of convex relaxations. Given a box-constrained nonlinear program (NLP)

$$\begin{aligned} &\max_{\mathbf{z}} h(\mathbf{z}) \\ &\text{s.t. } \mathbf{z} \in [\mathbf{z}^L, \mathbf{z}^U] \equiv Z \subset \mathbb{R}^{n_z} \end{aligned} \tag{3.4}$$

with a nonconvex objective function $h(\mathbf{z})$, the goal is to construct a convex maximization problem, i.e., a program with convex constraints and a concave objective function, whose optimal objective value overestimates the optimal solution value of (3.4). Convex and concave envelopes or tight relaxations are known for a variety of simple nonlinear terms [4, 110, 127] and this allows the construction of convex and concave relaxations for a quite general class of functions through several methods [5, 44, 81, 110]. All the methods proposed in the literature essentially rely on a few key ideas and components. McCormick's results [81] allow the construction of convex and concave relaxations of functions defined by recursive compositions of elementary operations and intrinsic functions. Floudas and coworkers [4, 5, 6] have proposed convex relaxations for arbitrary, twice continuously differentiable functions by the addition of a simple, sufficiently negative function that is known to be convex; concave relaxations are handled similarly. Both approaches can also use auxiliary variables and constraints. Smith and Pantelides [110] formalized the use of auxiliary variables, while Gatzke et al.[44] demonstrated how these methods can be combined and automated. Tawarmalani and Sahinidis [126, 127] proposed to further relax the convex relaxations via linearization to take advantage of the scalability of linear programming (LP) solvers. While many combinations of the above ideas are conceivable, we consider three extreme cases of convex relaxation that are of particular interest for the relaxation of the lower-level program. We also discuss how to construct linearizations of these convex relaxations.

3.3.1 Nonsmooth Concave Overestimation

The first alternative we consider is to construct a concave relaxation of the objective function in (3.4) by successively applying McCormick's composition theorem, without the addition of variables and constraints. McCormick [81] presents convex and concave relaxations of a function

$$h^t(\mathbf{z}) = T(t(\mathbf{z})) + U(u(\mathbf{z}))V(v(\mathbf{z})),$$

where $T, U, V : \mathbb{R} \to \mathbb{R}$ are continuous and $t, u, v : Z \to \mathbb{R}$ are continuous on Z. Assuming that convex and concave relaxations are known for all functions (t, u, v and T, U, V), and bounds are known for the ranges of the inner functions (t, u, v), McCormick's composition result provides convex and concave relaxations for h^t on Z. Recursive application of this result allows the derivation of convex and concave relaxations for complicated expressions termed *factorable* expressions. Assuming that the objective function h in (3.4) is factorable, we denote $h^{o,mc} : Z \to \mathbb{R}$ the concave relaxation constructed by the recursive application of the composition theorem. Since $h^{o,mc}(\mathbf{z}) \geq h(\mathbf{z})$ for all $\mathbf{z} \in Z$, the optimal objective value of

$$\max_{\mathbf{z} \in Z} h^{o,mc}(\mathbf{z}) \tag{3.5}$$

overestimates the optimal objective value of (3.4). While convex, (3.5) is not necessarily smooth, and therefore standard optimization techniques relying on the satisfaction of KKT conditions are not applicable in general. Since it is box-constrained, the linearization (using subgradients) at an arbitrary point $\bar{\mathbf{z}} \in Z$ results in a linear program which is a further relaxation.

Example 3.5. Consider the program

$$\max_{\mathbf{z} \in [-1,1]^2} z_1^2 + e^{z_1 z_2} \tag{3.6}$$

with an optimal objective value of $1 + e \approx 3.73$ and two optimal solutions $z_1 =$

$z_2 = -1$ and $z_1 = z_2 = 1$. A convex relaxation based on McCormick's composition theorem is constructed as follows. First the objective function is split to the terms z_1^2 and $e^{z_1 z_2}$. The first term is univariate convex and therefore its concave envelope is given by the secant which for $z_1 \in [-1, 1]$ evaluates to 1. The second term is a composition of the bilinear function $t : \mathbb{R}^2 \to \mathbb{R}$, $t(\mathbf{z}) = z_1 z_2$ and the exponential function $T : C \subset \mathbb{R} \to \mathbb{R}$, $T(w) = e^w$. The convex envelope of the bilinear function is given by $t^u(\mathbf{z}) = \max\{-1 - z_1 - z_2, -1 + z_1 + z_2\}$ and the concave envelope of the bilinear function is given by $t^o(\mathbf{z}) = \min\{1 - z_1 + z_2, 1 + z_1 - z_2\}$. The range of the bilinear function on $[-1, 1]^2$ is given by $C = [-1, 1]$. The exponential function is monotone increasing and therefore its maximum on $[-1, 1]$ is attained at $w = 1$. Since it is also univariate convex, its concave overestimator is given by the secant $e^{-1} + (e^1 - e^{-1})\frac{w+1}{2}$. McCormick's composition theorem contains the mid function which in this special case always evaluates to $t^o(\mathbf{z})$. McCormick's composition theorem therefore gives a concave relaxation as

$$\max_{\mathbf{z} \in [-1,1]^2} 1 + e^{-1} + (e^1 - e^{-1}) \frac{\min\{1 - z_1 + z_2, 1 + z_1 - z_2\} + 1}{2},$$

with the same optimal objective value as (3.6) and infinite optimal solutions $z_1 = z_2$. Note that in this case $h^{o,mc}$ is the concave envelope of h on Z.

3.3.2 Smooth Concave Overestimation

The second alternative we consider is based on the ideas of αBB relaxation by Adjiman et al. [2, 4, 5] and γBB relaxation by Arkotirianakis and Floudas [6]. To avoid the introduction of auxiliary variables and constraints, we deviate from the algorithmic framework presented in these references. Instead of splitting the nonlinear objective h into the sum of concave terms, special nonconcave terms and general nonconcave terms we apply the relaxation on the original function. Note also that we consider the simplest variant of uniform diagonal shift of the Hessian matrix.

Since univariate quadratic terms are convex

$$h^{o,\alpha}(\mathbf{z}) = h(\mathbf{z}) + \alpha \sum_{i=1}^{n_z} (z_i - z_i^L)(z_i^U - z_i)$$

is concave for sufficiently large values of α. Moreover for $\mathbf{z} \in Z$,

$$h^{o,\alpha}(\mathbf{z}) \geq h(\mathbf{z}), \qquad \forall \alpha \geq 0.$$

The smallest possible value for α is obtained by finding the largest eigenvalue of the Hessian matrix on Z, i.e., by the global solution of a nonconvex optimization problem. Instead, Adjiman et al.[4] have proposed efficient methods for overestimating α. One such method is the application of Gerschgorin's theorem and estimating

$$\frac{1}{2} \max_{\mathbf{z} \in Z} \max_i \max\{0, H_{ii}(\mathbf{z}) + \sum_{j \neq i} |H_{ij}(\mathbf{z})|\}$$

using interval arithmetic on the Hessian matrix. Note that $H_{ij} = \frac{\partial^2 h}{\partial z_i \partial z_j}$.

Since $h^{o,\alpha}(\mathbf{z}) \geq h(\mathbf{z})$ for all $\mathbf{z} \in Z$ and all $\alpha \geq 0$, the optimal objective value of

$$\max_{\mathbf{z} \in Z} h^{o,\alpha}(\mathbf{z}) \tag{3.7}$$

overestimates the optimal objective value of (3.4). The formulated relaxation (3.7) is a box-constrained maximization problem with a smooth concave objective function. The linear constraint qualification along with the concavity of the objective function make the first-order KKT conditions necessary and sufficient for a global maximum. Standard, gradient-based optimization algorithms can reliably solve (3.7). Finally, since (3.7) is box-constrained, the linearization at an arbitrary point $\bar{\mathbf{z}} \in Z$ results in a linear program which is a further relaxation.

The application of γBB relaxation [6] is analogous. In this method, relaxation is

achieved by the addition of exponential terms

$$h^{o,\gamma}(\mathbf{z}) = h(\mathbf{z}) + \sum_{i=1}^{n_z} \left(1 - e^{\gamma_i(z_i - z_i^L)}\right)\left(1 - e^{\gamma_i(z_i^U - z_i)}\right).$$

Example 3.6. Recall the program (3.6). A convex relaxation based on the αBB method is given by

$$\max_{\mathbf{z}\in[-1,1]^2} z_1^2 + \exp(z_1 z_2) + \alpha/2\left((z_1+1)(1-z_1) + (z_2+1)(1-z_2)\right)$$

where α can be calculated through application of Gerschgorin's theorem and interval extensions

$$\begin{aligned} \tfrac{1}{2}\max_{\mathbf{z}\in[-1,1]^2}\{0, 2 + z_2^2\, e^{z_1 z_2} + |e^{z_1 z_2} + z_2 z_1\, e^{z_1 z_2}|\} &= 1 + 1.5e \approx 5.08 \\ \tfrac{1}{2}\max_{\mathbf{z}\in[-1,1]^2}\{0, z_1^2\, e^{z_1 z_2} + |e^{z_1 z_2} + z_2 z_1\, e^{z_1 z_2}|\} &= 1.5e \approx 4.08, \end{aligned}$$

obtaining $\alpha = 5.08$. The optimal solution of the relaxation is $z_1 = z_2 = 0$ with an optimal objective value of 11.16.

3.3.3 Smooth Overestimation with Auxiliary Variables

The third alternative we consider is the introduction of auxiliary variables $\mathbf{w}$ and constraints as described in [44]. First, a factorable representation of the nonconcave function h is developed, introducing a new variable w_i for each distinct factor. Subsequently, the bounds for the auxiliary variables $\mathbf{w}$ are propagated via natural interval extensions by the bounds of $\mathbf{z}$ and the auxiliary variables already introduced. At the next step an equivalent equality constrained program is generated by introducing the definition of each factor as an equality constraint, and replacing each occurrence of a nonconvex function with the relevant factor. Finally, each nonlinear equality constraint is relaxed to pairs of inequalities. If the (smooth) convex and concave envelopes (or tight relaxations) of the nonlinear expression are known, these are introduced, otherwise convex and concave relaxations are computed by the αBB or

γBB method. Nonsmoothness in an envelope can be represented by multiple smooth convex inequalities (e.g., the bilinear case).

In the special case that the objective function contains additive univariate convex terms, these terms can be directly overestimated by the secant without auxiliary variables. Similarly additive concave terms in the objective are left unchanged. Thus a concave overestimating objective function $h^{o,ex}$ is obtained. The resulting program is

$$\begin{aligned}
&\max_{\mathbf{z},\mathbf{l},\mathbf{w}} h^{o,ex}(\mathbf{z},\mathbf{l},\mathbf{w}) \\
&\text{s.t. } t_i^u(\mathbf{z},\mathbf{l},\mathbf{w}) - w_i \leq 0, \quad i = 1,\ldots,n_w \\
&\qquad w_i - t_i^o(\mathbf{z},\mathbf{l},\mathbf{w}) \leq 0, \quad i = 1,\ldots,n_w \\
&\qquad \mathbf{t}^l(\mathbf{z},\mathbf{w},\mathbf{l}) = \mathbf{0} \\
&\qquad \mathbf{z} \in Z \\
&\qquad \mathbf{l} \in [\mathbf{l}^L, \mathbf{l}^U] \subset \mathbb{R}^{n_l} \\
&\qquad \mathbf{w} \in [\mathbf{w}^L, \mathbf{w}^U] \subset \mathbb{R}^{n_w},
\end{aligned} \tag{3.8}$$

where $\mathbf{t}^l$ denote affine, $\mathbf{t}^u$ convex and $\mathbf{t}^o$ concave functions respectively. By construction, the optimal solution value of (3.8) overestimates the optimal solution value of (3.4). It is a convex program with linear equality constraints and differentiable convex inequality constraints. Due to convexity, the KKT conditions are sufficient for a global minimum, and we employ this for the upper bounding procedure. The number of auxiliary variables and constraints introduced depends on the problem size and on the problem structure. Since it is bounded by a small number of factors in the McCormick factorization, it typically is a small multiple of the number of variables.

The existence of a Slater point provides a constraint qualification [20, p. 325] and in this case the first-order KKT conditions are also necessary for a local and global minimum. While typically the existence of a Slater point is expected, to our best knowledge it has not been proved in general for this type of convex relaxations.

Note that since the procedure described here is analogous to the procedure used

in natural interval extensions, which in turn are used to calculate bounds for the auxiliary variables, the relaxation provided by (3.8) is expected to be at least as tight as the interval extensions of h over Z. Moreover, by the introduction of auxiliary variables the relaxations can furnish tighter relaxations than the ones furnished by McCormick's composition theorem without auxiliary variables [126, p. 128].

A further relaxation of (3.8) can be performed via linearization of the objective function and the constraints [126]. A weaker linear relaxation can be obtained by dropping all nonlinear constraints. Finally, an even weaker linear relaxation is generated by dropping all constraints but the variable bounds and obtain a box-constrained program.

Example 3.7. Recall the program (3.6). A convex relaxation with auxiliary variables is constructed as follows. First the objective function is split to the terms z_1^2 and $e^{z_1 z_2}$. The first term is univariate convex and therefore its concave envelope is given by the secant which for $z_1 \in [-1, 1]$ evaluates to 1. For the second term, first an auxiliary variable w is introduced to replace the bilinear term $z_1 z_2$. Its bounds are calculated via natural interval extension to $[-1, 1]$ and are exact. Subsequently the term e^w is recognized as univariate convex and its concave overestimator is constructed by the secant $e^{-1} + (e^1 - e^{-1})\frac{w+1}{2}$. This leads to the nonlinear program

$$\begin{aligned} \max_{\mathbf{z}\in[-1,1]^2, w\in[-1,1]} \; & 1 + e^{-1} + (e^1 - e^{-1})\frac{w+1}{2} \\ \text{s.t. } & w = z_1 z_2, \end{aligned}$$

which is smooth but nonconvex due to the nonlinear equality constraint. It is further

relaxed to obtain

$$\begin{aligned}
\max_{\mathbf{z}\in[-1,1]^2, w\in[-1,1]} \; & 1 + e^{-1} + (e^1 - e^{-1})\frac{w+1}{2} \\
\text{s.t. } & w \leq 1 - z_1 + z_2 \\
& w \leq 1 + z_1 - z_2 \\
& w \geq -1 - z_1 - z_2 \\
& w \geq -1 + z_1 + z_2,
\end{aligned}$$

which is smooth and convex. It has the same optimal objective value as (3.6).

3.4 KKT-Based Upper Bound

In the following we describe how to obtain an upper bound by the solution of a MPEC program. The first step in obtaining the upper bound is to construct a relaxation of the lower-level program on X, i.e., a maximization program with constraints that are partially convex on $\mathbf{p} \in P$ for each $\mathbf{x} \in X$ and an objective function that is partially concave on $\mathbf{p} \in P$ for each $\mathbf{x} \in X$ and overestimates $g(\mathbf{x}, \cdot)$ for all $\mathbf{p} \in P$. As mentioned in Section 3.1, this relaxation results in a restriction of the semi-infinite program.

The next step is to replace the resulting SIP with a MPEC. A basic requirement for this transformation is differentiability of the relaxed lower-level program and therefore only the smooth relaxations described in Section 3.3 are applicable. Moreover, for the MPEC to be a valid restriction, the KKT conditions need to be sufficient for a global maximum. This is ensured by the (partial) convexity of the programs. Note that necessity of the KKT conditions is not required. If the constraint qualifications are violated and the relaxed lower-level program attains its maximum at a point which is not a KKT point, the formulated MPEC will be infeasible. Obtaining rigorous bounds on the KKT multipliers is not necessary but it is still helpful. If the bounds on the KKT multipliers are underestimated, the upper bounding problem is further restricted and therefore remains valid, but may be rendered infeasible.

At this point a comparison with the interval inclusion approach described in chapter 2 is warranted. The MPEC problems have additional variables and constraints. Therefore, they are significantly harder to solve than the ICR [26, 27]. Moreover the stationarity and complementary slackness constraints are equality constraints and state-of-the-art finitely terminating algorithms only guarantee the solution of nonlinear equality constrained problems within a tolerance. In some cases it can be shown that despite this approximation the generated points are guaranteed feasible. On the other hand, typically, convex relaxations are tighter than interval extensions. As a consequence the proposed upper bounds will typically be tighter than the ones furnished by the ICR.

3.4.1 Concave Overestimation without Auxiliary Variables

The first alternative we consider is smooth overestimation of the lower-level program via the addition of known concave terms. Without loss of generality we consider the αBB relaxations. Use of the γBB relaxations is analogous. The αBB overestimation of $g^{o,\alpha}(\mathbf{x}, \cdot)$ on P is given by

$$g^{o,\alpha}(\mathbf{x}, \mathbf{p}) = g(\mathbf{x}, \mathbf{p}) + \alpha \sum_{i=1}^{n_p} (p_i - p_i^L)(p_i^U - p_i).$$

For sufficiently large values of α, the overestimating function $g^{o,\alpha}(\mathbf{x}, \cdot)$ is partially concave on P for each $\mathbf{x} \in X$. A sufficiently large value of α is obtained via interval extensions of the eigenvalue estimates of the Hessian matrix on $X \times P$. Note that for convergence of $g^{o,\alpha}$ to g it is sufficient to subdivide P, without partitioning X.

As described in Section 3.3, the smooth relaxation of the lower-level program via αBB results in a box-constrained maximization program with a smooth concave objective function. Therefore, the first-order KKT conditions are necessary and sufficient

for a global maximum and

$$
\begin{aligned}
f^{UBD,\alpha} = \min_{\mathbf{x},\mathbf{p},\boldsymbol{\mu}} \; & f(\mathbf{x}) \\
\text{s.t.} \; & -g_{p_j}^{o,\alpha}(\mathbf{x},\mathbf{p}) + \mu_j - \mu_{n_p+j} = 0, \quad j = 1,\ldots,n_p \\
& \mu_j(p_j - p_j^U) = 0, \quad j = 1,\ldots,n_p \\
& \mu_{n_p+j}(-p_j + p_j^L) = 0, \quad j = 1,\ldots,n_p \\
& g^{o,\alpha}(\mathbf{x},\mathbf{p}) \leq 0 \\
& 0 \leq \mu_j \leq \mu_j^{max}, \quad j = 1,\ldots,2n_p \\
& \mathbf{x} \in X, \quad \mathbf{p} \in P
\end{aligned}
\tag{3.9}
$$

is equivalent to the restricted SIP for sufficiently large $\boldsymbol{\mu}^{max}$. Note that the number of variables in (3.9) is equal to the original number of variables n_x plus up to three times the number of parameters $3n_p$. In addition to the box constraints there are up to $3n_p$ (potentially nonconvex) equality constraints and one (potentially nonconvex) inequality constraint. A reformulation to a MINLP is possible by introducing binary variables and eliminating the KKT multipliers.

As stated above, typical finitely-terminating NLP solvers only approximate equality constraints. We will show that the feasibility of the points furnished can be easily verified, or the extent of constraint violation bounded. Suppose that an approximately feasible point $(\bar{\mathbf{x}}, \bar{\mathbf{p}})$ of (3.9) is obtained. Since $g^{o,\alpha}$ overestimates g and is partially concave on P we obtain

$$
O(\bar{\mathbf{x}}, P) \leq O^{o,\alpha}(\bar{\mathbf{x}}, P) \leq g^{o,\alpha}(\bar{\mathbf{x}}, \bar{\mathbf{p}}) + \sum_{j=1}^{n_p} \max\left\{ g_{p_j}^{o,\alpha}(\bar{\mathbf{x}}, \bar{\mathbf{p}})(p_j^U - \bar{p_j}), g_{p_j}^{o,\alpha}(\bar{\mathbf{x}}, \bar{\mathbf{p}})(p_j^L - \bar{p_j}) \right\},
$$

where O denotes the optimal solution value of the lower-level program and $O^{o,\alpha}$ the optimal solution value of the relaxed lower-level program. Evaluating the above sum gives the maximum constraint violation for $\bar{\mathbf{x}}$. If (3.9) is solved with sufficiently tight tolerances and $g^{o,\alpha}(\mathbf{x},\mathbf{p}) \leq -\varepsilon$ is used, guaranteed feasible points can be generated.

In order to calculate bounds for the KKT multipliers we first note that

$$g_{p_j}^{o,\alpha}(\mathbf{x},\mathbf{p}) = g_{p_j}(\mathbf{x},\mathbf{p}) + \alpha(p_j^U - p_j) - \alpha(p_j - p_j^L).$$

Therefore,

$$\begin{aligned}
\max_{\mathbf{x}\in X,\mathbf{p}\in P:p_j=p_j^U} g_{p_j}^{o,\alpha}(\mathbf{x},\mathbf{p}) &= -\alpha(p_j^U - p_j^L) + \max_{\mathbf{x}\in X,\mathbf{p}\in P:p_j=p_j^U} g_{p_j}(\mathbf{x},\mathbf{p}) \\
\max_{\mathbf{x}\in X,\mathbf{p}\in P:p_j=p_j^L} -g_{p_j}^{o,\alpha}(\mathbf{x},\mathbf{p}) &= \max_{\mathbf{x}\in X,\mathbf{p}\in P:p_j=p_j^L} \left(g_{p_j}(\mathbf{x},\mathbf{p}) - \alpha(p_j^U - p_j^L)\right) \\
&= -\alpha(p_j^U - p_j^L) - \min_{\mathbf{x}\in X,\mathbf{p}\in P:p_j=p_j^L} g_{p_j}(\mathbf{x},\mathbf{p}).
\end{aligned}$$

Whenever a bound is nonpositive (function monotone) the corresponding variable and complementary slackness conditions are eliminated. For the γBB relaxation [6] the second derivatives of the underestimating terms are variable-dependent, but, evaluated at the variable bounds, they are given by $\gamma^2(1+e^{-\gamma(p_j^U-p_j^L)})$. Therefore, the calculation of bounds on the KKT multipliers is analogous.

3.4.2 Concave Overestimation with Auxiliary Variables

We now consider the alternative of introducing auxiliary variables and constraints. Since this method can take advantage of known convex and concave envelopes, it often provides tighter relaxations than simply adding a convex term to the functions. On the other hand, the introduction of extra variables and constraints increases the size of the lower-level problem. Constraint qualification for this type of programs has not been shown in general. Moreover, obtaining upper bounds on the KKT multipliers is not always possible, so replacing the restricted SIP with a MPEC may be a further restriction and render the upper bounding program infeasible. To ensure convergence of the upper bounding problem this issue has to be addressed.

To obtain a compact presentation we augment the parameters $\mathbf{p}$ with the auxiliary variables and denote these $\tilde{\mathbf{p}}$. Also, we lump the box and auxiliary constraints to the

inequality constraints $\mathbf{u}(\mathbf{x}, \tilde{\mathbf{p}})$. The resulting restriction of (1.1)

$$\begin{aligned} f_{gsip}^{UBD,ex} = & \min_{\mathbf{x} \in X} f(\mathbf{x}) \\ & \text{s.t. } g^{o,ex}(\mathbf{x}, \tilde{\mathbf{p}}) \leq 0, \quad \forall \tilde{\mathbf{p}} : \mathbf{u}(\mathbf{x}, \tilde{\mathbf{p}}) \leq \mathbf{0} \end{aligned} \tag{3.10}$$

is a GSIP. Note that by construction, for all $\mathbf{x} \in X$ there exists $\tilde{\mathbf{p}}$, such that $\mathbf{u}(\mathbf{x}, \tilde{\mathbf{p}}) \leq 0$ and the GSIP can be reformulated to a bilevel program [119]. By convexity, the KKT conditions are sufficient for a global maximum in the lower-level program and

$$\begin{aligned} f^{UBD,ex} = & \min_{\mathbf{x}, \tilde{\mathbf{p}}, \boldsymbol{\mu}} f(\mathbf{x}) \\ & \text{s.t. } -\boldsymbol{\nabla}_{\tilde{\mathbf{p}}} g^{o,ex}(\mathbf{x}, \tilde{\mathbf{p}}) + \boldsymbol{\mu}^{\mathrm{T}} \boldsymbol{\nabla}_{\tilde{\mathbf{p}}} \mathbf{u}(\mathbf{x}, \tilde{\mathbf{p}}) = 0 \\ & \qquad \boldsymbol{\mu}^{\mathrm{T}} \mathbf{u}(\mathbf{x}, \tilde{\mathbf{p}}) = 0 \\ & \qquad g^{o,ex}(\mathbf{x}, \tilde{\mathbf{p}}) \leq 0 \\ & \qquad \mathbf{u}(\mathbf{x}, \tilde{\mathbf{p}}) \leq \mathbf{0} \\ & \qquad 0 \leq \mu_j \leq \mu_j^{max}, \quad j = 1, \ldots, n_\mu \\ & \qquad \mathbf{x} \in X, \qquad \tilde{\mathbf{p}} \in \mathbb{R}^{n_{\tilde{p}}} \end{aligned} \tag{3.11}$$

provides a valid upper bound of (1.1). Recall that the number of variables in the lower-level problem $n_{\tilde{p}}$ and the number of KKT multipliers n_μ depend on the number of linear and nonlinear expressions replaced. Here a reformulation as a MINLP is possible by the introduction of binary variables, but elimination of the KKT multipliers does not seem possible in general.

3.5 Linearization-Based Upper Bound

Similar to the MPEC-based upper bounds, the first step in the linearization-based upper bounds is to construct a convex relaxation of the lower-level program and thus

a restriction of (1.1)

$$\begin{aligned} &\min_{\mathbf{x}\in X} f(\mathbf{x}) \\ &\text{s.t. } g^o(\mathbf{x},\mathbf{p}) \leq 0, \quad \forall \mathbf{p} \in P, \end{aligned} \tag{3.12}$$

where $g^o : X \times P \to \mathbb{R}$ is partially concave on P for each $\mathbf{x} \in X$ and $g^o(\mathbf{x},\mathbf{p})$ overestimates $g(\mathbf{x},\mathbf{p})$. Note that for the approach involving auxiliary variables a somewhat different treatment is needed and described in Section 3.5.3.

The second step further restricts the generated SIP by linearizing at an arbitrary point $\bar{\mathbf{p}} \in P$, pointwise in X, and creating the following SIP

$$\begin{aligned} &\min_{\mathbf{x}\in X} f(\mathbf{x}) \\ &\text{s.t. } g^{o,lin}(\mathbf{x},\mathbf{p}) \leq 0, \quad \forall \mathbf{p} \in P, \end{aligned} \tag{3.13}$$

where $g^{o,lin}(\mathbf{x},\mathbf{p}) \equiv g^o(\mathbf{x},\bar{\mathbf{p}}) + \sum_{j=1}^{n_p} g^o_{p_j}(\mathbf{x},\bar{\mathbf{p}})(p_j - \bar{p}_j)$. An equivalent nonsmooth reformulation of (3.13) is the following problem

$$\begin{aligned} &\min_{\mathbf{x}\in X} f(\mathbf{x}) \\ &\text{s.t. } \max_{\mathbf{p}\in P} g^{o,lin}(\mathbf{x},\mathbf{p}) \leq 0. \end{aligned} \tag{3.14}$$

Since $g^{o,lin}$ is affine in $\mathbf{p}$, the maximum of $g^{o,lin}(\mathbf{x},\cdot)$ on P, will be attained at one of the vertices P_e of P for each $\mathbf{x}$ in X. Therefore an equivalent finite representation of (3.14) is

$$\begin{aligned} &\min_{\mathbf{x}\in X} f(\mathbf{x}) \\ &\text{s.t. } g^{o,lin}(\mathbf{x},\mathbf{p}) \leq 0, \quad \forall \mathbf{p} \in P_e. \end{aligned} \tag{3.15}$$

While for any $\bar{\mathbf{p}} \in P$, the formulated finite NLP (3.15) is a valid restriction of (1.1), the choice of $\bar{\mathbf{p}}$ greatly affects the strength of the generated upper bounds. Compared to the MPEC-based upper bound, this linearization approach presents the

inherent advantage that it avoids the use of equality constraints (complementarity and stationarity conditions) and any feasible point of (3.15) is guaranteed feasible for (1.1). On the other hand, the MPEC approach introduces a polynomial (in the number of inner variables or in the number of inner variables and nonconvex terms) number of constraints, whereas the linearization approach introduces a potentially exponential number of constraints. Moreover, the linearization approach produces bounds that are at best as tight as the MPEC-based ones, assuming that both problems are solved to global optimality.

If either of the two following relationships holds for variable p_j

$$\max_{\mathbf{x}\in X} g^o_{p_j}(\mathbf{x}, \bar{\mathbf{p}}) \leq 0 \tag{3.16}$$

$$\min_{\mathbf{x}\in X} g^o_{p_j}(\mathbf{x}, \bar{\mathbf{p}}) \geq 0 \tag{3.17}$$

the number of constraints can be reduced. The following procedure describes how to obtain the (sufficient) subset of extreme points P_{e^*} that needs to be considered in problem (3.15)

- Initialize $P_{e^*} = P_e$.
- FOR $j = 1, \ldots, n_p$ DO
  - IF $\max_{\mathbf{x}\in X} g^o_{p_j}(\mathbf{x}, \bar{\mathbf{p}}) \leq 0$ THEN $P_{e^*} = \{\mathbf{p} \in P_{e^*} : p_j = p_j^L\}$
    ELSE IF $\min_{\mathbf{x}\in X} g^o_{p_j}(\mathbf{x}, \bar{\mathbf{p}}) \geq 0$ THEN $P_{e^*} = \{\mathbf{p} \in P_{e^*} : p_j = p_j^U\}$.
- END

Evaluating the above optimization programs is expensive and we propose to estimate them with interval analysis.

Finally, it should be noted that for certain nonconvex terms, the concave overestimator (or envelope) g^o is given by more than one smooth function. Any of these functions that overestimates $g(\mathbf{x}, \cdot)$, for each $\mathbf{x} \in X$, can be used alone to provide the smooth overestimator g^o. Note that using both functions would result in a weaker upper bound. For instance, for the bilinear term $h(p_1, p_2) = p_1 p_2$ any of the two linear

overestimators $h^{o,1}$ and $h^{o,2}$ where

$$h^{o,1}(\mathbf{p}) = p_1^L p_2 + p_2^U p_1 - p_1^L p_2^U$$
$$h^{o,2}(\mathbf{p}) = p_1^U p_2 + p_2^L p_1 - p_1^U p_2^L$$

can be used as a valid overestimator for h on $[p_1^L, p_1^U] \times [p_2^L, p_2^U]$.

3.5.1 Smooth Concave Overestimation without Auxiliary Variables

Recall that the concave relaxation of g on P using αBB techniques has the form

$$g^{o,\alpha}(\mathbf{x}, \mathbf{p}) = g(\mathbf{x}, \mathbf{p}) + \alpha \sum_{j=1}^{n_p} (p_j - p_j^L)(p_j^U - p_j)$$

and the linearized approximation of the αBB concave relaxation around a point $\bar{\mathbf{p}} \in P$ is

$$g^{o,\alpha,lin}(\mathbf{x}, \mathbf{p}) = g^{o,\alpha}(\mathbf{x}, \bar{\mathbf{p}}) + \sum_{j=1}^{n_p} g_{p_j}(\mathbf{x}, \bar{\mathbf{p}})(p_j - \bar{p}_j) + \alpha \sum_{j=1}^{n_p} (-2\bar{p}_j + p_j^L + p_j^U)(p_j - \bar{p}_j).$$

Therefore, the αBB-based linearized upper bounding problem is of the form

$$\begin{aligned} &\min_{\mathbf{x} \in X} f(\mathbf{x}) \\ &\text{s.t. } g^{o,\alpha,lin}(\mathbf{x}, \mathbf{p}) \leq 0, \quad \forall \mathbf{p} \in P_{e^*}, \end{aligned}$$

where P_{e^*} is calculated by the following procedure

- Initialize $P_{e^*} = P_e$.
- FOR $j = 1, \ldots, n_p$ DO
  - IF $\max_{\mathbf{x} \in X} g_{p_j}(\mathbf{x}, \bar{\mathbf{p}}) \leq \alpha(2\bar{p}_j - p_j^L - p_j^U)$ THEN $P_{e^*} = \{\mathbf{p} \in P_{e^*} : p_j = p_j^L\}$ ELSE IF $\min_{\mathbf{x} \in X} g_{p_j}(\mathbf{x}, \bar{\mathbf{p}}) \geq \alpha(2\bar{p}_j - p_j^L - p_j^U)$ THEN $P_{e^*} = \{\mathbf{p} \in P_{e^*} : p_j = p_j^U\}$.

- END

3.5.2 Nonsmooth Concave Overestimation without Auxiliary Variables

Similar to the aforementioned technique, the goal of this method is to introduce a concave overestimator of the constraint g with respect to the inner variables $\mathbf{p}$ using McCormick techniques, and then to linearize the resulting expression around an arbitrary point $\bar{\mathbf{p}} \in P$. In order to explain the method we first analyze its application to a (trivial) SIP for which the constraint g does not depend on the variables $\mathbf{x}$

$$\begin{aligned} &\min_{\mathbf{x}\in X} f(\mathbf{x}) \\ &\text{s.t. } g(\mathbf{p}) \leq 0, \quad \forall \mathbf{p} \in P. \end{aligned} \tag{3.18}$$

The first step to create a finite and linearized form for this SIP is to create the concave overestimator g^o of g on P. In order to use McCormick's composition theorem, we first write g in the following form

$$g(\mathbf{p}) = H[h(\mathbf{p})],$$

where $h : P \to [g^L, g^U] \subset \mathbb{R}$ is a continuous multivariate intrinsic function on P and $H : [a, b] \to \mathbb{R}$, where $[a, b] \supset [g^L, g^U]$, is a continuous univariate function on $[a, b]$.

The second step is to construct a convex function h^u and a concave function h^o that satisfy

$$h^u(\mathbf{p}) \leq h(\mathbf{p}) \leq h^o(\mathbf{p}), \quad \forall \mathbf{p} \in P,$$

and also a concave function H^o that satisfies

$$H^o(z) \geq H(z), \quad \forall z \in [a, b].$$

Furthermore, let $z_{max} \in \arg\max_{z\in[a,b]} H^o(z)$. By McCormick's composition theorem the

function $g^{o,mc}$ defined as

$$g^{o,mc}(\mathbf{p}) = H^o[\text{mid}(h^u(\mathbf{p}), h^o(\mathbf{p}), z_{max})]$$

is a concave relaxation of H on $[a, b]$, and consequently of g on P. Thus a valid restriction of (3.18) is the following SIP

$$\begin{aligned} &\min_{\mathbf{x}\in X} f(\mathbf{x}) \\ &\text{s.t. } g^{o,mc}(\mathbf{p}) \leq 0, \quad \forall \mathbf{p} \in P. \end{aligned} \tag{3.19}$$

Provided that the mid function gives a unique result, i.e., h^u, h^o or z_{max} for all parameter values, an equivalent smooth reformulation of (3.19) is

$$\begin{aligned} &\min_{\mathbf{x}} f(\mathbf{x} \in X) \\ &\text{s.t. } H^o(h_{mid}(\mathbf{p})) \qquad \leq 0, \quad \forall \mathbf{p} \in P, \end{aligned} \tag{3.20}$$

where h_{mid} is exactly one of h^u, h^o or z_{max}.

However, in the general case, H^o is a nonsmooth function due to the existence of the mid function (see Appendix B.2.1). A simple way to alleviate this nonsmoothness is to enumerate all possible outcomes from the mid functions, and therefore, create the following SIP

$$\begin{aligned} &\min_{\mathbf{x}\in X} f(\mathbf{x}) \\ &\quad \text{s.t. } H^o(h^u(\mathbf{p})) \leq 0, \quad \forall \mathbf{p} \in P \\ &\qquad\quad H^o(h^o(\mathbf{p})) \leq 0, \quad \forall \mathbf{p} \in P \\ &\qquad\quad H^o(z_{max}) \leq 0, \quad \forall \mathbf{p} \in P, \end{aligned} \tag{3.21}$$

which is a further restriction of (3.20). It is obvious though that

$$H^o(z_{max}) = \max_{z\in[a,b]} H^o(z) \geq \max_{z\in[a,b]} H(z) \geq \max_{\mathbf{p}\in P} H(h(\mathbf{p})) = \max_{\mathbf{p}\in P} g(\mathbf{p}).$$

Therefore, $H^o(z_{max})$ is an overestimator of g and also a constant and thus it is a concave overestimator of g. Therefore, (3.21) can be reduced to

$$\begin{aligned} &\min_{\mathbf{x}\in X} f(\mathbf{x}) \\ &\text{s.t. } H^o(z_{max}) \leq 0, \quad \forall \mathbf{p} \in P, \end{aligned} \tag{3.22}$$

with a dummy dependence on $\mathbf{p}$. In conclusion, if the mid function provides a unique result for all parameter values, then we use (3.20). If, on the other hand, the result of the mid function changes or is not immediately obvious then we use form (3.22). In the general SIP (1.1), where the constraint depends on the variables, we have to make certain modifications to our analysis to incorporate the $\mathbf{x}$-dependence of the constraints. The example in Appendix B.2.2 shows that the value of $\mathbf{x}$ can influence the functional form of the convex and concave overestimators of h on P, the functional form of the concave overestimator of H on Z and finally the value of z_{max}. Furthermore, McCormick's composition furnishes a (finite) number of concave overestimators of $g(\mathbf{x}, \cdot)$ on P that are valid over different regions X^m of X. Therefore, for a nonlinear term $g(\mathbf{x}, \mathbf{p})$ there exists a finite number of functions $g^{o,mc,m}$, $1 \leq m \leq n$, that are partially concave on P and satisfy

$$\begin{aligned} g^{o,mc,m}(\mathbf{x}, \mathbf{p}) &\geq g(\mathbf{x}, \mathbf{p}), \quad \forall \mathbf{p} \in P, \text{ and for each } \mathbf{x} \in X^m \\ g^{o,mc,m}(\mathbf{x}, \mathbf{p}) &= H^{o,m}[\text{mid}\,(h^{u,m}(\mathbf{x}, \mathbf{p}), h^{o,m}(\mathbf{x}, \mathbf{p}), z^m_{max}(\mathbf{x}))] \\ \text{int}X^m \cap \text{int}X^{m'} &= \emptyset, \quad \forall m \neq m', \quad 1 \leq m \leq n \quad 1 \leq m' \leq n \\ \bigcup_{m=1}^{n} X^m &= X. \end{aligned}$$

For the case that the resulting expressions of $g^{o,mc,m}$ are, for each m, defined uniquely by one of h^u, h^o and z_{max}, denoted h^m_{mid}, then a smooth restricted SIP can be formu-

lated as

$$\begin{aligned} &\min_{\mathbf{x}\in X} f(\mathbf{x}) \\ &\text{s.t. } H^{o,m}[h^m_{mid}(\mathbf{x},\mathbf{p})] \leq 0, \quad \forall \mathbf{p} \in P,\ 1 \leq m \leq n. \end{aligned} \tag{3.23}$$

Then, by linearizing the constraints of (3.23) around an arbitrary point $\bar{\mathbf{p}}$, the following NLP provides an upper bound to (1.1)

$$\begin{aligned} &\min_{\mathbf{x}\in X} f(\mathbf{x}) \\ &\text{s.t. } H^{o,m}[h^m_{mid}(\mathbf{x},\bar{\mathbf{p}})] + \sum_{j=1}^{n_p} H^{o,m}_{p_j}[h^m_{mid}(\mathbf{x},\bar{\mathbf{p}})](p_j - \bar{p}_j) \leq 0, \quad \forall \mathbf{p} \in P_e,\ 1 \leq m \leq n. \end{aligned} \tag{3.24}$$

For every m for which $g^{c,mc,m}$ cannot be defined uniquely (either because of the McCormick composition or because it is not easily furnished by the comparison of the terms in the mid function) the term $h^m_{mid}(\mathbf{x},\mathbf{p}) = z^m_{max}(\mathbf{x})$ is used.

3.5.3 Smooth Concave Overestimation using Auxiliary Variables

A method to create a valid upper bound for (1.1) based on smooth concave overestimation of $g(\mathbf{x},\cdot)$ using auxiliary variables was presented in Section 3.4.2. Recall that the solution of the following GSIP is a restriction of (1.1)

$$\begin{aligned} &\min_{\mathbf{x}\in X} f(\mathbf{x}) \\ &\text{s.t. } g^{o,ex}(\mathbf{x},\tilde{\mathbf{p}}) \leq 0, \quad \forall \tilde{\mathbf{p}} : \mathbf{u}(\mathbf{x},\tilde{\mathbf{p}}) \leq \mathbf{0}, \end{aligned} \tag{3.25}$$

where the parameters $\tilde{\mathbf{p}} \in \tilde{P}$ contain the original parameters $\mathbf{p}$ and auxiliary parameters representing expressions of the variables and parameters. Bounds on the auxiliary parameters are propagated through interval extensions. The linearization approaches require that the set of parameter vertices is easily calculated which is not

the case here. Therefore, a further restriction of (3.25) is obtained by dropping the inner level constraints (with the exception of the bound constraints)

$$\begin{aligned} &\min_{\mathbf{x}\in X} f(\mathbf{x}) \\ &\text{s.t. } g^{o,ex}(\mathbf{x},\tilde{\mathbf{p}}) \leq 0, \quad \forall \tilde{\mathbf{p}} \in \tilde{P} \end{aligned} \tag{3.26}$$

and therefore further relaxing the lower-level program and thus further restricting (3.25). Taking into consideration that $g^{o,ex}(\mathbf{x},\cdot)$ is partially concave on $\tilde{P}$ for each $\mathbf{x} \in X$ and similar to the linearization approaches already presented, the following linearization of (3.26) around an arbitrary point $\bar{\mathbf{p}} \in \tilde{P}$ furnishes an upper bound for (1.1)

$$\begin{aligned} &\min_{\mathbf{x}\in X} f(\mathbf{x}) \\ &\text{s.t. } g^{o,ex}(\mathbf{x},\bar{\mathbf{p}}) + \sum_{j=1}^{n_{\tilde{p}}} g^{o,ex}_{p_j}(\mathbf{x},\bar{\mathbf{p}})(\tilde{p}_j - \bar{p}_j) \leq 0, \quad \forall \tilde{\mathbf{p}} \in \tilde{P}_e, \end{aligned} \tag{3.27}$$

where $\tilde{P}_e$ denotes the set of vertices of $\tilde{P}$. Recall that the set of vertices considered can be reduced if the functions are monotone with respect to some parameters.

