Disjunctive Aspects in Generalized Semi-infinite Programming

Zur Erlangung des akademischen Grades eines
Doktors der Ingenieurwissenschaften

(Dr.-Ing.)

von der
KIT-Fakultät für Wirtschaftswissenschaften
des Karlsruher Instituts für Technologie (KIT)

genehmigte

DISSERTATION

von

Dipl.-Wi.-Ing. Peter Kirst

Tag der mündlichen Prüfung:	12.06.2019
Referent:	Prof. Dr. Oliver Stein
Korreferent:	Prof. Dr. Gabriele Eichfelder

Karlsruhe 2019

Abstract

In this thesis the close relationship between generalized semi-infinite problems (GSIP) and disjunctive problems (DP) is considered. We start with the description of some optimization problems from timber industry and illustrate how GSIPs and DPs arise naturally in that field. Three different applications are reviewed.

Next, theory and solution methods for both types of problems are examined. We describe a new possibility to model disjunctive optimization problems as generalized semi-infinite programs. Applying existing lower level reformulations for the obtained semi-infinite program we derive conjunctive nonlinear problems without any logical expressions, which can be locally solved by standard nonlinear solvers.

In addition to this local solution procedure we propose a new branch-and-bound framework for global optimization of disjunctive programs. In contrast to the widely used reformulation as a mixed-integer program, we compute the lower bounds and evaluate the logical expression in one step. Thus, we reduce the size of the problem and work exclusively with continuous variables, which is computationally advantageous. In contrast to existing methods in disjunctive programming, none of our approaches expects any special formulation of the underlying logical expression. Where applicable, under slightly stronger assumptions, even the use of negations and implications is allowed.

Our preliminary numerical results show that both procedures, the reformulation technique as well as the branch-and-bound algorithm, are reasonable methods to solve disjunctive optimization problems locally and globally, respectively.

In the last part of this thesis we propose a new branch-and-bound algorithm for global minimization of box-constrained generalized semi-infinite programs. It treats the inherent disjunctive structure of these problems by tailored lower bounding procedures. Three different possibilities are examined. The first one relies on standard lower bounding procedures from con-

junctive global optimization. The second and the third alternative are based on linearization techniques by which we derive linear disjunctive relaxations of the considered sub-problems. Solving these by either mixed-integer linear reformulations or, alternatively, by disjunctive linear programming techniques yields two additional possibilities. Our numerical results on standard test problems with these three lower bounding procedures show the merits of our approach.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

This dissertation thesis is based on a series of three articles [65–67] in which two different types of optimization problems and their similarities are examined.

Firstly, so-called disjunctive programs (DP) are considered. Here, in contrast to standard conjunctive problems the description of the feasible set may contain different logical expressions, meaning that the constraints are not only connected by conjunctions but also disjunctive operators or negations. A large number of practical applications of DP arise in industrial and chemical engineering. DPs are explained in more detail in Section 1.2.

Secondly, so-called standard and generalized semi-infinite programs (GSIP) are considered whose feasible set is described by an infinite number of constraints. Classical applications of these problems are

- Chebyshev approximation,
- design centering,
- robust optimization and
- minimax problems.

Formally, GSIPs are described in Section 1.1. Although no logical expression is modelled explicitly, the feasible set of generalized semi-infinite problems may have so-called re-entrant corner points which typically arise in disjunctive programming. Additionally, the feasible set does not need to be closed, although no strict inequality occurs in the description of the problem at the first glance. Although both kinds of problems, DPs as well as GSIPs, are

usually treated completely independently from each other in literature, these are strong indications for a certain relationship that is known for a long time. In this dissertation thesis this is examined in more detail.

This document is structured as follows. In the next two sections, generalized semi-infinite as well as disjunctive problems are described formally.

Then, in Chapter 2 applications of disjunctive as well as generalized semi-infinite programming are explained. Three different examples from timber industry are reviewed and it is illustrated how disjunctive as well as semi-infinite aspects may arise in practice.