3.6 Relaxation over x and p

The upper bounding methodologies that have been presented so far rely on creating a function g^o that is partially concave with respect to the parameters $\mathbf{p}$ pointwise for each $\mathbf{x} \in X$. Another way of creating a valid overestimator of g is to construct a jointly concave function $g^{o,j}$ on $X \times P$, i.e., with respect to both the variables $\mathbf{x}$ and the parameters $\mathbf{p}$, using either McCormick or αBB concave relaxation methods, that satisfies

$$g^{o,j}(\mathbf{x},\mathbf{p}) \geq g(\mathbf{x},\mathbf{p}), \quad \forall (\mathbf{x},\mathbf{p}) \in X \times P.$$

Then, the following SIP is a restriction of (1.1)

$$\begin{aligned} &\min_{\mathbf{x}\in X} f(\mathbf{x}) \\ &\text{s.t. } g^{o,j}(\mathbf{x},\mathbf{p}) \leq 0, \quad \forall \mathbf{p} \in P. \end{aligned} \tag{3.28}$$

Note that for convergence both host sets (X and P) need to be refined.

3.6.1 Linearization

Similar to the linearization approaches that have been presented so far, and since $g^{o,j}$ is concave on $X \times P$ we can linearize (3.28) around an arbitrary point $(\bar{\mathbf{x}}, \bar{\mathbf{p}}) \in X \times P$ to obtain the following restriction of (1.1)

$$\begin{aligned} &\min_{\mathbf{x}\in X} f(\mathbf{x}) \\ &\text{s.t. } g^{o,j}(\bar{\mathbf{x}},\bar{\mathbf{p}}) + \sum_{j=1}^{n_x} g^{o,j}_{x_j}(\bar{\mathbf{x}},\bar{\mathbf{p}})(x_j - \bar{x}_j) + \sum_{j=1}^{n_p} g^{o,j}_{p_j}(\bar{\mathbf{x}},\bar{\mathbf{p}})(p_j - \bar{p}_j) \leq 0, \quad \forall \mathbf{p} \in P_e. \end{aligned} \tag{3.29}$$

Taking into consideration that the constraint function in (3.29) is separable in $\mathbf{x}$ and $\mathbf{p}$, a single inequality constraint is needed (i.e., P_e is a singleton). The corresponding parameter point $\mathbf{p}^*$ is calculated by the following procedure

- FOR $j = 1, \ldots, n_p$ DO
  - IF $g^{o,j}_{p_j}(\bar{\mathbf{x}},\bar{\mathbf{p}}) \leq 0$ THEN $p^*_j = p^L_j$ ELSE $p^*_j = p^U_j$.
- END

The following NLP with a single linear inequality constraint

$$\begin{aligned} &\min_{\mathbf{x}\in X} f(\mathbf{x}) \\ &\text{s.t. } g^{o,j}(\bar{\mathbf{x}},\bar{\mathbf{p}}) + \sum_{j=1}^{n_x} g^{o,j}_{x_j}(\bar{\mathbf{x}},\bar{\mathbf{p}})(x_j - \bar{x}_j) + \sum_{j=1}^{n_p} g^{o,j}_{p_j}(\bar{\mathbf{x}},\bar{\mathbf{p}})(p^*_j - \bar{p}_j) \leq 0 \end{aligned} \tag{3.30}$$

provides a valid upper bound for (1.1).

This approach will obviously furnish looser upper bounds than the ones produced by the MPEC and linearization approaches that rely on the concave overestimation of g only with respect to the parameters $\mathbf{p}$. However, a single linear inequality is required, compared to the polynomial or exponential number of nonlinear constraints. Again, the choice of $\bar{\mathbf{p}}$ greatly affects the tightness of the proposed upper bound.

3.6.2 MPEC formulation

Similar to the MPEC approach that was described in Section 3.4, a possible bounding problem is to replace the lower-level problem of (3.28) with its equivalent KKT conditions and solve the resulting problem to obtain an upper bound. Although this method would produce valid upper bounds, there are two distinct drawbacks compared to the MPEC approach that relies on concave relaxation of g only with respect to $\mathbf{p}$. First of all, the process of creating a concave overestimator of g on $X \times P$ will replace convex and nonconvex, with respect to $\mathbf{x}$, terms by concave ones which does not seem to simplify the solution of the resulting problem. Secondly the generated relaxation will be weaker. Note that even using αBB techniques, the value of α would be greater or equal to the value of α that corresponds to the concave relaxation only on P because the Hessian increases in size. In conclusion, this method does not seem to produce either tighter bounds or simpler constraint expressions and will, therefore, not be analyzed further.

3.7 Convergence of Upper Bounding Problems

The various alternatives described restrict the SIP (1.1) by overestimating the constraint $\mathbf{g}(\mathbf{x}, \mathbf{p})$ pointwise in $\mathbf{x}$. The parametric optimal solution value of the lower-level

program $O(\mathbf{x})$ is overestimated obtaining

$$\begin{aligned} &\min_{\mathbf{x}\in X} f(\mathbf{x}) \\ &\text{s.t.} \quad \tilde{g}(\mathbf{x}) \leq 0, \end{aligned}$$

with $\tilde{g}(\mathbf{x}) \geq O(\mathbf{x}, P)$. As described in Section 3.1 this relaxation of the lower-level program leads to a restriction of the SIP. In general, this restriction excludes some feasible points and may render the upper bounding problem infeasible. To ensure that the upper bound converges to the optimal solution value a subdivision of the parameter host set P, as in [27], is deemed necessary. For the subdivision additional variables and/or constraints will be introduced. Methods for efficient convergence are outside the scope of this paper and here we only briefly discuss basic convergence properties.

Similarly to the ICR by Bhattacharjee et al. [26, 27] and the proposal by Floudas and Stein [42], an exhaustive subdivision of the parameter set P leads to a pointwise convergence of $\tilde{g}$ to O. Therefore, points $\bar{\mathbf{x}}$ satisfying $\max_{\mathbf{p}\in P} g(\bar{\mathbf{x}}, \mathbf{p}) < 0$, i.e., SIP Slater points, become feasible in the upper bounding problems for sufficiently fine subdivision. As a consequence, if the upper bounding problems are solved to global optimality and SIP Slater points exist arbitrarily close to a global minimum of (1.1) the upper bound converges to the optimal solution value.

3.8 Implementation and Numerical Results

3.8.1 Implementation

The proposed upper bounding problems potentially contain nonconvex objective function and/or constraints. Aiming to obtain the best possible bounds, we solve all the problems globally with BARON version 7.5 [108] available through GAMS version 22.1 [30] on a 64-bit Xeon processor 3.2GHz running Linux 2.6.13.

As is typical in NLP and MINLP solvers, BARON allows the violation of inequal-

ity and equality tolerances by a positive tolerance. This is a limitation for the upper bounding problems involving equality constraints. Note that the inequality constraint $g(\mathbf{x}, \mathbf{p}) \leq 0$ can be further restricted to $g(\mathbf{x}, \mathbf{p}) \leq -\varepsilon$, for an ε equal to the constraint violation of the NLP solver, and therefore does not pose a significant problem. To obtain good estimates we set the smallest possible value (10^{-9}) for the relevant tolerances (`conttol, boxtol, inttol`). The absolute and relative termination criteria, i.e., the difference between the lower and upper bounds in the subproblems, are set to 10^{-4}. Our previous numerical experiments with similar programs have shown slow convergence for problems involving third order monomials, e.g., x^3 and for consistency purposes, we systematically encode third order monomials as a product of a square and a linear term, e.g. $x^2\, x$, and fourth order monomials as the product of two squares, e.g., $x^2\, x^2$.

The complementary slackness conditions that appear in the upper bounding problems are left as nonlinear equations. For the linearizations the midpoint of P is used.

For the implementation of the McCormick relaxations without introducing extra variables, there are a number of heuristics that need to be specified. For bilinear terms $p_1 p_2$ we use $p_1^U p_2 + p_2^L p_1 - p_1^L p_2^U$ as the concave overestimator. Similarly for negative bilinear terms $-p_1 p_2$ we use $(-p_1)^U p_2 + p_2^L p_1 - (-p_1)^U p_2^L$. For nonconvex terms of the form $w(x, p_1, p_2) = t(x)\, p_1\, p_2$ where $t(x)$ can hold both positive and negative values, we use both of the aforementioned forms to ensure a valid overestimation of $g(x, \cdot, \cdot)$ over X.

For the construction of the concave overestimator of g on $X \times P$, bilinear terms are handled in the same manner. For trilinear terms of the form $g(x, p_1, p_2) = x p_1 p_2$, the term is rewritten as $g_1(x, p_1 p_2) g_2(x, p_1, p_2)$ where $g_1(x, p_1, p_2) = x$ and $g_2(x, p_1, p_2) = p_1 p_2$. Convex and concave envelopes of these terms exist and using the constraints in Appendix B.2.3, the concave overestimator of $w_1 w_2$ that is used is $\gamma_1(x, p_1, p_2) + \gamma_2(x, p_1, p_2) - G_1^U G_2^L$. Furthermore, for compositions with respect to w, e.g., $(1 - g(x, p_1, p_2))^2$ then the convex and concave relaxations of w used are

$$\alpha_1(x, p_1, p_2) + \alpha_2(x, p_1, p_2) - G_1^U G_2^L$$

and

$$\gamma_1(x, p_1, p_2) + \gamma_2(x, p_1, p_2) - G_1^L G_2^L$$

respectively. Nonconvex terms of the form $w(x,p) = xp^2$ and $w(x,p) = x^2p^2$ are written as $w_1(x,p) = x$, $w_2(x,p) = p^2$ and $w_1(x,p) = x^2$, $w_2(x,p) = p^2$ respectively, and analyzed in a similar fashion.

Since the problems considered are relatively small, for the αBB relaxations we obtain the smallest possible α through the solution of a global optimization problem. This is done in the spirit of obtaining the tightest possible bounds. On the other hand, the bounds on the KKT multipliers and the second derivatives are estimated using the natural interval extensions capabilities of DAEPACK [128, 129]. For the MPEC-based upper bound using relaxation with extra variables, the upper bound of the KKT multipliers is set to 10^3. Note that overestimating the bounds of the multipliers typically increases the computational requirements to solve the problems.

3.8.2 Numerical Results

As a test set we use the well-established problems by Watson [135], summarized in Appendix B.1. Since BARON and DAEPACK currently do not support trigonometric functions, we only use those examples that do not involve trigonometric functions. For all problems we used $\mathbf{x} \in [-10, 10]^{n_x}$.

Tables 3.1 and 3.2 respectively contain the computational requirement as reported by BARON (through the GAMS attribute `resusd`) and the bounding values obtained. No distinction is made for times below 0.01s. In three cases (all KKT-based upper bounding problems) we distinguish between the time to find the optimal solution value and to confirm it (number in brackets) because the two computational requirements differ dramatically. The first column (Label) has the label of the problem, while the column labeled f^* contains the best known solution for the problem. The next six columns contain the upper bounds obtained by our upper bounding proposals, labeled by the corresponding sections. The final column (ICR) is the interval constrained reformulation by Bhattacharjee et al.[26] that was reproduced for the

sake of completeness.

Table 3.1: Numerical Results: Relaxation-Based Bounds

Problem		Upper Bounds						
Label	f^*	3.4.1	3.4.2	3.5.1	3.5.2	3.5.3	3.6	ICR
2	0.194	$+\infty$	0.194	$+\infty$	0.28	0.194	50.58	0.38
5	4.30	20.2	4.32	27.7	4.64	4.32	7890	4.72
6	97.2	$+\infty$	97.2	$+\infty$	306	97.2	$+\infty$	97.2
7	1.00	86.1	1.00	$+\infty$	1.60	1.00	$+\infty$	1.00
8	2.44	$+\infty$	3.13	$+\infty$	4.20	3.13	$+\infty$	7.39
9	-12	$+\infty$	-12.0	$+\infty$	-12.0	-12.0	$+\infty$	-12.0
N	0.00	$+\infty$	0.00	$+\infty$	0.00	0.00	$+\infty$	0.00

Table 3.2: Computational Requirements: Relaxation-Based Bounds

Problem	Upper Bounds						
Label	3.4.1	3.4.2	3.5.1	3.5.2	3.5.3	3.6	ICR
2	0.01	0.48	0.01	0.02	0.02	0.01	0.01
5	0.13	0.54	0.02	0.02	0.02	0.01	0.01
6	0.05	0.67	0.01	0.03	0.04	0.01	0.04
7	0.28	0.18 (121)	0.01	0.02	0.02	0.01	0.02
8	0.07	0.01 (273)	0.01	0.02	0.02	0.01	0.01
9	3.30	0.41 (1000)	0.01	0.01	0.01	0.01	0.01
N	0.01	0.04	0.01	0.01	0.02	0.01	0.01

3.8.3 Conclusions from Numerical Experiments

The upper bounds furnished are often exact. The computational requirement to obtain upper bounds is quite low for the small-scale problems considered; note that for the case of a KKT-based upper bound with extra variables, the computational requirement to confirm the global solution is quite high for three problems involving two parameters.

As expected, the KKT-based upper bounds using extra variables (Section 3.4.2) can be significantly tighter than the ICR-based at the expense of a higher computational cost. To our surprise the linearization-based bounds using auxiliary variables (Section 3.5.3) produced bounds as tight as the ones based on the KKT conditions and we believe that this is due to the problem structure. The bounds based on smooth relaxation without extra variables (Sections 3.4.1 and 3.5.1) are relatively weak. We

want to point out, once more, that we deviated from the αBB relaxation described by Adjiman et al. [3] and consider the constraint as a whole. Note finally, that the number of parameters in the problems considered is small ($n_p \in \{1, 2\}$) and therefore the effect of the exponential number of constraints in the linearization (Section 3.5.1, 3.5.2 and 3.5.3) is not apparent.

Chapter 4

Introduction to GSIP

Generalized semi-infinite programs (GSIP) are optimization problems that involve a finite number of decision variables subject to an infinite number of constraints the index set of which is dependent on the decision variables. The formulation of GSIP that we will be concerned with is:

$$\begin{aligned} &\inf_{\mathbf{x}\in X} f(\mathbf{x}) \\ &\text{s.t. } g(\mathbf{x},\mathbf{p}) \leq 0,\ \forall \mathbf{p} \in P(\mathbf{x}) \\ &\qquad P(\mathbf{x}) = \{\mathbf{p} \in D : u_j(\mathbf{x},\mathbf{p}) \leq 0, j \in J\} \\ &\qquad\quad X \subset \mathbb{R}^{n_x},\ D \subset \mathbb{R}^{n_p}, |J| < \infty. \end{aligned} \tag{4.1}$$

First of all, we provide some examples of engineering applications that give rise to GSIP in section 4.1. We provide a list of helpful definitions in Section 4.2 and a comparison of GSIP and SIP in Sections 4.3 and 4.4. In Section 4.5 we outline the irregularities of the feasible set in GSIP and in Section 4.6 we provide a literature review on the global optimization methods for GSIP. Finally, in Section 4.7 we comment on the limitations of the numerical procedures that have been proposed in the literature and analyze the contribution of our work.

4.1 Origin and Engineering Applications

The term "Generalized Semi-Infinite Programs" first appeared in [59] but to our best knowledge this work has not been published yet. The term officially appears in [51] where the computation of the acceleration radius of robots is formulated as a GSIP.

A generic engineering application that gives rise to GSIP is the reverse Chebyshev approximation problem (RCAP) [68]. Let $f : D \to \mathbb{R}$ be a twice continuously differentiable function on $D \subset \mathbb{R}^2$. Let $y = f(\mathbf{p})$ describe the dependence of a physical quantity y on the input parameters $\mathbf{p}$. Let $g : \mathbb{R}^2 \times \mathbb{R}^6 \to \mathbb{R}$ be an approximating function for f that is parameterized in $\mathbf{x}$, e.g. a second-order polynomial of the form $g(\mathbf{x}, \mathbf{p}) = x_1 p_1^2 + x_2 p_1 + x_3 p_2^2 + x_4 p_2 + x_5 p_1 p_2 + x_6$. Consider a fixed approximation error ε. The goal is to maximize the volume of the parameter set $P(\mathbf{y}) \subset D$ denoted by $\mathrm{Vol}(\mathbf{y})$ for which the Chebyshev norm of the difference between the original function f and the approximating function $g(\mathbf{x}, \cdot)$ on $P(\mathbf{y})$ is less than ε. This formulation gives rise to the following GSIP:

$$\begin{aligned} &\max_{\mathbf{x},\mathbf{y}} \mathrm{Vol}(\mathbf{y}) \\ &\text{s.t. } |f(\mathbf{p}) - g(\mathbf{x}, \mathbf{p})| \leq \varepsilon, \forall \mathbf{p} \in P(\mathbf{y}) \subset D \\ &\qquad\qquad \mathbf{x} \in X \subset \mathbb{R}^6, \end{aligned}$$

where the set $P(\mathbf{y})$ is described in terms of a finite number of inequalities, i.e., $P(\mathbf{y}) = \{\mathbf{p} \in P : u_j(\mathbf{y}, \mathbf{p}) \leq 0,\ j \in J\}$.

Another engineering application that gives rise to GSIP is design centering [117]. For the source of the original contribution and the notation we refer the reader to [117]. We are including this analysis here for completeness.

Assume that $C \subset \mathbb{R}^m$ is a compact set with a nonempty interior, known as the container or host set, defined as:

$$C = \{\mathbf{p} \in \mathbb{R}^m : u_j(\mathbf{p}) \leq 0, j \in J\} \tag{4.2}$$

Consider a parameterized body $B(\mathbf{x}) \subset \mathbb{R}^m$ with $\mathbf{x} \in \mathbb{R}^n$ and a measure of the body

$f(\mathbf{x})$. The goal in design centering is to maximize the measure of the parameterized body subject to the body being fully inscribed in the host set C. This gives rise to the following optimization problem:

$$\begin{aligned} &\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) \\ &\text{s.t. } B(\mathbf{x}) \subset C. \end{aligned} \tag{4.3}$$

As suggested by [117], relation (4.3) gives rise to the following GSIP:

$$\begin{aligned} &\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) \\ &\text{s.t. } u_j(\mathbf{p}) \leq 0, \forall \mathbf{p} \in B(\mathbf{x}), j \in J. \end{aligned} \tag{4.4}$$

Within the context of design centering, a very interesting application of GSIP is gemstone cutting [138] in which a prototype diamond is cut from a rough gemstone. Therefore, the container (host) set is the rough diamond, the parameterized body is the prototype diamond and the objective is the maximization of the volume of the cut diamond. Another major application of design centering is kinetic model reduction (KMR) [25, 84, 90]. In KMR, the full kinetic model of a combustion mechanism is replaced by a reduced kinetic model in which some of the reactions and/or species have been eliminated from the model. The container set is the subset of the concentration and temperature space for which the difference between the full and the kinetic model are less than some given tolerance. The parameterized body is a hyper-rectangle and the objective is to maximize its volume. In other words, in KMR, the goal is to maximize some measure of a hyper-rectangle in the concentration and temperature space such that for every point in this hyper-rectangle the reduced kinetic model reproduces the full kinetic model within some given tolerance. This application will be discussed in more detail in Chapter 7.

Another class of applications that gives rise to GSIP problems originates from reformulating optimization problems with uncertain parameters as worst-case scenario design problems. We will revisit the example mentioned in section 1.1 to illustrate

this point. Consider the problem

$$\begin{aligned} &\min_{\mathbf{x}\in X} f(\mathbf{x}) \\ &\text{s.t. } g(\mathbf{x},\mathbf{p}) \leq 0 \\ &\qquad \mathbf{x} \in X, \end{aligned} \tag{4.5}$$

where, for example, f is the total production cost of a pharmaceutical, $\mathbf{x}$ are the decision variables and $\mathbf{p}$ are parameters the values of which are determined upstream in the production. If the parameters $\mathbf{p}$ are certain, then problem (4.5) is an ordinary finite nonlinear program. However, if the parameters are uncertain and can vary within a range, e.g., a n_p - dimensional interval $P = [\mathbf{p}^L, \mathbf{p}^U]$ and, furthermore, we want to ensure that the safety constraint g is satisfied for all $\mathbf{p} \in P$ (worst-case scenario design) then (4.5) is reformulated as SIP (1.1). If, furthermore, the uncertainty in the parameters is dependent on the optimization variables $\mathbf{x}$, e.g., the parameters are allowed to vary in the interval $P(\mathbf{x}) = [p_L(\mathbf{x}), p_U(\mathbf{x})]$ then the problem becomes generalized semi-infinite (4.1). Within the context of worst-case scenario design or robust optimization, a very interesting application is found in portfolio optimization [17, 120]. Assume that an investor has an initial capital of $\$C$ to invest in n shares. Furthermore, assume that the unit return of investment for share i, $1 \leq i \leq n$ is p_i. If the returns p_i were certain then the solution of the following optimization problem would guarantee maximum revenue:

$$\begin{aligned} &\max_{\mathbf{x}\in\mathbb{R}^n} \sum_{i=1}^{n} p_i x_i \\ &\text{s.t. } \sum_{i=1}^{n} x_i = C\,. \end{aligned}$$

However, if the returns p_i are uncertain and vary within an uncertainty set P then the

following problem provides a worst-case scenario estimate of the optimal portfolio:

$$\begin{aligned} &\max_{\mathbf{x}\in\mathbb{R}^n} \min_{\mathbf{p}\in P} \sum_{i=1}^{n} p_i x_i \\ &\text{s.t. } \sum_{i=1}^{n} x_i = C\,. \end{aligned} \tag{4.6}$$

Problem (4.6) can be reformulated as the following SIP:

$$\begin{aligned} &\max_{\mathbf{x}\in\mathbb{R}^n,\, z\in\mathbb{R}} z \\ &\text{s.t. } z - \sum_{i=1}^{n} p_i x_i \leq 0,\ \forall \mathbf{p} \in P \\ &\qquad \sum_{i=1}^{n} x_i = C. \end{aligned}$$

If the uncertainty, also known as the volatility, of the returns p_i, also depends on the initial investments x_i then the worst-case scenario optimal portfolio is given by the following generalized semi-infinite program:

$$\begin{aligned} &\max_{\mathbf{x}\in\mathbb{R}^n,\, z\in\mathbb{R}} z \\ &\text{s.t. } z - \sum_{i=1}^{n} p_i x_i \leq 0,\ \forall \mathbf{p} \in P(\mathbf{x}) \\ &\qquad \sum_{i=1}^{n} x_i = C. \end{aligned}$$

We refer the reader to [17, 120] for the original source, a more detailed description and the consideration of various uncertainty sets $P(\mathbf{x})$.

4.2 Definitions

Below are some necessary definitions for the analysis of the test set:

Definition 4.1. (General Form of GSIP).

Recall, that the general formulation of GSIP that we are attempting to solve is:

$$\begin{aligned} &\inf_{\mathbf{x}\in X} f(\mathbf{x}) \\ &\text{s.t. } g(\mathbf{x},\mathbf{p}) \leq 0, \forall \mathbf{p} \in P(\mathbf{x}) \\ &P(\mathbf{x}) = \{\mathbf{p} \in D : u_j(\mathbf{x},\mathbf{p}) \leq 0,\ \forall j \in J\}. \end{aligned} \tag{4.7}$$

Therefore, a GSIP will be completely defined by specifying f, g, $\mathbf{u}$, X and D.

Definition 4.2. (Lower-Level Problem)
For a given $\bar{\mathbf{x}} \in X$ the lower-level problem is defined as:

$$\begin{aligned} O(\bar{\mathbf{x}}, D) = \max_{\mathbf{p}\in D} & g(\bar{\mathbf{x}},\mathbf{p}) \\ s.t.\ & u_j(\bar{\mathbf{x}},\mathbf{p}) \leq 0, \forall j \in J. \end{aligned}$$

For $\bar{\mathbf{x}} \in X$ for which the lower-level problem is infeasible, then by definition $O(\bar{\mathbf{x}}, D) = -\infty$. The feasible set of the GSIP, F, is defined as:

$$F = \{\mathbf{x} \in X \mid O(\mathbf{x}, D) \leq 0\}.$$

Definition 4.3. (Upper-level Problem).
For a given $\bar{\mathbf{p}} \in D$ the upper-level problem is:

$$\begin{aligned} &\max_{\mathbf{x}\in X} f(\mathbf{x}) \\ &\text{s.t. } g(\mathbf{x},\bar{\mathbf{p}}) \leq 0. \end{aligned}$$

Therefore, when an upper-level problem is referred to as being convex at $\bar{\mathbf{p}} \in D$ it is implied that f is convex in $\mathbf{x} \in X$ and $g(\cdot, \bar{\mathbf{p}},)$ is convex on $\mathbf{x} \in X$.

Definition 4.4. (Lower-Level Feasible Set)
For a given $\bar{\mathbf{x}} \in X$ the lower-level feasible set is given by

$$P(\bar{\mathbf{x}}) = \{\mathbf{p} \in D \mid u_j(\bar{\mathbf{x}},\mathbf{p}) \leq 0, \forall j \in J\}.$$

Definition 4.5. (Joint Lower-Level Feasible Set)
The joint lower-level feasible set is defined as:

$$LL(X, D) = \{(\mathbf{x}, \mathbf{p}) : \mathbf{x} \in X,\ \mathbf{p} \in P(\mathbf{x})\}.$$

Definition 4.6. (Host Sets)
$X \subset \mathbb{R}^{n_x}$ and $D \subset \mathbb{R}^{n_p}$ are the host sets for the optimization variables and the parameters, respectively. J is the index set for the finite set of lower-level inequality constraints.

Definition 4.7. (Defining Functions)
f: $X \to \mathbb{R}$ is the objective function of the GSIP, g: $X \times D \to \mathbb{R}$ is the upper-level constraint while $u_j : X \times D \to \mathbb{R}$, $j \in J$, are the lower-level inequality constraints.

Definition 4.8. (Lower- and Upper-Level Feasibility)
A set $Q \subset X \times D$ is lower-level feasible if $Q \subset LL(X, D)$.
A set $Q \subset X \times D$ is upper-level feasible if $g(\mathbf{x}, \mathbf{p}) \leq 0$, $\forall (\mathbf{x}, \mathbf{p}) \in Q$.

Definition 4.9. (Active Index Set - GSIP Slater Point)
For a given $\bar{\mathbf{x}} \in F$, the index set of the active constraints is defined as

$$P_0(\bar{\mathbf{x}}) = \{\mathbf{p} \in D \mid g(\bar{\mathbf{x}}, \mathbf{p}) = 0,\ u_j(\bar{\mathbf{x}}, \mathbf{p}) \leq 0,\ \forall j \in J\}.$$

A point $\mathbf{x} \in F$ is a GSIP Slater point if $P_0(\mathbf{x}) = \emptyset$. Finally the set of Slater points of the GSIP is denoted by X_s.

Definition 4.10. (Global Solution - Points with Infimum Objective Function Value)
The global solution value of the GSIP is denoted by f^{GSIP}. The set of points $\mathbf{x}$ for which $f(\mathbf{x}) = f^{GSIP}$ are denoted by the set X_{inf}.

Definition 4.11. (Infeasible - Superoptimal Points)
The set of infeasible $\mathbf{x} \in X$ for which the objective function value is less than the infimum value of the GSIP is denoted by the set H:

$$H = \{\mathbf{x} \in X \mid f(\mathbf{x}) < f^{GSIP}\}.$$

4.3 Similarities Between SIP and GSIP

The major similarities between SIP and GSIP are:

1. Both problems involve the optimization of a finite number of decision variables subject to a, potentially, infinite number of constraints. In degenerate cases, SIP and GSIP can be transformed to finite nonlinear programs.

2. To determine feasibility of any point $\bar{\mathbf{x}}$, an optimization problem (lower-level problem) needs to be solved to global optimality. This is the key point in both SIP and GSIP, namely that to guarantee feasibility the global solution of an auxiliary problem is needed.

3. Both problems involve the optimization of one problem (lower-level problem) inside another optimization problem (outer problem). SIP and GSIP are, therefore, optimization problems with optimization problems embedded.

4. In contrast with finite programming, even if all the defining functions are affine and the problem is feasible, the feasible set of SIP and GSIP is, in general, not a polytope.

4.4 Differences Between SIP and GSIP

On the other hand, GSIP exhibit many differences compared to SIP. Specifically:

1. Four elements are required to fully define a SIP, including the objective function f, the semi-infinite constraint g, the host set of the decision variables X and the host set of the parameters D. To fully define a GSIP, five elements are required. The first four are the same as in SIP. The additional fifth is the vector of the lower-level inequality constraints u_j or, equivalently, the lower-level set-valued mapping $P : X \to D$.

2. In semi-infinite optimization the index set of the constraints is independent of the decision variables. In GSIP this set is dependent on the decision variables.

The explicit dependence of this index set is given by the lower-level constraints (or equivalently the lower-level set-mapping P). A way to illustrate this major difference is by asking the question: "If we were to sketch the set of points $(\mathbf{x}, \mathbf{p})$ for which the upper-stage constraint g should be introduced, how would this set look like?" In Figure 4-1 we provide an answer to this question. Assume that

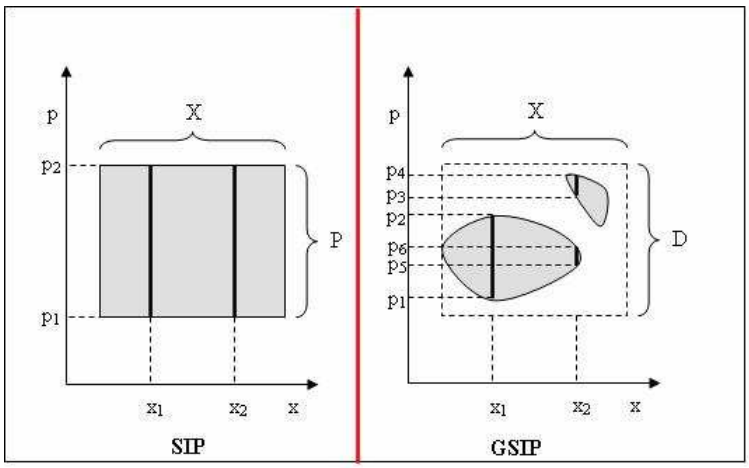

Figure 4-1: Uncertainty Set with respect to the Decision Variables

the dimensionality of the host sets X, P and D is 1. Then, for the semi-infinite case, for all $\mathbf{x} \in X$, the index set of the constraints is the same, i.e. $P(\mathbf{x}) = P = [p_1, p_2]$. However, for the generalized semi-infinite case, the index set of the constraints is not the same for each $\mathbf{x} \in X$. To illustrate this, consider two arbitrary points $\mathbf{x}_1$ and $\mathbf{x}_2$. The index set of the constraints at $\mathbf{x}_1$ and $\mathbf{x}_2$ are given by $P(\mathbf{x}_1) = [p_1, p_2]$ and $P(\mathbf{x}_2) = [p_5, p_6] \cup [p_3, p_4]$, respectively.

3. In semi-infinite programming, similar to finite programming, under mild assumptions there are two possible outcomes: either a problem is infeasible or it is feasible and a minimum exists. However, in generalized semi-infinite programming there is a third possible scenario; the problem may be feasible but the infimum is not attained [121]. Indeed, it does not seem possible to establish mild assumptions under which the infimum is attained for GSIP. We refer to Section 4.5 for more details.

4. If all the defining functions are affine, then the feasible set in semi-infinite

programming is convex. However, this result does not hold, in general, for generalized-semi infinite programming [106].

4.5 The Feasible Set in GSIP

The feasible set of GSIP problems can exhibit very unusual properties not usually encountered in finite or even ordinary semi-infinite optimization problems. We will outline the major aspects of this irregular behavior; we refer the reader to [106, 115, 121] for a more thorough analysis.

1. Even if all defining functions of a GSIP, i.e. f, g and u_j, $j \in J$, are continuous on X, $X \times D$ and $X \times D$ respectively and the host sets of the decision variables and parameters, X and D respectively, are compact, the feasible set of the GSIP is, in general, not closed. To illustrate this, consider the following example [65]:

$$\begin{aligned} &\min_{\mathbf{x}\in[-1,1]^2} x_2 \\ &\text{s.t. } -p^3 + x_2 \leq 0,\ \forall p \in P(\mathbf{x}) \\ &\qquad P(\mathbf{x}) = \{p \in [-1,0] : 2x_2 - p^3 - x_1^2 \leq 0\}. \end{aligned}$$

   The feasible set is given by $M = \{\mathbf{x} \in [-1,1]^2 : x_1^2 < 2x_2\} \cup \{0,0\}$ and is illustrated in Figure 4-2.

2. Similarly, even if the defining functions of the GSIP are continuous on their domains and the host sets X and D are compact, the infimum value of the GSIP is not necessarily attained. To illustrate this point consider the following example [121]:

$$\begin{aligned} &\min_{x\in[-1,1]} x^2 \\ &\text{s.t. } x - p \leq 0,\ \forall p \in P(x) \\ &\qquad P(x) = \{p \in [-1,1] : (p+1)^2 + x^2 \leq 0\}. \end{aligned}$$

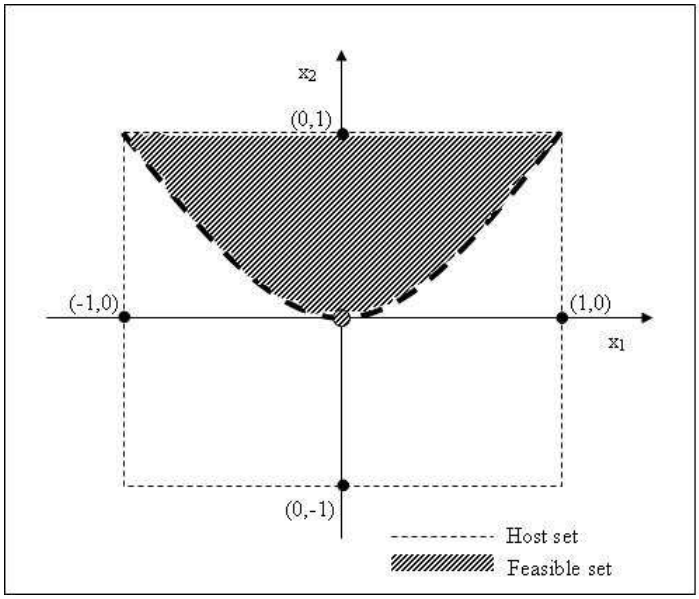

Figure 4-2: Non-closedness of the Feasible Set in GSIP

The feasible set is given by $F = \{x \in [-1,1] : x \neq 0\}$. Clearly, the infimum objective function value is 0 and attained for $\bar{x} = 0$. However, since $\bar{x}$ is not feasible the infimum the minimum of the GSIP does not exist while the problem is infeasible.

3. The feasible set of GSIP may contain *re-entrant corner points* which are spurious points for stationarity-based optimality conditions [64]. To illustrate this behavior we analyze the following GSIP from [106].

$$\begin{aligned}
&\min_{x\in[-1,1]^2} x_1 + x_2 \\
&\quad \text{s.t. } -p \leq 0,\ \forall p \in P(x) \\
&\qquad P(x) = \{p \in [-1,0] : u_j(x,p) \leq 0, j = 1,2\}. \\
&\quad u_1(x,p) = x_1 - p \\
&\quad u_2(\mathbf{x},\mathbf{p}) = x_2 - p.
\end{aligned}$$

The feasible set is $F = \{\mathbf{x} \in [-1,1]^2 : \max\{x_1, x_2\} \geq 0\}$. In [106] it is shown that although $\bar{\mathbf{x}} = \{0,0\}$ is not a local minimum of the GSIP it satisfies the first order optimality condition presented in [64]. Figure 4-3 illustrates the feasible

set of this problem and the re-entrant corner point at the origin.

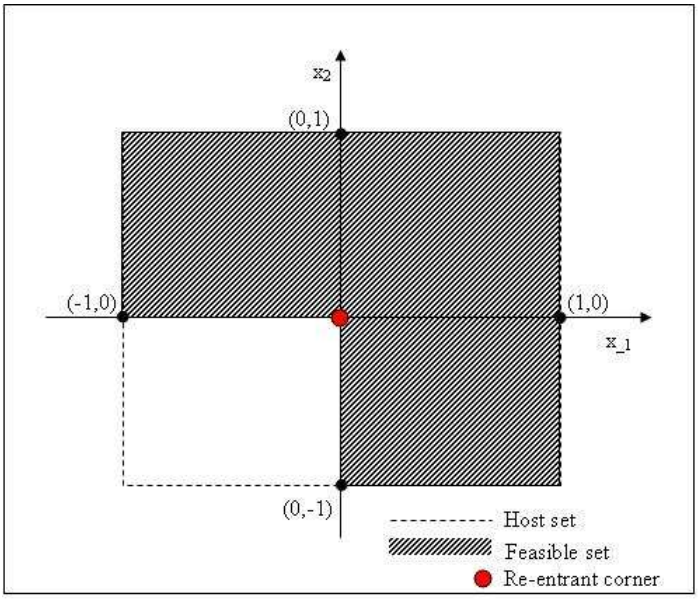

Figure 4-3: Re-entrant Corner Points in GSIP

4. Even if all the defining functions are affine, the feasible set of GSIP is not convex, in general. To illustrate this consider the example that was analyzed above. The feasible region shown in Figure 4-3 is nonconvex.

In general, GSIP exhibit irregular behavior, such as nonclosed feasible sets, infimum values not being attained, re-entrant corners and nonconvex feasible sets even when all the defining functions are affine and the host sets compact. For a detailed analysis on first and second order optimality conditions and on the topological structure of the feasible set in GSIP we refer the reader to [60, 65, 104, 105, 113, 114, 115, 116, 118, 134, 136, 137].

4.6 Global Optimization Methods

Similar to semi-infinite programming, in generalized semi-infinite programming there are two optimization problems to be solved:

1. The feasibility or lower-level problem (see Definition 4.3). The global solution value, or a valid upper bound on the global solution value, of this problem is

required in order to determine feasibility of any point $\mathbf{x}$. A local, KKT, or stationary approach for the lower-level problem cannot guarantee feasibility for a GSIP with an arbitrary structure (i.e., the lower-level problem is nonconvex).

2. The outer problem. This problem involves the minimization of the objective function subject to the feasible set. In contrast to the lower-level problem, a local-, global-, stationary- or KKT-based approach could be devised.

Based on this principle, conceptual and implementable global optimization algorithms for GSIP have been suggested in [45, 78, 120, 122, 137].

In [45] a conceptual method based on exact discontinuous penalization is proposed. The authors show that under mild assumptions on the defining functions and sets, the GSIP and a finite unconstrained problem (FP) possess the same minimizing sequences. While this result is quite strong theoretically it appears to suffer from certain practical drawbacks. First of all, the objective function of the finite problem (FP) is discontinuous and standard solvers cannot be expected to solve this problem globally. Therefore, upon finite termination, a certificate of ε-global optimality cannot be provided. Secondly, the objective function of the equivalent finite problem contains a term that is defined as the global solution value of an auxiliary optimization problem. The host set of the decision variables for this auxiliary problem is given by the solution of a second auxiliary problem. Clearly, this set can be nonconvex since the second auxiliary problem need not be convex. Therefore, for a single evaluation of the objective function of the finite program two optimization problems need to be solved globally and a nonconvex set needs to be calculated exactly. In conclusion, while this approach offers a strong theoretical result it seems incapable of providing a practical numerical procedure and to guarantee global optimality, even conceptually, upon finite termination.

In [78] a branch-and-bound approach for a specialized class of GSIP is developed. The authors assume that the upper-level constraint is concave in the parameters and the lower-level constrains are linear in the parameters, quadratic in the decision variables and separable in the decision variables and parameters. Under these assump-

tions, strong duality on the lower-level problem reduces the GSIP to an equivalent single-stage nonconvex optimization problem. Levitin and Tichatschke [78] propose the use of branch-and-bound to solve this problem globally. On the one hand, [78] proposes the first implementable global optimization algorithm for GSIP. On the other hand, the assumptions on the structure of the defining functions (convexity, linearity and separability) are very strong and are satisfied only in a limited number of engineering applications.

Stein and Still [119] have demonstrated that GSIP can be exactly transformed to bilevel programs under the assumption that the lower-level problem is feasible for all $\mathbf{x} \in X$. Using this transformation, Stein and Still [120] provide a method to further transform GSIP with convex lower-level problems to equivalent single-level nonlinear programs. In order to make this transformation the lower-level problem is replaced by its necessary and sufficient first-order optimality conditions. To our best knowledge [120] contains the first numerical results for GSIP while the algorithm has also been implemented for the diamond cutting problem by Winterfeld [138]. On the other hand, the authors assume that the lower-level problem is feasible and convex for all $\mathbf{x} \in X$. While feasibility of the lower-level is a reasonable assumption for engineering applications, convexity of the lower-level problem is restrictive for the problem of kinetic model reduction, the main application that we are targeting with this thesis.

In [122] a conceptual method based on discretization is proposed. Still [122] emphasizes the difficulty in generalizing discretization from SIP to GSIP. Specifically, as illustrated in Figure 4-1, because the index set of the constraints is different for each value of the decision variables it follows that a uniform grid on the host of the parameters does not provide a relaxation nor a restriction of the original GSIP. To alleviate the problem of non-closedness of the feasible set, the author assumes that the lower-level set-valued mapping is continuous and the host set of decision variables is compact. Therefore, the feasible set of the GSIP is compact and the minimum of the GSIP exists (infimum value is attained) [137]. To prove convergence of the discretization procedure, the author assumes that the discretization grids are given by continuous functions and therefore reduces the GSIP to a finite problem

the constraints of which are implicitly defined with respect to the decision variables. Still [122] proposes a conceptual method for constructing these continuous functions, however the method relies on sampling the parameter set and the construction of local linearizations around points in the decision variable space. Overall, we believe that the first attempt to generalize discretization from SIP to GSIP is developed in [122]. However, the method proposed in [122] cannot guarantee the generation of feasible points nor provide a certificate of global optimality on finite termination. Furthermore, it appears that the method attempts to create a relaxation of the GSIP based on discretization but the computation of the continuous functions defining the discretization based on local linearization may violate this property.

Weber [137] proposes a conceptual global optimization method for GSIP. With the standard assumption that the lower-level set-valued mapping is well-behaved, the author shows that GSIP can be transformed to SIP, globally in the decision-variable space, using diffeomorphisms. Therefore, any global optimization algorithm devised for SIP with continuously differentiable functions could be applied to solve the transformed problem. The main drawback with this approach is that the semi-infinite constraints that correspond to the equivalent SIP are implicitly defined. Therefore, this method does not seem to provide a practical numerical approach for GSIP.

4.7 Limitations in the GSIP Literature

In summary, the analysis of Section 4.6 leads to the following conclusions:

1. For GSIP with nonconvex defining functions and/or with lower-level set-valued mappings that are not lower-semi continuous (well-behaved) there is currently no method to provide guaranteed feasible points nor a ε-certificate of global optimality on finite termination.

2. Either methods will assume a special structure of the defining functions (convexity and/or linearity) and lower-semi continuity of the lower-level set-valued mapping or they will avoid these assumptions and be conceptual.

In the following chapter we will attempt to develop a global optimization algorithm that will avoid any special assumption on the structure of the defining functions, such as convexity, and will also avoid the assumption of lower semi-continuity of the set-valued mapping.

Chapter 5

Global Solution of GSIP using Interval Methods

5.1 Introduction

The goal of this chapter is to present a global optimization method for GSIP that avoids the requirement of convexity for any of the participating functions and furthermore does not require the inner problem to be feasible for all values of $\mathbf{x}$. In avoiding such requirements, we will attempt to solve nonconvex GSIP for which the infimum is not attained. We refer the reader to [76] for the original contribution.

In Section 5.2 we outline the definitions and assumptions that are necessary for convergence of our algorithm, while in Section 5.3 we present the general properties and the specific steps of the global optimization procedure. In Section 5.4 we comment on the nature of the approach and in Section 5.5 we establish the convergence of the algorithm. Finally, in Section 5.6 we present the major criteria for creating the test set, the implementation details of our algorithm and the numerical results on the test set.

5.2 Definitions and Assumptions

The following definitions and assumptions are necessary to present the global optimization procedure, its theoretical convergence and also to analyze the numerical results from the application of the algorithm to the test set.

5.2.1 Definitions

Definition 5.1. (Categorization of X_{inf})
The set X_{inf} can be categorized in the following way:

$$X_{inf} = X_{inf,1} \cup X_{inf,2}$$
$$X_{inf,1} = \{\bar{x} \in X_{inf} : \exists \varepsilon > 0 \text{ such that } \forall x \in N_\varepsilon(\bar{x}), x \text{ is infeasible}\}$$
$$X_{inf,2} = X_{inf} \setminus X_{inf,1}.$$

Definition 5.2. (B&B levels for GSIP Slater Points)
For fixed $\bar{\mathbf{x}} \in X_{inf}$ and for every $\mathbf{x} \in X_s$, assign $q^1_{\mathbf{x}}$ to be the earliest level of the branch-and-bound tree for which $\mathbf{x}$ is found feasible in the upper bounding operation. Call $q^2_{\mathbf{x},\bar{\mathbf{x}}}$ the earliest level of the branch-and-bound tree for which $\mathbf{x}$ and $\bar{\mathbf{x}}$ do not belong to the same node.

We will prove that $q^1_{\mathbf{x}}$ and $q^2_{\mathbf{x}}$ are well defined.

Definition 5.3. (Interval Extensions)
$U_j : \mathbb{IR}^{n_x} \times \mathbb{IR}^{n_p} \to \mathbb{IR}$, $\forall j \in J$, are interval-valued functions and refer to interval extensions of the lower-level constraints with respect to both the optimization variables $\mathbf{x}$ and the parameters $\mathbf{p}$. Specifically, if X and D are intervals, $U_j(X, D) = [u^L_j(X, D), u^U_j(X, D)]$, where u^L_j and u^U_j are real-valued functions.
$G : X \times \mathbb{IR}^{n_p} \to \mathbb{IR}$ is an interval-valued function that refers to an interval extension of $g(\mathbf{x}, \mathbf{p})$ with respect $\mathbf{p}$. If D is an interval, then $G(\mathbf{x}, D) = [g^L(\mathbf{x}, D)\ ,\ g^U(\mathbf{x}, D)]$, where g^L and g^U are real-valued functions.
Throughout the algorithm, natural interval extensions are employed. In [85] it is shown that natural interval extensions are continuous.

We refer to [26, 55, 85, 98] for notation and a detailed theoretical analysis of interval methods and their applications to global optimization.

5.2.2 Assumptions

Assumption 5.4. (Nature of Host Sets)
The decision-variable and parameter host sets, $X = [x^L, x^U]$ and $D = [p^L, p^U]$, respectively, are n_x- and n_p- dimensional intervals respectively.

Assumption 5.5. (Continuity & Differentiability of Defining Functions)
f is continuously differentiable on an open set $X' \supset X$, g is continuously differentiable on X' for each $p \in D$ and continuous on D for each $x \in X$. Finally u_j is continuous on X for each $p \in D$ and continuous on D for each $x \in X$, for each $j \in J$.

Assumption 5.6. (Convergence of Lower Bound)
For every $x \in H$, there must exist $p \in D$ for which $g(x,p) > 0$ and $u_j(x,p) < 0$, $\forall j \in J$.

Assumption 5.7. (Convergence of Upper Bound - Breadth First Search)
There exists $x' \in X_{inf}$ for which there exists a sequence of GSIP Slater points $\{x_n\}$ satisfying: $\lim_{n\to\infty} x_n = x'$.

Assumption 5.8. (Convergence of Upper Bound - Best Bound Search)

1. *For every $x \in X_{inf,1}$, there exists $p \in D$, for which $g(x,p) > 0$ and $u_j(x,p) < 0$, $\forall j \in J$.*

2. *For every $x' \in X_{inf,2}$ there exists a sequence of GSIP Slater points $\{x_n\}$ for which:*

$$\lim_{n\to\infty} x_n = x' \quad \textit{and} \quad q^1_{x_n} \leq q^2_{x_n,x'}, \forall n.$$

5.3 Description of Branch-and-Bound Algorithm

The present work concerns generalized semi-infinite programs of the form:

$$\begin{aligned} &\inf_{x \in X \subset \mathbb{R}^{n_x}} \; f(x) \\ &\text{s.t. } g(x,p) \leq 0, \; \forall p \in P(x) \\ &P(x) = \{p \in D \subset \mathbb{R}^{n_p} : u_j(x,p) \leq 0, \; \forall j \in J\}. \end{aligned} \tag{5.1}$$

The branch-and-bound procedure by which convergence to the GSIP infimum is achieved will now be described. Two different node selection heuristics are presented: breadth-first and best-bound search. In the convergence proof and in the numerical results it is shown that while the breadth-first search requires weaker theoretical assumptions, it is typically more computationally expensive than the best-bound approach.

5.3.1 General Properties

The general properties of the algorithm are the following:

1. (a) For the case of breadth-first search, all the nodes at a specific level (except the ones that are fathomed due to infeasibility or value dominance) are examined before moving on to the next level of the B&B tree.

   (b) In the case of the best bound approach, the node selected for branching and bounding will be amongst those with the lowest lower bound (best bound).

2. Associated with each level of the B&B tree q is the index set S_q of uniform subdivisions of the parameter set D defined as $S_q = \{1, 2, \ldots, 2^q\}^{n_p}$ and also the set of grid points T_q which are chosen to be the top right-hand corner points of the subintervals indexed by S_q. Each dimension j of D is subdivided into 2^q subintervals of equal width:

$$D_j^k = [p_j^L + \frac{(k-1)\,\omega(P_j)}{2^q}, \; p_j^L + \frac{k\omega(P_j)}{2^q}], \; \forall k = 1, \ldots, 2^q.$$

Now, there exists $\tau = (k_1, \ldots, k_{n_p}) \in S_q$ such that:

$$P_\tau = [p_1^L + \frac{(k_1 - 1)\,\omega(P_1)}{2^q},\; p_1^L + \frac{k_1\omega(P_1)}{2^q}] \times \ldots \\ \times [p_{n_p}^L + \frac{(k_{n_p} - 1)\,\omega(P_{n_p})}{2^q},\; p_{n_p}^L + \frac{k_{n_p}\omega(P_{n_p})}{2^q}]. \tag{5.2}$$

In compact notation, relation (5.2) can be written as:

$$P_\tau = D_1^{k_1} \times \ldots \times D_{n_p}^{k_{n_p}}.$$

3. Associated with node $M \subset X$, located at level q of the B&B tree, are the following, level-dependent, sets:

   (a) $S_{M,P_{pos}} \subset S_q$, defined as the subset of S_q for which it has been established that $\forall \tau \in S_{M,P_{pos}}$, $u_j(x,p) \leq 0$, $\forall (j,x,p) \in J \times M \times P_\tau$,

   (b) $S_{M,P_{neg}} \subset S_q$, defined as the subset of S_q for which it has been established that $\forall \tau \in S_{M,P_{neg}}$, $\exists j(\tau) \in J$ for which $u_j(x,p) > 0$, $\forall (x,p) \in M \times P_\tau$,

   (c) $S_{M,P_{uns}} \subset S_q$, defined as the subset of S_q for which neither of the above conclusions has been reached yet.

4. The iteration k is used to indicate the set of active nodes I_k. The level q is used to indicate the index set S_q and the set of grid points T_q.

   (a) For the breadth-first search, the iteration number k and the level of the branch-and-bound tree q increase monotonically. This is because each level of the B&B tree is examined before the next level is considered.

   (b) For the best bound approach, while the iteration number, by construction, will increase monotonically, this is not the case with the level of the B&B tree which may oscillate (increase/decrease) in the search for the best bound.

5. Each node M in the B&B tree has a unique nonnegative integer $\lambda(M)$ that is assigned to it when the node is created. Whenever the notion of *a sequence of*

nested nodes is encountered the notation of $\{M_{\lambda_q}\}$ is used. Thus, while $\{M_\lambda\}$ denotes the sequence of all nodes that are created in the $B\&B$ tree, $\{M_{\lambda_q}\}$, $q = 1, 2, \ldots$ refers to a subsequence, in the form of a sequence of nested nodes and obviously M_{λ_1} is located at the level above M_{λ_2} and $M_{\lambda_2} \subset M_{\lambda_1}$, etc. Whenever a single node is isolated (e.g., in the convergence proof for the upper bound) a superscript notation on the node M is used. Thus, it will be stated, for instance, there exists a node M^1 that satisfies a given property.

6. Two more nonnegative integers will be associated with each node. $q(M)$ denotes the level of the tree at which M is located (the same q which is used for the subscript of the sequence of nested nodes) and $k(M)$ denotes the iteration number of the algorithm at which node M was created. Obviously, these two integers are not unique to a particular node because at level q of the B&B tree there are 2^q nodes and at iteration k either no node will be created (fathoming) or two child nodes will be created sharing the same iteration number.

5.3.2 Detailed Algorithm

1. **Initialization**. Set $k = 0$, $\lambda = 1$, $S_{X,P_{pos}} = S_{X,P_{neg}} = \emptyset$, $S_{X,P_{uns}} = S_0$, $S_{-1} = S_0$, $I_0 = I_1 = \{X\}$, $\alpha_0 = \alpha_1 = +\infty$, $\beta_0 = \beta_1 = -\infty$, $q(X) = 0$, $\lambda(X) = 1$, $k(X) = 1$, $f_X^{LBD} = -\infty$, $f_X^{UBD} = +\infty$.

2. **Termination Test**. Delete from I_k all nodes M for which $f_M^{LBD} \geq \alpha_k$. If $\alpha_k - \beta_k \leq \varepsilon$ or $I_k = \emptyset$ then terminate. If $\alpha_k = +\infty$ then the instance is infeasible. Otherwise, α_k is an $\varepsilon-$optimal estimate of the solution value, and $\mathbf{x}^{GSIP}$ is a feasible point for the GSIP (1) at which α_k is attained.

3. **Node Selection**. Set $k = k + 1$. Depending on the node selection heuristic consider the following options:

   (a) Best Bound: select the node $M \in I_k$ for which:

$$M \in \arg\min_{M \in V_1} \lambda(M), \;\; V_1 := \arg\min_{M \in I_k} f_M^{LBD}.$$

(b) Breadth-First Search: select the node $M \in I_k$ for which:

$$M \in \arg\min_{M \in V_2} \lambda(M), \;\; V_2 := \arg\min_{M \in I_k} q(M).$$

Remove M from I_k.

4. **Lower Level Calculations**. Let $q = q(M)$. Set $S_{M,P_{pos}} = S_{M,P_{neg}} = S_{M,P_{uns}} = \emptyset$. $\forall \tau \in S_q$ there exist $M^* \in I_{k-1}$ and $P_{\tau'}$, $\tau' \in S_{q-1}$ for which $M \subset M^*$ and $P_\tau \subset P_{\tau'}$, respectively. Then:

   (a) if $\tau' \in S_{M^*,P_{pos}}$ then $S_{M,P_{pos}} = S_{M,P_{pos}} \cup \tau$,

   (b) if $\tau' \in S_{M^*,P_{neg}}$ then $S_{M,P_{neg}} = S_{M,P_{neg}} \cup \tau$,

   (c) if $\tau' \in S_{M^*,P_{uns}}$ then $S_{M,P_{uns}} = S_{M,P_{uns}} \cup \tau$.

   Evaluate an interval extension of u_j, $\forall j \in J$, on $M \times P_\tau$ for each P_τ for which $\tau \in S_{M,P_{uns}}$.

   (a) If $u_j^U(M, P_\tau) \leq 0, \;\; \forall j \in J$, then $S_{M,P_{pos}} = S_{M,P_{pos}} \cup \tau$ and $S_{M,P_{uns}} = S_{M,P_{uns}} - \tau$.

   (b) Else if $u_j^L(M, P_\tau) > 0$ for some $j \in J$, then $S_{M,P_{neg}} = S_{M,P_{neg}} \cup \tau$ and $S_{M,P_{uns}} = S_{M,P_{uns}} - \tau$.

5. **Lower bounding problem**. The solution value of the following semi-infinite program (SIP):

$$\begin{aligned} &\min_{x \in M} f(x) \\ &\text{s.t. } g(x,p) \leq 0, \forall p \in P_\tau, \; \forall \tau \in S_{M,P_{pos}} \end{aligned} \tag{5.3}$$

   provides a lower bound for (5.1) restricted to M (See Lemma 5.11). Instead of solving (5.3) to global optimality, a convex relaxation of (5.3) using discretiza-

tion (i.e., with a finite number of constraints) is solved instead:

$$\begin{aligned} &\min_{x\in M} f_{mc}(x) \\ &\text{s.t. } g_{mc}(x,p) \leq 0, \forall p \in P_\tau \cap T_q,\ \forall \tau \in S_{M,P_{pos}}. \end{aligned} \tag{5.4}$$

where f_{mc}, $g_{mc}(\cdot, p)$ are convex relaxations, in the sense of McCormick, of f and $g(\cdot, p)$, respectively, on M. (5.4) can be solved to guaranteed global optimality with convex NLP solvers. Assign to f_M^{LBD} the solution value of problem (5.4) if it is feasible. Otherwise set $f_M^{LBD} = +\infty$. Set $\beta_k = \min\limits_{M\in I_k} f_M^{LBD}$.

6. **Fathoming**. If $f_M^{LBD} = +\infty$ (fathoming by infeasibility) or $f_M^{LBD} \geq \alpha_{k-1}$ (fathoming by value dominance) then set $\alpha_k = \alpha_{k-1}$ and go to step 2.

7. **Upper Bounding Problem**. Any feasible point of the following SIP:

$$\begin{aligned} &\min_{x\in M} f(x) \\ &\text{s.t. } g(x,p) \leq 0, \forall p \in P_\tau,\ \forall \tau \in S_{M,P_{pos}} \cup S_{M,P_{uns}} \end{aligned} \tag{5.5}$$

provides an upper bound for (1) (see Theorem 5.22). Instead of solving (5.5) to feasibility, an upper bound is generated using the interval constrained reformulation introduced in [26]:

$$\begin{aligned} &\min_{\mathbf{x}\in M} f(x) \\ &\text{s.t. } g^U(x,P_\tau) \leq 0, \forall \tau \in S_{M,P_{pos}} \cup S_{M,P_{uns}}. \end{aligned} \tag{5.6}$$

If a feasible point $\bar{x}$ is found, assign $f_M^{UBD} = f(\bar{x})$, otherwise set $f_M^{UBD} = +\infty$. Assign $\alpha_k = \min\{f_M^{UBD}, \alpha_{k-1}\}$. If $\alpha_k = f_M^{UBD}$ then set $x^{GSIP} = \bar{x}$. For instance, a local minimum found by a local NLP solver can supply this feasible point.

8. **Branching**. Recall that n_x is the dimension of X and $x_j^U - x_j^L$, $j = 1, \ldots, n_x$, is the width of coordinate j. Bisect M along the coordinate that maximizes

$x_j^U - x_j^L$. Specifically, let node $M \in I_k$ be the following n_x-dimensional interval:

$$M = [x_1^L\ ,\ x_1^U] \times \ldots \times [x_{j-1}^L\ ,\ x_{j-1}^U] \times [x_j^L\ ,\ x_j^U] \times [x_{j+1}^L\ ,\ x_{j+1}^U] \times \ldots \times [x_{n_x}^L\ ,\ x_{n_x}^U].$$

Based on the bisection process described above and assuming that $x_j^U - x_j^L = \max\limits_{1 \le i \le n_x} x_i^U - x_i^L$ then the coordinate j is bisected and the nodes created at level $q+1$ of the B&B tree as a result will be:

$$M_L = [x_1^L\ ,\ x_1^U] \times \ldots \times [x_j^L\ ,\ \tfrac{(x_j^U + x_j^L)}{2}] \times \ldots \times [x_{n_x}^L\ ,\ x_{n_x}^U].$$
$$M_U = [x_1^L\ ,\ x_1^U] \times \ldots \times [\tfrac{(x_j^U + x_j^L)}{2}\ ,\ x_j^U] \times \ldots \times [x_{n_x}^L\ ,\ x_{n_x}^U].$$

Set $q(M_L) = q(M_U) = q(M)+1$, $\lambda(M_L) = \lambda+1$, $\lambda(M_U) = \lambda+2$, $k(M_L) = k(M_U) = k$. $f_{M_L}^{LBD} = f_{M_U}^{LBD} = f_M^{LBD}$. Set $\lambda = \lambda + 2$. Finally, $I_{k+1} = (I_k \cup M_L \cup M_U)$. Go to step 2.

5.4 Nature of the Algorithm

5.4.1 Set Description

Consider the sets Q and R that satisfy:

$$Q \subset P(x),\ \forall x \in X$$
$$P(x) \subset R,\ \forall x \in X.$$

1. Q is a restriction of the lower-level feasible set for all $x \in X$. The following SIP

$$\begin{aligned} &\min_{x \in X} f(x) \\ &\text{s.t. } g(x,p) \le 0, \forall p \in Q, \end{aligned} \tag{5.7}$$

provides a relaxation of the original GSIP. Therefore, a restriction of the lower-level feasible set, and therefore of the lower-level problem, pointwise in X, leads to a relaxation of the outer problem. Q is obtained using interval extensions on the lower-level inequality constraints. The resulting SIP is further relaxed

using discretization and convexification.

2. R is a relaxation of the lower-level feasible set for all $x \in X$. The following SIP

$$\begin{aligned} &\min_{x \in X} f(x) \\ &\text{s.t. } g(x,p) \leq 0, \forall p \in R, \end{aligned} \tag{5.8}$$

provides a restriction of the original GSIP. Therefore, a relaxation of the lower-level feasible set, and therefore of the lower-level problem, pointwise in X, leads to a restriction of the outer problem. R is obtained using interval extensions on the lower-level constraints. The resulting SIP is further restricted using the interval-constrained reformulation.

Mitsos et al. [83] have used the notion of relaxing and restricting the lower-level problem in order to restrict and relax the outer program, respectively, for the global solution of bilevel programs.

If Q and R are compact then either the minimum of (5.7) and (5.8) will exist or (5.7) and (5.8) will be infeasible, even if the infimum of the original GSIP is not attained.

5.4.2 Comparison with methods using Optimality Conditions

It should be noted that the proposed approach does not make use of any optimality conditions for GSIP. In avoiding these optimality conditions, problems that involve a constraint qualification violation in the lower-level problem, problems with a non-closed feasible set and also problems which exhibit re-entrant corner points can be tackled. Specifically, in any optimality condition, a local or global minimum is a feasible point of the GSIP. Therefore, the optimality conditions are not applicable to GSIP for which the infimum is not attained. To bypass this difficulty, methods that rely on optimality conditions make the assumption that the lower-level problem is feasible for all $x \in X$. This assumption ensures that the infimum of the GSIP is attained. Furthermore, re-entrant corner points are known to be spurious points for most optimality conditions and therefore, algorithms depending on these conditions

could terminate upon finding such points. Our approach bypasses all these difficulties by using interval extensions on the lower-level functions so that our method can get arbitrarily close to spurious points such as infeasible points with the infimum objective function value or re-entrant corner points. In the latter case, since re-entrant corner points can only be strictly suboptimal for the GSIP, they will be fathomed using the principle of fathoming by value dominance within the B&B procedure.

5.5 Convergence of the B&B scheme

Prior to introducing the convergence proof, a brief outline is provided. First of all, the property that for each node examined in the branch-and-bound tree, the lower- and upper-bounding methodologies that were described in the algorithm (See Section 5.3) provide a valid lower and upper bound, respectively, on the optimal solution value of the GSIP at each node, will be proved. Then, the notions of deletion-by-infeasibility being semi-certain in the limit and of the bounding operation being semi-consistent will be demonstrated. Furthermore, the convergence of the lower-bounding operation to the infimum of the GSIP will be shown. Next, auxiliary lemmas for the convergence of the upper-bounding operation will be provided. These lemmas refer to the existence of a well-behaved neighborhood around a GSIP Slater point and to the uniform nature of the subdivisions of the parameter host set D. Last, the finite ε-convergence of the upper-bounding methodology to the infimum of the GSIP for both the breadth-first and the best-bound node selection heuristics is demonstrated.

Lemma 5.9. *The upper and lower bounding problems that were described in Section 5.3.2 provide a valid upper and lower bound, respectively, to the global solution value of the GSIP, f^{GSIP}.*

Proof. Consider any node M in the branch-and-bound tree and the GSIP restricted

to M, i.e., the problem:

$$\begin{aligned} f_M^{GSIP} &= \inf_{x \in M} f(x) \\ \text{s.t. } & g(x,p) \leq 0, \forall\, p \in P(x) \\ P(x) &= \{p \in D : u_j(x,p) \leq 0,\ \forall j \in J\}. \end{aligned} \tag{5.9}$$

Let $f_{M,U}^{SIP}$ be the solution value of (5.5). For the upper-bounding problem, for every $x \in M$, $P(x) \subset \bigcup_{\tau \in S'} P_\tau$, $S' = S_{M,P_{pos}} \cup S_{M,P_{uns}}$. This implies that (5.5) is a restriction of (5.9) and therefore:

$$f_M^{GSIP} \leq f_{M,U}^{SIP}. \tag{5.10}$$

Let f_M^{UBD} be the solution value of (5.6). From [26] and [27] is it known that (5.6) is a restriction of (5.5). Thus:

$$f_{M,U}^{SIP} \leq f_M^{UBD}. \tag{5.11}$$

Furthermore, (5.9) is exactly the same formulation as (5.1) on a subset of X. Clearly,

$$f^{GSIP} \leq f_M^{GSIP}. \tag{5.12}$$

Finally, from (5.10), (5.11) and (5.12):

$$f^{GSIP} \leq f_M^{UBD}.$$

Furthermore, if the infimum of (5.1) is not attained, then exactly one of (5.10) or (5.12) would hold as a strict inequality (if M contains one of the points with infimum objective function value then (5.10) is satisfied as a strict inequality, because SIP always attain a minimum if they are feasible, while if not then (5.12) would be satisfied as a strict inequality) and thus:

$$f^{GSIP} < f_M^{UBD}.$$

Thus, the upper bounding methodology described in section 2.2 provides an upper

bound to the global solution value of the original GSIP.

Let $f_{M,L}^{SIP}$ be the solution value of (5.3). For the lower-bounding problem, for every $\mathbf{x} \in M$, $P(x) \supset \bigcup_{\tau \in S'} P_\tau$, $S' = S_{M,P_{pos}}$. This implies that (5.3) is a relaxation of (5.9) and therefore:

$$f_{M,L}^{SIP} \leq f_M^{GSIP}. \tag{5.13}$$

Let f_M^{LBD} be the solution value of (5.4). By construction, (5.4) is a relaxation of (5.3). Furthermore by convexity, (5.4) can be solved reliably using local solvers to obtain f_M^{LBD}. This implies that:

$$f_M^{LBD} \leq f_{M,L}^{SIP}. \tag{5.14}$$

Combining (5.13) and (5.14), at node M a solution value f_M^{LBD} is obtained which satisfies:

$$f_M^{LBD} \leq f_M^{GSIP}. \tag{5.15}$$

At any given iteration k, the incumbent lower bound is the minimum over all lower bounding values for the active nodes I_k. Consider the set of nodes I_k' for which for every $M \in I_k'$, M was deleted either by infeasibility or by value dominance at one of the iterations $0,\ldots,k-1$. Clearly,

$$\bigcup_{M \in I_k \cup I_k'} M = X,\ \forall k \in \mathbb{N}. \tag{5.16}$$

Therefore, combining (5.15) and (5.16):

$$\min_{M \in I_k \cup I_k'} f_M^{LBD} \leq f^{GSIP}. \tag{5.17}$$

Since it has already been shown that the upper-bounding methodology is valid:

$$f_M^{LBD} \geq f^{GSIP},\ \forall M \in I_k'.$$

Hence (5.17) is equivalent to:

$$\min_{M \in I_k} f_M^{LBD} \leq f^{GSIP}.$$

Therefore, the incumbent lower bound value is a lower bound to the infimum of the GSIP:

$$\beta_k = \min_{M \in I_k} f_M^{LBD} \leq \min_{M \in I_k} f_M^{GSIP} = f^{GSIP}.$$

□

5.5.1 Convergence of Lower-Bound

Definition 5.10. Let H be the set of infeasible-superoptimal points as given in Definition 4.11. The deletion-by-infeasibility rule will be called semi-certain in the limit if for every infinite sequence of nested nodes $\{M_{\lambda_q}\}$ with accumulation point $\{\bar{x}\}$, $\bar{x} \notin H$.

Lemma 5.11. *Under Assumption 5.6 the deletion-by-infeasibility rule is semi-certain in the limit.*

Proof. Consider an infinite sequence of nested nodes $\{M_{\lambda_q}\}$ with accumulation point $\bar{x}$ and further assume that $\bar{x} \in H$. From the assumptions of the Lemma, there exists $\bar{p} \in D : u_j(\bar{x}, \bar{p}) < 0, \forall\, j \in J$ and $g(\bar{x}, \bar{p}) > 0$. Consider a sequence of intervals X_n and P_n for which $\bar{x} \in X_n$, $\forall n \in \mathbb{N}$, $X_{n+1} \subset X_n$, $\lim_{n\to\infty} X_n = \{\bar{x}\}$, $\bar{p} \in P_n$, $\forall n \in \mathbb{N}$, $P_{n+1} \subset P_n$ and $\lim_{n\to\infty} P_n = \{\bar{p}\}$. From continuity of the natural interval extension $U_j(X, P)$ [85] and treating $\bar{x}$ and $\bar{p}$ as degenerate intervals: $\lim_{n\to\infty} u_j^U(X_n, P_n) = u_j^U(\bar{x}, \bar{p}) = u_j(\bar{x}, \bar{p})$. Thus, for each $j \in J$, there exists finite n_j^* such that $u_j^U(X_{n_j^*}, P_{n_j^*}) < 0$. Thus, there exists finite $n^* = \max_{j \in J} n_j^*$ for which $u_j^U(X_{n^*}, P_{n^*}) < 0, \forall\, j \in J$.

Since the branching procedure is exhaustive, there exists a level q^1 of the B&B tree for which node $M_{\lambda_{q^1}}$ containing $\bar{x}$ will satisfy $M_{\lambda_{q^1}} \subset X_{n^*}$. The subdivision procedure is such that there exists a level q^2 of the B&B tree and $\tau \in S_{q^2}$ for which $\bar{p} \subset P_\tau \subset P_{n^*}$. For level $q^3 = \max\{q^1, q^2\}$, there exists a node $M_{\lambda_{q^3}} \subset X_{n^*}$ containing $\bar{x}$ and $\tau' \in S_{q^3}$ for which $\bar{p} \subset P_{\tau'} \subset P_{n^*}$. From inclusion monotonicity of the interval

extensions, $u_j^U(M_{\lambda_{q^3}}, P_{\tau'}) \leq u_j^U(X_{n^*}, P_{n^*}) < 0$, $\forall\ j \in J$. This implies that for $q \geq q^3$ the lower bounding problems for the descendant nodes of $M_{\lambda_{q^3}}$ will consider $P_{\tau'}$ and its subdivisions as lower-level feasible.

By continuity of $g(\bar{x}, \cdot)$, there exists an open ball around $\bar{p}$ of radius δ, namely $N_\delta(\bar{p})$, for which $\forall p \in N_\delta(\bar{p})$, $g(\bar{x}, p) > 0$. Since the discretization of the parameter host set D is exhaustive, i.e. $\lim_{q\to\infty} \sup_{\mathbf{p}_1 \in T_q} \inf_{p_2 \in T_q} ||p_1 - p_2|| = 0$, the discretization of $P_{\tau'}$ is also exhaustive. This ensures that there exists a level $q^4 \geq q^3$ of the B&B tree for which a point $p' \in N_\delta(\bar{p})$ will be incorporated in the set of grid points T_{q^4}. From the subdivision heuristic (Step 2 of detailed algorithm) it can be easily shown that one constraint for the function g at p' will be present in the lower-bounding problem at every subsequent level. Let $g'(x)$ correspond to the constraint function at p'. From Lemma 1 in [27], $\lim_{q\to\infty} \min_{x \in M_{\lambda_q}} g'_{mc,q}(x) = g'(\bar{x})$ where $g'_{mc,q}$ is the McCormick relaxation for the constraint function g' on M_{λ_q}. The statement above along with the fact that $g'(\bar{x}) > 0$ imply that there exists some finite $q^5 \geq q^4$ for which $q > q^5$ implies that $\min_{x \in M_{\lambda_q}} g'_{mc,q}(x) > 0$. This finally implies that the lower bounding problem is infeasible for $\{M_{\lambda_q}\}$ with $q > q^5$ and node $M_{\lambda_{q^5+1}}$ containing $\bar{x}$ will be fathomed from the B&B tree. Thus, for an infinite sequence of nested nodes M_{λ_q} converging to $\bar{x}$, $\bar{x} \notin H$. $\square$

Definition 5.12. A lower bounding operation is called strongly semi-consistent if at every step any undeleted partition element can be further refined, and if any infinite sequence of nested nodes $\{M_{\lambda_q}\}$ satisfies

$$\bar{x} \notin H \ \ and \ \ \lim_{q\to\infty} \beta(M_{\lambda_q}) = f(\bar{x})$$

where $\bar{x} = \bigcap_q M_{\lambda_q}$ and $\beta(M_{\lambda_q})$ is the solution value of the lower-bounding problem at node M_{λ_q}.

Lemma 5.13. *The lower-bounding operation described in the B&B algorithm is strongly semi-consistent.*

Proof. From the rectangular partitioning of the variable space, any undeleted partition element can be further refined. Assume an infinite sequence of nested nodes

$\{M_{\lambda_q}\}$ with an accumulation point $\{\bar{x}\}$. From Lemma 5.11 it is known that $\bar{x} \notin H$. Furthermore, if an infinite sequence of nested nodes exists this implies that $\beta(M)$ exists for all $M \in \{M_{\lambda_q}\}$. Therefore, from Lemmas 1 and 3 in [27], $\lim_{q\to\infty} \beta(M_{\lambda_q}) = f(\bar{x})$. □

Definition 5.14. A node selection heuristic is said to be bound improving if, at least each time after a finite number of iterations, the partition element where the actual lower bound is attained is selected for further partitioning within the B&B tree.

Lemma 5.15. *The breadth-first and best-bound node selection heuristics are bound improving.*

Proof. Clearly, the best-bound node selection heuristic is bound improving because by construction, at each iteration a node with the lowest lower bound is chosen for further partitioning. The breadth-first node selection heuristic is also bound improving because the number of partition elements is always finite which assures that any partition element will be chosen for further partitioning after a finite number of steps. □

Theorem 5.16. *The lower bounding operation that was described in section 2.2 converges to the GSIP infimum value, i.e.,* $\beta = \lim_{k\to\infty} \beta_k = f^{GSIP}$.

Proof. The B&B procedure that has been described in Section 5.3.2 satisfies the following properties:

1. The subdivision of the partition sets is exhaustive,
2. The selection of the partition sets to be refined is bound improving (Lemma 5.15),
3. The lower bounding operation is strongly semi-consistent (Lemma 5.13),
4. The deletion-by-infeasibility rule is semi-certain in the limit (Lemma 5.11).

Assume that the B&B procedure does not terminate in a finite number of steps and consider the sequence of lower bounds β_k. Based on the proof of Theorem 2.1 in [62]

with simple arguments to extend for the strong semi-consistency of the lower bounding operation (instead of strong consistency) and for the deletion-by-infeasibility rule being semi-certain in the limit (instead of certain) we conclude that:

$$\beta = \lim_{k\to\infty} \beta_k = f(\bar{x}) = f^{GSIP}. \tag{5.18}$$

.

□

5.5.2 Convergence of Upper Bound

Let $CC(p, \varepsilon^*)$ denote a closed cube with center p and edge ε^*.

Lemma 5.17. *Assume a Slater point $\bar{x} \in X_s$ for which $P(\bar{x}) \neq \emptyset$. Then there exists $\varepsilon^* > 0$ for which for all $p \in P(\bar{x})$, the set $Q(p) = \{(\bar{x}, \bar{p}) : \bar{p} \in CC(p, \varepsilon^*) \cap D\}$ is upper-level feasible.*

Proof. $g(\bar{x}, \cdot)$ is continuous on D and thus it is uniformly continuous on D. Under continuity of u, it is well known that for all $x \in X$, $P(x)$ is a closed set. Since $P(\bar{x}) \subset D$ and D is compact, $P(\bar{x})$ is also compact. Therefore, $g(\bar{x}, \cdot)$ is a continuous function on the non-empty compact set $P(\bar{x})$ and thus it attains its supremum on $P(\bar{x})$. Choose $\delta = \left| \max_{p\in P(\bar{x})} g(\bar{x}, p) \right|$. Note that $\delta > 0$ by the definition of a GSIP Slater point. From uniform continuity, there exists $\varepsilon^* > 0$, such that for all $p \in D$ and $p' \in D$ for which $p' \in CC(p, \varepsilon^*)$ implies that $|g(\bar{x}, p) - g(\bar{x}, p')| < \delta$. This implies that for all $p \in P(\bar{x})$ and for $p' \in CC(p, \varepsilon^*) \cap D$, $g(\bar{x}, p') < 0$. Therefore, for all $p \in P(\bar{x})$, the set $Q(p)$ is upper-level feasible. □

Lemma 5.18. *Assume a Slater point $\bar{x} \in X_s$ for which $P(\bar{x}) \neq \emptyset$. Then, there exists $\varepsilon > 0$ such that for every $x \in CC(\bar{x}, \varepsilon) \cap X$, $g(x, p) < 0$, $\forall p \in P(\bar{x})$.*

Proof. From Lemma 5.17, the function $g_{\bar{x}}^{\max}(x) \equiv \max_{p\in P(\bar{x})} g(x, p)$ is well defined for all $x \in X$. It is also well known that this function is continuous on X since g is continuous on $X \times D$. Since $\bar{x}$ is a Slater point, this implies that $g_{\bar{x}}^{\max}(\bar{x}) < 0$. Since

$g_{\bar{x}}^{\max}$ is continuous on X this implies that there exists $\varepsilon > 0$ for which for every $x \in CC(\bar{x}, \varepsilon) \cap X$, $g_{\bar{x}}^{max}(x) < 0$. □

The next Lemma proves that with the subdivision heuristic described in Section 5.3, the subdivision of the parameter host set D is uniform as defined in [91].

Lemma 5.19. *Assume $\tau \in S_q$. Then, according to the subdivision approach of the algorithm above, there exists a set S defined as: $S = \{\tau' \in S_{q+1} : P_{\tau'} \subset P_\tau\}$ for which $\bigcup_{\tau' \in S} P_{\tau'} = P_\tau$. Furthermore, S defines a uniform subdivision of P_τ.*

Proof. Each dimension j of D is divided into 2^q subintervals of equal width:

$$D_{j,q}^k = [p_j^L + \frac{(k-1)\,\omega(P_j)}{2^q},\ p_j^L + \frac{k\omega(P_j)}{2^q}],\ \forall k = 1, \ldots, 2^q.$$

Now, there exist $\tau = (k_1, \ldots, k_{n_p}) \in S_q$ such that:

$$\begin{aligned} P_\tau = [p_1^L + \tfrac{(k_1-1)\,\omega(P_1)}{2^q},\ p_1^L + \tfrac{k_1\omega(P_1)}{2^q}] \times \ldots \\ \times [p_{n_p}^L + \tfrac{(k_{n_p}-1)\,\omega(P_{n_p})}{2^q},\ p_{n_p}^L + \tfrac{k_{n_p}\omega(P_{n_p})}{2^q}] \end{aligned} \tag{5.19}$$

In compact notation, relation (5.19) can be written as:

$$P_\tau = D_{1,\,q}^{k_1} \times \ldots \times D_{n_p,\,q}^{k_{n_p}}.$$

By construction, at level $(q+1)$ each dimension of D will be subdivided into 2^{q+1} subintervals of equal width:

$$D_{j,q+1}^k = [p_j^L + \frac{(k-1)\,\omega(P_j)}{2^{q+1}},\ p_j^L + \frac{k\omega(P_j)}{2^{q+1}}],\ \forall k = 1, \ldots, 2^{q+1}.$$

Obviously,

$$\begin{aligned} &[p_j^L + \tfrac{(2k_j-2)\omega(P_j)}{2^{q+1}},\ p_j^L + \tfrac{(2k_j-1)\omega(P_j)}{2^{q+1}}] \cup [p_j^L + \tfrac{(2k_j-1)\omega(P_j)}{2^{q+1}},\ p_j^L + \tfrac{2k_j\omega(P_j)}{2^{q+1}}] = \\ &[p_j^L + \tfrac{(k_j-1)\omega(P_j)}{2^q},\ p_j^L + \tfrac{k_j\omega(P_j)}{2^q}] = D_{j,q}^{k_j},\ \forall\, j : 1 \le j \le n_p. \end{aligned} \tag{5.20}$$

Let $P_j^{'}$ and $P_j^{''}$ be the intervals on the left hand-side of relation (5.20). Rewrite (5.20) as:

$$P_j^{'} \cup P_j^{''} = D_{j,\,q+1}^{k_j},\ \forall\, j : 1 \leq j \leq n_p. \tag{5.21}$$

Create the sets $A_1 = \{P_1^{'}, P_1^{''}\}$, ..., $A_{n_p} = \{P_{n_p}^{'}, P_{n_p}^{''}\}$. Choosing one element from each set A_j, the n_p - dimensional interval, $P_{\tau'}$, formulated as the Cartesian product of the n_p one-dimensional intervals will satisfy:

$$P_{\tau'} \subset P_{\tau}. \tag{5.22}$$

Obviously, there are 2^{n_p} such combinations of choices from the sets A_1,..., A_{n_p} all of which satisfy relation (5.22). Denote this set of choices as S, i.e., $S = \{\tau' \in S_{q+1} : P_{\tau'} \subset P_{\tau}\}$. Since relation (5.22) holds for every element of S this implies that:

$$\bigcup_{\tau' \in S} P_{\tau'} \subset P_{\tau}. \tag{5.23}$$

Suppose that there is an element of P_τ, say p^*, that does not belong to $\bigcup\limits_{\tau' \in S} P_{\tau'}$. This implies that if $p^* = (p_1^*, \ldots, p_{n_p}^*)$ there exists k, $1 \leq k \leq n_p$, such that $p_k^* \notin P_k^{'} \cup P_k^{''}$. This directly contradicts relation (5.21). This coupled with relation (5.23) results in:

$$\bigcup_{\tau' \in S} P_{\tau'} = P_{\tau}.$$

Thus, the set S defines a subdivision of P_τ. Furthermore, since each dimension of P_τ is divided into equal subintervals, this implies that S defines a uniform subdivision of P_τ.

□

Lemma 5.20. *Consider the branching rule mentioned in the algorithm above along with one complementary rule, i.e.:*

1. *Assume that the node selection heuristic has supplied node M located at level $q(M)$ of the B&B tree. In the branching procedure at node M bisect a coordinate*

j for which:

$$x_j^U - x_j^L = \max_{1 \le i \le n_x} x_i^U - x_i^L. \tag{5.24}$$

2. *If more than one coordinates satisfies relation (5.24) then the coordinate with the smallest index will be selected for bisection.*

Then the following results hold:

1. *For each level q of the B&B tree, the coordinate branched upon is uniquely defined, i.e., for each node M with q(M) = q, the same coordinate will be bisected. Furthermore, for each level of the B&B tree q, each coordinate will have the same width for all M with q(M) = q.*

2. *There exists a level q^* of the B&B tree such that for every M for which q(M) = q^*, the sequence of levels defined by*

$$q_k = q^* + k \cdot n_x, \;\; k = 0, 1, ...$$

where n_x is the dimension of X, provides a uniform subdivision of M.

Proof.

1. This result is proved through induction. Assume $q = 0$. At this level (root node), the choice of coordinate is well defined by the two branching rules and since there is only one node in the level, only one coordinate will be selected for branching throughout the level. Furthermore, since there is only one node at this level, each coordinate has the same width throughout the level. Assume these two properties hold for $q = k$. It will be demonstrated that they hold for $q = k + 1$. If coordinate m is selected throughout level q for bisection at level $q = k + 1$ each node M with $q(M) = k + 1$f will have:

$$b_{j,M} = b_j, \; j \neq m$$
$$b_{j,M} = \tfrac{b_j}{2}, \; j = m$$

where $b_{j,M}$ is the width of coordinate j at level $q = k + 1$ for node M and b_j is the uniform width of coordinate j at level $q = k$ of the B&B tree. This clearly implies that all nodes at level $q = k + 1$ will have the same width for all the coordinates. By a similar argument, since all the nodes at $q = k + 1$ have the same coordinate widths, the choice of coordinate for bisection is uniquely defined throughout this level.

2. Let $b_{m,q}$ be the width of coordinate m at level q of the B&B tree. It is well known that the branching rules stated above are exhaustive [63], i.e.:

$$\lim_{q\to\infty} b_{m,q} = 0, \ 1 \leq m \leq n_x.$$

Consider the index set for the coordinates $A = \{1, ..., n_x\}$. Pick $i^* \in A$ such that $b_{i^*,0} = \min_{m\in A} b_{m,0}$. If there are more than one coordinates satisfying this then choose the one with the highest index (in order for this coordinate to be last to be branched on based on the branching rules). Let the sets A_1 and A_2 be defined as follows:

$$A_1 = \{m \in A : b_{m,0} > b_{i^*,0}\},$$
$$A_2 = \{m \in A : b_{m,0} = b_{i^*,0}\}.$$

For each $m \in A_1$, let q_m be the level of the B&B tree for which:

$$\begin{aligned} \tfrac{b_{i^*,0}}{2} < b_{m,q_m+1} \leq b_{i^*,0}, \\ b_{m,q_m} > b_{i^*,0}. \end{aligned} \tag{5.25}$$

For each $m \in A_2 \setminus i^*$, let q_m be the level of the B&B tree for which:

$$\begin{aligned} b_{m,q_m} = b_{i^*,0}, \\ b_{m,q_m+1} = \tfrac{b_{i^*,0}}{2}. \end{aligned} \tag{5.26}$$

Assume $q' = \max_{m\in(A_1\cup A_2)\setminus i^*} q_m$. This implies that coordinate i^* will certainly be

bisected at level $q' + 1$, and thus:

$$\begin{aligned} q_{i^*} &= q' + 1 \\ b_{i^*,q_{i^*}} &= b_{i^*,0} \\ b_{i^*,q_{i^*}+1} &= \tfrac{b_{i^*,0}}{2}. \end{aligned} \tag{5.27}$$

Since each time a coordinate of X is chosen it is bisected, q_m is well defined, i.e., for all m for which $1 \leq m \leq n_x$, there is exactly one level q_m that satisfies either relation (5.25) or (5.26) or (5.27). It is also clear that:

$$\begin{aligned} m_i \neq m_j \;\; &\Rightarrow \;\; q_{m_i} \neq q_{m_j}, \\ q_{m_i} > q_{m_j} \;\; &\Rightarrow \;\; b_{m_i,q_{m_i}} \leq b_{m_j,q_{m_j}}. \end{aligned} \tag{5.28}$$

Without loss of generality, suppose that $q_{n_x} > q_{n_x-1} > \ldots > q_1$ so that $i^* = n_x$. Relation (5.28) implies that:

$$b_{i^*,q_{i^*}} \leq b_{n_x-1,q_{n_x-1}} \leq \ldots \leq b_{1,q_1}. \tag{5.29}$$

It is obvious from relations (5.25), (5.26), (5.27), (5.29), and from the bisection rules that at level $q^* = q' + 2$, coordinate 1 will be chosen for bisection. At level $q^* + 1$, coordinate 2 will be bisected and with the same argument at levels $q^* + (n_x - 2)$ and $q^* + (n_x - 1)$, coordinates $n_x - 1$ and n_x will be bisected, respectively.

Consider a node X_1 located at level q^*. Let S_{n_x} be the index set for the 2^{n_x} descendant nodes of X_1 at node $q^* + n_x$. From the bisection process described above it is clear that:

$$\begin{aligned} \bigcup_{X_i \in S_{n_x}} X_i &= X_1 \\ b_{m,q^*+n_x} &= \tfrac{b_{m,q^*}}{2}, \forall\, 1 \leq m \leq n_x. \end{aligned}$$

This clearly implies that the set of nodes indexed by S_{n_x} provides a uniform

subdivision of X_1. It is obvious that from level $q^* + n_x$ up to $q^* + (2n_x - 1)$ the same sequence of coordinates will be chosen for bisection as was chosen from levels q^* up to $q^* + (n_x - 1)$. Thus, the index set S_{2n_x} of descendant nodes of X_1 at level $q^* + 2n_x$ also provides a uniform subdivision for X_1. Thus, the sequence of levels defined by $q_k = q^* + k \cdot n_x$ provides a uniform subdivision for X_1. Since the choice of X_1 was arbitrary, obviously the same result holds for every node at level q^*.

□

Corollary 5.21. *For every node M for which $q(M) \geq q^*$ (for the definition of q^* see Lemma 5.20) the sequence of levels defined by*

$$q_k = q(M) + kn_x, k = 0, 1, \ldots \tag{5.30}$$

where n_x is the dimension of X, provides a uniform subdivision of M.

Proof. Follows immediately from Lemma 5.20. □

Theorem 5.22. *Assume the GSIP formulation* (5.1) *and let f^{GSIP} be its solution value. Then, the upper bounding operation described in Section 5.3 converges to the solution value of the GSIP provided that for the breadth-first search Assumption 5.7 holds, while for the best-bound search Assumption 5.8 holds.*

Proof. Suppose that $\bar{x}$ is a Slater point with $P(\bar{x}) \neq \emptyset$. From Lemma 5.17 there exists $\varepsilon^* > 0$ for which for all $p \in P(\bar{x})$, the set $\bar{x} \times CC(p,\varepsilon^*) \cap D$ is upper-stage feasible. Define $\omega_q(D) = \max\limits_{1 \leq j \leq n_p} \frac{\omega(P_j)}{2^q}$. Since $\lim\limits_{q \to \infty} \omega_q(D) = 0$ (the subdivision of D is exhaustive), there exists a level $\bar{q}$ for which $\omega_q(D) \leq \varepsilon^*$ for all $q > \bar{q}$. Fix $q^1 > \bar{q}$: $\omega_{q^1}(D) \leq (\frac{\varepsilon^*}{2})$. The index set for the subdivision of the parameter set D at level q^1 can be written as:

$$S_{q^1} = S_f \cup S_{inf}, \tag{5.31}$$

where the index sets S_f and S_{inf} are defined as:

$$S_f = \{\tau \ \in S_{q^1} : \exists\, p \in P_\tau : u_j(\bar{x}, p) \leq 0, \forall j \in J\} \tag{5.32}$$

$$S_{inf} = \{\tau \ \in S_{q^1} : \forall\, \mathbf{p} \in P_\tau : u_j(\bar{x}, p) > 0, \text{for some } j \in J\}. \tag{5.33}$$

Relation (5.31) holds directly through relations (5.32) and (5.33). Furthermore,

$$S_f \cap S_{inf} = \emptyset.$$

Let $\tau' \in S_{inf}$. This implies that $\exists\ j \in J$ for which $u_j(\bar{x}, p) > 0$, $\forall\ p \in P_{\tau'}$. From Lemma 5.18 there exists a neighborhood of $\bar{x}$, namely $CC(\bar{x}, \varepsilon) \cap X$, such that $\forall\ (x, p) \in CC(\bar{x}, \varepsilon) \cap X \times P_{\tau'}$, $u_j(x, p) > 0$ (the correspondence with Lemma 5.18 is $-u_j(x, p)$ for $g(x, p)$, $P_{\tau'}$ for $P(\bar{x})$ and $\bar{x}$ for $\bar{x}$).

Certain results from interval analysis are going to be used at this point. Assume an arbitrary continuous function $t : X \times D \to \mathbb{R}$, where X and D are n_x- and n_p-dimensional intervals respectively. Consider any continuous interval extension of t on $X \times D$, namely $T : X \times D \to \mathbb{IR}$, $T(X, D) = [t^L(X, D), t^U(X, D)]$. Assume a set of uniform subdivisions of both X and D, defined by the index sets Σ_i and S_i, respectively, and also assume that as $i \to \infty$ the cardinality of Σ_i and S_i tends to infinity. Then, it is well known that $\lim\limits_{i \to \infty} t_i^L(X, D) \equiv \lim\limits_{i \to \infty} \min\limits_{\sigma \in \Sigma_i, \tau \in S_i} t^L(X_\sigma, P_\tau) = \min\limits_{(x,p) \in X \times D} t(x, p)$.

Therefore, since $\min\limits_{(x,p)\, \in\, CC(\bar{x}, \varepsilon) \cap X \times P_{\tau'}} u_j(x, p) > 0$, $j \in J$, there exists $\bar{i}_j$ such that $i \geq \bar{i}_j$ implies that $u^L_{j,\, i_j}(CC(\bar{x}, \varepsilon) \cap X, P_{\tau'}) > 0$. Choose $i' = \max\limits_{j \in J} \bar{i}_j$ (recall that J is the index set of the lower-level constraints). Consider a node M for which $q(M) \geq q^*$ (q^* was defined in Corollary 5.21) and $M \subset CC(\bar{x}, \varepsilon) \cap X$. Natural interval extensions are inclusion monotonic and, therefore:

$$\exists i' \text{ such that } u^L_{j,\, i}(M, P_{\tau'}) > 0, \forall j \in J \text{ and } i \geq i'. \tag{5.34}$$

Corollary 5.21 shows that for every node M of the tree for which $q(M) \geq q^*$ there exists a set of nodes at each of the levels $q(M) + kn_x, k = 1, 2, ...$, the union of which

constitutes a uniform subdivision of M. Therefore, taking relationship (5.34) into consideration:

$$\exists k' \text{ such that } u^L_{j,\,k}(M, P_{\tau'}) > 0, \forall j \in J \text{ and } k \geq k', \tag{5.35}$$

where k denotes the uniform subdivision of M at level $q(M) + kn_x$ of the B&B tree. Assume that at the subdivision of M defined by k' the element of the partition of M that contains $\bar{x}$ is $\bar{X}$ (i.e., $\bar{X} \subset M$, $\bar{X}$ contains $\bar{x}$ and from the uniform subdivision of M, $\bar{X}$ is an interval) and the index set for the subdivision of $P_{\tau'}$ is defined by $S_{i'}$. Taking relationship (5.35) into consideration it can finally be shown that $u^L_j(\bar{X}, P_{\tau''}) > 0$, for all $\tau'' \in S_{i'}$.

Since the branching procedure is exhaustive, there exists a level q^2 of the branch-and-bound tree for which a node M^1, $q(M^1) = q^2$, contains $\bar{x}$ and satisfies $M^1 \subset \bar{X}$. From Lemma 5.19, the exists a level of the tree $q^3 > q^1$, for which the subdivision of $P_{\tau'}$ given by $S_{P_{\tau'}}$, $S_{P_{\tau'}} \subset S_{q^3}$, is denser than the one defined by $S_{i'}$.

Combining all the previous results, it is clear that at level $q' = \max\{q^2, q^3\}$, there exists a node M^2, $q(M^2) = q'$, that contains $\bar{x}$ and satisfies $M^2 \subset M^1$ and furthermore $u^L_j(M^2, P_{\tau''}) > 0$, $\forall\ \tau'' \in S_{P_{\tau'}}$. Clearly, for $q \geq q'$, $P_{\tau'}$ will be lower-level infeasible for all nodes X satisfying $X \subset M^2$, and therefore no descendant partition of $P_{\tau'}$ will be considered by the upper bounding problems.

Let $S_{inf} = \{\tau'_1, \ldots, \tau'_p\}$, for some finite p. Since the choice of τ' was arbitrary, for every $\tau'_m \in S_{inf}$, $1 \leq m \leq p$, there must exist a finite level q'_m for which $P_{\tau'_m}$ will no longer be considered for the upper bounding problems for nodes containing $\bar{x}$. Let $q_{inf} = \max\limits_{1 \leq m \leq p} q'_m$. For $q \geq q_{inf}$, the only part of the parameter set D that will be considered for the upper-bounding problems, for the nodes containing $\bar{x}$, will be a subset of D defined by S_f. In other words, for $q \geq q_{inf}$ the following relationship holds:

$$\forall \tau \in S_q \text{ for which } P_\tau \subset P_{\tau'}, \text{ for some } \tau' \in S_{inf}, P_\tau \text{ will not be considered for the upper bounding problem in the nodes containing } \bar{x}.$$

Fix $\tau' \in S_f$. Since $\omega_{q^1}(P) \leq (\frac{\varepsilon^*}{2})$, this implies that $P_{\tau'}$ is upper-level feasible for $\bar{x}$, i.e., $g(\bar{x}, p) < 0$ for every $p \in P_{\tau'}$ (Lemma 5.17). From Theorem 2 in [27], there exists a uniform subdivision of $P_{\tau'}$ for which $g^U_{q^i}(\bar{x}, P_{\tau'}) < 0$, where $g^U_{q^i}$ denotes the upper bound of the interval extension of g on $P_{\tau'}$ under a uniform subdivision defined by q^i. From Lemma 5.19, there exists a level q^4 of the B&B tree for which the uniform subdivision of $P_{\tau'}$, $S_{P_{\tau'}} \subset S_{q^4}$, is denser than the one defined by q^i. This implies that:

$$\text{For all } \tau'' \in S_{P_{\tau'}},\ g^U(\bar{x}, P_{\tau''}) < 0. \tag{5.36}$$

Relation (5.36) implies that for every subsequent level, all of the partitions of $P_{\tau'}$ will be upper-level feasible for $\bar{\mathbf{x}}$.

Let $S_f = \{\tau'_1, \ldots, \tau'_r\}$, for some finite r. Since the choice of τ' was arbitrary, for every $\tau'_m \in S_f$, $1 \leq m \leq r$, there must exist a level q_m for which the partition of $P_{\tau'_m}$ at this level will satisfy relation (5.36). Let $q_{feas} = \max\limits_{1 \leq m \leq r} q_m$. For $q \geq q_{feas}$, for which the partition of D is given by the index set S_q, the following relation will hold:

$$\forall \tau \in S_q \text{ for which } P_\tau \subset P_{\tau'}, \text{ for some } \tau' \in S_f, : g^U(\bar{x}, P_\tau) < 0. \tag{5.37}$$

Relations (5.36) and (5.37) show that at level $q_{Slater} = \max\{q_{inf}, q_{feas}\}$, $\bar{x}$ will be feasible to the upper bounding formulation at node M containing $\bar{x}$.

Similarly, suppose that $\bar{x}$ is Slater point with $P(\bar{x}) = \emptyset$. This implies that:

$$\forall p \in D : \exists j \in J : u_j(\bar{x}, p) > 0. \tag{5.38}$$

Define $r^* = \min\limits_{p \in D} \max\limits_{j \in J} u_j(\bar{x}, p)$. Taking relation (5.38) into consideration, $r^* > 0$ should hold. Consider an infinite sequence of nested nodes $\{M_{\lambda_q}\}$ such that the accumulation point of $\{M_{\lambda_q}\}$ is $\{\bar{x}\}$. Define $U_q(M_{\lambda_q}, D) = \min\limits_{\tau \in S_q} \max\limits_{j \in J} u^L_j(M_{\lambda_q}, P_\tau)$. By similar arguments, it is clear that the following relation holds:

$$\lim_{q \to \infty} U_q(M_{\lambda_q}, D) = r^* > 0.$$

Therefore, there has to exist q^*, and for each $\tau \in S_q$ a lower-level inequality index $j(\tau)$ such that the node $M_{\lambda_{q^*}}$ and the subdivision of D defined by S_{q^*} satisfy:

$$u^L_{j(\tau)}(M_{\lambda_{q^*}}, P_\tau) > 0, \; \forall \tau \in S_q.$$

Note the dependence of the lower-level inequality constraint j on the partition element τ of the index set of subdivisions S_q. Therefore, for $M_{\lambda_{q^*}}$ and for all of its descendant partitions, $\bar{x}$ will be considered feasible to the upper-bounding problem.

Note that since $\bar{x} \in X_s$ has been proved to be found feasible finitely for the upper bounding problem in both cases ($P(\bar{x}) = \emptyset$ and $P(\bar{x}) \neq \emptyset$) the distinction between GSIP Slater points with an empty or an non-empty lower-level feasible set is no longer made.

Recall that $\{x_n\}$ is a sequence of GSIP Slater points converging to a point x' with $f(x') = f^{GSIP}$. Depending on the node selection heuristic the proof continues in the following way:

1. **Breadth First Search**. Fix $\varepsilon > 0$. From continuity of f on X and from the convergence of $\{x_n\}$ to x', there obviously exists n^* for which, $n > n^*$ implies that $|f(x') - f(x_n)| < \varepsilon$.

   Since the branching process is exhaustive, there exists an infinite sequence of nested nodes $\{M_{\lambda_q}\}$ such that $\lim_{q\to\infty} \omega(M_{\lambda_q}) = 0$, $M_{\lambda_{q+1}} \subset M_{\lambda_q}$ and $x_{n'} \in M_{\lambda_q}$ for every q.

   Finally, for q_{Slater}, x_n will be found feasible to the upper bounding problem and it will provide an objective value function better than or equal to $f(x_n)$. Furthermore, since our node selection heuristic is based on breadth-first search, each node is guaranteed to be be branched on at some finite iteration. Thus, $M_{\lambda_{q_{Slater}}}$ will be visited at some finite iteration k^*. Therefore:

   $$\forall k > k^* \; , \; \alpha_k \leq f^{GSIP} + \varepsilon.$$

   Thus, for $\varepsilon > 0$, $\exists \; k^*$ for which for all $k > k^*$, $f^{GSIP} \leq \alpha_k \leq f^{GSIP} + \varepsilon$. This

implies that:

$$\lim_{k\to\infty} \alpha_k = f^{GSIP}. \tag{5.39}$$

2. **Best Bound Approach**. It has already been shown that for every point $x \in X_s$, q_x^1 is well defined, i.e., x will be found feasible to the upper bounding operation at some level of the branch-and-bound tree ($q_{Slater}(x)$). $q_{x,x'}^2$ is well defined because the root node ensures that $x' \in X_{inf}$ and x are together initially, and the exhaustive partitioning of X ensures that these two points will finally belong to different nodes (if $x \neq x'$).

   Consider, the lower bounding operation. From Theorem 5.16, $\lim_{k\to\infty} \beta_k = f^{GSIP}$. Since $k \to \infty$ there exists an infinite sequence of nested nodes $\{M_{\lambda_q}\}$ converging to a point $\{x'\}$. From Theorem 5.16, $f(x') = f^{GSIP}$. Using the same approach as in Lemma 5.11 it can easily shown that $x' \notin X_{inf,1}$ ($x \in H$ and $x \in X_{inf,1}$ satisfy the same assumptions). In other words if $x' \in X_{inf,1}$ the lower bounding operation would eventually fathom the node containing x' because of infeasibility. Therefore, $x' \in X_{inf,2}$.

   From Assumption 5.8.2 at least one of the two following relationships must hold:

   (a) $\exists$ a subsequence of $\{x_n\}$, $\{x_{n_k}\}$ that satisfies:

   $$\lim_{n_k\to\infty} x_{n_k} = x' \quad \text{and} \quad q_{x_{n_k}}^1 < q_{x_{n_k},x'}^2, \forall n. \tag{5.40}$$

   (b) $\exists$ a subsequence of $\{x_n\}$, $\{x_{n_k}\}$ that satisfies:

   $$\lim_{n_k\to\infty} x_{n_k} = x' \quad \text{and} \quad q_{x_{n_k}}^1 = q_{x_{n_k},x'}^2, \forall n. \tag{5.41}$$

   If relationship (5.40) holds, then by Lemma 5 in [27] it can now be easily shown that:

   $$\lim_{k\to\infty} \alpha_k = f^{GSIP}. \tag{5.42}$$

   If relationship (5.41) holds, then for an arbitrary member $\hat{x}$ of $\{x_{n_k}\}$, $\hat{x}$ is not found feasible when it belongs in the same node as x' but it is found feasible

in the first level of the B&B tree for which $\hat{x}$ and x' do not belong to the same node. Then, obviously $\hat{x}$ and x' belong to sister nodes. To understand the later assume a node M that contains $\hat{x}$ and x' defined by:

$$M = [x_1^L\ ,\ x_1^U] \times \ldots \times [x_{j-1}^L\ ,\ x_{j-1}^U] \times [x_j^L\ ,\ x_j^U] \times [x_{j+1}^L\ ,\ x_{j+1}^U] \times \ldots \times [x_{n_x}^L\ ,\ x_{n_x}^U].$$

Then, according to the bisection process, in level $q(M)+1 = q^2_{\hat{x},x'}$ two nodes, called sister nodes, will be created as follows:

$$M_1 = [x_1^L\ ,\ x_1^U] \times \ldots \times [x_j^L\ ,\ \tfrac{(x_j^U + x_j^L)}{2}] \times \ldots \times [x_{n_x}^L\ ,\ x_{n_x}^U].$$
$$M_2 = [x_1^L\ ,\ x_1^U] \times \ldots \times [\tfrac{(x_j^U + x_j^L)}{2}\ ,\ x_j^U] \times \ldots \times [x_{n_x}^L\ ,\ x_{n_x}^U].$$

Without loss of generality, assume $x' \in M_1$ and $\hat{x} \in M_2$ and also that $\lambda(M_1) = y$, for some $y \in \mathbb{N}$. According to the bisection process this implies that $\lambda(M_2) = y+1$. At the time of bisection, the lower bounds assigned by the algorithm to these two nodes, $f_{M_1}^{LBD}$ and $f_{M_2}^{LBD}$ respectively, are equal. Since an infinite sequence of nodes $\{M_{\lambda_q}\}$ that converges to x' is created, at some finite iteration of the algorithm, M_1 is going to be bisected and examined. Suppose that the children nodes of this bisection are M_3 and M_4 and that $\lambda(M_3) = y'$ and $x' \in M_3$. Obviously, the following relationships hold:

(a) $f_{M_1}^{LBD} = f_{M_2}^{LBD} \leq f_{M_3}^{LBD}$ (non-decreasing lower bounds),

(b) $y+1 < y'$,

(c) At some finite iteration k^*, M_3 is chosen and examined.

Recall that with the best-bound node selection heuristic, a node M is chosen such that $M = \arg\min_{M \in V_1} \lambda(M),\ V_1 := \arg\min_{M \in I_k} f_M^{LBD}$.

At iteration k^* of the algorithm where M_3 is chosen, assume $M_2 \in I_{k^*}$. Then because of (a) and (c), $M_2, M_3 \in V_1$. However, (b) implies that $\lambda(M_2) < \lambda(M_3)$. Therefore, $M_3 \neq \arg\min_{M \in V_1} \lambda(M)$ and M_3 would not have been chosen. Therefore, $M_2 \notin I_{k^*}$. There are two possibilities for this, either M_2 was fathomed by value dominance or it was chosen and examined prior to iteration k^*. However

$f_{M_2}^{LBD} \leq f(x') = f^{GSIP}$ and, therefore, the node could not have been fathomed due to value dominance. Thus, M_2 was chosen and examined prior to iteration k^*. Therefore, for any $M' \in \{M_{\lambda_q}\}$ that is chosen and examined, its sister node M'' is also chosen and examined. Thus, there exists an infinite sequence of nodes $\{M_n\}$, each member of which is a sister node to a member of $\{M_{\lambda_q}\}$ and is examined at some finite iteration.

Based on the existence of $\{M_n\}$, relationship (5.41) and the continuity argument of Lemma 5 in [26], relationship(5.42) holds in this case too.

□

Corollary 5.23. *The B&B algorithm guarantees ε-optimality in a finite number of iterations.*

Proof. Fix $\varepsilon > 0$. From relations (5.18), (5.39) and (5.42):

$$\lim_{k\to\infty} \alpha_k = \lim_{k\to\infty} \beta_k = f^{GSIP}.$$

This implies that there exists a finite iteration k^* for which

$$\alpha_k - \beta_k < \varepsilon, \forall k \geq k^*.$$

□

5.6 Numerical Implementation

5.6.1 Test Set

The main goal of the test set it to represent all possible scenarios for the structure and optimality of GSIP. Certain problems are drawn from the GSIP literature while others are original. The test set is created based on the following criteria:

1. **Closed Feasible Set**. The test set includes problems where the feasible set is closed (3,5,8,10,12,14,15) and not closed (1,2,4,6,7,9,11,13,16).

2. **Convex Lower-Level Problem**. The test set includes problems where the lower-level problem is convex on D for each $\mathbf{x} \in X$ (1,4,6,10,11) and also problems where the lower-level problem is nonconvex on D for some $\mathbf{x} \in X$ (2,3,5,7,8,9,12,13,14,15,16).

3. **Empty Lower-Level Feasible Set**. The test set includes problems where for every $\mathbf{x} \in X$ the lower-level feasible set is not empty (3,5,8,10,12, 14,15) and also problems where there exists $\mathbf{x} \in X$ for which the lower-level feasible set is empty (1,2,4,6,7,9,11,13,16).

4. **Re-entrant corner points**. Problem 8 involves a re-entrant corner point.

   Statements of the test problems can be found in Appendix A.

5.6.2 Numerical Implementation & Results

The GSIP B&B algorithm was implemented in C++ using an in-house B&B code, while the upper and lower bounding problems were solved to local optimality using SNOPT 6.1-1 [46]. To calculate the inclusion bounds on $g(\mathbf{x}, D)$ (upper-stage constraints) and also on u_j (lower-level constraints) natural interval extensions were employed. In terms of numerical implementation the relative and absolute tolerances for retaining a node within the tree were set to 0.01. A node was fathomed from the tree if either of the two tolerances was met. The tolerance for SNOPT for both upper and lower bounding problems was set to 10^{-7}.

Table 5.1: Convergence results for GSIP with Best-Bound Search

Problem	Nodes	Depth	CPU	f	x	Minimum	Feasible Set
1	23	8	0.22	0.0674	(0.00781, 0.09375)	Attained	Non-Closed
2	7	2	0.02	0	(0, 0)	Attained	Non-Closed
3	59	7	42.05	-0.5	(1, 1)	Attained	Closed
4	16	5	0.04	0.0039	(-0.0625)	Not Attained	Non-Closed
5	30	8	0.23	-5	(5, -4.6875)	Attained	Closed
6	4	2	0.02	-6	(0, -3)	Attained	Non-Closed
7	116	13	15.69	-0.4922	(0.4922, 0.0078)	Attained	Non-Closed
8	1	1	0.003	-1	(1, 0)	Attained	Closed
9	17	7	0.05	0.04785	(-0.21875)	Not Attained	Non-Closed
10	67	15	1.59	-1	(-1, 0)	Attained	Closed
11	45	10	0.39	5.039	(-0.5019, -0.5019, 0)	Not-Attained	Non-Closed
12	19	5	0.04	0.5	(-0.707)	Attained	Closed
13	323	16	16.75	2.937	(-1, 0.2506, 0.2506)	Not Attained	Non-Closed
14	77	13	3.72	0.386	(-0.6214, 0, 0)	Attained	Closed
15	119	11	1.42	-3.70	(2, 1.4527)	Attained	Closed
16	10	4	0.03	-10.67	(2, 0.25, 1, 2, 1, 2)	Attained	Non-Closed

Tables 1 and 2 summarize the main results from the numerical procedure presented in this paper. Table 1 shows results for the best-bound node selection heuristic while Table 2 shows the corresponding results with the breadth-first node selection heuristic. In both tables, Column 1 lists the index of the problem, Column 2 the number of required nodes to achieve ε-optimality, Column 3 the maximum depth explored in the B&B tree, Column 4 the required CPU time, Column 5 the ε-optimal objective function value that was obtained, Column 6 the feasible point at which this objective function value was obtained, Column 7 indicates whether the infimum of the GSIP is attained and finally Column 8 comments on the closedness of the feasible set of the GSIP.

An immediate conclusion from the numerical results presented here is that, as expected, the best-bound node selection heuristic outperforms the breadth-first node selection heuristic. However, because the test set comprises small problems this difference is not crucial to the efficiency of the algorithm. For both node selection heuristics the CPU time ranges from 0.02 to 42s. There seems to be no correlation between the CPU time and the whether the feasible set of the GSIP is closed or not. Furthermore, there seems to be no correlation between the CPU time and whether the minimum of the GSIP is attained or not.

Table 5.2: Convergence results for GSIP with Breadth-First Search

Problem	Nodes	Depth	CPU	f	x	Minimum	Feasible Set
1	35	8	0.22	0.0664	(0, 0.0625)	Attained	Non-Closed
2	7	2	0.02	0	(0, 0)	Attained	Non-Closed
3	59	7	42.05	-0.5	(1, 1)	Attained	Closed
4	16	5	0.04	0.0039	(-0.0625)	Not Attained	Non-Closed
5	184	8	2.38	-5	(5, -4.6875)	Attained	Closed
6	5	2	0.02	-6	(0, -3)	Attained	Non-Closed
7	116	13	15.69	-0.4922	(0.4922, 0.0078)	Attained	Non-Closed
8	1	1	0.003	-1	(1, 0)	Attained	Closed
9	25	7	0.06	0.04785	(-0.21875)	Not Attained	Non-Closed
10	67	15	1.59	-1	(-1, 0)	Attained	Closed
11	55	10	0.43	5.039	(-0.5019, -0.5019, 0)	Not-Attained	Non-Closed
12	19	5	0.04	0.5	(-0.707)	Attained	Closed
13	325	16	16.77	2.937	(-1, 0.2506, 0.2506)	Not Attained	Non-Closed
14	119	13	6.27	0.386	(-0.6214, -0.03125, 0)	Attained	Closed
15	143	11	2.99	-3.69	(1.9375, 1.4392)	Attained	Closed
16	19	4	0.05	-10.67	(2 , 0.25, 1, 2, 1, 2)	Attained	Non-Closed

Chapter 6

Test Set for Generalized Semi-Infinite Programs

6.1 Introduction

The goal of this chapter is to present a novel test set for generalized semi-infinite programs. In Section 6.2 we discuss the notion of replacing the lower-level problem with its KKT conditions and in Section 6.3 we discuss the main criteria for the development of the test set. Furthermore, we provide the test problems along with a detailed analysis in Section 6.4.

6.2 Replacing Lower-Level Problem with its KKT Conditions

For convex lower-level problems that are feasible for each $\mathbf{x} \in X$, an equivalent single-level representation of the GSIP has been suggested in [120]

$$\begin{aligned} &\min_{\mathbf{x} \in X} f(\mathbf{x}) \\ &\text{s.t. } g(\mathbf{x}, \bar{\mathbf{p}}) \leq 0 \\ &\bar{\mathbf{p}} \text{ is a KKT point of (1.4) for } \mathbf{x}. \end{aligned} \tag{6.1}$$

However, if the lower-level problem is convex for each $\mathbf{x} \in X$ but infeasible for some $\mathbf{x} \in X_1 \subset X$, then it is clear that X_1 is an infeasible subset of (6.1) but a feasible subset of (4.7). Since the set $X \setminus X_1$ will have the same behavior in the two formulations, it is clear that the KKT representation of the lower-level problem provides a restriction of the GSIP. Furthermore, if the lower-level problem is nonconvex for some $\mathbf{x} \in X$ but feasible for all $\mathbf{x} \in X$, then the KKT conditions are only a necessary condition for the optimality of the lower-level problem (assuming the Slater condition is satisfied for each $\mathbf{x} \in X$). Thus, a KKT representation of (4.7) would only provide a relaxation of the GSIP. Finally, if the lower-level problem is nonconvex for some $\mathbf{x} \in X$ and also infeasible for some $\mathbf{x} \in X$ then (6.1), in general, provides neither a relaxation nor a restriction of (4.7).

6.3 Criteria for the Test Set

The main goal of the test set it to represent all possible scenarios for the structure and optimality of GSIP. Certain problems are drawn from the GSIP literature while others are original. The test set is created based on the following criteria:

1. **Closedness of Feasible Set**. The test set includes problems where the feasible set is closed (3,5,8,10,12,14,15) and not closed (1,2,4,6,7,9,11,13,16).

2. **Existence of the Minimum of the GSIP**. The test set includes examples where the infimum of the GSIP is attained (1,,2,3,5,6,7,8,10,12,14,15,16) and also examples where the infimum is not attained (4,9,11,13).

3. **Convexity of Lower-Level Problem**. The test set includes problems where the lower-level problem is convex on D for each $\mathbf{x} \in X$ (1,4,6,10,11) and also problems where the lower-level problem is nonconvex on D for some $\mathbf{x} \in X$ (2,3,5,7,8,9,12,13,14,15,16).

4. **Emptiness of Lower-Level Feasible Set**. The test set includes problems where for every $\mathbf{x} \in X$ the lower-level feasible set is not empty (3,5,8,10,12,

14,15) and also problems where there exists $\mathbf{x} \in X$ for which the lower-level feasible set is empty (1,2,4,6,7,9,11,13,16).

5. **Existence of Re-entrant corner points**. Problem 8 involves a re-entrant corner point.

It should be mentioned that while the literature examples include the original objective and the lower- and upper-level constraints, the host sets of the decision variables and parameters, X and D respectively, are typically altered.

6.4 Test Problems

1. ([65] pg. 157 , Ex. 4-2)

$$\begin{aligned} f(\mathbf{x}) &= (x_1 - \frac{1}{4})^2 + x_2^2 \\ g(\mathbf{x}, \mathbf{p}) &= p + x_2 \\ u(\mathbf{x}, \mathbf{p}) &= p^2 - x_1 \\ X = [-1, 1]^2 \, , \; D &= [-1, 1]. \end{aligned}$$

The upper-level problem is convex in $\mathbf{x}$ for each $\mathbf{p} \in D$. The lower-level problem is convex in $\mathbf{p}$ for each $\mathbf{x} \in X$. The lower-level feasible set is given by:

$$P(\mathbf{x}) = \begin{cases} [-\sqrt{x_1}, \sqrt{x_1}] & \text{if } x_1 \geq 0 \\ \emptyset & \text{if } x_1 < 0. \end{cases}$$

The feasible set is given by:

$$M = \{\mathbf{x} \in [-1, 1]^2 \mid x_1 < 0\} \cup \{\mathbf{x} \in [-1, 1]^2 \mid x_1 \geq 0, \sqrt{x_1} \leq x_2\}.$$

The feasible set is not closed but the minimum is unique and attained at $\bar{\mathbf{x}} = (0, 0)$. At the optimal solution, the Slater constraint qualification is

violated for the lower-level problem:

$$\min_{p\in[-1,1]} p$$
$$\text{s.t.} - p^2 \leq 0.$$

The Slater constraint qualification is violated because the lower-level feasible set is empty for certain values of the optimization variables. This implies that the lower-level problem is infeasible for some $\mathbf{x} \in X$.

2. ([65] pg. 156 , Ex. 4-1)

$$\begin{aligned} f(\mathbf{x}) &= x_2 \\ g(\mathbf{x},\mathbf{p}) &= -p^3 + x_2 \\ u(\mathbf{x},\mathbf{p}) &= 2x_2 - p^3 - x_1^2 \\ X = [-1,1]^2 \ , \ D &= [-1,0]. \end{aligned}$$

The upper-level problem is convex in $\mathbf{x}$ for each $\mathbf{p} \in D$. The lower-level problem is nonconvex in $\mathbf{p}$ for each $\mathbf{x} \in X$. The lower-level feasible set is given by:

$$P(\mathbf{x}) = \begin{cases} [(2x_2 - x_1^2)^{1/3}, 0] & \text{if } 2x_2 - x_1^2 \leq 0 \\ \emptyset & \text{if } 2x_2 - x_1^2 > 0. \end{cases}$$

The feasible set is given by:

$$M = \{\mathbf{x} \in [-1,1]^2 \mid x_1^2 < 2x_2\} \cup \{0\}.$$

The feasible set is not closed but the minimum is unique and attained at $\bar{\mathbf{x}} = (0,0)$. At the optimal solution, the Slater constraint qualification is vi-

olated for the lower-level problem:

$$\min_{p\in[-1,1]} p$$
$$\text{s.t.} - p^2 \leq 0.$$

The Slater constraint qualification is violated because the lower-level feasible set is empty for certain values of the optimization variables. This implies that the lower-level problem is infeasible for some $\mathbf{x} \in X$.

3. ([105] Pg. 184, Ex. 5.2)

$$f(\mathbf{x}) = -\frac{1}{2}x_1^4 + 2x_1x_2 - 2x_1^2$$
$$g(\mathbf{x},\mathbf{p}) = p_1^2 + p_2^2 - x_1 + x_1^2 - x_2$$
$$u(\mathbf{x},\mathbf{p}) = p_1^2 + p_2^2 + p_3^2 - x_1$$
$$X = [0,1]^2\ ,\ D = [0,1]^3.$$

The upper-level problem is nonconvex in $\mathbf{x}$ for each $\mathbf{p} \in D$. The lower-level problem is nonconvex in $\mathbf{p}$ for each $\mathbf{x} \in X$. The lower-level feasible set is given by:

$$P(\mathbf{x}) = \begin{cases} \{\mathbf{p} \in [0,1]^3 : p_1^2 + p_2^2 + p_3^2 \leq x_1\} & \text{if } x_1 \geq 0 \\ \emptyset & \text{if } x_1 < 0. \end{cases}$$

The feasible set is given by:

$$M = \{\mathbf{x} \in [0,1]^2 \mid x_1^2 \leq x_2\}.$$

The feasible set is closed and the minimum is unique and attained at $\bar{\mathbf{x}} = (1,1)$. For the global minimizer $\bar{\mathbf{x}}$ the active lower-level set is given by:

$$P_o(\bar{\mathbf{x}}) = \{\mathbf{p} \in [0,1]^3 \mid p_3 = 0\ ,\ p_1^2 + p_2^2 = 1\}.$$

This implies that infinitely many constraints, described by the smooth subman-

ifold $P_o(\bar{\mathbf{x}})$ are needed to describe the feasible set locally around $\bar{\mathbf{x}}$. This implies that the reduction approach suggested in [60] is not applicable.

4. ([121] Pg. 303)

$$\begin{aligned} f(\mathbf{x}) &= x^2 \\ g(\mathbf{x}, \mathbf{p}) &= x - p \\ u(\mathbf{x}, \mathbf{p}) &= (p+1)^2 + x^2 \\ X = [-1, 1]\,,\ D &= [-2, 2]. \end{aligned}$$

The upper-level problem is convex in $\mathbf{x}$ for each $\mathbf{p} \in D$. The lower-level problem is convex in $\mathbf{p}$ for each $\mathbf{x} \in X$. The lower-level feasible set is non-empty for all $\mathbf{x} \in [-1, 0) \cup (0, 1]$ and empty for $\mathbf{x} = 0$. Specifically the lower-level feasible set is given by:

$$P(\mathbf{x}) = \begin{cases} 1 \text{ if } x = 0 \\ \emptyset \text{ if } x \neq 0. \end{cases}$$

The feasible set is given by:

$$M = \{\mathbf{x} \in [-1, 1] \mid x \neq 0\}.$$

Although the problem is feasible, the minimum is not attained. This is because the point for which the infimum value of the GSIP is attained, i.e. $\bar{\mathbf{x}} = 0$, is not feasible. However, there exists a sequence of feasible points converging to $\bar{\mathbf{x}}$ that have an empty lower-level feasible set. From a constraint qualification perspective, the source of this irregular behavior comes from the violation of MFCQ at $\mathbf{p} = 1 \in Y(\bar{\mathbf{x}})$. Specifically, at $(x, p) = (0, -1)$ we have $u(x, p) = 0$ and $D_p u(x, p) = 2(p + 1) = 0$.

5. ([104] Pg. 683 , Ex. 3.1)

$$
\begin{aligned}
f(\mathbf{x}) &= -x_1 \\
g(\mathbf{x},\mathbf{p}) &= p_2 \\
u_1(\mathbf{x},\mathbf{p}) &= p_2 - x_1 - x_2 p_1 \\
u_2(\mathbf{x},\mathbf{p}) &= p_2 - p_1^2 - x_2 \\
X = [-5,5]^2 \,,\; D &= [-2,2] \times [-4,4].
\end{aligned}
$$

The upper-level problem is convex in $\mathbf{x}$ for each $\mathbf{p} \in D$. The lower-level problem is nonconvex in $\mathbf{p}$ for each $\mathbf{x} \in X$. With a short calculation it is easy to verify that:

$$
\begin{cases}
\mathbf{y} = (0,-4) \in P(\mathbf{x}),\; x_1 \geq -4 \,,\; x_2 \geq -4 \\
\mathbf{y} = (-2,-4) \in P(\mathbf{x}),\; -5 \leq x_1 \leq -4 \,,\; -5 \leq x_2 \leq -4.
\end{cases}
$$

This implies that $P(\mathbf{x})$ is non-empty for all $\mathbf{x} \in [-5,5]^2$. Thus, the feasible set of the GSIP is a closed subset of the compact set X. This combined with the fact that the problem is feasible implies that the minimum of the GSIP is attained.

The feasible set of the problem is:

$$
\begin{aligned}
M = \{\mathbf{x} \in [-5,5]^2 \mid x_2 \leq -4\} \cup \{\mathbf{x} \in [-5,5]^2 \mid -4 \leq x_2 \leq 0 \,,\; x_2 \geq \frac{1}{2} x_1\} \\
\cup \{\mathbf{x} \in [-5,5]^2 \mid x_2 \geq 0 \,,\; x_2 \leq -\frac{1}{2} x_1\}.
\end{aligned}
$$

The set of global minimizers is:

$$
M_o = \{\mathbf{x} \in [-5,5]^2 \mid x_2 \leq -4 \,,\; x_1 = 5\}.
$$

6. ([105] Pg. 182 , Ex. 5.1)

$$
\begin{aligned}
f(\mathbf{x}) &= 4x_1^2 - x_2 - x_2^2 \\
g(\mathbf{x},\mathbf{p}) &= x_2 - p_2 \\
u_1(\mathbf{x},\mathbf{p}) &= p_1 - x_1 \\
u_2(\mathbf{x},\mathbf{p}) &= p_2 - x_1 \\
u_3(\mathbf{x},\mathbf{p}) &= (p_1 + p_2)^2 - p_3 \\
X = [-3,2]^2 \,,\; D &= [-4,4]^2 \times [0,16].]
\end{aligned}
$$

The upper-level problem is nonconvex in $\mathbf{x}$ for each $\mathbf{p} \in D$. The lower-level problem is convex in $\mathbf{p}$ for each $\mathbf{x} \in X$. With a short calculation it is easy to verify that the lower-level feasible set is:

$$
P(\mathbf{x}) = \begin{cases} [-4,x_1] \times [-4,x_1] \times [0,16] \cap \{\mathbf{p} \in D \mid (p_1+p_2)^2 - p_3 \leq 0\} & \text{if } x_1 < -2 \\ \emptyset & \text{if } x_1 \geq -2. \end{cases}
$$

The feasible set of the problem is:

$$
\begin{aligned}
M = \{\mathbf{x} \in [-3,2]^2 \mid x_1 \geq 0 \,,\; x_2 \leq 0\} \;\cup\; \{\mathbf{x} \in [-3,2]^2 \mid -2 \leq x_1 \leq 0 \,,\; x_2 \leq 4x_1^2\} \\
\cup \{\mathbf{x} \in [-3,2]^2 \mid x_1 < -2\}.
\end{aligned}
$$

As stated in [105] "the feasible set is not closed and not open but it is the union of sets where each is defined by finitely many differentiable functions. Both topological properties cannot appear simultaneously in semi-infinite or finite optimization problems."

Although the feasible set is not closed, the minimum is attained at $\bar{\mathbf{x}} = (0,-3)$. Furthermore, this global minimum is the minimum of the unconstrained problem.

7. ([134] pg. 931 , Ex. 3.3)

$$
\begin{aligned}
f(\mathbf{x}) &= -x_1 \\
g(\mathbf{x},\mathbf{p}) &= 3x_2^2 - p^5 \\
u(\mathbf{x},\mathbf{p}) &= -p^5 - 4x_1^2 - x_2^2 + 1 \\
X = [0,1]^2\ ,\ D &= [-2,0].
\end{aligned}
$$

The upper-level problem is convex in $\mathbf{x}$ for each $\mathbf{p} \in D$. The lower-level problem is nonconvex in $\mathbf{p}$ for each $\mathbf{x} \in X$. With a short calculation it is easy to verify that the lower-level feasible set is:

$$
P(\mathbf{x}) = \begin{cases} [(-4x_1^2 - x_2^2 + 1)^{\frac{1}{5}}, 0] & \text{if } -4x_1^2 - x_2^2 + 1 \leq 0 \\ \emptyset & \text{if } -4x_1^2 - x_2^2 + 1 > 0. \end{cases}
$$

The feasible set of the problem is the union of the following open and closed sets:

$$
M = \{\mathbf{x} \in [0,1]^2 \mid 4x_1^2 + x_2^2 < 1\} \cup (\tfrac{1}{2}, 0).
$$

Although the feasible set is not closed, the minimum is unique and attained at $\bar{\mathbf{x}} = (\frac{1}{2}, 0)$. A way to resolve the apparent nonconvexity from the lower-level problem is to replace p^5 in the both the upper-level and the lower-level constraints with a new variable and make the inner problem linear in p. Finally, it is easy to verify that the extended MFCQ is violated at the global minimizer. However, and as is pointed out in [134], violation of the EMFCQ does not force the violation of the extended Kuhn-Tucker Constraint Qualification (EKTCQ) which actually holds at the global minimizer.

8. ([134] pg. 929 , Ex. 3.2)

$$
\begin{aligned}
f(\mathbf{x}) &= -x_1 \\
g(\mathbf{x},\mathbf{p}) &= -px_2 \\
u(\mathbf{x},\mathbf{p}) &= x_1 - p^2 \\
X = [-1,1]^2 \,,\; D &= [-1,1].
\end{aligned}
$$

The upper-level problem is convex in $\mathbf{x}$ for each $\mathbf{p} \in D$. The lower-level problem is nonconvex in $\mathbf{p}$ for each $\mathbf{x} \in X$. With a short calculation it is easy to verify that the lower-level feasible set is:

$$
P(\mathbf{x}) = \begin{cases} [(-\sqrt{x_1}, \sqrt{x_1}] & \text{if } x_1 \geq 0 \\ \emptyset & \text{if } x_1 < 0. \end{cases}
$$

The feasible set of the problem is the union of the following two closed sets:

$$
M = \{\mathbf{x} \in [-1,1]^2 \mid x_1 \leq 0\} \cup \{\mathbf{x} \in [-1,1]^2 \mid x_1 \geq 0 \,,\; x_2 = 0\}.
$$

The feasible set is closed and the minimum is unique and attained at $\bar{\mathbf{x}} = (1,0)$. It is clear that for each $\mathbf{x} \in X = \{\mathbf{x} \in [-1,1]^2 \mid x_1 \geq 0 \,,\; x_2 = 0\} \subset M$ the index set of active constraints $P_o(\mathbf{x}) = \{\mathbf{p} \in P(\mathbf{x}) \mid g(\mathbf{x},\mathbf{y}) = 0\}$ is non-empty. This implies that essentially the assumption of the existence of a sequence of GSIP slater points arbitrarily close to the global minimizer is violated. However, because the interval extensions are exact for each $\mathbf{x} \in X$ the upper bounding methodology finds the global solution and it does so in one iteration (root node).

9. (Own example)

$$
\begin{aligned}
f(\mathbf{x}) &= x^2 \\
g(\mathbf{x}, \mathbf{p}) &= \exp(x)p^2 - x^2 p \\
u(\mathbf{x}, \mathbf{p}) &= p^2 x^3 - x - 0.2 \\
X = [-1, 1]\ ,\ D &= [0, 1].
\end{aligned}
$$

The upper-level problem is nonconvex in $\mathbf{x}$ for each $\mathbf{p} \in D$. The lower-level problem is nonconvex in $\mathbf{p}$ for each $\mathbf{x} \in X$. With a short calculation it is easy to verify that the lower-level feasible set is:

$$
P(\mathbf{x}) = \begin{cases} \emptyset & \text{if } x < -0.208 \\ [\frac{x+0.2}{x^3}, 1] & \text{if } -0.208 \leq x < 0 \\ [0, 1] & \text{if } x \geq 0. \end{cases}
$$

For $x \geq 0.208$, $p = 1$ is lower-level feasible. Substituting in the upper-level constraint we have:

$$
g(x, 1) = exp(x) - x^2 > 0, \forall -0.208 \leq x \leq 1.
$$

Therefore, feasible set of the problem is the following open set:

$$
M = \{\mathbf{x} \in [-1, 1] \mid x_1 < -0.208\}.
$$

The feasible set is not closed, it is open, and the minimum is not attained at $\bar{\mathbf{x}} = -0.208$. The fact that the $\bar{\mathbf{x}}$ is not attained is due to the violation of the Slater constraint qualification, and thus of the MFCQ, at $\bar{\mathbf{x}}$ for the lower-level

problem:

$$\min_{p\in[0,1]} 0.812p^2 - 0.043p$$
$$\text{s.t.} \quad -0.008p^2 + 0.008 \leq 0.$$

10. ([106] Pg. 203 , Ex. 4-5)

$$f(\mathbf{x}) = x_1 + x_2$$
$$g(\mathbf{x}, \mathbf{p}) = -p$$
$$u_1(\mathbf{x}, \mathbf{p}) = x_1 - p$$
$$u_2(\mathbf{x}, \mathbf{p}) = x_2 - p$$
$$X = [-1, 1]\ ,\ D = [-1, 1].$$

The upper-level problem is convex in $\mathbf{x}$ for each $\mathbf{p} \in D$. The lower-level problem is convex in $\mathbf{p}$ for each $\mathbf{x} \in X$. With a short calculation it is easy to verify that the lower-level feasible set is:

$$P(\mathbf{x}) = [\min\{x_1, x_2\}, 1].$$

The feasible set of the problem is the following open set:

$$M = \{\mathbf{x} \in [-1, 1] \mid \max\{x_1, x_2\} \geq 0\}.$$

The feasible set is closed since for every $\mathbf{x} \in [-1, 1]$, $P(\mathbf{x})$ is not empty. The minimum is attained at two points $\mathbf{x}_1 = (-1, 0)$ and $\mathbf{x}_2 = (0, -1)$, while the algorithm finds the first as the answer for both heuristics. This is an example of a re-entrant corner point at the origin $\mathbf{x} = (0, 0)$.

11. (Own example, extension of test problem #3 in [135])

$$\begin{aligned} f(\mathbf{x}) &= x_1^2 + x_2^2 + x_3^2 \\ g(\mathbf{x}, \mathbf{p}) &= x_1 + x_2 \exp(x_3 p) + \exp(2p) - 2\sin(4p) \\ u(\mathbf{x}, \mathbf{p}) &= 2p - x_2 - 1 \\ X &= [-5, 5]^3 \ , \ D = [0, 1]. \end{aligned}$$

To analyze this example we are going to consider $x_3 = 0$ henceforth. The upper-level problem in convex in $\mathbf{x}$ for each $\mathbf{p} \in D$. The lower-level problem is nonconvex in $\mathbf{p}$ for each $\mathbf{x} \in X$. With a simple calculation the lower-level feasible set is given by:

$$P(\mathbf{x}) = \begin{cases} [0, \frac{x_2+1}{2}] & \text{if } x_2 \geq -1 \\ \emptyset & \text{if } x_2 < -1. \end{cases}$$

It can be seen that for $x_2 = -1$, where $P(\mathbf{x}) = \{0\}$ we obtain:

$$g((x_1, -1, 0), 0) = x_1 - \exp(0) + \exp(0) - 2\sin(0) = x_1.$$

Therefore, the feasible set at $x_3 = 0$, $x_2 = -1$ is

$$M_{x_3=0, x_2=-1} = \{x_1 \in [-5, 5] \mid x_1 \leq 0\}.$$

Assume a point $\bar{\mathbf{x}} = (\bar{x}_1, \bar{x}_2, \bar{x}_3)$ for which $\bar{x}_1 > 0$, $\bar{x}_2 = -1$ and $\bar{x}_3 = 0$. This point is clearly infeasible. However, allowing x_2 to be arbitrarily smaller than -1, say $x_2 = \bar{x}_2 - \epsilon$, $\epsilon > 0$, the point $\mathbf{x} = (\bar{x}_1, -1 - \epsilon, 0)$ is feasible, and thus there exists a sequence of feasible points (with an empty lower-level set) converging to $\bar{\mathbf{x}}$ (which is infeasible for the GSIP). This clearly implies that the feasible set of the GSIP is not closed.

Now we will show that although the feasible set in not-closed, the minimum of

the GSIP is indeed attained. Specifically consider the point $\bar{\mathbf{x}} = (-0.5, -0.5, 0.0)$. The lower-level feasible set is given by $P(\bar{\mathbf{x}}) = [0, 0.25]$. The lower-level problem for $\bar{\mathbf{x}}$ is formulated as:

$$G = \max_{p \in [0,0.25]} -1 + \exp(2p) - 2\sin(4p).$$

It can be easily calculated that $G = 0$ and thus $\bar{\mathbf{x}}$ is feasible for the GSIP. The objective function value for the current point is $f(\bar{\mathbf{x}}) = 0.5$. The only possibility for the minimum to not be attained is if the set of global minimizers consists of points each of which is infeasible but arbitrarily close to points with an empty lower-level feasible set which are, of course, feasible. The only candidates to satisfy this condition are points of the form $\mathbf{x} = (x_1, -1, x_3)$. However, the objective function value of such points is greater or equal to 1 and since we already have obtained a feasible point with objective function value of 0.5, these points cannot have a objective function value equal to the GSIP infimum. Thus, the minimum is attained and we will show that the point $\bar{\mathbf{x}}$ is the unique global minimizer.

The search for the minimum of the GSIP will be performed only for points $\mathbf{x}$ for which the lower-level feasible set in non-empty. This is because all points $\mathbf{x}$ for which the lower-level feasible set is empty have an objective function value greater than $f(\bar{\mathbf{x}}) = 0.5$. For all points with a non-empty lower-level feasible set, $p = 0$ is lower-level feasible and as such the points $\mathbf{x} = (x_1, x_2, x_3)$ that are feasible must satisfy:

$$g(\mathbf{x}, 0) = x_1 + x_2 + 1 \leq 0 \;\; \Rightarrow \;\; x_1 + x_2 \leq -1. \tag{6.2}$$

With the use of (6.2) we now form the following optimization problem

$$\begin{aligned} \min_{\mathbf{x} \in [-1,0]^3} \; & f(\mathbf{x}) = x_1^2 + x_2^2 + x_3^2 \\ \text{s.t.} \; & x_1 + x_2 \leq -1. \end{aligned} \tag{6.3}$$

The argument of (6.3) is obviously $\bar{\mathbf{x}} = (-0.5, -0.5, 0.0)$. Since this point is indeed feasible, it is the unique global minimizer of the GSIP.

Furthermore, for $x_3 = 0$, the lower-level problem involves the maximization of g which is convex $\in \mathbf{p}$ for all $\mathbf{x} \in X$. This implies that for $x_2 \geq -1$, the infinite number of constraints at each $\mathbf{x}$ can be substituted with the constraints at the bounds of the lower-level feasible interval. Therefore, feasibility can be checked by the constraints at $p_1 = 0$ and at $p_2 = \frac{x_2+1}{2}$. Thus, the feasible set of the GSIP at $x_3 = 0$ is:

$$M = \{\mathbf{x} \in [-5,5]^3 \mid x_2 < -1\} \cup$$
$$\{\mathbf{x} \in [-5,5]^3 \mid x_1 + x_2 \leq -1\,,\; x_1 + x_2 + \exp(x_2 + 1) - 2\sin(2x_2 + 2) \leq 0\}.$$

12. (Own example)

$$f(\mathbf{x}) = x^2$$
$$g(\mathbf{x}, \mathbf{p}) = \frac{1}{2}p^3 - x^2$$
$$u(\mathbf{x}, \mathbf{p}) = x^2 - p^2$$
$$X = [-1, 1]\,,\; D = [0, 1].$$

The upper-level problem is nonconvex in $\mathbf{x}$ for each $\mathbf{p} \in D$. The lower-level problem is nonconvex in $\mathbf{p}$ for each $\mathbf{x} \in X$. With a short calculation it is easy to verify that the lower-level feasible set is:

$$P(\mathbf{x}) = [\sqrt{x^2}, 1].$$

The feasible set of the problem is the following closed set:

$$M = \{\mathbf{x} \in [-1, 1] \mid x \geq -\tfrac{\sqrt{2}}{2}\}.$$

The feasible set is closed since for every $\mathbf{x} \in [-1, 1]$, the lower-level feasible set $P(\mathbf{x})$ is not empty. Specifically, $\mathbf{p} = 1$ is lower-level feasible for each $\mathbf{x} \in X$.

The minimum is attained at two points $\mathbf{x}_1 = \frac{\sqrt{2}}{2}$ and $\mathbf{x}_2 = -\frac{\sqrt{2}}{2}$, while the algorithm finds the latter as the answer.

13. (Own example, extension of test problem #5 in [135])

$$\begin{aligned} f(\mathbf{x}) &= \exp(x_1) + \exp(x_2) + \exp(x_3) \\ g(\mathbf{x}, \mathbf{p}) &= \frac{1}{1+p^2} - x_1 - x_2 p - x_3 p^2 \\ u(\mathbf{x}, \mathbf{p}) &= x_2 + x_3 - \frac{1}{2} p \\ X &= [-1, 1]^3 \, , \; D = [0, 1]. \end{aligned}$$

The upper-level problem in convex in $\mathbf{x}$ for each $\mathbf{p} \in D$. The lower-level problem is convex in $\mathbf{p}$ for each $\mathbf{x} \in X_1 = \{\mathbf{x} = (x_1, x_2, x_3) \mid x_3 \geq 0.25\}$ and nonconvex in $\mathbf{p}$ for each $\mathbf{x} \in X \setminus X_1$.

With a simple calculation the lower-level feasible set is given by:

$$P(\mathbf{x}) = \begin{cases} [2(x_2 + x_3), 1] & \text{if } x_2 + x_3 \leq \frac{1}{2} \\ \emptyset & \text{if } x_2 + x_3 > \frac{1}{2}. \end{cases}$$

Next, we will show that the feasible set is not closed. To prove this, consider the point $\bar{\mathbf{x}} = (-1, 0.25, 0.25)$. The lower-level feasible set for this point is given by the singleton set $P(\bar{\mathbf{x}}) = \{1\}$. Evaluating the upper-level constraint we see that $g(\bar{\mathbf{x}}, 1) = 1$. This implies that $\bar{\mathbf{x}}$ is infeasible for the GSIP. However, for $\mathbf{x}$ for which $x_2 + x_3 > 0.5$, $\mathbf{x}$ has an empty lower-level feasible set, which implies that $\mathbf{x}$ is feasible for the GSIP. This suggests that there exists a sequence of points that are feasible for the GSIP converging to an infeasible point. From this, we conclude that the feasible set of the GSIP is not closed.

We are going to examine the feasible set at the projection on x_1 and specifically at $x_1 = -1$. Furthermore, we are going to focus on $\mathbf{x}$ for which the lower-level feasible set is non-empty. Thus we will consider $\mathbf{x}$ for which $x_2 + x_3 \leq 0.5$. This implies that $p = 1$ is lower-level feasible and as such the upper-level constraint

can be written as:

$$g(\mathbf{x}, 1) = \frac{3}{2} - (x_2 + x_3) \leq 0 \;\; \Rightarrow \;\; x_2 + x_3 \geq \frac{3}{2}. \tag{6.4}$$

(6.4) suggests that for $\mathbf{x}$ for which $x_2 + x_3 \leq \frac{1}{2}$, $\mathbf{x}$ is infeasible for the GSIP. Finally, for $x_1 = -1$ the feasible set of the GSIP is given by the following open set:

$$M_{x_1=-1} = \{(x_2, x_3) \in [-1, 1]^2 \mid x_2 + x_3 > 0.5\}.$$

Furthermore, we are going to show that $\bar{\mathbf{x}}$ is indeed a point of infimum objective function value. We have already showed that there exists a sequence of feasible points converging to $\bar{\mathbf{x}}$. To complete the proof we need to show that all feasible points of the GSIP have an objective function value greater than $f(\bar{\mathbf{x}})$ $= \exp(-1) + 2\exp(0.25) \approx 2.93$.

For $\mathbf{x}$ for which $x_2 + x_3 > 0.5$ it is obvious that $f(\bar{\mathbf{x}})$ is the infimum. Assume $\mathbf{x}$ for which $x_2 + x_3 = \alpha$, $\alpha \leq 0.5$. $p = 1$ is lower-level feasible for $\mathbf{x}$ and the corresponding upper-level constraint is:

$$g(\mathbf{x}, 1) = \frac{1}{2} - x_1 - (x_2 + x_3) \leq 0 \;\; \Rightarrow \;\; x_1 \geq \frac{1}{2} - \alpha. \tag{6.5}$$

(6.5) is a necessary condition for feasibility of $\mathbf{x}$. The optimum objective function value would be attained at $\mathbf{x} = (\frac{\alpha}{2}, \frac{\alpha}{2}, \frac{1}{2} - \alpha)$. Furthermore, calculating the optimum solution with respect to α we obtain the following problem:

$$F = \min_{\alpha \in [-2, 0.5]} \exp(\frac{1}{2} - a) + 2\exp(\frac{a}{2}). \tag{6.6}$$

where the lower bound of α is determined by the lower bounds on x_2 and x_3. Solving (6.6) we obtain that $F = 3.55 > 2.93$. This implies that the necessary conditions for feasibility enforce that the minimum possible objective function value for feasible points to be greater than the objective function value of $\bar{\mathbf{x}}$. This concludes the proof that $f(\bar{\mathbf{x}})$ is the infimum objective function value for

the GSIP.

14. (Own example, extension of test problem #7 in [135])

$$
\begin{aligned}
f(\mathbf{x}) &= x_1^2 + x_2^2 + x_3^2 \\
g(\mathbf{x}, \mathbf{p}) &= x_1(p_1 + p_2^2 + 1) + x_2(p_1 p_2 - p_2^2) + x_3(p_1 p_2 + p_2^2 + p_2) + 1 \\
u(\mathbf{x}, \mathbf{p}) &= x_1^2 - p_1^2 \\
X &= [-1, 0]^3 \, , \; D = [0, 1]^2.
\end{aligned}
$$

The upper-level problem is convex in $\mathbf{x}$ for each $\mathbf{p} \in D$. The lower-level problem is nonconvex in $\mathbf{p}$ for each $\mathbf{x} \in X$.

With a short calculation it is easy to see that the lower-level feasible set is:

$$P(\mathbf{x}) = \{\mathbf{p} \in [0, 1]^2 \mid [\sqrt{x_1^2}, 1] \times [0, 1]\}. \tag{6.7}$$

Therefore, for all $\mathbf{x} \in [-1, 1]$ the lower-level feasible set is not empty and specifically for each $\mathbf{x} \in [-1, 1]$ there exists $p_1 \in [0, 1]$ such that $(p_1, 0)$ is lower-level feasible. Using this point and evaluating g we obtain that

$$g = x_1 p_1 + x_1 + 1. \tag{6.8}$$

From (6.7) it is obvious that $\mathbf{p} = (-x_1, 0)$ is lower-level feasible for each $\mathbf{x} \in X$. From (6.8), substituting with $p_1 = -x_1$ and satisfying (6.8) as an inequality constraint (taking into consideration the bound constraints for x_1) we obtain that:

$$-1 \leq x_1 \leq \frac{1 - \sqrt{5}}{2} \approx -0.6180.$$

Thus an outer approximation, i.e. a relaxation of the feasible set M is the following set $\bar{M}$:

$$M \subset \bar{M} = \{\mathbf{x} \in [-1, 1]^3 \mid x_1 \leq -0.6180\}. \tag{6.9}$$

We will show that indeed $M = \bar{M}$. Taking into consideration the functional form of g we see that g can be written as:

$$g = x_1 P_1(\mathbf{p}) + x_2 P_2(\mathbf{p}) + x_3 P_3(\mathbf{p}) + 1. \tag{6.10}$$

In (6.10) we observe that $P_1(\mathbf{p}), P_3(\mathbf{p}) \geq 0, \forall\ \mathbf{p} \in [0,1]^2$. Furthermore, since for every $\mathbf{x} \in X$, there exists $p_1 \in [0,1]$ such that $\mathbf{p} = (p_1, 1)$ is lower-level feasible, $\min_{\mathbf{p} \in P(\mathbf{x})} P_2(\mathbf{p}) \leq 0\ \forall\ \mathbf{x} \in [-1,0]$. Thus, if $\bar{\mathbf{x}} = (-0,6180, -1.0, 0.0)$ is feasible then every point $\mathbf{x}$ for which $x_1 \leq -0.6180$, $-1 \leq x_2 \leq 0$, $-1 \leq x_3 \leq 0$ will also be feasible. In other words we are fixing x_1 and x_3 to their upper bounds and x_2 to its lower bound (based on the observation on the signs of $P_1(\mathbf{p}), P_2(\mathbf{p})$ and $P_3(\mathbf{p})$).

Performing a global minimization procedure on $g(\bar{\mathbf{x}}, \mathbf{p})$ on $P(\bar{\mathbf{x}})$ we find that indeed $\bar{\mathbf{x}}$ is feasible. This implies that every point in $\bar{M}$ is feasible and thus:

$$M \subset \bar{M} = \{\mathbf{x} \in [-1,1]^3 \mid x_1 \leq -0.6180\}. \tag{6.11}$$

From relations (6.9) and (6.11) we obtain that

$$M = \bar{M}.$$

This implies that the feasible set is closed and since it is non-empty, the minimum of this problem is attained. Furthermore, it is easy to see that the minimum is unique and attained at $\bar{\mathbf{x}} = (-0.6180, 0.0, 0.0)$.

15. (Own example, extension of test problem K in [135])

$$\begin{aligned} f(\mathbf{x}) &= x_2^2 - 4x_2 \\ g(\mathbf{x}, \mathbf{p}) &= x_1 \cos(p) + x_2 \sin(p) - 1 \\ u(\mathbf{x}, \mathbf{p}) &= -p_1^2 - \frac{7}{4}x_2 + \frac{23}{4} \\ X &= [0, 2]^2 \, , \; D = [0, \pi]. \end{aligned}$$

The upper-level problem is convex in $\mathbf{x}$ for each $\mathbf{p} \in D$. The lower-level problem is nonconvex in $\mathbf{p}$ for each $\mathbf{x} \in X$.

With a short calculation it is easy to see that the lower-level feasible set is:

$$P(\mathbf{x}) = \{\mathbf{p} \in [0, \pi] \;\; | \;\; [\sqrt{\tfrac{23}{4} - \tfrac{7}{4}x_2}, \pi]\}.$$

Therefore, for all $\mathbf{x} \in [0, 2]^2$ the lower-level feasible set is not empty. Taking into consideration that the GSIP is indeed feasible (as we will show later on), this implies that the minimum of the GSIP is attained.

Next, we will show that the unconstrained minima are not attained for this problem. Specifically, the set of unconstrained minima is

$$M_{unc} = \{\mathbf{x} \in [0, 2]^2 \; | \; x_2 = 2\}.$$

The lower-level feasible set for each $\mathbf{x} \in M_{unc}$ is given by:

$$P(\mathbf{x})_{unc} = \{\mathbf{p} \in [0, \pi] \; | \; \tfrac{3}{2} \leq p \leq \pi\}.$$

It is clear that $\mathbf{p} = \frac{\pi}{2}$ is lower-level feasible for the set of unconstrained minima. This implies that the upper-level constraint

$$g = x_1 \cdot 0 + 2 \cdot 1 - 1 = 1 > 0.$$

Therefore, the set of unconstrained minima is infeasible for the constrained

GSIP.

Next, we will attempt to describe the feasible set of the problem. For this purpose we will define two complementary sets, X_1 and X_2 that satisfy:

$$X_1 \cup X_2 = [0,2]^2 \quad , \quad X_1 \cap X_2 = \emptyset$$
$$X_1 = \{\mathbf{x} \in [0,2]^2 \mid \tfrac{\pi}{2} \notin P(\mathbf{x})\} = \{\mathbf{x} \in [0,2]^2 \mid x_2 < 1.8757\}$$
$$X_2 = \{\mathbf{x} \in [0,2]^2 \mid \tfrac{\pi}{2} \in P(\mathbf{x})\} = \{\mathbf{x} \in [0,2]^2 \mid x_2 \geq 1.8757\}.$$

For $\mathbf{x} \in X_2$ the upper-level constraint g at $\bar{p} = \frac{\pi}{2}$ evaluates at:

$$g(\mathbf{x}, \bar{p}) = x_2 - 1 > 0.$$

This implies that for each $\mathbf{x} \in X_2$, $\mathbf{x}$ is infeasible. Next we will consider $\mathbf{x} \in X_1$. For all these points it is clear that $P(\mathbf{x}) \subset [\frac{\pi}{2}, \pi]$. This implies that for any given $\mathbf{x} \in X_1$, $\cos(\mathbf{p})$ and $\sin(\mathbf{p})$ are monotonically decreasing functions on the interval described by $P(\mathbf{x})$. This implies that $g(\mathbf{x}, \cdot)$ is also monotonically decreasing on the interval described by $P(\mathbf{x})$. Therefore, a necessary and sufficient condition for feasibility of $\mathbf{x} \in X_1$ is that the constraint function g evaluates to a non-positive number for the lower bound on the lower-level feasible interval described by $P(\mathbf{x})$. The lower bound of this interval is given by $p = \sqrt{\frac{23}{4} - \frac{7}{4}x_2}$. Thus the feasible set of the GSIP can be described as follows:

$$M = \{\mathbf{x} \in [0,2]^2\} \cap \{x_2 < 1.8757\}$$
$$\cap \{x_1 \cos(\sqrt{\frac{23}{4} - \frac{7}{4}x_2}) + x_2 \sin(\sqrt{\frac{23}{4} - \frac{7}{4}x_2}) - 1 \leq 0\}.$$

Furthermore, we will calculate, algebraically, the global minimum of the problem. Keeping in mind that f is monotonically decreasing with respect to x_2 on $[0,2]$ we conclude that we are seeking for the maximum feasible value of $x_2 \in [0,2]$. Furthermore, since $M \subset X_1$, $\cos(\mathbf{p})$ is non-positive for all $\mathbf{p} \in P(\mathbf{x})$, $\mathbf{x} \in M$. Finally, since x_1 does not appear in f we will fix $x_1 = 2$ and locate the largest value of x_2, namely $\bar{x}_2$ for which $(x_1, \bar{x}_2) \in M$. We claim that, provided

such a value of x_2 exists, the point $(x_1, \bar{x}_2)$ will be the unique global minimizer of the problem. In conclusion, the global minimizer will be the solution of the following optimization problem:

$$\begin{aligned} \min_{\mathbf{x}\in[0,2]\times[0,1.8757]} \; & x_2^2 - 4x_2 \\ \text{s.t. } & x_1\cos(\sqrt{\frac{23}{4} - \frac{7}{4}x_2}) + x_2\sin(\sqrt{\frac{23}{4} - \frac{7}{4}x_2}) \leq 0 \\ & x_1 = 2. \end{aligned}$$

The solution to this problem is $\bar{\mathbf{x}} = \{2, 1.4619\}$ and that is the unique global minimizer of the original GSIP. Since the lower-level problem is not convex for some $\mathbf{x} \in X$ and feasible for all $\mathbf{x} \in X$, replacing this problem by its KKT conditions provides only a relaxation of the original GSIP.

16. (Own example, extension of test problem #9 in [135])

$$\begin{aligned} f(\mathbf{x}) &= -4x_1 - \frac{2}{3}(x_4 + x_6) \\ g(\mathbf{x}, \mathbf{p}) &= x_1 + x_2p_1 + x_3p_2 + x_4p_1^2 + x_5p_1p_2 + x_6p_2^2 - 1.0 \\ u(\mathbf{x}, \mathbf{p}) &= x_1\cos(p_1) - x_2\sin(p_1) \\ X &= [0,2]^6 \,,\; D = [-1,1]^2. \end{aligned}$$

The upper-level problem is convex in $\mathbf{x}$ for each $\mathbf{p} \in D$. The lower-level problem is convex in $\mathbf{p}$ for each $\mathbf{x} \in X_0 = \{\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6) \mid x_1 = x_2 = 0,\; 4x_4x_6 - x_5^2 \geq 0\}$ and not convex in $\mathbf{p}$ for each $\mathbf{x} \in X \setminus X_0$.

With a short calculation it is easy to see that the lower-level feasible set is:

$$P(\mathbf{x}) = \begin{cases} \{\mathbf{p} \in [-1,1]^2 \;\mid\; [\max\{\tan^{-1}(\frac{x_1}{x_2}), 0\}, 1] \times [-1,1]\} & \text{if} \quad \tan^{-1}(\frac{x_1}{x_2}) \leq 1, x_2 \neq 0 \\ \emptyset & \text{if} \quad \tan^{-1}(\frac{x_1}{x_2}) > 1, x_2 \neq 0 \\ \emptyset & \text{if} \quad x_2 = 0, x_1 \neq 0 \\ [-1,1]^2 & \text{if} \quad x_2 = 0, x_1 = 0 \end{cases}$$

We are going to examine the feasible subset of the points $\mathbf{x}$ that belong in the first and fourth categories, namely X_1 and X_4 respectively, in the lower-level feasible set that was described above. Obviously, the second and third categories are subsets of the feasible set (points with an empty lower-level feasible set).

Clearly, $\mathbf{p} = (1, 1)$ is lower-level feasible for each $\mathbf{x} \in X_1$. Furthermore, for each $\mathbf{x} \in X_1$ the maximum of the upper-level constraint g is attained at $p = (1, 1)$:

$$g(\mathbf{x}, (1, 1)) = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 - 1.0.$$

It is clear that a necessary and sufficient condition for $\mathbf{x} \in X_1$ to be feasible is that $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 1$. It is trivial to show that the same condition along with the extra requirement that $x_1 = x_2 = 0$ has to hold for $\mathbf{x} \in X_4$.

Thus the feasible set of the GSIP is:

$$\begin{gathered} M = \{\mathbf{x} \in [0, 2]^6 \mid \tan^{-1}(\tfrac{x_1}{x_2}) \leq 1 \, , \; x_2 \neq 0 \, , \; x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 1\} \\ \cup \{\mathbf{x} \in [0, 2]^6 \mid \tan^{-1}(\tfrac{x_1}{x_2}) > 1 \, , \; x_2 \neq 0\} \cup \{\mathbf{x} \in [0, 2]^6 \mid x_1 \neq 0 \, , \; x_2 = 0\} \\ \cup \{\mathbf{x} \in [0, 2]^6 \mid x_1 = x_2 = 0 \, , \; x_3 + x_4 + x_5 + x_6 \leq 1\}. \end{gathered}$$

The set of the unconstrained minima of f is the following:

$$M_{unc} = \{\mathbf{x} \in [0, 2]^2 \;\mid\; x_1 = x_4 = x_6 = 2\}.$$

Both the heuristics locate the point $\mathbf{x} = (2, 0.25, 1, 2, 1, 2)$ as the candidate for global optimality. It is easy to see that this point belongs in the 2nd category of feasible points and since it is indeed a minimum for the unconstrained problem it is a minimum of the GSIP.

However, using the set M_{unc} we will show that the feasible set of the GSIP is not closed. Specifically we will show that the set of global minima is not closed which implies that the feasible set is also not closed. Specifically, consider point $\bar{\mathbf{x}} = (2, 1.2841, 1, 2, 1, 2)$. The coordinates x_1 and x_2 were chosen in order to satisfy that $\tan^{-1}(\frac{x_1}{x_2}) = 1$ and that $\mathbf{x} \in M_{unc}$. By a quick examination,

$\mathbf{p} = (1, 1)$ is lower-level feasible for this point but provides a positive value for the upper-level constraint. This implies that $\bar{\mathbf{x}}$ is infeasible for the GSIP. For each $\mathbf{x} = (2, 1.2841 - \epsilon, 1, 2, 1, 2)$, $\epsilon > 0$ it follows that $\mathbf{x} \in X_2 \subset M$. This implies that there exists a sequence of feasible points for the GSIP that converge to an infeasible point. This concludes our observation that the feasible set of the GSIP is not closed.

The set of global minima for the GSIP is set:

$$M_{con} = \{\mathbf{x} \in [0, 2]^6 \mid x_1 = x_4 = x_6 = 2, x_2 < 1.2841\}.$$

Chapter 7

Kinetic Model Reduction

The goal of this chapter is to analyze the problem of kinetic model reduction (KMR). The analysis will focus on equivalent reformulations of KMR within the context of global optimization and also on a new methodology to provide valid reduction ranges for KMR. In Section 7.1 we provide a brief introduction on KMR and in Section 7.2 we demonstrate that KMR is a design centering problem. In Section 7.3 we will analyze and comment on literature methods that provide estimates on regions of valid reduction. Then, we will formulate KMR as a global optimization problem and demonstrate that this problem is a special case of a generalized semi-infinite program (GSIP) in Section 7.4. In Section 7.5 we provide equivalent reformulations of KMR and in Section 7.6 we comment on the limitations of global optimization methods in semi-infinite, generalized semi-infinite and bilevel programming to provide a tractable approximation for KMR. In Section 7.7 we provide a new method to calculate ranges of guaranteed valid reduction for KMR and in Section 7.8 we analyze the application of the aforementioned method to the combustion of hydrogen.

7.1 Introduction

Reacting flow simulations are known to be computationally expensive and sometimes intractable especially when they incorporate both the chemistry (thousands of chemical reactions and species) and the transport phenomena often occurring at different

time scales [11, 25, 90, 96, 111]. Kinetic model reduction (KMR) is a methodology by which full kinetic models that describe the reacting flow are replaced by reduced kinetic models in regions of the concentration-temperature space where it is believed that they accurately represent the full model.

Consider a full kinetic model, denoted by Γ, that describes the combustion kinetics of a chemical species. Assume that Γ involves n_s participating species and n_r participating reactions. Now consider a reduced model, denoted by $\bar{\Gamma}$ that also describes the combustion of this chemical species but which involves a reduced number of participating species $\bar{n}_s$, $\bar{n}_s \leq n_s$ and/or a reduced number of participating reactions $\bar{n}_r$, $\bar{n}_r \leq n_r$. Within the context of KMR, there are two important questions:

1. (Optimally reduced model): Given a specific point in temperature and concentration space, a full kinetic model Γ describing the combustion mechanism and a description tolerance ε, what is the smallest possible reduced mechanism $\bar{\Gamma}$ that reproduces (to some metric) the full model within ε?

2. (Optimal ranges of valid reduction): Given a full kinetic model Γ, a reduced kinetic model $\bar{\Gamma}$ and a description tolerance ε, what is the maximum range of validity of $\bar{\Gamma}$? Therefore, this question can be posed as: what is the maximum range in temperature and concentration space such that the difference between the full model Γ and the reduced model $\bar{\Gamma}$ for every point in that range is less than ε?

The question of finding optimally reduced kinetic models is out of the scope of this thesis. The reader is referred to [9, 10, 12, 28, 39, 41, 88] for a number of different algorithms that have been developed to address this matter.

In the following sections we are going to emphasize on finding optimal ranges of valid reduction for KMR.

7.2 KMR as a Design Centering Problem

The problem of finding a (optimal) valid range of reduction is actually a design centering problem. To illustrate this point we provide the following example.

Example 7.1. Assume a hypothetical combustion mechanism that involves two species, A and B, and four reactions. We also make the following assumptions:

1. Γ is the full kinetic model of the mechanism, $\bar{\Gamma}$ is the reduced one and ε is the description accuracy.

2. There exist lower and upper bounds for the concentrations of the participating species, $C_{A,low}, C_{B,low}$ and $C_{A,up}, C_{B,up}$ respectively.

3. There exists a known point $\bar{C} = (\bar{C}_A, \bar{C}_B)$ such that the reduced model $\bar{\Gamma}$ reproduces the full model Γ within the prescribe accuracy ε. We will refer to this point as the nominal point of reduction.

4. There exists a non-degenerate region (i.e. a geometric region with a non-empty interior, e.g. not a line) around $\bar{C}$ for which the full model can be reproduced with the reduced model within the prescribed accuracy ε.

The first goal is to inscribe a box (in this 2-D case a rectangle) within the total valid reduction range (feasibility problem). The second and numerically more challenging goal is to maximize the area of the inscribed box to capture as much of the total valid reduction range as possible (flexibility problem).

Figure 7-1 illustrates the feasibility problem. Note that $\bar{C}$ does not actually have to lie in the interior of the valid reduction range. However, this reduction range should be non-degenerate. To ensure that, the point $\bar{C}$ has to be chosen such that the maximum difference between the full model Γ and the reduced kinetic model $\bar{\Gamma}$ is strictly less than ε. Then, by simple continuity of the participating functions, there exists some set around $\bar{C}$ that belongs to the interior of the valid reduction range.

Figure 7-2 illustrates the flexibility problem. In the general case, the valid region of reduction is nonconvex. Therefore, inscribing a box and maximizing its volume

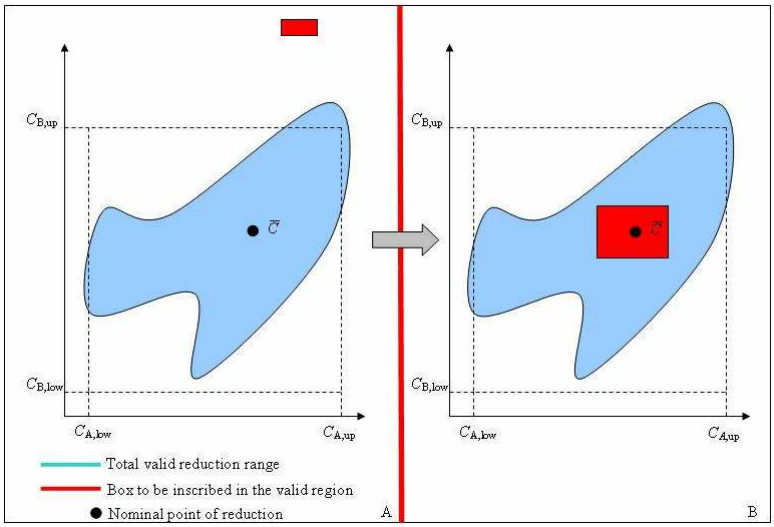

Figure 7-1: Feasibility Problem in KMR

(area in the 2-D case) amounts to solving, at the very best, a nonconvex optimization problem globally.

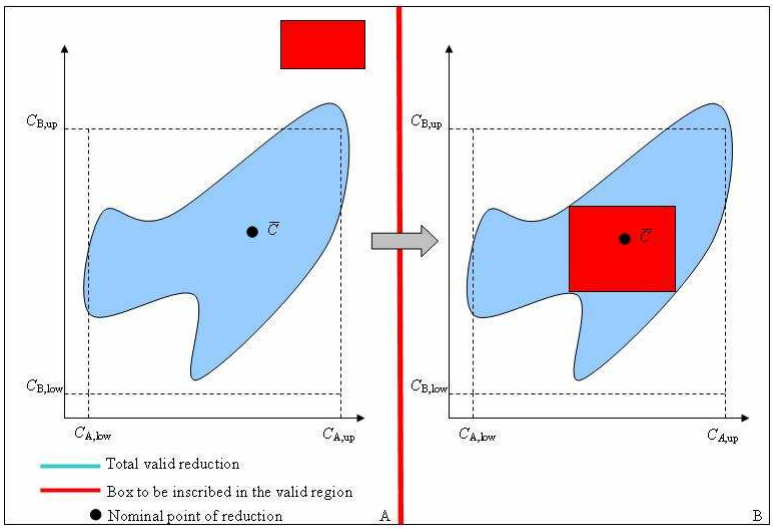

Figure 7-2: Flexibility Problem in KMR

It is immediately clear that KMR is a design centering problem. Following the notation of Section 4.1:

1. The container (host) set is the total valid range of reduction.

2. The parameterized body is the cube (rectangle in the 2-case).

3. The optimization variables are obviously the concentrations of the two species C_A and C_B, the lower and upper bounds of which are used to parameterize the inscribed box.

7.3 Literature Review

Numerical algorithms that address the feasibility and flexibility subproblems of KMR have been developed in [12, 49, 50, 90, 109, 112].

The common element in [49, 50, 109] is the approximation of the valid range of reduction using the convex hull of a finite number of sample points. More specifically, the authors generate a finite number of sample points either uniformly or by an active-set strategy and evaluate them for feasibility. Then, they take the convex hull of the feasible points and claim that this set (along with certain heuristic rules) is an approximation to the valid range of reduction. Figure 7-3 illustrates the method: It can be seen that although the convex hull may give a rough approximation of the

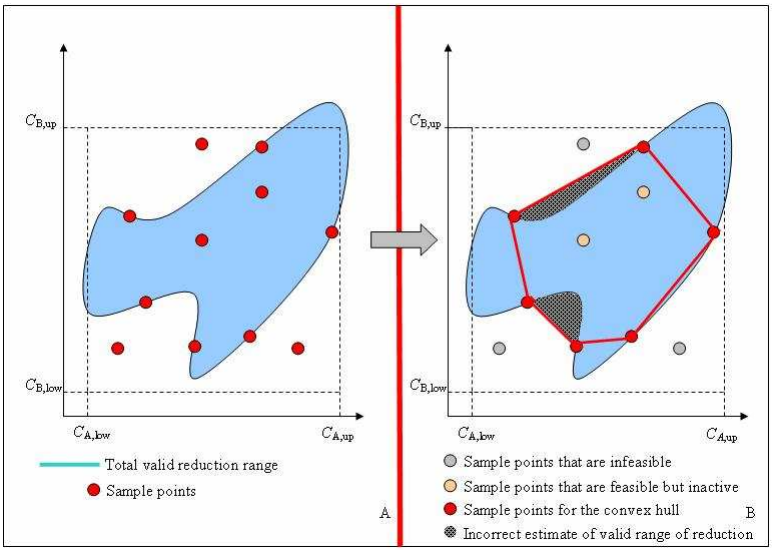

Figure 7-3: Convex Hull for the Flexibility Problem

valid range of reduction it can also overestimate this region. For the case of KMR which is inherently nonconvex and for which the valid region of reduction comprises, very often, disconnected sets, this wrong estimation leads to considerable numerical

errors.

Banerjee and Ierapetritou [12] recognize the limitation of this method and propose to use the α-hulls instead of the convex hulls. The α-hulls, having their origin in pattern recognition and shape reconstruction, approximate the feasible set in a more precise way than the convex hulls by introducing curvature (with negative convexity) between the sample points. However this method suffers from two drawbacks:

1. It relies on reconstructing the feasible set from a finite set of points. Even with the addition of curvature, this method can still overestimate, or miscalculate, the valid range of reduction.

2. The form of the feasible set of the approximated range of reduction is not particularly useful to numerical software. This is because the range is no longer given in a box-form but as a nonconvex set.

Song et al. [112] propose the use of local linearizations of the constraint functions (rate expressions) around some nominal values of the model parameters to extrapolate the rates in a parameter range. In order to control the error of the adaptation of the rates for a larger parameter set, the authors test the vertices of the identified hyper-rectangle. Despite the fact that this method tends to account for the numerical error introduced by the linearization, examining the vertices of the hyper-rectangle can guarantee feasibility of the extrapolation only when the feasible set is convex. As mentioned before, KMR is inherently nonconvex so the maximal error can occur in the boundary of the hyper-rectangle that is not an edge or even in the interior of the hyper-rectangle.

Oluwole et al. [90] address the issue of feasibility in KMR and introduce the first, to our best knowledge, numerical procedure that guarantees the generation of a valid range of reduction. The authors use the interval-constrained reformulation by Bhattacharjee et al. [26] and adaptively (using sensitivities at the edge of the hyper-rectangle) shrink the estimated box of validity until it is rendered feasible. The authors choose to implement Taylor models for the interval extensions which, especially for the polynomial character of the kinetic expressions, provide a much tighter

inclusion than the natural interval extensions. We emphasize that the approach in [90] targets feasibility and not flexibility for two reasons:

1. The procedure ends when the first feasible box has been found. Therefore, there is no procedure to extend the box when it is ensured that a feasible one has already been found.

2. The interval-constrained reformulation was only used to evaluate the constraint functions (difference between the full and kinetic models) and not to formulate the feasible-box search as an optimization problem.

7.4 KMR Formulated as a GSIP

Recall that the flexibility problem in KMR formulated as an optimization problem can be stated in the following way: "Given a full kinetic model Γ, a reduced kinetic model Γ_{red} and a description tolerance ε, what is the maximum volume of a hyper-rectangle for which the reduced model reproduces the full kinetic model within ε for every point in that hyper-rectangle?". This leads to the following optimization problem:

$$\begin{aligned}
&\max_{\mathbf{x}^l,\mathbf{x}^u} V(\mathbf{x}^l,\mathbf{x}^u)\\
&\text{s.t. } |\Gamma_{red,j}(\mathbf{p}) - \Gamma_j(\mathbf{p})| \leq \varepsilon_{tol}, j = 1,\ldots,n_s+1,\ \forall \mathbf{p} \in P(\mathbf{x}^l,\mathbf{x}^u)\\
&\qquad P(\mathbf{x}^l,\mathbf{x}^u) = \{\mathbf{p} \in [\mathbf{x}_{low},\mathbf{x}_{up}] : x_j^l \leq p_j \leq x_j^u\}. \qquad (7.1)\\
&\qquad \mathbf{x}^l,\mathbf{x}^u \in [\mathbf{x}_{low},\mathbf{x}_{up}].
\end{aligned}$$

Recall that the general form of a GSIP is:

$$\begin{aligned} &\inf_{\mathbf{x}\in\mathbf{X}} f(\mathbf{x}) \\ &\text{s.t. } g(\mathbf{x},\mathbf{p}) \leq 0,\ \forall \mathbf{p} \in P(\mathbf{x}) \\ &\qquad P(\mathbf{x}) = \{\mathbf{p} \in D : u_j(\mathbf{x},\mathbf{p}) \leq 0, j \in J\} \\ &\qquad\qquad X \subset \mathbb{R}^{n_x},\ D \subset \mathbb{R}^{n_p}, |J| < \infty. \end{aligned} \tag{7.2}$$

Comparing relations (7.1) and (7.2) we observe the following:

1. The objective function f in problem (7.2) is the volume of the hyper-rectangle defined by the lower and upper bounds of the optimization variables x_j in (7.1). Specifically $V(x_j^l, x_j^u) = \prod_1^{n_s+1}(x_j^u - x_j^l)$.

2. The optimization variables $\mathbf{x}$ refer to the concentrations of the n_s species and to the temperature of the system. Since for each of the participating species and temperature we define a lower-bounding and an upper-bounding variable, the total number of decision variables is $2(n_s + 1)$ ($X \subset \mathbb{R}^{2(n_s+1)}$).

3. The number of parameters of the problem, $\mathbf{p}$, is $n_s + 1$ ($P \subset \mathbb{R}^{n_s+1}$). For each pair of variables (x_j^l, x_j^u), $j = 1, \ldots, n_s + 1$ corresponds a parameter p_j.

4. The upper-level constraint g in (7.2) is actually a vector of $n_s + 1$ constraints in KMR. This does not add any complexity to the problem because all the $n_s + 1$ generalized semi-infinite constraints are defined with respect to the same lower-level feasible set.

5. The lower-level feasible set P is defined as simple bound inequalities on the parameters. Essentially these relations define a box the extreme points of which are the 2^{n_s+1} points generated by the values of x_j^u and x_j^l, $j = 1, \ldots, n_s + 1$.

6. Notice that the generalized-semi infinite constraints do not depend directly on the optimization variables. The dependence stems from the $\mathbf{x}$-dependence of the lower-level feasible set. This is consistent with the fact that KMR is a design centering problem.

7. The host set of the decision variables $\mathbf{x}$ is $X = \{\mathbf{x} \in \mathbb{R}^{2(n_s+1)} : x_{j,low} \leq x_j^{u,l} \leq x_{j,up},\ j = 1, \ldots, n_s + 1\}$.

8. The cardinality of the lower-level inequality constraints is $n_s + 1$. Therefore, $|J| = n_s + 1$.

9. Denote $\mathbf{p} = (\mathbf{y}, T)$. Then the full kinetic model describing the evolution of the concentration of species j, y_j for a constant-volume adiabatic process is given by [25]:

$$\Gamma_j = \frac{\sum_{i=1}^{n_r} \nu_{j,i} z_i r_i(y,T)}{C(y,T)}, j = 1, \ldots, n_s,$$

where $\nu_{j,i}$ is the stoichiometric coefficient of species j in reaction i, r_i is rate expression for reaction i and $C(y,T)$ is the total concentration of the mixture. The underlying equation for the evolution of temperature is [25]:

$$\Gamma_{n_s+1} = \frac{\sum_{j=1}^{n_s} h_j(T) \sum_{i=1}^{n_r} \nu_{i,j} r_i(\mathbf{y},T)}{C_p(\mathbf{y},T)},$$

where h_j is the molar enthalpy of species j at temperature T and $C_p(\mathbf{y},T)$ is the heat capacity of the mixture.

A reduced model Γ_{red} is expressed in terms of the full model with the addition of binary variables z_i, $i = 1, \ldots, n_r$ indicating whether reaction i is included in the model ($z_i = 1$) or has been deleted ($z = 0$). Therefore, the reduced kinetic model can be expressed in terms of the following equations [25]:

$$\Gamma_j = \frac{\sum_{i=1}^{n_r} \nu_{j,i} z_i r_i(y,T)}{C(y,T)}, j = 1, \ldots, n_s$$

$$\Gamma_{n_s+1} = \frac{\sum_{j=1}^{n_s} h_j(T) \sum_{i=1}^{n_r} \nu_{j,i} z_i r_i(\mathbf{y},T)}{C_p(\mathbf{y},T)},$$

7.5 KMR as a SIP and as a Bi-level program

We mentioned that kinetic model reduction is a design centering problem. As such, it can be formulated as a special case of a generalized-semi infinite program. However,

there do exist other equivalent formulations of KMR.

First of all, if we take advantage of the simple linear and separable, in the decision variables and parameters, structure of the lower-level feasible set and introduce the following variable transformation [137]:

$$\mathbf{s}(\mathbf{x}^l, \mathbf{x}^u, \mathbf{a}) = \mathbf{x}^l + \mathbf{a}(\mathbf{x}^u - \mathbf{x}^l), \tag{7.3}$$

then, the flexibility problem in kinetic model reduction can be expressed as an ordinary semi-infinite program:

$$\begin{aligned}
&\max_{\mathbf{x}^l, \mathbf{x}^u} V(\mathbf{x}^l, \mathbf{x}^u) \\
&\text{s.t. } \left|\Gamma_{red,j}(\mathbf{s}(\mathbf{x}^u, \mathbf{x}^l, \mathbf{a})) - \Gamma_j(\mathbf{s}(\mathbf{x}^u, \mathbf{x}^l, \mathbf{a}))\right| \leq \varepsilon_{tol}, j = 1, \ldots, n_s + 1, \ \forall \mathbf{a} \in [0,1]^{n_s+1}. \\
&\qquad\qquad \mathbf{x}^l, \mathbf{x}^u \in [\mathbf{x}_{low}, \mathbf{x}_{up}].
\end{aligned}$$

The SIP formulation presents certain advantages and drawbacks with respect to the original GSIP formulation:

1. SIP are considered to be numerical more tractable than GSIP. Therefore, it seems that an easier formulation has been postulated.

2. In the SIP reformulation, the decision variables $\mathbf{x}$ appear on the upper-level constraint. Recall that since KMR is a design centering problem, the decision variables appeared only in the description of the lower-level feasible set. Furthermore, the SIP reformulation introduces bilinearities between the decision variables $\mathbf{x}$ and the transformed parameters a. It is well-known that bilinearities are not desirable in any optimization problem, whether that is solved local or global.

Overall, it is possible to transform the original GSIP into a SIP at the cost of introducing bilinearities in the upper-level constraints.

Another possible way to formulate the flexibility problem in KMR is to use bi-level programming. There are two ways to generate the bi-level program; directly from the

original GSIP and indirectly after first reformulating the GSIP as an equivalent SIP. We are going to analyze the latter case. For this purpose, assume an ordinary semi-infinite program consisting of n (multiple) semi-infinite constraints:

$$\begin{aligned} &\min_{\mathbf{x}\in X} f(\mathbf{x}) \\ &\text{s.t. } g_1(\mathbf{x},p) \leq 0,\ \forall p \in P \\ &\qquad g_2(\mathbf{x},p) \leq 0,\ \forall p \in P \\ &\qquad\qquad \dots \\ &\qquad g_n(\mathbf{x},p) \leq 0,\ \forall p \in P, \end{aligned} \tag{7.4}$$

where $P \subset \mathbb{R}$ is the compact host set of the parameters. In order to transform this SIP into an equivalent bi-level program we introduce the following notation:

1. g_1, ..., g_n are multiple lower-level non-interacting constraints (also known as non-interacting players in bi-level programming).

2. We introduce as many inner variables as the number of constraints present times the dimensionality of the host set, which in this case is 1.

The equivalent bi-level reformulation of (7.4) is:

$$\begin{aligned} &\min_{\mathbf{x}\in X} f(\mathbf{x}) \\ &\text{s.t. } g_1(\mathbf{x},p_1^*) \leq 0, \\ &\qquad\qquad \dots \\ &\qquad g_n(\mathbf{x},p_n^*) \leq 0, \\ &p_1^*,\dots,p_n^* \in \arg\max_{p_1,p_2,\dots,p_n} G(\mathbf{x},p_1,\dots,p_n) \\ &\qquad\qquad \text{s.t. } G(\mathbf{x},p_1,\dots,p_n) = g_1(\mathbf{x},p_1) + + \dots + g_n(\mathbf{x},p_n) \\ &\qquad\qquad\qquad p_1,\dots,p_n \in P. \end{aligned} \tag{7.5}$$

In a similar way, the bi-level equivalent reformulation of KMR is:

$$
\begin{aligned}
&\max_{\mathbf{x}^l,\mathbf{x}^u} V(\mathbf{x}^l,\mathbf{x}^u)\\
&\text{s.t. } \Gamma_{red,1}(\mathbf{s}(\mathbf{x}^u,\mathbf{x}^l,\mathbf{b}_1^*)) - \Gamma_1(\mathbf{s}(\mathbf{x}^u,\mathbf{x}^l,\mathbf{b}_1^*)) \le \varepsilon_{tol},\\
&\qquad \Gamma_1(\mathbf{s}(\mathbf{x}^u,\mathbf{x}^l,\mathbf{b}_2^*)) - \Gamma_{red,1}(\mathbf{s}(\mathbf{x}^u,\mathbf{x}^l,\mathbf{b}_2^*)) \le \varepsilon_{tol},\\
&\qquad\qquad\qquad \ldots\\
&\qquad \Gamma_{red,2n_s+1}(\mathbf{s}(\mathbf{x}^u,\mathbf{x}^l,\mathbf{b}_{2n_s+1}^*)) - \Gamma_{2n_s+1}(\mathbf{s}(\mathbf{x}^u,\mathbf{x}^l,\mathbf{b}_{2n_s+1}^*)) \le \varepsilon_{tol},\\
&\qquad \Gamma_{2n_s+2}(\mathbf{s}(\mathbf{x}^u,\mathbf{x}^l,\mathbf{b}_{2n_s+2}^*)) - \Gamma_{red,2n_s+2}(\mathbf{s}(\mathbf{x}^u,\mathbf{x}^l,\mathbf{b}_{2n_s+2}^*)) \le \varepsilon_{tol},\\
&\mathbf{b}_1^*,\mathbf{b}_2^*,\ldots,\mathbf{b}_{2n_s+1}^*,\mathbf{b}_{2n_s+2}^* \in \arg\max_{\mathbf{b}_1,\mathbf{b}_2,\ldots,\mathbf{b}_{2n_s+1},\mathbf{b}_{2n_s+2}} G(\mathbf{x}^l,\mathbf{x}^u,\mathbf{b}_1,\mathbf{b}_2,\ldots,\mathbf{b}_{2n_s+1},\mathbf{b}_{2n_s+2})\\
&G(\mathbf{x}^l,\mathbf{x}^u,\mathbf{b}_1,\mathbf{b}_2,\ldots,\mathbf{b}_{2n_s+1},\mathbf{b}_{2n_s+2}) = g_1(\mathbf{x}^l,\mathbf{x}^u,\mathbf{b}_1) + \ldots + g_{2n_s+2}(\mathbf{x}^l,\mathbf{x}^u,\mathbf{b}_{2n_s+2})\\
&g_1(\mathbf{x}^l,\mathbf{x}^u,\mathbf{b}_1) = \Gamma_{red,1}(\mathbf{s}(\mathbf{x}^u,\mathbf{x}^l,\mathbf{b}_1^*)) - \Gamma_1(\mathbf{s}(\mathbf{x}^u,\mathbf{x}^l,\mathbf{b}_1^*)) - \varepsilon_{tol}\\
&\qquad\qquad \ldots\\
&g_{2n_s+2}(\mathbf{x}^l,\mathbf{x}^u,\mathbf{b}_{2n_s+2}) = \Gamma_{2n_s+2}(\mathbf{s}(\mathbf{x}^u,\mathbf{x}^l,\mathbf{b}_{2n_s+2}^*)) - \Gamma_{red,2n_s+2}(\mathbf{s}(\mathbf{x}^u,\mathbf{x}^l,\mathbf{b}_{2n_s+2}^*)) - \varepsilon_{tol}\\
&\mathbf{x}^l,\mathbf{x}^u \in [\mathbf{x}_{low},\mathbf{x}_{up}]\\
&\mathbf{b}_1,\ldots,\mathbf{b}_{2n_s+2} \in [0,1]^{n_s+1}.
\end{aligned}
$$

7.6 Is KMR Tractable with SIP, GSIP or Bilevel Programming?

In this section we will comment on the applicability of the global, or feasible-point, methods that have been proposed for semi-infinite [27, 84], generalized semi-infinite [77] and bi-level programming [83]. The main motivation to analyze only these algorithms is that, to our best knowledge, they are the first numerical procedures to provide a guaranteed feasible point for semi-infinite, generalized semi-infinite and bi-level programming, respectively.

7.6.1 Applicability of SIP Algorithms

Both algorithms that have been proposed for semi-infinite programs [27, 84] require the subdivision of the parameter host set in order to guarantee global optimality. Bhattacharjee et al. use the subdivision in order to converge the interval extensions while Mitsos et al. use the subdivision in order to converge the relaxation-based bounds. Therefore, there is a continuous increase of the number of constraints in the approximating problems and in order to guarantee optimality this number can be unbounded. Within the context of branch-and-bound suppose that the subdivision of the parameter set is defined by the empirical rule: $S_q = \{1, 2, \ldots, 2^q\}^{n_s+1}$, where S_q is the index set of the partitions of the parameter set at level q of the branch-and-bound tree and $n_s + 1$ are the total number of parameters in the problem. This implies that at level q the number of generated constraints is $(2n_s + 2)(2^{q(n_s+1)})$. The first term corresponds to the number of semi-infinite constraints in problem while the second refers to the number of partitions that each of the constraints needs to be evaluated on.

To illustrate the explosion in the number of generated constraints, consider the hydrogen combustion mechanism in [9]. The proposed model consists of 8 participating species and 20 chemical reactions. At the second level in the tree, i.e.$q = 2$, the total number of generated constraints amounts to $(2(8 + 1))(2^{2*9}) = 4718592$! It is clear that standard numerical solvers would not be able to handle such large systems robustly. Notice, also, that the explosion in the number of constraints occurs very early in the tree. This is mainly due to the large number of parameters $(n_s + 1)$ in the system.

7.6.2 Applicability of GSIP Algorithm

The algorithm that was developed and analyzed in Chapter 5 also relies on the subdivision principle in order to guarantee global optimality. Therefore, if the flexibility problem in KMR is formulated as a GSIP one should expect the aforementioned limitations, i.e. an explosion in the number of the generated constraints.

A question that may arise in the applicability of the GSIP algorithm comes from a close examination of the functional form of the lower-level inequality constraints. One could argue that as we shrink the space of decision variables, because these variables bound the parameters, then the space of parameter values would shrink too. In other words, by shrinking the space of the decision variables, e.g. by going deeper in the branch-and-bound tree we could avoid subdividing the parameter set since this set shrinks as well.

In order to answer this question, suppose that we are given a full kinetic model Γ, a reduced kinetic model $\bar{\Gamma}$ as well as a nominal point for reduction x_{nom}. For simplicity also suppose there is only one species participating and that the lower and upper bound of the species concentration in the flexibility program are x_{low} and x_{up}, respectively. We define two auxiliary variables x^u and x^l for which $x_{low} \leq x^l \leq x_{nom}$ and $x_{nom} \leq x^u \leq x_{up}$. There are two possibilities:

1. In a feasibility approach assume that we are shrinking the interval of validity around the point x_{nom}. Figure 7-4 illustrates this case. The approach by Olu-

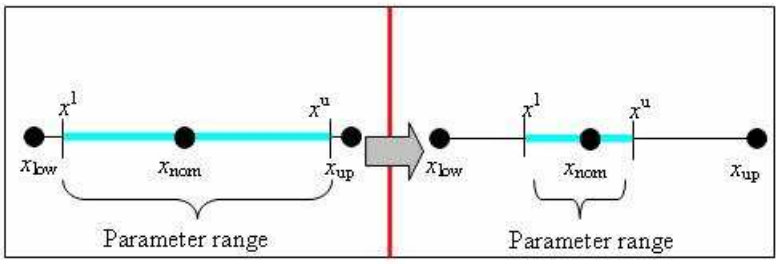

Figure 7-4: Decision and Parameter Ranges Shrinking

wole et al. [90] is based upon this principle. Therefore, as the approximated region of valid reduction shrinks around a nominal point of valid reduction, then no subdivision of the parameter set is needed. From simple continuity statements, there exists a non-degenerate range (in the 1-D case, a non-degenerate range is a line, a degenerate one is a point) for which the reduction is valid. Recall that shrinking an approximated region of reduction until it is feasible and stopping upon finding such solution addresses the feasibility issue in KMR.

2. In a flexibility approach this claim is not correct. Specifically, as the decision-variable space shrinks, the host set of the parameters does not need to shrink. To illustrate this point consider a fixed node in the branch-and-bound tree that has lower and upper bounds x_l^l,x_l^u and x_u^l, x_u^u respectively, for the decision variables x^l and x^u. Figure 7-5 shows that as the decision-variable set shrinks the parameter set remains finite. Therefore, in order to guarantee global optimality,

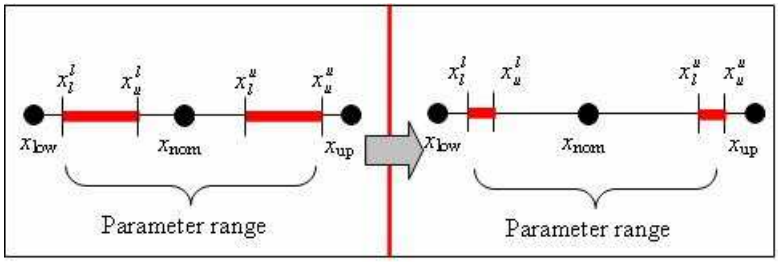

Figure 7-5: Parameter Set in the Flexibility Problem

even for this a special of GSIP, subdivision of the parameter set is necessary.

7.6.3 Applicability of Bi-level Algorithm

In Section 7.5 we mentioned that one possible reformulation of the flexibility problem in KMR is given by a bi-level program. Recently, Mitsos et al. [83] proposed the first global optimization procedure for bilevel optimization problems with a nonconvex inner problem. The drawback in using bi-level programming for KMR is that the bilevel equivalent introduces $2(n_s + 1)^2$ inner variables. This is in addition to the equality constraints and the Lagrange multipliers that the method employs through the KKT conditions. Therefore, for the hydrogen combustion mechanism consisting of 8 participating species, a bi-level reformulation would introduce 162 inner variables, which already exceeds the practical limits of the algorithm proposed in [83].

Overall, the algorithms that have been proposed for semi-infinite [27, 84], generalized semi-infinite [77] and bi-level programming [83] do not seem to be directly applicable to the flexibility problem in KMR.

7.7 KMR using Global Optimization

In this section, we will analyze a new method to provide valid regions of reduction that has the following characteristics:

1. It addresses the flexibility problem in KMR. Therefore, the method will find feasible regions of valid reduction and will attempt to maximize, in some sense, the volume of this region.

2. It will sacrifice global optimality with tractability. The analysis provided in Section 7.6 demonstrated that guaranteed global optimality along with guaranteed feasible regions within the context of semi-infinite, generalized semi-infinite and bi-level programming renders the feasibility problem in KMR intractable.

Suppose we are given the following information:

1. The full and a reduced kinetic model of the combustion mechanism Γ and Γ_{red}, respectively.

2. The description tolerance between the full and the kinetic model ε.

3. A nominal point of reduction, i.e. a point $\mathbf{x}_{nom}$ for which the full model Γ is reproduced by the reduced kinetic model Γ_{red} within the description accuracy ε.

Then, the method consists of three phases:

1. **Interval Guess using Symbolic Interval Extensions**.
   Using symbolic interval extensions, we will formulate a problem to find an initial feasible box. This will give us both an initial estimate of a feasible region and will furthermore, take advantage of the ICR to its fullest extent. Recall that the approach in [90] also uses interval extensions but it does not formulate an optimization problem because it targets the feasibility and not the flexibility problem. In order to construct the restricted problem, we employ interval extensions on the upper-level constraints which involve only the parameters of KMR

(recall that KMR is a design centering problem and as such the decision variables do not appear in the upper-level constraints). Fortunately enough, we can create a symbolic interval extension of the upper-level constraints because there are explicit lower and upper bounds on the parameters given by the lower-level inequality constraints. Therefore, we create the following optimization problem:

$$\begin{aligned}
&\max_{\mathbf{x}^l,\mathbf{x}^u} V(\mathbf{x}^l,\mathbf{x}^u) \\
&\text{s.t. } \Gamma^U_{red,j}(\mathbf{x}^u,\mathbf{x}^l) - \Gamma^L_j(\mathbf{x}^u,\mathbf{x}^l) - \varepsilon_{tol} \leq 0, j = 1,\ldots,n_s+1, \\
&\quad \Gamma^U_j(\mathbf{x}^u,\mathbf{x}^l) - \Gamma^L_{red,j}(\mathbf{x}^u,\mathbf{x}^l) - \varepsilon_{tol} \leq 0, j = 1,\ldots,n_s+1, \\
&\qquad\qquad \mathbf{x}^l,\mathbf{x}^u \in [\mathbf{x}_{low},\mathbf{x}_{up}].
\end{aligned} \tag{7.6}$$

where $\Gamma^L_{red,j}$ and Γ^L_j are the lower bounds on the interval extensions of $\Gamma_{red,j}$ and Γ_j with respect to $\mathbf{p} \in [\mathbf{x}^l,\mathbf{x}^u]$ and $\Gamma^U_{red,j}$ and Γ^U_j are the upper bounds of these interval extensions. Notice that is a finite optimization problem with decision variables $\mathbf{x}^l$ and $\mathbf{x}^u$. Let $[\bar{\mathbf{x}}^l,\bar{\mathbf{x}}^u]$ define the hyper-rectangle that is created from the interval approach.

2. **Finding Critical Points**

   In a combustion mechanism consisting of n_s species plus the system temperature we have $2n_s+2$ total decision variables (a lower and an upper bounding decision variable on each of the species and the system temperature). To avoid confusion we will, henceforth, lump species and temperature to species.

   In order to fully define the notion of a critical point assume that we have a feasible range of reduction given by a hyper-rectangle. The fact that we have $2n_s+2$ options for extending the hyper-rectangle stems from the n_s+1 potential species and whether to decrease their lower bounding value or increase their upper bounding value. Assume $J = \{1,\ldots,n_s+1\}$ and $I = \{1,2\}$, where J is the index set of the species and I is the index set of the choice to decrease the lower-bounding value, which we will correspond to 1 or to increase the upper-bounding value, which we will correspond to 2. Let $H = J \times I$ describe all the

$2n_s + 2$ options for extending the hyper-rectangle. Define as $\bar{H}$ the subset of non-active directions of H. An active direction is one that touches a subset of the constraints of the problem and who's extensions is likely to cause violations of these to one or more of these constraints.

For a fixed $(j', i') \in H$, consider, for each $j \in J$, the optimization problems of minimizing and maximizing the difference between the full and the reduced kinetic model on the hyper-rectangle subject to the direction defined by (j', i') held constant at the corresponding extreme value of the hyper-rectangle (lower bound when $i' = 1$, upper bound when $i' = 2$):

$$\begin{aligned} &\max_{\mathbf{p}\in[\bar{\mathbf{x}}^l,\bar{\mathbf{x}}^\mathbf{u}]} \Gamma_{red,j}(\mathbf{p}) - \Gamma_j(\mathbf{p}) \\ &\text{s.t.} p^l_{j'} = \bar{x}^l_j \quad \text{if} \quad i' = 1 \quad \text{or} \quad p^u_{j'} = \bar{x}^u_j \quad \text{if} \quad i' = 2. \end{aligned} \tag{7.7}$$

$$\begin{aligned} &\min_{\mathbf{p}\in[\bar{\mathbf{x}}^l,\bar{\mathbf{x}}^\mathbf{u}]} \Gamma_{red,j}(\mathbf{p}) - \Gamma_j(\mathbf{p}) \\ &\text{s.t.} p^l_{j'} = \bar{x}^l_j \quad \text{if} \quad i = 1' \quad \text{or} \quad p^u_{j'} = \bar{x}^u_j \quad \text{if} \quad i' = 2. \end{aligned} \tag{7.8}$$

The solution points of (7.7) and (7.8) are $n_s + 1$-dimensional arrays, namely $optval_{\max,j}$ and $optval_{\min,j}$ respectively. The solution values of these problems are denoted $valmax_j$ and $valmin_j$, respectively. To fully define a critical point we need to define/specify the following:

(a) A direction in extending the hyper-rectangle (region of validity). To fully specify this direction we need to define a tuple $(j', i') \in H$.

(b) An index set $K = \{1, 2\}$. For $k \in K$ with $k = 1$ we refer to problem (7.7) and with $k = 2$ we refer to problem (7.8).

(c) An index set $\bar{J} = \{1, \ldots, n_s + 1\}$. For $\bar{j} \in \bar{J}$ we refer to the consideration of the optimization problems (7.7) and (7.8) for species $\bar{j}$.

(d) An index set $\hat{J} = \{1, \ldots, n_s + 1\}$. For $\hat{j} \in \hat{J}$ we refer to the $\hat{j}$-entry in the array of $optval_{\max,j}$ and $optval_{\min,j}$.

(e) The space of $n_s + 1$-dimensional intervals $\mathbb{IR}^{n_s+1}$. We assume that the hyper-rectangle is described by X_{cur}: $X_{cur} \subset \mathbb{IR}^{n_s+1}$ and $X_{cur} = [\mathbf{x}^l, \mathbf{x}^u]$.

We define the following mapping: $C : H \times K \times \bar{J} \times \hat{J} \times \mathbb{IR}^{n_s+1} \rightarrow \mathbb{R}$. $C((j', i'), k, \bar{j}, \hat{j}, X_{cur})$ refers to the $\hat{j}$-entry of the array $optval_{\max,\bar{j}}$ of the optimization problem to maximize (if $k = 1$) or of the array $optval_{\min,\bar{j}}$ of the optimization problem to minimize the difference between the full and the reduced kinetic models for species $\bar{j}$ in an effort to increase (if $i' = 2$) or decrease (if $i' = 1$) the value of the concentration (or temperature) of species j' and thus to extend the feasible region defined by X_{cur}. C is a mapping from $H \times K \times \bar{J} \times \hat{J} \times \mathbb{IR}^{n_s+1}$ to the critical points of the problem (subset of the real numbers).

To find the total number of critical points for a given (j', i') we need to solve $2(n_s + 1)$ unconstrained (box-constrained) optimization problems of the form (7.7) globally. For a user-specified direction (j', i'), the critical points are the points in the existing hyper-rectangle for which the difference (without an absolute norm) between the full and the reduced kinetic model is either maximized or minimized for any one of the participating species. If the system is well-behaved, then the critical points can give us a good suggestion on how to expand the hyper-rectangle without violating the constraints.

3. **Performing Line Optimization**

In order to expand the hyper-rectangle we take advantage of the critical points that were calculated in the previous steps. Line optimization implies that the bounds of only one direction will be expanded while the rest of the directions will maintain the bounds from the existing hyper-rectangle.

Before we describe the underlying equations, it should be noted that upon a user-specified direction (j', i') there is a uniquely defined optimization variable $x_{(j',i')}$ that corresponds to the specified direction.

For each $j \in J$ we define $\bar{\mathbf{x}}^j = (\bar{x}^j_1, \ldots, \bar{x}^j_{j'-1}, x_{(j',i')}, \bar{x}^j_{j'+1}, \ldots, \bar{x}^j_{n_s+1})$ and $\hat{\mathbf{x}}^j = (\hat{x}^j_1, \ldots, \hat{x}^j_{j'-1}, x_{(j',i')}, \hat{x}^j_{j'+1}, \ldots, \hat{x}^j_{n_s+1})$.

Without loss of generality we will assume that $i' = 1$, i.e. we are attempting to decrease the value of the lower bound of one of the participating species.

With the help of the critical points that were calculated in the previous step, we formulate the following optimization problem:

$$\begin{aligned}
&\min_{x_{(j',i')}} x_{(j',i')} \\
&\text{s.t. } \Gamma_j(\bar{\mathbf{x}}^j) - \Gamma_{j,red}(\bar{\mathbf{x}}^j) \leq \frac{\varepsilon + valmax_j}{2},\ j = 1, \ldots, n_s + 1 \\
&\quad \Gamma_j(\hat{\mathbf{x}}^j) - \Gamma_{j,red}(\hat{\mathbf{x}}^j) \geq \frac{-\varepsilon + valmax_j}{2},\ j = 1, \ldots, n_s + 1 \\
&\quad \bar{x}^j_{j^*} = optval_{(\max,j)}(j^*), \forall j^* \in J, j^* \neq j' \\
&\quad \hat{x}^j_{j^*} = optval_{(\min,j)}(j^*), \forall j^* \in J, j^* \neq j' \\
&\quad x_{(j',i')} \in [x_{(j',i'),low}, x_{(j',i'),cur}].
\end{aligned} \tag{7.9}$$

Note that the line optimization problem in (7.9) involves a single variable subject to $2(n_s + 1)$ nonlinear constraints.

After this analysis we present an algorithm that addresses the flexibility problem in KMR. The algorithm consists of two phases; the interval guess and the expansion of the valid range of reduction using critical points and line optimization.

Recall, J is the index set of the participating species, H is the index of all directions, $\bar{H}$ is the subset of H containing non-active directions, $X_{cur,k}$ stores the temporary estimate of the valid range of reduction at iteration k of the algorithm.

Specifically, the detailed steps of the algorithm are:

1. (Initialization). Set $k = v = c_{act} = 0$, $\bar{H} = H$. Choose a description tolerance ε and a constraint activation tolerance ε_{con}. Also create two, $n_s + 1$-dimensional arrays *maxu* and *minu* to store the global solution values of the unconstrained problems in equations (7.7) and (7.8), respectively. Also create $n_s + 1$, $n_s + 1$-dimensional arrays $optval_{\max,j}$ and $n_s + 1$, $n_s + 1$-dimensional arrays $optval_{\min,j}$ to store the solution points of (7.7) and (7.8), respectively. Last, create 2, $n_s + 1$-dimensional arrays to store the global solution values of (7.7) and (7.8).

2. (Interval Guess). Employ a symbolic interval extension on the upper-level constraints (with respect to the parameters) and solve problem (7.6) to obtain the initial feasible box $X_{cur,0}$.

3. (Choice of direction & phase) Set k = k + 1 and $X_{cur,k} = X_{cur,k-1}$. Specify a tuple $(j', i') \in \bar{H}$. Reset the arrays $optval_{\max,j}$, $optval_{\min,j}$, $j \in J$, $valmax$, $valmin$, $maxu$, $minu$, v. Go to step 4 or 6.

4. (Critical Points). For each $j \in J$, solve problems (7.7) and (7.8) to populate the arrays $optval_{\max,j}$ and $optval_{\min,j}$ with the critical points and the arrays $valmax$ and $valmin$ with the global solution values of these problems.

5. (Line minimization). Solve problem (7.9) to obtain an update on the decision variable $x_{(j',i')}$. Update the feasible box $X_{cur,k}$ based on this new value. Go to step 7 to check for feasibility of the incumbent.

6. (Brute-force extension). Choose $(j', i') \in \bar{H}$. Extend the corresponding direction $\mathbf{x}_{(j',i')}$ from its current bound to its original lower bound (if i' =1) or to its original upper bound (if i'=2). Update the feasible box $X_{cur,k}$ based on this new value. Go to step 7 to check for feasibility of the incumbent.

7. (Guarantee feasibility). For each $j \in J$ solve the unconstrained version of equations (7.7) and (7.8), i.e. the problems

$$\begin{aligned} &\max_{\mathbf{p} \in X_{cur,k}} \Gamma_{red,j}(\mathbf{p}) - \Gamma_j(\mathbf{p}) \\ &\min_{\mathbf{p} \in X_{cur,k}} \Gamma_{red,j}(\mathbf{p}) - \Gamma_j(\mathbf{p}) \end{aligned} \tag{7.10}$$

   Populate the arrays $maxu$ and $minu$ with the global solution values of these problems.

   (a) If $v = \max\{\max_{j \in J} maxu, \left|\min_{j \in J} minu\right|\} \leq \varepsilon$, then the new estimate $X_{cur,k}$ is feasible. Go to step 8.

   (b) On the other hand, if there exists $j \in J$ such that either $maxu(j) \geq \varepsilon$ or $minu(j) \leq -\varepsilon$ then the new box is not feasible and it needs to be shrunk in

order to become feasible again. Use a rule, e.g. the Armijo rule, to shrink the extended direction back to the value of the last updated feasible box. Go to step 7.

8. (Checking for active constraints & directions at their bounds). Check the two following conditions:

   (a) If $v \leq |\varepsilon - \varepsilon_{con}|$ then (j', i') is an active direction.

   (b) If the variable $x_{(j',i)}$ attains its original lower bound ($i' = 1$) or upper bound ($i' = 2$) then this direction cannot be extended further.

   If any of these statements is true then delete (j', i') from $\bar{H}$. Set $c_{act} = c_{act} + 1$.

9. (Checking for feasible directions). If $c_{act} < 2(n_s + 1)$ go to step 3.

10. (Termination). There are no more feasible directions to extend the box (all directions are active). Therefore, with a user-specified accuracy of ε, a user-specified constraint activation tolerance ε_{con}, a full kinetic model Γ, a reduced kinetic model Γ_{red} and a user-specified set of directions $(j', i')_k$ the final estimate of the region of valid reduction is $X_{cur,k}$.

7.8 Application: Hydrogen Combustion

In order to test the performance of this algorithm we chose to implement it for the combustion of hydrogen. For this purpose we are using the full and reduced kinetic models suggested in [9] and included here. We note the following:

1. The full kinetic model has 46 reactions and 8 participating species, namely H_2, O_2, H_2O, H_2O_2, HO_2, OH, H and O. The reduced kinetic model has 21 reactions and all 8 of the participating species.

2. Species M represents the third-body enhancement in certain association or dissociation reactions.

Table 7.1: Full Kinetic Model for the Combustion of H_2

Reaction	A	β	E	Reaction	A	β	E
1. $H_2 + O \rightarrow OH + H$	5.12E4	2.67	26	2. $OH + H \rightarrow H_2 + O$	3.53E4	2.62	19
3. $H_2 + OH \rightarrow H_2O + H$	1.02E8	1.60	14	4. $H_2O + H \rightarrow H_2 + OH$	4.52E8	1.60	77
5. $O_2 + H + M \rightarrow HO_2 + M$	2.1E18	-0.80	0	6. $HO_2 + M \rightarrow H + O_2 + M$	1.16E20	-1.26	211
7. $O_2 + H + H_2O \rightarrow HO_2 + H_2O$	6.89E15	0.00	-9	8. $HO_2 + H_2O \rightarrow O_2 + H + H_2O$	3.80E17	-0.46	203
9. $O_2 + H \rightarrow OH + O$	9.76E13	0.00	62	10. $OH + O \rightarrow O_2 + H$	1.45E13	0.00	3
11. $H_2O_2 + H \rightarrow HO_2 + H_2$	1.69E12	0.00	16	12. $HO_2 + H_2 \rightarrow H_2O_2 + H$	1.51E9	0.78	84
13. $H_2O_2 + H \rightarrow OH + H_2O$	1.02E13	0.00	15	14. $OH + H_2O \rightarrow H_2O_2 + H$	6.72E7	1.28	296
15. $H_2O_2 + O \rightarrow OH + HO_2$	6.62E11	0.00	17	16. $OH + OH_2 \rightarrow H_2O_2 + O$	4.07E8	0.72	78
17. $H_2O_2 + OH \rightarrow H_2O + HO_2$	7.83E12	0.00	6	18. $H_2O + HO_2 \rightarrow H_2O_2 + OH$	4.74E11	0.45	141
19. $H_2O_2(+M) \rightarrow 2OH(+M)$	3.00E14	0.00	203	20. $2OH(+M) \rightarrow H_2O_2(+M)$	7.23E13	-0.37	0
21. $OH + O \rightarrow O_2 + H$	1.45E13	0.00	3	22. $H_2 + M \rightarrow 2H + M$	6.27E18	-0.98	437
23. $2H + H_2 \rightarrow 2H_2$	9.79E16	-0.60	0	24. $2H_2 \rightarrow 2H + H_2$	3.28E17	-0.58	437
25. $H + O + M \rightarrow OH + M$	1.18E19	-1.00	0	26. $OH + M \rightarrow H + O + M$	2.73E19	-1.03	429
27. $H + OH + M \rightarrow H_2O + M$	5.53E22	-2.00	0	28. $H_2O + M \rightarrow H + OH + M$	1.26E25	-2.3	503
29. $H + HO_2 \rightarrow H_2 + O_2$	4.28E13	0.00	6	30. $H_2 + O_2 \rightarrow H + HO_2$	2.60E12	0.48	231
31. $H + HO_2 \rightarrow 2OH$	1.69E14	0.00	4	32. $2OH \rightarrow H + HO_2$	1.84E10	0.83	150
33. $H + HO_2 \rightarrow H_2O + O_2$	3.01E12	0.00	7	34. $H_2O + O \rightarrow H + HO_2$	3.23E11	0.56	227
35. $2O + M \rightarrow O_2 + M$	5.40E13	0.00	-7	36. $O_2 + M \rightarrow 2O + M$	4.82E16	-0.43	494
37. $O + HO_2 \rightarrow O_2 + OH$	3.19E13	0.00	0	38. $O_2 + OH \rightarrow O + HO_2$	1.34E12	0.43	218
39. $2OH \rightarrow O + H_2O$	1.51E9	1.14	0	40. $O + H_2O \rightarrow 2OH$	1.49E11	0.87	75
41. $OH + HO_2 \rightarrow H_2O + O_2$	2.89E13	0.00	-2	42. $H_2O + O_2 \rightarrow OH + HO_2$	1.19E14	0.16	290
43. $2HO_2 \rightarrow H_2O_2 + O_2$	4.22E14	0.00	50	44. $H_2O_2 + O_2 \rightarrow 2OH_2$	2.88E16	-0.29	207
45. $2HO_2 \rightarrow H_2O_2 + O_2$	1.32E11	0.00	-7	46. $H_2O_2 + O_2 \rightarrow 2OH_2$	9.00E12	-0.29	150

3. We are assuming that the combustion of hydrogen is isothermal. This is the reason why temperature does not appear in the participating species.

4. The pre-exponential factor A, the pressure dependence factor β and the activation energy E refer to a kinetic rate constant expression of the following form: $k = AT^{\beta} \exp(-\frac{E}{RT})$.

Tables 7.1 and 7.2 provide the full kinetic model for hydrogen combustion and the corresponding enhancement factors for certain multi-body interactions that are encountered in the mechanism, respectively [9]. The reduced kinetic model contains reactions 1-10, 17, 19, 23, 24, 27, 29, 30, 31, 39, 40 and 41. Furthermore, the third body in multi-body interactions is denoted M.

7.8.1 Implementation Details

There are several implementation details that require a brief analysis.

1. First of all, we computed a nominal point of valid reduction from the full to the reduced kinetic mechanism. In order to achieve this we performed an isothermal, isochoric simulation of the full kinetic mechanism using Jacobian [1]. After examination of the profiles of the species concentrations we selected the candidate nominal point to be the one corresponding to a simultaneous peak in

Table 7.2: Enhancement Factors

Reaction	H_2O	O_2	Reaction	H_2O	O_2
1	—	—	2	—	—
3	—	—	4	—	—
5	—	0.4	6	0.4	—
7	—	—	8	—	—
9	—	—	10	—	—
11	—	—	12	—	—
13	—	—	14	—	—
15	—	—	16	—	—
17	—	—	18	—	—
19	—	0.4	20	—	0.4
21	6.5	0.4	22	6.5	0.4
23	—	—	24	—	—
25	6.5	0.4	26	6.5	0.4
27	2.54	0.4	28	2.54	0.4
29	—	—	30	—	—
31	—	—	32	—	—
33	—	—	34	—	—
35	6.5	0.4	36	6.5	0.4
37	—	—	38	—	—
39	—	—	40	—	—
41	—	—	42	—	—
43	—	—	44	—	—
45	—	—	46	—	—

the concentration of radicals (H, O, HO_2, OH). In order to verify feasibility of this point we calculated the difference between the conservation equations of the full and the reduced kinetic mechanisms for each participating species and made sure that this difference is less than the threshold for the maximum absolute difference between the full and the reduced kinetic mechanism that we set based on the time-scale of the simulation.

2. The threshold for the maximum absolute difference between the full and the reduced kinetic mechanisms was chosen to be, approximately, the inverse of the time scale of the entire simulation. Note that since the difference of the full and the reduced kinetic mechanisms is given as the difference in the conservation equations of the two mechanisms divided by the total concentration of the mixture, the units of this difference are s^{-1}.

3. Furthermore, we set the boundaries for the decision variables. For simplicity, if $\mathbf{x}_{nom}$ was the nominal point of reduction, then the lower and upper bound

for each decision variable x_i, $i = 1, \ldots, 8$ were set to $0.1x_{nom,i}$ and $10x_{nom,i}$, respectively.

4. Moreover, for the initial box $X_{initial} = [0.1\mathbf{x}_{nom}, 10\mathbf{x}_{nom}]$ we verified that it is not a valid range of reduction. If that was the case then there would essentially be no optimization problem to solve. Furthermore, to test the efficiency of the algorithm we would like to create active directions or in other words boxes for which at least one of the constraints is active on the edges of the box. Indeed, the initial hyper-rectangle is not entirely feasible.

5. In the aforementioned algorithm, each time the user specifies a search direction, a choice between the approach using critical points and line minimization and the approach using brute-force extension of the feasible box has to be made. We chose a fixed number of active directions c^*_{act} for which for $c_{act} < c^*_{act}$ we chose the former approach while for $c_{act} \geq c^*_{act}$ we chose the latter approach.

7.8.2 Implementation and Numerical Results

We begin this section by providing a quantitative description of all the implementation aspects previously described.

1. The temperature of the isothermal combustion mechanism is $T = 1014$K.

2. The initial conditions for the isothermal, isochoric simulation of the hydrogen combustion mechanism were set to:

$$P = 1\text{atm, initial mixture} : \mathrm{H_2} - \mathrm{O_2}\ (50\%, 50\%). \tag{7.11}$$

3. The nominal point of reduction was chosen as the point corresponding to a simultaneous peak in the concentration of the radical species. We denote

$$\mathbf{x}_{nom} = (\mathrm{H}_{2,nom}, \mathrm{O}_{2,nom}, \mathrm{H_2O}_{nom}, \mathrm{H_2O}_{2,nom}, \mathrm{HO}_{2,nom}, \mathrm{OH}_{nom}, \mathrm{H}_{nom}, \mathrm{O}_{nom}) =$$
$$(3.01\text{E-}3,\ 4.55\text{E-}3,\ 2.97\text{E-}3,\ 3.12\text{E-}8,\ 3.11\text{E-}6,\ 4.08\text{E-}6,\ 1.64\text{E-}4,\ 4.35\text{E-}5),$$

where the concentrations are given in units of mol/l.

4. The time scale for the simulation was $10^{-4}s$. Therefore, the threshold for the difference between the full and the reduced kinetic mechanisms, ε, was set to $3 \cdot 10^4$.

5. The threshold for constraint activation, ε_{con} was set to $\varepsilon_{con} = 0.01\varepsilon = 3 \cdot 10^{-6}s$.

6. The number of activations c^*_{act} for which the switch from the critical-point/line minimization approach to the brute-force extension of the feasible directions is made, was chosen to be 1. In other words, after one direction has been activated, the extension of the other feasible directions is done via brute-force.

7. Natural interval extensions were used to identify an initial feasible box in Step 2(interval guess) of the aforementioned algorithm.

To keep the report of the numerical results compact and insightful we report 6 different runs of the algorithm with different user-specified directions. Furthermore, we provide a comparison of the volume of the box at the end of the algorithm with the volume of the feasible box after the interval guess. We will quantify this comparison by a ratio of the two guesses called the *expansion ratio*. The reason for providing this comparison is to show that the interval guess in [90] can be substantially improved by using global optimization techniques.

Table 7.3 provides these numerical results. Column 1 provides the index of the simulation. Column 2 provides the sequence of user-supplied directions and Column 3 provides the expansion ratio.

Table 7.3: Numerical Procedure Applied to H_2 Combustion

#	User-specified Directions	Expansion Ratio
1	$O_{2,lo}, H_{2,lo}, H_2O_{lo}, H_2, O_{up}, 0_{up}HO_{2,up}, H_2O_{lo}, H_{up}, H_{2,lo}, O_{2,lo}, O_{up}$	46
2	$O_{2,lo}, H_{up}, O_{up}, H_{up}, 0_{up}H_2, O_{lo}, O_{up}, H_{up}, H_{2,lo}, H_2O_{up}, HO_{2,up}, O_{2,up}, O_{2,lo}$	62
3	$O_{up}, O_{up}, O_{up}, O_{up}, O_{up}, HO_{2,up}, H_2O_{up}, O_{2,up}, O_{2,lo}, H_{up}, H_{2,lo}, H_2O_{lo}$	77

The expansion ratio can reach the value of 80, a large improvement of the estimation of a valid region of reduction with respect to the estimate using interval

extensions.

Chapter 8

Conclusions & Future Work

The goal of this chapter is to provide the main conclusions and suggestions for future work in the areas of semi-infinite programming, generalized semi-infinite programming and kinetic model reduction.

8.1 Semi-Infinite Programming (SIP)

8.1.1 Conclusions & Contribution

In summary, semi-infinite programming involves the optimization of a a finite number of decision variables subject to an infinite number of constraints. Prior to this thesis, there were two outstanding issues for the numerical treatment of SIP with nonconvex functions participating. The first issue is the generation of guaranteed feasible points for which the only existing method was proposed by Bhattacharjee et al. [26]. The second issue is the development of an optimization algorithm that guarantees finite global optimality under assumptions that do not make use of convexity of any of the defining functions.

The first contribution of this thesis was the establishment of the first optimization procedure that locates guaranteed feasible points and provides a certificate of ε-global optimality for SIP with nonconvex functions participating. More specifically:

1. We demonstrated that the traditional means of proving finite convergence of

global optimization algorithms within a branch-and-bound framework (using consistency of the bounding operation) was not applicable to our approach. On the contrary, the lower- and upper- bounding operations were treated separately and were shown to converge finitely under mild assumptions on the problem structure.

2. An implementation of this global optimization procedure was developed and interfaced with several nonlinear solvers, some of which make use of sequential-quadratic programming techniques such as SNOPT and SLSQP and others that use interior-point methods such as IPOPT. For an arbitrary problem there is no definite conclusion on which nonlinear solver will perform better and therefore, incorporating more options provides an opportunity for better results.

3. Several problems for the well-known Watson test set were solved and the algorithm was shown to converge even when local reduction approaches would not even be applicable.

The second contribution of this thesis was the development of alternative methods to provide guaranteed feasible points for SIP with nonconvex functions participating. More specifically:

1. We demonstrated that the feasible-point method by Bhattacharjee et al. [26] is based on a restriction of the lower-level problem in SIP which leads to a relaxation of the overall (outer) problem.

2. It was shown that alternative means of restricting the lower-level problem could also provide guaranteed feasible points for SIP. Specifically, McCormick and α-BB relaxation techniques were used to construct concave overestimators (with respect to the parameters for each given value of the decision variables) of the semi-infinite constraints. Then, we proposed feasible-point methods that rely on either a KKT- or a linearization- based approach on the convexified constraints.

3. A numerical implementation of these algorithms was completed and the performance of these methods was compared to the interval constrained reformulation

that was suggested by Bhattacharjee et al. It was shown that the KKT- based approach using McCormick techniques could generate tighter upper bounds the global solution value of the SIP at the cost of solving more computationally expensive upper-bounding subproblems.

8.1.2 Future Work

With regards to the global optimization approach using interval methods [27] the following extensions/improvements are conceivable:

1. Strengthening the effectiveness of the exclusion heuristic. In the numerical implementation of this heuristic, the evaluation of an upper bound of the semi-infinite constraint on the examined domains relied on the use of interval methods. An alternative approach would be to use standard global optimization techniques to calculate the exact image of the semi-infinite constraint. While more computationally expensive, this approach could reveal redundant constraints much earlier in the tree when the effect of dropping constraints would aid significantly in making the subproblems in the subsequent levels more computationally tractable.

2. Relaxing the necessary assumptions for finite convergence. In order to prove the finite ε-convergence of the numerical procedure, an assumption was made regarding the relation between two quantities $q^1_{\mathbf{x}}$ and $q^2_{\mathbf{x}}$. The main drawback of this assumption is that it is hard to verify a priori. It is conceivable that the algorithm can be modified to incorporate an adaptive subdivision strategy that would enable this assumption to be dropped.

With regards to the feasible-point methods using McCormick and α-BB techniques the following extensions are conceivable:

1. Implement the proposed algorithms in a branch-and-bound framework and demonstrate their performance on the Watson test set.

2. Combine the tighter and more expensive upper bounding methods, such as the KKT-based upper bound, with looser and computationally more tractable methods, such as the linearization-based or interval-extension based upper bounds. A simple combination is to periodically employ the tighter and more expensive bounding problems. A more elaborate combination is to use a KKT-based upper bounding problem but only solve the resulting MPEC approximately to obtain a point $\bar{\mathbf{x}}$ and an estimate for the corresponding optimal solution of the relaxed lower-level problem $\bar{\mathbf{p}}$. Then the feasibility of $\bar{\mathbf{x}}$ can be probed by linearizing the concave lower-level problem around $\bar{\mathbf{p}}$. This approach is difficult to implement with black-box NLP solvers, but could be easily implemented in a framework such as the NCP approach by Floudas and Stein [42]. The promise of the combination, is that an approximate solution of the MPEC will provide a point $\bar{\mathbf{p}}$ which is suitable for linearization.

8.2 Generalized Semi-Infinite Programming (GSIP)

8.2.1 Conclusions & Contribution

In summary, generalized semi-infinite programming involves the optimization of a a finite number of decision variables subject to an infinite number of constraints the index set of which is dependent on the optimization variables. Prior to this thesis, there were two outstanding issues for the numerical treatment of GSIP with nonconvex functions participating. The first issue is the generation of guaranteed feasible points and the second issue is the development of an optimization algorithm that guarantees finite global optimality under assumptions that do not make use of convexity of any of the defining functions and which do not require the feasible set to be closed or the infimum to be attained.

The main contribution of this thesis was the establishment of the first optimization procedure that locates guaranteed feasible points and provides a certificate of ε-global optimality for GSIP with nonconvex functions participating and for which the feasible

set need not be closed and the infimum need not be attained. More specifically:

1. We demonstrated that discretization methods and the interval-constrained reformulation that were proved capable of providing lower and upper bounds for SIP, respectively, could be used for the same purpose for GSIP. However, we also showed that the limitation of any numerical procedure for GSIP is the tractable description of the lower-level feasible set. In order to overcome the difficulty of the dependence of the lower-level feasible set on the decision variables we proposed to use interval extensions on the lower-level inequalities.

2. We provided a convergence proof in which the aforementioned numerical procedure guarantees ε-global optimality to the GSIP infimum. The underlying assumptions do not require convexity in any of the functions participating, nor do they require that the feasible set of the GSIP is closed or that the minimum is attained.

3. We have proposed the first, to our best knowledge, test set for the general case of GSIP. This test set includes both literature and original examples. Furthermore, and more importantly, it involves examples that capture all the irregular behavior of GSIP. Therefore, the test set contains problems with a non-closed feasible set, problems where the infimum is not attained even though the GSIP is feasible, problems with re-entrant corner points, problems in which a constraint qualification is violated in the lower-level problem, and above all, problems that involve nonconvex functions participating.

4. Similar to the numerical procedures that were developed for SIP, it was shown that the upper-bounding approach for GSIP relies on the relaxation of the lower-level feasible set and the overestimation of the upper-level constraint, i.e. on the restriction of the lower-level problem, pointwise in $\mathbf{x}$. This creates a valid restriction of the outer (overall) problem. Similarly, discretization methods construct a restriction of the lower-level problem (restriction of the lower-level feasible set and underestimation of the upper-level constraint). This creates a valid relaxation of the outer (overall) problem.

8.2.2 Future Work

Future work in the area of generalized semi-infinite programming could focus on the following:

1. Examine the applicability of the global optimization algorithm for bilevel programs with a nonconvex lower-level problem that was suggested by Mitsos et al. [83]. It is already known [119] that GSIP with a non-empty lower-level feasible set for all $\mathbf{x}$ can be equivalently transformed to bi-level programs. Therefore, under these assumptions the algorithm suggested by [83] could be used to solve nonconvex GSIP. An interesting extension of this work would be to examine applicability of the algorithm when the lower-level feasible set is indeed empty for some values of $\mathbf{x}$. A hybrid method between the one recommended by Lemonidis and Barton [76] and Mitsos et al [83] is plausible.

2. Examine the reverse applicability. Specifically, it seems that interval methods could be used to provide an alternative lower-bounding methodology for bi-level programs without any convexity assumptions. This lower-bound combined with the upper bound suggested by Mitsos et al. [83] or with a simple feasibility check solving the inner problem globally (assuming that the minimum of the lower-level problem is unique, pointwise in $\mathbf{x}$) could lead to a global optimization procedure for nonconvex bilevel programs.

8.3 Kinetic Model Reduction (KMR)

8.3.1 Conclusions & Contribution

Kinetic model reduction is a general framework in which a full kinetic model describing the evolution of a certain process, e.g. the combustion of a chemical, is replaced with a reduced kinetic model with the ultimate goal of making the numerical simulation and therefore, analysis of the chemical process, more tractable to computers and engineers. Within the framework of combustion chemistry (especially for combustion

incorporated in reacting flow systems) there are two discrete subproblems:

1. Given a full kinetic model Γ describing a combustion mechanism, a region P of temperature and composition and a description accuracy ε, find the reduced model Γ_{red} with the minimum number of reactions and chemical species that can reproduce the full kinetic model for every point in P within the description accuracy ε.

2. Given a full kinetic model Γ, a reduced kinetic model Γ_{red} and a description accuracy ε, find the maximum volume of a hyper-rectangle in temperature and concentration space for which the reduced kinetic model reproduces the full kinetic model within ε for every point in the hyper-rectangle.

While there have been many algorithms to target the first issue, the second is still outstanding. Within the second issue our contribution focused on the flexibility aspect, i.e. on providing as large of a region of valid reduction as possible. More specifically:

1. We posed the flexibility problem in KMR as a GSIP and demonstrated that is essentially a classical example of a design centering problem.

2. We proposed alternative formulations of the KMR. Specifically, we demonstrated that KMR can be equivalently written as both an ordinary SIP and as a bi-level program with multiple non-interacting lower-level players.

3. Both the SIP and the GSIP formulations can provide the global solution, at least theoretically, of KMR using the algorithms that were developed in this thesis. However, global optimality in these algorithms is achievable when the parameter set is subdivided. Taking into consideration that even the small-scale mechanisms involve $10^1 - 10^2$ species, this subdivision will quickly render the problem intractable without a certificate of global optimality. Furthermore, it was shown that the bilevel algorithm suggested by Mitsos et al. [83] would also explode in the number of extra variables it needs to introduce mainly because of the underlying inner variables.

4. We proposed an alternative method that address the feasibility problem in KMR. This method sacrifices global optimality but provides good (at least not as conservative as the interval methods do) estimates of valid regions of reduction. The method is based on finding an initial feasible box using an interval approach and then extending the edges of the box either using critical points and line minimization or using brute-force methods. We demonstrated that using this algorithm, the estimates on the valid ranges of reduction could improve by two orders of magnitude over the estimates given by interval methods.

8.3.2 Future Work

Future work in the area of kinetic model reduction could focus on the following:

1. Using the SIP equivalent reformulation of KMR, the algorithm suggested by Blankenship and Falk [29] along with their exchange method heuristics could be implemented. This would provide an outer approximation method to the feasibility problem. Therefore, on finite termination, the box is not guaranteed to be feasible. However, the method suggested in [29] minimizes, with increasing iteration, the violation of the constraints. Once the constraints are slightly infeasible, in which case the box is almost optimal but quite not feasible, any "small shrink" of the feasible box could render an estimate that is not only feasible but also with a volume close to the optimal one. To my opinion, this is the most promising solution to the flexibility problem in KMR.

2. The other major focus area is the improvement of the procedure that was described in Section 7.7. First of all, many of the global optimization subproblems do not have to be considered. Specifically the global optimization problems that refer to the maximization and minimization of the difference between the full and the kinetic model with respect to a species, only have to be considered when previous iterations are showing that the species are active (either one of the directions that stem from the species are active). Furthermore, one could certainly employ Taylor models to formulate the interval guess. This is guar-

anteed to provide a better initial estimate than the one obtained by the use of natural interval extensions.

Chapter 9

Bilevel Programming in Game Theory: Capstone Paper

The goal of this chapter is to examine market duopolies and show that they can be modeled and solved as bilevel programs with nonconvex functions participating. Section 9.1 provides an introduction to the game theory problem. Section 9.2 gives a general description of the mathematical formulation of bilevel games. Section 9.3 provides a background and motivation for this work. Section 9.4 proposes a bilevel formulation of the new entrant game. Section 9.5 provides a detailed explanation of the model assumptions. Section 9.6 describes a list of scenarios that were studied. Section 9.7 provides a comprehensive list of numerical results and corresponding conclusions. Finally, Section 9.8 provides final remarks and comments on the contribution of this paper in modeling the strategic interactions in duopoly markets.

The author would like to express his warmest gratitude to Dr. Alexander Mitsos for providing the numerical implementation for all the studied scenarios and for all his insight throughout this work including the proposed mathematical formulation of the new entrant game. The author would also like to thank Professor David McAdams from the Sloan School of Management for his guidance in the early stages of this work and especially in providing conceptual ideas on bilevel games in industry.

9.1 Monopoly Markets with New Entrants

Market monopolies are frequently encountered in many industrial sectors including the pharmaceutical industry. In order to introduce the industrial setting, let's assume that Discovercorp has a pharmaceutical drug, Sinadim, designed to cure lung cancer. Let's also assume that Discovercorp has obtained an orphan drug exclusivity. The orphan drug status is provided to drugs that are targeting a small population in the United States (below 200,000 patients) and as such would not be financially advantageous for the firms to pursue. They provide tax deductions and a guaranteed monopoly status to the first incumbent for seven years [47]. However, after this period of time new entrants can enter the market offering similar products. Let's assume that it is currently the end of the 7-year protection period and Genecorp, a provider of generic drugs, is considering launching Keradim, to cure the same disease and is considering the best way to compete with Discovercorp and its monopoly. Obviously, Genecorp has three strategic decisions to make:

1. whether to enter the market and if so,
2. how much to spend on marketing Keradim and
3. how much to charge for Keradim.

Discovercorp also has three strategic choices:

1. how much to spend on marketing Sinadim,
2. how to charge for Sinadim and
3. whether to continue competing in this market or exit.

For both players in the market, namely Discovercorp and Genecorp, their choices on these strategic decisions depend on the other firm's strategy. As we will demonstrate in the subsequent sections, assuming that one firm will act first on choosing its pricing and marketing strategy, and the other second (i.e., there will be a leader and a follower), the interdependence of Discovercorp's and Genecorp's choices will generate a bilevel game between the two firms.

9.2 Bilevel Games

The general formulation of bilevel programs (henceforth called bilevel games) is the following:

$$\begin{aligned} &\max_{\mathbf{x},\mathbf{y}} f(\mathbf{x},\mathbf{y}) \\ &\text{s.t. } \mathbf{g}(\mathbf{x},\mathbf{y}) \leq \mathbf{0} \\ &\qquad \mathbf{y} \in \arg\max_{\mathbf{y}} h(\mathbf{x},\mathbf{y}) \\ &\qquad\qquad \text{s.t. } \mathbf{p}(\mathbf{x},\mathbf{y}) \leq \mathbf{0} \\ &\mathbf{x} \in [\mathbf{x}^L, \mathbf{x}^U] \subset \mathbb{R}^{n_x}, \quad \mathbf{y} \in [\mathbf{y}^L, \mathbf{y}^U] \subset \mathbb{R}^{n_y}. \end{aligned} \tag{9.1}$$

In game theory terminology, bilevel programs consist of the following rules:

1. The leader (the player that moves first) is trying to maximize his objective function f under his resource constraints $\mathbf{g}$. This sub-problem is called the leader's problem (outer or upper-level problem).

2. The follower (the player that moves second) is trying to maximize his own objective function h under his resource constraints $\mathbf{p}$. This sub-problem is called the follower's problem (inner or lower-level problem).

3. The leader's decision variables are $\mathbf{x}$ while the follower's are $\mathbf{y}$.

4. The game between the leader and the follower consists of the following steps (assuming, for simplicity, a single choice for each players x and y):

   (a) The leader chooses a value of x.

   (b) The follower formulates a best response based on x and generates the set $Y(x)$ (each $y \in Y(x)$ is an optimal response for the follower given x from the leader).

   (c) The leader evaluates his objective function and constraints given x and the set $Y(x)$ and chooses a pair (x, y), $y \in Y(x)$, that results in an optimal strategy for him.

(d) The leader repeats steps (a)-(c) until his objective function is maximized across all values of x in his design space.

The game between the leader and follower ends when the leader has created a strategy (value of x) that maximizes his best response across different values of x (his decision variable). For a thorough review of practical applications of bilevel programs we refer the reader to [14, 36]. In [82] the first global optimization algorithm for the general class of bilevel games involving nonconvex functions (objective and constraints) in both the leader's and more importantly in the follower's problem has been proposed. All the numerical results provided in this work have been furnished by the algorithm suggested in [82].

9.3 Background & Motivation

In the academic literature, the strategic interactions in market oligopolies are explored in two different contexts:

1. Cournot & Bertrand Competition (Cournot and Bertrand equilibria). In both cases, the market participants move simultaneously and there is no 1^{st} or 2^{nd} mover advantage, they do not cooperate, they have the same marginal cost and the demand is linear. In Cournot competition, firms compete on quantities and choose them simultaneously, while in Bertrand they compete on price. Cournot's ground-breaking work (1838) is considered as the start of modern game theory [8].

2. Stackelberg Games. In this case, the market participants move sequentially and, typically, these games are recognized by a leader-follower relationship [43]. Due to their inherent hierarchical structure, Stackelberg games are very closely connected and often modeled as bilevel programming problems [13].

The problem we are attempting to solve has a leader-follower relationship and, therefore, we will continue the literature review only on Stackelberg games.

In [67] the government-private sector relationship is modeled as a Stackelberg game in which the private sector (follower) is represented by several small agents, and optimizes its cost function give the decisions from the government (leader). The authors assume a finite-time linear quadratic differential game which ultimately leads decoupling of the leader's (upper-level) and follower's (lower-level) optimization problems.

In [102] an extended Bertrand competition in which the market participants choose the time of entrance and their pricing strategy is studied. The limiting assumption in this work is the characterization of market participants as either Stackelberg followers or leaders without knowing their cost or profit functions. The goal of the work is to locate Nash equilibria along the discrete possible times for entry.

In [35] equilibrium configurations between quantity Stackelberg games and price Stackelberg games in which the demand is concave (with respect to price) and the costs are convex are compared. These assumptions effectively create concave profit functions for which first-order closed form solutions are available. The two limiting assumptions in this work are:

1. Only one out of price/quantity is allowed to vary in the Stackelberg game. In other words, the dimensionality of the Stackelberg game is 1.

2. The convexity assumption on the functional form of the demand vs. price and the firm costs is too restrictive.

In [72] the strategic preference of firms to have a first- or second-mover advantage in Stackelberg price games is examined. Once again, the limiting assumption is the dimensionality of the firm's choices and the convexity of the functions involved in order for the 1st order conditions to furnish the competitor's true best response strategy.

In [87] the strategic choice of firms to play a Cournot (simultaneous) or Stackelberg (sequential) game is explored and a discussion of whether there is a strategic advantage to being the first mover is presented.

In general, the literature on Stackelberg games has attempted to provide answers to 3 questions (under the assumption of linearity or, at most, convexity of the func-

tions involved):

1. What are the optimal strategies in pure pricing or pure quantity Stackelberg games (firms are only allowed to have one degree of freedom)?

2. Is there a benefit to being a first- or second-mover in sequential games?

3. Is there a reason that market participants should prefer sequential over simultaneous games or vice versa?

These three goals are fundamentally different from the one we are trying to achieve. Our goal is to demonstrate that market duopoly problems can actually include highly nonconvex functions and that they can still be solved using the latest advancements in bilevel programming [82].

9.4 Formulating Monopoly Markets with New Entrants as Bilevel Games

The strategic decisions that Genecorp and Discovercorp can make and their interactions can be formulated as a bilevel game. The following sections explain the formulation of this game.

9.4.1 General Formulation

Without discussing the nature of the functions involved, the general characteristics of the game are:

1. Genecorp is the leader (player that moves first) because it is the one who decides whether to enter the market of selling its lung cancer drug. Genecorp has three strategic choices:

   (a) Whether to enter the market (binary variable x_1).

   (b) How much to sell the drug for (continuous variable x_2).

(c) How much to spend on marketing (continuous variable x_3).

2. Discovercorp is the follower (player that moves second) and based on Genecorp's strategic choices of x_1, x_2 and x_3 makes its own strategic choices:

   (a) How much to sell the drug for (continuous variable y_2).

   (b) How much to spend on marketing (continuous variable y_3).

   (c) Whether to exit the market (this option is not modeled here).

3. The objective function for both Discovercorp and Genecorp is the maximization of profits from selling their drug (revenues minus manufacturing and marketing costs).

4. The resource constraints for the problem are:

   (a) The upper and lower bounds on prices that are set for the drugs (the price must be higher than the cost of producing the drug and lower than the acceptable country standards).

   (b) The upper and lower bounds on marketing (marketing expenses cannot be greater than the available cash for the firms at the time of Genecorp's entrance in the market and cannot be lower than some minimum threshold to place the drug on the pharmacy shelves).

9.4.2 Formulation of Objectives and Constraints

In order to complete the mathematical formulation of the new entrant game we have to define the objective functions and the resource constraints formally. More specifically:

1. Genecorp's objective function (f) is assumed to be of the form:

$$f(x_1, x_2, x_3, y_2, y_3) = x_1 \cdot (MS_G \cdot Q \cdot (x_2 - C_{M,G}) - C_{A,G} - A_{0,G}),$$

   where:

(a) MS_G represents the total fraction of the market that Genecorp will capture with its strategy:

$$MS_G(x_2, x_3, y_2, y_3) = \frac{1}{1 + \frac{y_3 x_2}{y_2 x_3}} \; x_1.$$

The choice of MS_G is based on three desired properties:

i. If Genecorp spends more on marketing (x_3) then its share in the market should go up, i.e., MS_G should increase with x_3.

ii. If Genecorp charges less, i.e., x_2 decreases, then its share of the total market goes up.

iii. If Genecorp and Discovercorp charge the same and spend the same on marketing then they will share the market equally.

(b) Q represents the total market demand of the product. The functional form of the market demand is assumed to be a logistic one:

$$Q(x_3, y_3) = b_0 + b_1 \frac{1 + me^{-(x_1 x_3 + y_3)/\tau}}{1 + ne^{-(x_1 x_3 + y_3)/\tau}}.$$

The functional form of the demand satisfies a number of desired properties:

i. The total demand is a function of the aggregate marketing $x_3 + y_3$. The product of $x_1 x_3$ is introduced because x_3 is meaningful (and therefore added to y_3) only when $x_1 = 1$.

ii. With no marketing the market demand will be positive ($b_0 \geq 0$).

iii. With little marketing the second term is relatively small (beginning of logistic curve).

iv. With more marketing the influence on market demand is higher (sharp rise on logistic curve).

v. With excess marketing the second term approaches b_1 and the logistic curve flattens out.

vi. Unlike the case of market share in which there are direct means of

competition, the marketing of Discovercorp and Genecorp are complementary to each other in this case because the total demand is a function of the aggregate marketing of the two firms.

The functional form of (2) is designed to provide a market demand that varies with total marketing as shown in Figure 9-1:

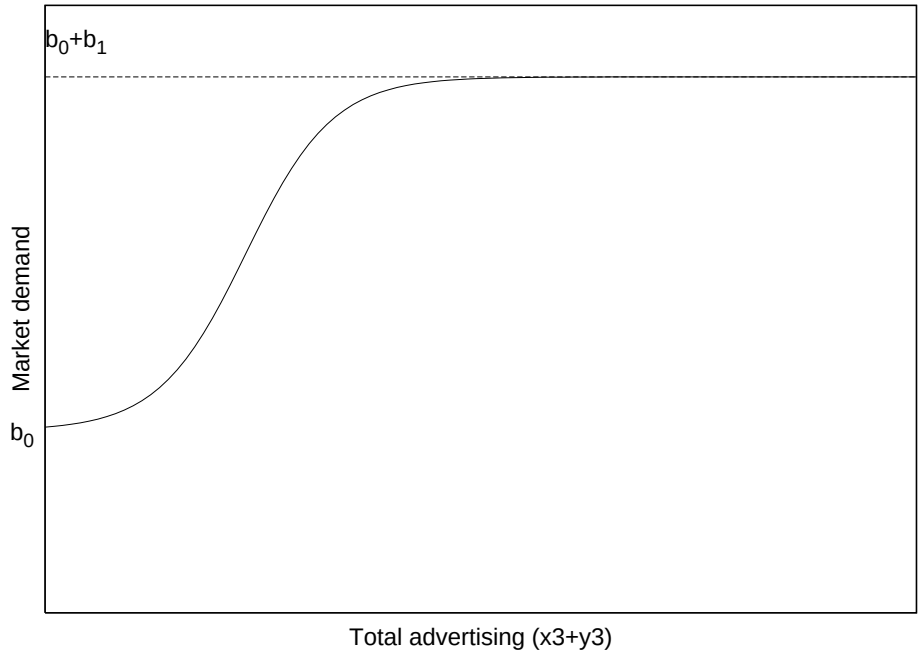

Figure 9-1: Total Market Demand vs. Aggregate Marketing

(c) $C_{A,G}(x_3)$ represents the total cost of marketing for Genecorp. We will assume that the total cost to the firm is some multiple of the pure marketing costs. Therefore,

$$C_{A,G} = a_G x_3,$$

where a_G is a constant that depends on Genecorp's operations.

(d) $A_{0,G}$ represents the total fixed capital cost for Genecorp in order to enter the market.

(e) $C_{M,G}$ represents the variable cost of manufacturing a single unit of product for Genecorp. We will assume no economies of scale, so therefore $C_{M,G}$ is independent of the quantity produced.

2. Genecorp's resource constraints (bound constraints in this case) are of the form:

$$x_1 \in \{0,1\}, x_2 \in [x_2^L, x_2^U], x_3 \in [x_3^L, x_3^U].$$

3. Discovercorp's objective function looks very similar to Genecorp's with the exception that we don't model the choice of exiting the market. Therefore, no fixed costs need to be included. Everything else is symmetric:

$$h(x_1, x_2, x_3, y_2, y_3) = MS_D \cdot Q \cdot (y_2 - C_{M,D}) - C_{A,D},$$

$$MS_D(x_2, x_3, y_2, y_3) = (1 - x_1) + x_1 \frac{1}{1 + \frac{x_3 y_2}{x_2 y_3}},$$

$$C_{A,D} = a_D y_3.$$

4. Similarly, Discovercorp's resources constraints (bound constraints in this case) are:

$$y_2 \in [y_2^L, y_2^U], y_3 \in [y_3^L, y_3^U].$$

Therefore, the bilevel formulation of the "new entrant in the market" game is the following:

$$\begin{aligned} &\max_{x_1,x_2,x_3,y_2,y_3} x_1 \cdot (MS_G(x_1, x_2, x_3, y_2, y_3) \cdot Q(x_3, y_3) \cdot (x_2 - C_{M,G}) - a_G x_3 - A_{0,G}) \\ &\{y_2, y_3\} \in \arg\max_{y_2,y_3} MS_D(x_1, x_2, x_3, y_2, y_3) \cdot Q(x_3, y_3) \cdot (y_2 - C_{M,D}) - a_D y_3 \quad (9.2) \\ &x_1 \in \{0,1\}, x_2 \in [x_2^L, x_2^U], x_3 \in [x_3^L, x_3^U], y_2 \in [y_2^L, y_2^U], y_3 \in [y_3^L, y_3^U] \end{aligned}$$

9.4.3 Specification of Game Parameters

In order to complete the formulation of the bilevel game between Genecorp and Discovercorp, we need to specify the numerical values on all the parameters in the aforementioned relations. Table 9.1 contains these numerical values.

Table 9.1: Parameters of the Bilevel Game

Parameter	Value	Unit	Description
$C_{M,D}$	15	\$/drug	Manufacturing Cost for Sinadim (Discovercorp)
$C_{M,G}$	12	\$/drug	Manufacturing Cost for Keradim (Genecorp)
a_G	2	—	Marketing Costs Constant for Genecorp
a_D	2	—	Marketing Costs Constant for Discovercorp
$A_{0,G}$	1000	\$	Fixed Cost for Entering Market (Genecorp)
x_2^L	15	\$/drug	Minimum Price for Keradim (Genecorp)
x_2^U	100	\$/drug	Maximum Price for Keradim (Genecorp)
x_3^L	100	\$	Minimum Marketing Costs for Keradim (Genecorp)
x_3^U	10000	\$	Maximum Marketing Costs for Keradim (Genecorp)
y_2^L	12	\$/drug	Minimum Price for Sinadim (Discovercorp)
y_2^U	100	\$/drug	Maximum Price for Sinadim (Discovercorp)
y_3^L	100	\$	Minimum Marketing Costs for Sinadim (Discovercorp)
y_3^U	10000	\$	Maximum Marketing Costs for Sinadim (Discovercorp)
b_0	200	# of drugs	Market Demand with no Marketing
b_1	200	# of drugs	Total Extra Market Demand due to Marketing
m	1	—	Logistic Demand - Numerator Constant
n	100	—	Logistic Demand - Denominator Constant
τ	500	—	Logistic Demand - Regularization constant

9.5 Model Assumptions

The formulation of the strategic interaction between Genecorp and Discovercorp as a bilevel game has many assumptions built-in. More specifically:

1. **Complete Information.** We are assuming that Genecorp and Discovercorp know perfectly each other's resource constraints and objective. Therefore, this game is a complete information one in which no incentives or constraints are hidden.

2. **Non-repeated Game.** We are implicitly assuming that the leader and follower will play this game once. In reality, as the numbers change (e.g., market size, marginal costs, etc.) the game will be played continuously between the leader and the follower.

3. **Co-operative Formulation.** The mathematical formulation in (9.2) implies that the leader will choose the $\mathbf{y} \in Y(\mathbf{x})$ that maximizes his objective. In reality, this is a best-case scenario approach, since the leader will probably not be able to choose amongst the follower's equally best options. The follower will make that choice instead. Nevertheless, if the set $Y(\mathbf{x})$ is a singleton for each $\mathbf{x}$ then we do not have to revert to the pessimistic formulation. In practice, for all the scenarios that will be described, we actually checked that at the equilibrium solution $(\mathbf{x}, \mathbf{y})$ the set $Y(\mathbf{x})$ is indeed a singleton.

4. **Strategic Decisions.** The two basic strategic decisions for Genecorp and Novartis have been assumed to be the pricing and marketing strategy. In reality, product differentiation and price discrimination (based on local income) are also possible strategies.

5. **Functional Forms and Parameters**. We have assumed a very specific structure for the market share split between Discovercorp and Genecorp depending on relative marketing and pricing strategies. We have also assumed that the total demand is a logistic function of aggregate marketing of both firms and more importantly, that the demand does not depend on price (but it does depend on total advertising). Finally, the parameter values have been chosen arbitrarily.

The mathematical model proposed can easily be adapted to incorporate objective functions with more strategic decisions (point 4) and with different functional forms (point 5). The numerical algorithm in [82] can be used with no alterations at all.

9.6 Overview of Scenarios

In order to study the bilevel game between Discovercorp and Genecorp, we have generated a number of scenarios:

1. **Asymmetric Base Case**. This is the base case scenario as described in Sections 9.4.2 and 9.4.3 in which Genecorp is the leader (entrant in the market) and Discovercorp is the follower (reacts to Genecorp's entrance). The asymmetry comes from the different manufacturing costs to produce the drug and the one-time fixed cost that Genecorp has to pay in order to enter the market.

2. **Asymmetric Base Case - Reversed**. This is the base case scenario (1) after reversing the sequence of moves. Specifically, Discovercorp is the leader and Genecorp is the follower.

3. **Symmetric Case**. In this case, the marginal costs of production are actually the same (15$/drug) and there is no fixed cost for Genecorp to enter the market. Therefore, the objective function and bound constraints for the leader and the follower are exactly the same.

4. **Symmetric Case - Reversed**. This is the symmetric case in which Discovercorp is the leader and Genecorp is the follower.

5. **Asymmetric Hostile Case**. This case is very similar to the asymmetric base case with the only difference that the objective function of Discovercorp (follower in this case) has a severe one-time penalty if Genecorp enters the market. This is actually modeled adding a $-6000x_1$ term in Discovercorp's objective function. This term is only activated if Genecorp enters the market, i.e., when $x_1 = 1$. The scenario includes the term hostile because it appears that Discovercorp does not want to accommodate Genecorp's entry in the market.

6. **Asymmetric Hostile Case - Reversed**. The same as before, only Discovercorp is now the leader and Genecorp is the follower.

9.7 Numerical Results & Comments

The numerical results on the aforementioned scenarios are summarized in Table 9.2. Column 1 provides the current scenario (as described in Section 9.6), while columns 2 and 3 identify the leader and the follower in that scenario. Columns 4-8 report the optimal values of the strategic choices for Discovercorp and Genecorp (i.e., x_1, x_2, x_3, y_2 and y_3) and columns 9 and 10 include the overall profits for Genecorp and Discovercorp based on the optimal strategies.

Table 9.2: Numerical Results: Bilevel Scenarios

Scenario	Leader	Follower	x_1	x_2 ($)	x_3 ($)	y_2 ($)	y_3 ($)	Genecorp Profits ($)	Discovercorp Profits ($)
1	G	D	1	100	4557	100	4244	8110	7907
2	D	G	1	100	4395	100	4108	8404	8210
3	G	D	1	100	4253	100	4250	8499	8495
4	D	G	1	100	4250	100	4253	8495	8499
5	G	D	1	100	4557	100	4244	8110	1907
6	D	G	0	0	0	82	10000	0	6799

Based on Table 9.2 we reach the following conclusions:

1. The only scenarios in which Discovercorp and Genecorp actually do have equal market shares are 3 and 4. In these scenarios, the objective function and constraints are exactly the same between the two firms. They collude (without consulting each other) on the higher pricing strategy (100 $/drug) and they have equal marketing ($4250 \neq 4253$ but the difference is within the optimization tolerance level) the aggregate of which is sufficient to capture essentially the entire market ($b_0 + b_1 = 400$ drugs). The fact that Genecorp and Discovercorp have the same objective function values in scenarios 3 and 4, however, is completely predictable by the properties of min-min problems (see point 3(a) for more details).

2. **In scenarios 1 and 2 the two firms actually do not have equal market share**. In these two scenarios Genecorp chooses to advertise at a quantity that Discovercorp' best response strategy is to advertise less. Since they collude on

the highest possible price (100$/drug) Genecorp wins approximately 52% of the market while Discovercorp receives 48%.

3. A closer examination between scenarios 1 and 2 yields an extremely interesting result. While Genecorp and Discovercorp receive the same market share between the two scenarios (52% and 48% respectively), the overall profits are not the same. In scenario 2, when Discovercorp is the leader and moves first, the equilibrium optimal strategies are to advertise less for both firms. Since the aggregate marketing is close to saturating market demand, less marketing actually generates more profits for both firms ($8404 vs. $8110 for Genecorp and $8210 vs. $7907 for Discovercorp). Therefore, **there is a strategic advantage to both firms if Discovercorp moves first in the asymmetric base case**. In order to explain this, we will analyze from a mathematical and practical standpoint:

   (a) From a mathematical standpoint, the bilevel problem need not have the same values for the leader and the follower when their order is switched. This would only happen if both the leader and the follower had the same objective and no constraints. In this case $\max_{\mathbf{x}} \max_{\mathbf{y}} f(\mathbf{x}, \mathbf{y}) = \max_{\mathbf{y}} \max_{\mathbf{x}} f(\mathbf{x}, \mathbf{y})$ (max-max problem). This relationship completely predicts the results of scenarios 3 and 4 in which Genecorp and Discovercorp have the same objective and no constraints (assuming Discovercorp accommodates Genecorp's entry in the market).

   (b) From a practical standpoint, the apparent discrepancy in scenarios 3 and 4 comes from the fact that this is not a simultaneous game between the two firms but a sequential one instead. Therefore, the follower reacts to the leader's actions and the leader anticipating the follower's reaction selects his optimal strategy. However, when the order is reversed, then the former leader (now follower) responds to a given strategy in his optimal way. In order to reinforce this point, assume that we are looking at scenario 1 and Genecorp (leader) chooses its optimal reaction strategy from scenario

2 (where Genecorp is the follower), i.e. $x_1 = 1, x_2 = 100, x_3 = 4395$. Discovercorp's (follower) optimal reaction strategy is $y_2 = 100, y_3 = 4213$ in which case the profits for Genecorp are \$7969 which is clearly suboptimal with regards to its optimal strategy $x_1 = 1, x_2 = 100, x_3 = 4557$ with profits of \$8110 (line 1 of Table 9.2).

4. In the asymmetric hostile cases (scenarios 5 and 6) there are two very important results:

   (a) In the case of Genecorp being the leader, the introduction of a one-time penalty for Discovercorp if Genecorp enters the market actually does not change the equilibrium strategies of both firms. They advertise and price exactly the same way as the asymmetric base case and they get the same market shares (52% for Genecorp and 48% for Discovercorp). The only difference are the overall profits for Discovercorp that are reduced by the exact amount of the one-time penalty for the firm. **Therefore, in the case of a one-time penalty, when Genecorp moves first, Discovercorp will not change its optimal response strategy. The reason for that is because if Genecorp enters then all the outcomes will be equally less profitable by the fixed penalty amount.** Therefore, the optimal response strategy of Discovercorp does not actually change.

   (b) In the case of Discovercorp being the leader, this is the only outcome where Discovercorp actually does not allow Genecorp to enter the market. To do so, it actually prices at a lower price than the highest possible one (82 \$/drug) and advertises at the maximum rate (10000 \$). Genecorp's optimal strategy is not to enter the market under these conditions and Discovercorp gets 100% of the market. It is worth mentioning that if Discovercorp was forced to allow entry for Genecorp (i.e., $x_1 = 1$) then Discovercorp's optimal strategy is the exact same as the reverse asymmetric case with no penalty in which Discovercorp receives 48% of the market while Genecorp receives 52%. **Therefore, if Discovercorp's objective**

has a one-time penalty from Genecorp's entrance then Discovercorp has an incentive to move first strategically and not allow entry.

5. Overall, apart from the symmetric scenarios (3,4) there is a an incentive for the two firms to move either first or second. In the asymmetric non-hostile case (1,2) both firms have a strategic advantage if Discovercorp moves first. However, in the asymmetric hostile case (5,6) Discovercorp has a incentive to move first and Genecorp also has an incentive to move first.

6. Our results verify the fact that in duopoly competition it is not always advantageous to move first in the market [80].

9.8 Final Remarks & Contribution

We have attempted to model the strategic interactions in duopoly markets using Stackelberg games with nonconvex objectives for both the leader and the follower and to solve the resulting nonconvex bilevel optimization problem using the latest numerical method in the field [82]. The main contributions of this chapter are:

1. The market participants are allowed to have more than one strategic choice. Unlike the current literature in which either price or quantity selection is allowed, we allow for higher dimensional strategic spaces.

2. The profit functions of both the leader and more importantly, the follower, are highly nonconvex. The recent advances in bilevel optimization [82] allow for the introduction of nonconvex functions whereas the current literature, in an attempt to provide closed-form solutions, assumes linearity or at best, convexity, of both the leader's and the follower's sub-problem.

We believe that the strategic interaction of duopolists is inherently nonconvex (especially in higher dimensional strategic spaces including pricing, marketing and quantity selection strategies) and therefore the current work can serve as a tool for exploring these interactions.

Appendix A

SIP Test Set using Interval Methods (Chapter 2)

The problems 1, 2, 3 and H

1.

$$\begin{aligned} f(\mathbf{x}) &= \frac{1}{3}x_1^2 + \frac{1}{2}x_1 + x_2^2 - x_2 \\ g(\mathbf{x}, p) &= x_1^2 + 2x_1x_2p - \sin(p) \\ X &= [-10, 10]^2 \\ P &= [0, 2]. \end{aligned}$$

2.

$$\begin{aligned} f(\mathbf{x}) &= \frac{1}{3}x_1^2 + x_2^2 + \frac{1}{2}x_1 \\ g(\mathbf{x}, p) &= (1 - x_1^2\,p^2)^2 - x_1\,p^2 - x_2^2 + x_2 \\ X &= [-1, 1] \\ P &= [0, 1]. \end{aligned}$$

3.

$$
\begin{aligned}
f(\mathbf{x}) &= x_1^2 + x_2^2 + x_3^2 \\
g(\mathbf{x}, p) &= x_1 + x_2 \exp(x - 3p) + \exp(2p) - 2\sin(4p) \\
X &= [-10, 10]^3 \\
P &= [0, 1].
\end{aligned}
$$

H.

$$
\begin{aligned}
f(\mathbf{x}) &= x_2 \\
g(\mathbf{x}, p) &= -(x_1 - p)^2 - x_2 \\
X &= [0, 1] \times [-100, 100] \\
P &= [0, 1].
\end{aligned}
$$

Appendix B

B.1 SIP Test Set using Relaxation-Based Bounds (Chapter 3)

For consistency purposes we use the problem labels of Watson [135]

2.

$$\begin{aligned} P &= [0,1] \\ f(\mathbf{x}) &= \frac{1}{3}x_1^2 + x_2^2 + \frac{1}{2}x_1 \\ g(\mathbf{x},p) &= (1 - x_1^2\,p^2)^2 - x_1\,p^2 - x_2^2 + x_2. \end{aligned}$$

5.

$$\begin{aligned} P &= [0,1] \\ f(\mathbf{x}) &= e^{x_1} + e^{x_2} + e^{x_3} \\ g(\mathbf{x},p) &= \frac{1}{1+p^2} - x_1 - x_2\,p - x_3\,p^2. \end{aligned}$$

6.

$$
\begin{aligned}
P &= [0,1] \\
f(\mathbf{x}) &= (x_1 - 2\,x_2 + 5\,x_2^2 - x_2^2\,x_2 - 13)^2 + (x_1 - 14x_2 + x_2^2 + x_2^3 - 29)^2 \\
g(\mathbf{x},p) &= x_1^2 + 2\,x_2\,p^2 + e^{x_1+x_2} - e^p.
\end{aligned}
$$

Note that in [26] the exponent is missing in the first term of the objective function.

7.

$$
\begin{aligned}
P &= [0,1]^2 \\
f(\mathbf{x}) &= x_1^2 + x_2^2 + x_3^2 \\
g(\mathbf{x},\mathbf{p}) &= x_1\,(p_1 + p_2^2 + 1) + x_2\,(p_1\,p_2 - p_2^2) + x_3\,(p_1\,p_2 + p_2^2 + p_2) + 1.
\end{aligned}
$$

8.

$$
\begin{aligned}
P &= [0,1]^2 \\
f(\mathbf{x}) &= x_1 + \frac{1}{2}x_2 + \frac{1}{2}x_3 + \frac{1}{3}x_4 + \frac{1}{4}x_5 + \frac{1}{3}x_6. \\
g(\mathbf{x},\mathbf{p}) &= e^{p_1^2+p_2^2} - x_1 - x_2\,p_1 - x_3\,p_2 - x_4\,p_1^2 - x_5\,p_1\,p_2 - x_6\,p_2^2.
\end{aligned}
$$

Note that presumably in Watson's collection [135] the coefficient of x_4 in the objective function is mistyped. This is suggested by the optimal solution value reported in [135] and by the symmetry of the problem with respect to the variables x_4 and x_6.

9.

$$
\begin{aligned}
P &= [-1,1]^2 \\
f(\mathbf{x}) &= -4\,x_1 - \frac{2}{3}(x_4 + x_6) \\
g(\mathbf{x},\mathbf{p}) &= x_1 + x_2\,p_1 + x_3\,p_2 + x_4\,p_1^2 + x_5\,p_1\,p_2 + x_6\,p_2^2 - 3 - (p_1 - p_2)^2\,(p_1 + p_2)^2.
\end{aligned}
$$

N.

$$P = [-1, 1]$$
$$f(\mathbf{x}) = x_2$$
$$g(\mathbf{x}, p) = 2\,x_1^2\,p^2 - p^4 + x_1^2 - x_2.$$

B.2 McCormick Relaxations

B.2.1 Nonsmooth Example

We will state McCormick's composition theorem and then provide an example that demonstrates that McCormick relaxation can be nonsmooth.

Theorem B.1 (McCormick's Composition Theorem)**.** *Let $X \subset \mathbb{R}^n$ be a nonempty convex set. Consider the function $H[h(\cdot)]$, where $h : X \to \mathbb{R}$ is a multivariate continuous function on X, and let $H : Z = [a, b] \supset h(X) \to \mathbb{R}$ be a univariate function. Suppose that there exists a convex function $c^u : X \to \mathbb{R}$ and a concave function $c^o : X \to \mathbb{R}$ that satisfy*

$$h^u(\mathbf{x}) \leq h(\mathbf{x}) \leq h^o(\mathbf{x}), \quad \forall \mathbf{x} \in X.$$

Furthermore, assume that there exists a convex function $H^u : [a, b] \to \mathbb{R}$ and a concave function $H^o : [a, b] \to \mathbb{R}$ that satisfy

$$H^u(z) \leq H(z) \leq H^o(z), \quad \forall z \in [a, b].$$

Let z_{min} be a point at which C^u attains its minimum and z_{max} be a point at which C^o attains its maximum.

$$z_{min} = \arg\min_{z \in [a,b]} C^u(z)$$
$$z_{max} = \arg\max_{z \in [a,b]} C^o(z)$$

Then, the following results hold

$$H^u(z(\mathbf{x})) = H^u[\text{mid}\{h^u(\mathbf{x}), h^o(\mathbf{x}), z_{min}\}]$$

is a convex relaxation of $H(h(\mathbf{x}))$ *on* X *and,*

$$C^o(z(\mathbf{x})) = H^o[\text{mid}\{h^u(\mathbf{x}), h^o(\mathbf{x}), z_{max}\}]$$

where the mid function selects the middle value between the three scalar arguments.

The following example demonstrates that a McCormick relaxation can be nonsmooth.

Example B.2. Let $X \subset \mathbb{R}^2$ be a nonempty convex set defined as $X = \{\mathbf{x} \in \mathbb{R}^2 : -1 \leq x_1 \leq 1, -1 \leq x_2 \leq 1\}$. Consider the function $g(\mathbf{x}) = \frac{1}{2}(x_1^2 + x_2^2)$. This function can be written in composition form in the following way

$$\begin{aligned} h(\mathbf{x}) &= \frac{1}{2}(x_1^2 + x_2^2) \\ H(z) &= z,\ z \in Z = [0, 1], \end{aligned}$$

which obviously is not the optimal choice, but will demonstrate the possibility of nonsmoothness. A valid underestimator, h^u, and overestimator, h^o of h on X are

$$\begin{aligned} h^u(\mathbf{x}) &= x_1 + x_2 - 1 \\ h^o(\mathbf{x}) &= 1. \end{aligned}$$

Furthermore, a convex underestimator, H^u, of H on Z is obviously

$$H^u(z) = z.$$

Finally, it is clear that $z_{min} = 0$. This implies that the McCormick convex underesti-

mator of g is given by

$$H^u(z(\mathbf{x})) = \text{mid}(x_1 + x_2 - 1, 1, 0).$$

Evaluating the mid function

$$H^u(z(\mathbf{x})) = \begin{cases} 0, & \text{if } x_1 + x_2 - 1 \leq 0 \\ x_1 + x_2 - 1, & \text{otherwise.} \end{cases} \tag{B.1}$$

This clearly shows that H^u is a nonsmooth function.

B.2.2 Influence of x-dependence

This example shows how the $\mathbf{x}$-dependence of the constraints in semi-infinite programming, i.e., the functional form $g(\mathbf{x}, \mathbf{p})$, influences the McCormick composition in creating a concave overestimator g^o on P, for each $\mathbf{x} \in X$.

Example B.3. Consider the function $g : X \times P \to \mathbb{R}$ defined as $g(\mathbf{x}, \mathbf{p}) = x_3 e^{(x_1 p_1^2 + x_2 p_2^2)}$, where $X = [-1, 1]^3$ and $P = [0, 1]^2$. Depending on the sign of the three $\mathbf{x}$ variables, we proceed as follows (we will only list the options for x_1 and x_3, since there is a symmetry between x_1 and x_2)

1. $x_1 \geq 0$, $x_2 \geq 0$, $x_3 \geq 0$:

   For a fixed $\mathbf{x}$ satisfying these conditions we define the following functions

   $$\begin{aligned} h(\mathbf{p}) &= x_1 p_1^2 + x_2 p_2^2, \quad \mathbf{p} \in P = [0, 1]^2, \\ H(z) &= x_3 e^z, \quad z \in Z = [0, x_1 + x_2]. \end{aligned}$$

   Based on this observation, we can define the following convex and concave relaxations, h^u and h^o respectively, for h on P

   $$\begin{aligned} h^u(\mathbf{p}) &= x_1 p_1^2 + x_2 p_2^2, \quad \mathbf{p} \in P \\ h^o(\mathbf{p}) &= x_1 p_1 + x_2 p_2, \quad \mathbf{p} \in P. \end{aligned}$$

The concave overestimator H^o of H on Z is given by the secant

$$H^o(z) = x_3 \left(\frac{(e^{x_1+x_2} - 1)z}{x_1 + x_2} + 1 \right), \quad z \in Z.$$

Finally, z_{max} is defined as one of the points for which H^o attains its maximum

$$z_{max} \in \arg\max_{z \in Z} H^o(z) = \{x_1 + x_2\}.$$

Therefore, the concave overestimator of g with respect to p is defined as

$$H^o(z(\mathbf{p})) = H^o[h^o(\mathbf{p})] = x_3 \left(\frac{(e^{x_1+x_2} - 1)(x_1 p_1 + x_2 p_2)}{x_1 + x_2} + 1 \right).$$

For the other cases we similarly obtain

2. $x_1 \geq 0,\ x_2 \geq 0,\ x_3 < 0$

$$\begin{aligned}
h^u(\mathbf{p}) &= x_1 p_1^2 + x_2 p_2^2, \quad \mathbf{p} \in P \\
h^o(\mathbf{p}) &= x_1 p_1 + x_2 p_2, \quad \mathbf{p} \in P \\
H^o(\mathbf{z}) &= x_3 e^z, \quad z \in Z = [0, x_1 + x_2], \\
z_{max} &\in \arg\max_{z \in Z} H^o(z) = \{0\} \\
H^o(z(\mathbf{p})) &= H^o[h^u(\mathbf{p})] = x_3 e^{x_1 p_1^2 + x_2 p_2^2}.
\end{aligned}$$

3. $x_1 \geq 0,\ x_2 < 0,\ x_3 \geq 0$

$$\begin{aligned}
h^u(\mathbf{p}) &= x_1 p_1^2 + x_2 p_2, \quad \mathbf{p} \in P \\
h^o(\mathbf{p}) &= x_1 p_1 + x_2 p_2^2, \quad \mathbf{p} \in P \\
H^o(\mathbf{z}) &= x_3 \left(\frac{(e^{x_1} - e^{x_2})(z - x_2)}{x_1 - x_2} + e^{x_2} \right), \quad z \in Z = [x_2, x_1], \\
z_{max} &\in \arg\max_{z \in Z} H^o(z) = \{x_1\} \\
H^o(z(\mathbf{p})) &= H^o[h^o(\mathbf{p})] = x_3 \left(\frac{(e^{x_1} - e^{x_2})(x_1 p_1 + x_2 p_2^2 - x_2)}{x_1 - x_2} + e^{x_2} \right).
\end{aligned}$$

4. $x_1 \geq 0,\ x_2 < 0,\ x_3 < 0$

$$\begin{aligned} h^u(\mathbf{p}) &= x_1 p_1^2 + x_2 p_2, \quad \mathbf{p} \in P \\ h^o(\mathbf{p}) &= x_1 p_1 + x_2 p_2^2, \quad \mathbf{p} \in P \\ H^o(\mathbf{z}) &= x_3 e^z, \quad z \in Z = [x_2, x_1], \\ z_{max} &\in \arg\max_{z \in Z} H^o(z) = \{x_2\} \\ H^o(z(\mathbf{p})) &= H^o[h^u(\mathbf{p})] = x_3 e^{x_1 p_1^2 + x_2 p_2}. \end{aligned}$$

B.2.3 Relaxations of the Product of Two Functions

In this section we will show the convex and concave relaxations for a product of two functions $g_1(\mathbf{x})g_2(\mathbf{x})$ on X, compare also the treatment of trilinear terms in [79].

Assume that there exist convex functions g_1^u and g_2^u and concave functions g_1^o and g_2^o that satisfy

$$\begin{aligned} g_1^u(\mathbf{x}) \leq g_1(\mathbf{x}) \leq g_1^o(\mathbf{x}), \quad \forall \mathbf{x} \in X \\ g_2^u(\mathbf{x}) \leq g_2(\mathbf{x}) \leq g_2^o(\mathbf{x}), \quad \forall \mathbf{x} \in X. \end{aligned}$$

Furthermore, let $G_1^L, G_1^U, G_2^L, G_2^U$ satisfy

$$\begin{aligned} G_1^L \leq g_1(\mathbf{x}) \leq G_1^U, \quad \forall \mathbf{x} \in X \\ G_2^L \leq g_2(\mathbf{x}) \leq G_2^U, \quad \forall \mathbf{x} \in X. \end{aligned}$$

Then using the following definitions

$$\alpha_1(\mathbf{x}) = \begin{cases} G_2^L g_1^u(\mathbf{x}), & \text{if } G_2^L \geq 0 \\ G_2^L g_1^o(\mathbf{x}), & \text{otherwise.} \end{cases}$$

$$\alpha_2(\mathbf{x}) = \begin{cases} G_1^L g_2^u(\mathbf{x}), & \text{if } G_1^L \geq 0 \\ G_1^L g_2^o(\mathbf{x}), & \text{otherwise.} \end{cases}$$

$$\beta_1(\mathbf{x}) = \begin{cases} G_2^U g_1^u(\mathbf{x}), & \text{if } G_2^U \geq 0 \\ G_2^U g_1^o(\mathbf{x}), & \text{otherwise.} \end{cases}$$

$$\beta_2(\mathbf{x}) = \begin{cases} G_1^U g_2^u(\mathbf{x}), & \text{if } G_1^U \geq 0 \\ G_1^U g_2^o(\mathbf{x}), & \text{otherwise.} \end{cases}$$

$$\gamma_1(\mathbf{x}) = \begin{cases} G_2^L g_1^u(\mathbf{x}), & \text{if } G_2^L \leq 0 \\ G_2^L g_1^o(\mathbf{x}), & \text{otherwise.} \end{cases}$$

$$\gamma_2(\mathbf{x}) = \begin{cases} G_1^U g_2^u(\mathbf{x}), & \text{if } G_1^U \leq 0 \\ G_1^U g_2^o(\mathbf{x}), & \text{otherwise.} \end{cases}$$

$$\delta_1(\mathbf{x}) = \begin{cases} G_2^U g_1^u(\mathbf{x}), & \text{if } G_2^U \leq 0 \\ G_2^U g_1^o(\mathbf{x}), & \text{otherwise.} \end{cases}$$

$$\delta_2(\mathbf{x}) = \begin{cases} G_1^L g_2^u(\mathbf{x}), & \text{if } G_1^L \leq 0 \\ G_1^L g_2^o(\mathbf{x}), & \text{otherwise.} \end{cases}$$

A valid convex g^u and concave g^o relaxation of g on X are given by

$$g^u(\mathbf{x}) \geq \max\{\alpha_1(\mathbf{x}) + \alpha_2(\mathbf{x}) - G_1^L G_2^L, \beta_1(\mathbf{x}) + \beta_2(\mathbf{x}) - G_1^U G_2^U\}$$
$$g^o(\mathbf{x}) \leq \min\{\gamma_1(\mathbf{x}) + \gamma_2(\mathbf{x}) - G_1^U G_2^L, \delta_1(\mathbf{x}) + \delta_2(\mathbf{x}) - G_1^L G_2^U\}.$$

Bibliography

[1] Numerica Technology, Cambridge, MA, www.numericatech.com.

[2] C. S. Adjiman, I. P. Androulakis, and C. A. Floudas. A global optimization method, αBB, for general twice-differentiable constrained nlps - II: Implementation and computational results. *Computers and Chemical Engineering*, 22:1159–1179, 1998.

[3] C. S. Adjiman, I. P. Androulakis, and C. A. Floudas. Global optimization of mixed-integer nonlinear problems. *AIChE Journal*, 46(9):1769–1797, 2000.

[4] C. S. Adjiman, S. Dallwig, C. A. Floudas, and A. Neumaier. A global optimization method, αBB, for process design. *Computers and Chemical Engineering*, 22:1137–1158, 1998.

[5] C. S. Adjiman and C. A. Floudas. Rigorous convex underestimators for general twice-differentiable problems. *Journal of Global Optimization*, 9(1):23–40, 1996.

[6] I. G. Akrotirianakis and C. A. Floudas. A new class of improved convex underestimators for twice continuously differentiable constrained nlps. *Journal of Global Optimization*, 30(367-390), 2004.

[7] G. Alefeld and G. H. Mayer. Interval analysis: Theory and Applications. *Journal of Computational and Applied Mathematics*, 121:421–464, 2000.

[8] R. Amir. Cournot oligopoly and the theory of supermodular games. *Games and Economic Behavior*, 15(2):132–148, 1996.

[9] I. P. Androulakis. Kinetic mechanism reduction based on an integer programming approach. *AIChE Journal*, 46(2):361–371, 2000.

[10] S. Balakrishnan, P. Georgopoulos, I. Banerjee, and M. Ierapetritou. Uncertainty considerations for describing complex reaction systems. *AIChE Journal*, 48(12):2875–2889, 2002.

[11] I. Banerjee and M. G. Ierapetritou. Feasibility evaluation of nonconvex systems using shape reconstruction techniques. *Industrial & Engineering Chemistry Research*, 44(10):3638–3647, 2005.

[12] I. Banerjee and M. G. Ierapetritou. An adaptive reduction scheme to model reactive flow. *Combustion and Flame*, 144(3):619–633, 2006.

[13] J. F. Bard. Technical note: Some properties of the bilevel programming problem. *Journal of Optimization Theory and Applications*, 68(2):371–378, 1991.

[14] J. F. Bard. *Practical Bilevel Optimization*. Kluwer Academic Publishers, Dordrecht, 1998.

[15] E. Baumann. Optimal centered forms. *BIT*, 28(1):80–87, 1988.

[16] A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Mathematics of Ooperations Research*, 23(4):769–805, 1998.

[17] A. Ben-Tal and A. Nemirovski. Robust solutions of uncertain linear programs. *Operations Research Letters*, 25(1):1–13, 1999.

[18] A. Ben-Tal and A. Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming*, 88(3):411–424, 2000.

[19] A. Ben-Tal, A. Nemirovski, and C. Roos. Robust solutions of uncertain quadratic and conic-quadratic problems. *SIAM Journal on Optimization*, 13(2):535–560, 2002.

[20] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, Massachusetts, Second edition, 1999.

[21] D. Bertsimas, D. Pachamanova, and M. Sim. Robust linear optimization under general norms. *Operations Research Letters*, 32(6):510–516, 2004.

[22] D. Bertsimas and M. Sim. Robust Discrete Optimization and Network Flows. *mathematical programming*, 98(1-3):49–71, 2003.

[23] D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.

[24] D. Bertsimas and M. Sim. Tractable approximations to robust conic optimization problems. *Mathematical Programming*, 107(1-2):5–36, 2006.

[25] B. Bhattacharjee. *Kinetic Model Reduction using Integer and Semi-Infinite Programming, Ph.D. Thesis*. Massachusetts Institute of Technology, Cambridge, Massachusetts, 2003.

[26] B. Bhattacharjee, W. H. Green Jr., and P. I. Barton. Interval methods for semi-infinite programs. *Computational Optimization and Applications*, 30(1):63–93, 2005.

[27] B. Bhattacharjee, P. Lemonidis, W. H. Green Jr., and P. I. Barton. Global solution of semi-infinite programs. *Mathematical Programming*, 103(2):283–307, 2005.

[28] B. Bhattacharjee, D. A. Schwer, P. I. Barton, and W. H. Green Jr. Optimally-reduced kinetic models: reaction elimination in large-scale kinetic mechanisms. *Combustion and Flame*, 135:191–208, 2003.

[29] J. W. Blankenship and J. E. Falk. Infinitely constrained optimization problems. *Journal of Optimization Theory and Applications*, 19(2):261–281, 1976.

[30] A. Brooke, D. Kendrick, A. Meeraus, and R. Raman. GAMS 250: A user's guide. http://www.gams.com/, 2004.

[31] A. W. Charnes, W. Cooper, and K. O. Kortanek. Duality, Haar programs and finite sequence spaces. *U.S. Proceedings of the National Academy of Sciences*, 48:783–786, March 1962.

[32] I. D. Coope and C. J. Price. *Exact Penalty Function Methods for Nonlinear Semi-Infinite Programming, in: Semi-Infinite Programming, eds. R. Reemtsen and J. -J. Rückmann.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.

[33] I. D. Coope and G. A. Watson. A projected Lagrangian algorithm for semi-infinite programming. *Mathematical Programming*, 32(3):337–356, 1985.

[34] R. Correa and H. C. Ramirez. A global algorithm for nonlinear semidefinite programming. *SIAM Journal on Optimization*, 15(1):303–318, 2004.

[35] K. G. Dastidar. On Stackelberg games in a homogeneous product market. *European Economic Review*, 48:549–562, 2004.

[36] Stephan Dempe. *Foundations of Bilevel Programming.* Nonconvex Optimization and its Applications. Kluwer Academic Publishers, Dordrecht, 2002.

[37] V. Deshmukh, H. T. Ma, and E. A. Butchert. Optimal control of parametrically excited linear delay differential systems via Chebyshev polynomials. *Optimal Control Applications & Methods*, 27(3):123–136, 2006.

[38] M. Diehl, H. G. Bock, and E. Kostina. An Approximation Technique For Robust Nonlinear Optimization. *Mathematical Programming*, 107(1-2):213–230, 2006.

[39] K. Edwards, T. F. Edgar, and V. I. Manousiouthakis. Kinetic model reduction using genetic algorithms. *Computers and Chemical Engineering*, 22(1-2):239–246, 1998.

[40] A. El-Kady and E. M. E. Elbarbary. A Chebyshev expansion method for solving nonlinear optimal control problems. *Applied Mathematics and Computation*, 129(2-3):171–182, 2002.

[41] L. Elliott, D. B. Ingham, A. G. Kyne, N. S. Mera, M. Pourkashanian, and C. W. Wilson. Reaction mechanism reduction and optimization using genetic algorithms. *Industrial and Engineering Chemistry Research*, 44(4):658–667, 2005.

[42] C. Floudas and O. Stein. The adaptive convexification algorithm: A feasible point method for semi-infinite programming. *SIAM Journal of Optimization*, 18(4):1187–1208, 2007.

[43] K. Fujiwara. A Stackelberg game model of dynamic duopolistic competition with sticky prices. *Economics Bulletin*, 12(12):1–9, 2006.

[44] E. P. Gatzke, P. Kesavan, and P. I. Barton. Parallel nonconvex mixed integer nonlinear optimization using outer approximation. In *AIChE Annual Meeting*, 2001.

[45] A. Geletu and A. Hoffmann. A conceptual method for solving generalized semi-infinite programming problems via global optimization by exact discontinuous penalization. *European Journal of Operational Research*, 157:3–15, 2004.

[46] P. E. Gill, W. Murray, and M. A. Saunders. SNOPT: An SQP algorithm for large scale constrained optimization. *SIAM Review*, 47(1):99–131, 2005.

[47] G. J. Glover. The influence of market exclusivity on drug availability and medical innovations. *The AAPS Journal 2007*, 9(3):E312–E316, 2007.

[48] C. Gonzaga, E. Polak, and R. Trahan. Improved algorithm for optimization problems with functional inequality constraints. *IEEE Transactions On Automatic Control*, 25(1):49–54, 1980.

[49] V. Goyal and M. Ierapetritou. Determination of operability limits using simplicial approximation. *AIChE Journal*, 48(12):2902–2909, 2002.

[50] V. Goyal and M. Ierapetritou. Framework for evaluating the feasibility/operability of nonconvex processes. *AIChE Journal*, 49(5):1233–1240, 2003.

[51] T. J. Graettinger and B. H. Krogh. The acceleration radius: a global performance measure for robotic manipulators. *Robotics and Automation, IEEE Journal of Robotics and Automation*, 4(1):60–69, 1988.

[52] R. Gramlich, G. Hettich and E.W. Sachs. Local convergence of SQP methods in semi-infinite programming. *SIAM Journal On Optimization*, 5(3):641–658, 1995.

[53] A. Haar. Über lineare ungleichungen. *Acta. Math.*, 2:1–14, 1924.

[54] E. Haaren-Retagne. *Semi-Infinite Programming Algorithm for Robot Trajectory Planning*. PhD thesis, Universitat Trier, 1992.

[55] E. Hansen. *Global Optimization Using Interval Analysis*. Marcel Dekker, New York, NY, 1993.

[56] P. Hansen, B. Jaumard, and J. Xiong. Decomposition and interval arithmetic applied to global minimization of polynomial and rational functions. *Journal of Global Optimization*, 3(4):421–437, 1993.

[57] R. Hettich. A review of numerical-methods for semi-infinite optimization. *Lecture Notes in Economics and Mathematical Systems*, 215:158–178, 1983.

[58] R. Hettich. An implementation of a discretization method for semi-infinite programming. *Mathematical Programming*, 34(3):354–361, 1986.

[59] R. Hettich and G. Still. On generalized semi-infinite programs. Working paper, University of Trier, 1986.

[60] R. Hettich and G. Still. Second order optimality conditions for generalized semi-infinite programming problems. *Optimization*, 34:195–211, 1995.

[61] R. P. Hettich and H. Th. Jongen. Semi-infinite programming: Conditions of optimality and applications. In *Optimization Techniques, Proc. 8th IFIP Conf. on Optimization Techniques*, Part 2. Springer-Verlag, New York, 1977.

[62] R. Horst. Deterministic global optimization with partition sets whose feasibility is not known: Application to concave minimization, reverse convex constraints, DC-programming and lipschitzian optimization. *Journal of Optimization Theory and Applications*, 58(1):11–37, 1988.

[63] R. Horst and H. Tuy. *Global Optimization: Deterministic Approaches*. Springer, Berlin, Third edition, 1996.

[64] H. T. Jongen, J.-J. Rückmann, and O. Stein. Disjunctive optimization: Critical point theory. *Journal of Optimization Theory and Applications*, 93(2):321–336, 1997.

[65] H. T. Jongen, J.-J Rückmann, and O. Stein. Generalized semi-infinite optimization: A first order optimality condition and examples. *Mathematical Programming*, 83(1):145–158, 1998.

[66] D. H. Jung and B. C. Lee. Development of a simple and efficient method for robust optimization. *International Journal for Numerical Methods in Engineering*, 53(9):2201–2215, 2002.

[67] H. A. Kandil and P. Bertrand. Government-private sector relations as a stackelberg game, degenerate case. *Journal of Economic Dynamics and Control*, 11:513–517, 1987.

[68] A. Kaplan and R. Tichatschke. *On a class of terminal variational problems, in: Guddat et al. (Eds.), Parametric Optimization and Related Topics IV*. Peter Lang Verlag, 1997.

[69] D. Kleis and E. W. Sachs. Optimal control of the sterilization of prepackaged food. Technical report, Universitat Trier, 1997.

[70] K. O. Kortanek. *On the 1962-1972 Decade of Semi-Infinite Programming: a Subjective View, in: Semi-Infinite Programming, Recent Advances, M. A. Goberna and M. A. Lopez*. Kluwer Academic Publishers, 2001.

[71] K. O. Kortanek and W. L. Gorr. Numerical aspects of pollution abatement problems: Optimal control strategies for air quality standards. In R. Wartmann M. Henke, A. Jaeger and J. H. Zimmerman, editors, *Proceedings in Operations Research*, pages 34–58. Physic-Verlag, 1972.

[72] L. Lambertini. Extended games played by managerial firms. *The Japanese Economic Review*, 51(2):274–283, 2000.

[73] C. T. Lawrence and A. L. Tits. *Feasible Sequential Quadratic Programming for Finely Discretized Problems from SIP, in: Semi-Infinite Programming, eds. R. Reemtsen and J. -J. Rückmann.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.

[74] C. K. Lee. *Global Optimization of Hybrid Systems.* PhD thesis, Department of Chemical Engineering Massachusetts Institure of Technology, July 2006.

[75] P. Lemonidis and P. I. Barton. Corrigendum to global solution of semi-infinite programs mathematical programming, series b, 103(2):283-307,2005]. In preparation, 2006.

[76] P. Lemonidis and P. I. Barton. Global solution of generalized-semi infinite programs. In preparation, 2008.

[77] P. Lemonidis, W. H. Green Jr., and P. I. Barton. Kinetic model reduction using generalized semi-infinite programming. In preparation, 2008.

[78] E. Levitin and R. Tichatschke. A branch-and-bound approach for solving a class of generalized semi-infinite programming problems. *Journal of Global Optimization*, 13:299–315, 1998.

[79] C. D. Maranas and C. A. Floudas. Finding all solutions of nonlinearly constrained systems of equations. *Journal of Global Optimization*, 7(2):143–182, 1995.

[80] T. Matsumura. Stackelberg mixed duopoly with a foreign competitor. *Bulletin of Economic Research*, 55(3):275–287, 2003.

[81] G. P. McCormick. Computability of global solutions to factorable nonconvex programs. Convex underestimating problems. *Mathematical Programming*, 10(2):147–175, 1976.

[82] A. Mitsos. Global solution of nonlinear mixed-integer bilevel programs. *Submitted to: Journal of Global Optimization*, April 3, 2008.

[83] A. Mitsos, P. Lemonidis, and P. I. Barton. Global optimization of bilevel programs with a nonconvex inner program. In Press: Journal of Global Optimization http://dx.doi.org/10.1007/s10898-007-9260-z, Oct 12, 2007.

[84] A. Mitsos, P. Lemonidis, C. K. Lee, and P. I. Barton. Relaxation-based bounds for semi-infinite programs. *SIAM Journal of Optimization*, 19(1):77–113, 2008.

[85] R. Moore. *Methods and Applications of Interval Analysis*. SIAM, Philadelphia, PA, 1979.

[86] S. Nordebo, I. Claesson, and S. Nordholm. Weighted Chebyshev approximation for the design of broadband beamformers using quadratic programming. *IEEE Singal Processing Letters*, 46(7):103–105, 1999.

[87] H.-T. Normann. Endogenous timing with incomplete information and with observable delay. *Games and Economic Behavior*, 39:282–291, 2002.

[88] M. S. Okino and M. L. Mavrovouniotis. Simplification of mathematical models of chemical reaction systems. *Chemical Reviews*, 98(2):391–408, 1998.

[89] M. Okuda, K. Kiyose, M. Ikehara, and S. Takahashi. Equiripple design in complex domain for FIR digital filters by transforming desired response. *Electronics and Communications in Japan Part III-Fundamental Electronic Science*, 84(2):30–37, 2001.

[90] O. O. Oluwole, B. Bhattacharjee, J. E. Tolsma, P. I. Barton, and W. H. Green Jr. Obtaining accurate solutions using reduced chemical kinetic models: A new model reduction method for models rigorously validated over ranges. *Combustion Theory and Modelling*, 146:348–365, 2006.

[91] N. S. V. Paluri and S. Sondur. Extrapolation based interval approach to compute the frequency responses of non-rational transfer functions with nonlinear parametric dependencies. *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Conference Control Conference*, 2005.

[92] E. R. Panier and A. L. Tits. A globally convergent algorithm with adaptively refined discretization for semi-infinite optimization problems arising in engineering design. *IEEE Transactions on Automatic Control*, 34(8):903–908, 1989.

[93] E. Polak. On the mathematical foundations of nondifferentiable optimization in engineering design. *SIAM Review*, 29(1):21–89, 1987.

[94] E. Polak. On the use of consistent approximations in the solution of semi-infinite optimization and optimal-control problems. *Mathematical Programming*, 62(2):385–414, 1993.

[95] E. Polak and D. Q. Mayne. Algorithm for optimization problems with functional inequality constraints. *IEEE Transactions on Automatic Control*, 21(2):184–193, 1976.

[96] S. B. Pope. Computationally efficient implementation of combustion chemistry using in situ adaptive tabulation. *Combustion Theory and Modelling*, 1(1):41–63, 1997.

[97] C. J. Price and I. D. Coope. Numerical experiments in semi-infinite programming. *Computational Optimization and Applications*, 6:169–189, 1995.

[98] H. Ratschek and J. Rokne. *Computer Methods for the Range of Functions*. Ellis Horwood Series, Mathematics and its Applications, New York, 1984.

[99] R. Reemtsen. A cutting plane method for solving minimax problems in the complex plane. *Numerical Algorithms*, 2:409–436, 1992.

[100] R. Reemtsen. Some outer approximation methods for semi-infinite optimization problems. *Journal of Computational and Applied Mathematics*, 53(1):87–108, 1994.

[101] R. Reemtsen and Görner. *Numerical Methods for Semi-Infinite Programming: A Survey, in: Semi-Infinite Programming, R. Reemtsen and J.-J. Rückmann.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.

[102] A. J. Robson. Duopoly with endogenous strategic timing: Stackelberg regained. *International Economic Review*, 31(2):263–274, 1990.

[103] T. Roh and L. Vandenberghe. Discrete transforms, semidefinite programming, and sum-of-squares representations of nonnegative polynomials. *SIAM Journal on Optimization*, 16(4):939–964, 2006.

[104] J.-J. Rückmann and A. Shapiro. First-order optimality conditions in generalized semi-infinite optimization. *Journal of Optimization Theory and Applications*, 101(3):677–691, 1999.

[105] J.-J. Rückmann and A. Shapiro. Second-order optimality conditions in generalized semi-infinite optimization. *Set-Valued Analysis*, 9:169–186, 2001.

[106] J.-J. Rückmann and O. Stein. On linear and linearized generalized semi-infinite optimization problems. *Annals of Operations Research*, 101:191–208, 2001.

[107] E. W. Sachs. *Semi-Infinite Programming in Control, in: Semi-Infinite Programming, R. Reemtsen and J.-J. Rückmann.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.

[108] N. Sahinidis and M. Tawarmalani. BARON. Published Online: http://www.gams.com/solvers/baron.pdf, 2005.

[109] A. R. Sirdeshpande, M. G. Ierapetritou, and I. P. Androulakis. Design of flexible reduced kinetic mechanisms. *AIChE Journal*, 47(11):2461–2473, 2001.

[110] E. M. B. Smith and C. C. Pantelides. A symbolic reformulation / spatial branch-and-bound algorithm for the global optimization of nonconvex MINLPs. *Computers and Chemical Engineering*, 142:457–478, 1999.

[111] M. D. Smooke, R. E. Mitchell, and D. E. Keyes. Numerical-solution of 2-dimensional axisymmetric laminar diffusion flames. *Combustion Science and Technology*, 67(4-6):85–122, 1989.

[112] J. Song, G. Stephanopoulos, and W. H. Green Jr. Valid parameter range analyses for chemical reaction kinetic models. *Chemical Engineering Science*, 57:4475–4491, 2002.

[113] O. Stein. On the level sets of marginal functions. University of Aachen, Preprint No. 80, 1998.

[114] O. Stein. *The Reduction Ansatz in absence of lower semi-continuity, in: Parametric Optimization and Related Topics V*, pages 165–178. Peter Lang Verlag, Frankfurt a.M., 2000.

[115] O. Stein. *Approximation, Optimization and Mathematical Economics*, pages 309–327. Physica, Heidelberg, 2001.

[116] O. Stein. First-order optimality conditions for degenerate index sets in generalized semi-infinite optimization. *Mathematics of Operations Research*, 26(3):565–582, 2001.

[117] O. Stein. *Optimization with Multivalued Mappings Theory, Applications and Algorithms in: S. Dempe and V. Kalashnikov (Eds.)*, volume 2, pages 209–228. Springer US, 2006.

[118] O. Stein and G. Still. On optimality conditions for generalized semi-infinite programming problems. *Journal of Optimization Theory and Applications*, 104(2):443–458, 2000.

[119] O. Stein and G. Still. On generalized semi-infinite optimization and bilevel optimization. *European Journal of Operational Research*, 142(3):444–462, 2002.

[120] O. Stein and G. Still. Solving semi-infinite optimization problems with interior point techniques. *SIAM Journal on Control and Optimization*, 42(3):769–788, 2003.

[121] G. Still. Generalized semi-infinite programming: Theory and methods. *European Journal of Operational Research*, 119(2):301–313, 1999.

[122] G. Still. Generalized semi-infinite programming: Numerical aspects. *Optimization*, 49(3):223–242, 2001.

[123] K. Suyama, K. Takada, R. Hirabayashi, and H. Iwakura. Complex Chebyshev approximation for FIR filters using linear semi-infinite programming method. *Electronics And Communications in Japan Part III-Fundamental Electronic Science*, 87(3):49–57, 2004.

[124] Y. Tanaka. A trust region method for semi-infinite programming problems. *International Journal of Systems Science*, 30(2):199–204, 1999.

[125] Y. Tanaka, M. Fukushima, and T. Ibaraki. A globally convergent SQP method for semi-infinite nonlinear optimization. *Journal of Computational and Applied Mathematics*, 23(2):141–153, 1988.

[126] M. Tawarmalani and N. V. Sahinidis. *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming.* Nonconvex Optimization and its Applications. Kluwer Academic Publishers, Boston, 2002.

[127] M. Tawarmalani and N. V. Sahinidis. Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. *Mathematical Programming*, 99(3):563–591, 2004.

[128] J. E. Tolsma and P. I. Barton. http://yoric.mit.edu/daepack/daepack.html.

[129] P. I. Tolsma, J. E. Barton. DAEPACK: An open modeling environment for legacy models. *Industrial and Engineering Chemistry Research*, 39(6):1826–1839, 2000.

[130] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.

[131] L. Vandenberghe and S. Boyd. Applications of semidefinite programming. *Applied Numerical Mathematics*, 29(3):283–299, 1999.

[132] A. I. F. Vaz, E. M. G. P. Fernandes, and M. P. S. F. Gomes. A quasi-Newton interior point method for semi-infinite programming. *Optimization Methods & Software*, 18(6):673–687, 2003.

[133] A. I. F. Vaz, E. M. G. P. Fernandes, and M. P. S. F. Gomes. SIPAMPL: Semi-infinite programming with AMPL. *ACM Transactions on Mathematical Software*, 30(1):47–61, 2004.

[134] F. G. Vazquez and J.-J. Rückmann. Extensions of the Kuhn-Tucker constraint qualification to generalized semi-infinite programming. *SIAM Journal of Optimization*, 15(3):926–937, 2005.

[135] G. A. Watson. Numerical experiments with globally convergent methods for semi-infinite programming-problems. *Lecture Notes in Economics and Mathematical Systems*, 215:193–205, 1983.

[136] G. W. Weber. On the topology of generalized semi-infinite optimization. *Journal of Convex Analysis*, 9(2), 2002.

[137] G. W. Weber. *Generalized Semi-Infinite Optimization and Related Topics*, volume 29. Helderman Verlag, 2003.

[138] A. Winterfeld. Application of general semi-infinite programming to lapidary cutting problems. Technical report, Fraunhofer Institut Techno-und Wirtschaftmathematik, 2006.

[139] S. Zakovic, B. Rustem, and S. P. Asprey. A parallel algorithm for semi-infinite programming. *Computational Statistics & Data Analysis*, 44(1-2):377–390, 2003.