Next, in Chapter 3 it is shown, how disjunctive problems can be reformulated as generalized semi-infinite programs, which reveals the close relationship of both problems in more detail. Solving these reformulations by existing GSIP solution techniques leads to local solution procedures for disjunctive programs. Three different such possibilities are considered.

The global solution of disjunctive problems is addressed in Chapter 4. In order to achieve this, a tailored branch-and-bound algorithm is developed based on classical lower bounding techniques from standard conjunctive programming.

In Chapter 5 a global solution algorithm for GSIPs based on discretization is explained. During this algorithm several disjunctive problems have to be solved to global optimality so that the branch-and-bound algorithm from Chapter 4 can be applied. Additionally, lower bounding procedures tailored to the proposed discretization method are considered.

Chapter 6 concludes this thesis with some final remarks.

The notation is standard. In particular, Df denotes the row vector of partial derivatives of the function f and the Hessian of a twice differentiable function f is denoted by $D^2 f$. Moreover, regarding a box $B \subset \mathbb{R}^n$ with $B = [\underline{x}_1, \overline{x}_1] \times \ldots \times [\underline{x}_n, \overline{x}_n]$ for the diagonal length we put $\operatorname{diag}(B) = \|\overline{x} - \underline{x}\|_2$ and for the geometrical midpoint we put $\operatorname{mid}(B) = \frac{1}{2}(\underline{x} + \overline{x})$.

1.1 Generalized Semi-infinite Programming

A generalized semi-infinite optimization problem is defined by

$$GSIP: \qquad \min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad x \in M_{GSIP}$$

with the feasible set

$$M_{GSIP} := \{x \in \mathbb{R}^n : \ g(x,y) \leq 0 \text{ for all } y \in Y(x)\}$$

with

$$Y(x) = \left\{ y \in \mathbb{R}^m \ \middle| \ v_i(x,y) \geq 0 \text{ for } i \in I \right\}.$$

In case that the set-valued mapping $Y(x)$ is constant we have a so called standard semi-infinite program, which is an optimization problem with a finite number of variables and a possibly infinite number of constraints. As basic references we mention [53] for an introduction to semi-infinite optimization, [54,92] for numerical methods in SIP, and the monographs [44] for linear semi-infinite optimization, as well as [88] for algorithmic aspects.

One approach to tackle standard semi-infinite problems is to consider a finite subset of the possibly infinite number of constraints and solve the resulting standard optimization problem. Subsequently adding new constraints to the finite approximation of the original index set leads to improved approximations of the (standard) semi-infinite problem. By iterating this process, a so-called discretization method is obtained (see, e.g., [25]). This procedure naturally yields an outer approximation at the feasible set and thus lower bounds at the optimal value. In contrast, in [23] an inner approximation of the feasible set is derived such that upper bounds at the globally minimal value can also be computed. The combination of lower and upper bounding techniques leads to a global optimization procedure for standard semi-infinite programs with an exact termination criterion as described in [24,75]. Similarly, the combination of the lower bounding procedure from [25] with a different upper bounding technique is used in [82]. Local solvers for standard semi-infinite programs are introduced in [42] and [115] via an MPCC reformulation of a relaxed lower level problem which is adaptively refined.

Even more difficult to handle are the so-called generalized semi-infinite programs (GSIP), where the index set of constraints may vary for different points in space. The monograph [112] contains a detailed study of generalized semi-infinite optimization. The most recent comprehensive reviews on theory, applications and numerical methods of standard and generalized semi-infinite optimization are [50,77,116].

In both cases, the standard as well as generalized one, infinitely many constraints appear, so that for checking feasibility of a given point $\overline{x} \in \mathbb{R}^n$ it turns out that the global supremal value of the following optimization problem has to be checked for nonpositivity:

$$Q(\overline{x}): \qquad \max_{y \in \mathbb{R}^m} g(\overline{x}, y) \qquad \text{s.t.} \qquad v_i(\overline{x}, y) \geq 0, \ i \in I.$$

Here, as usual, the case of an inconsistent feasible set is covered by the supremal value $-\infty$. The feasible set of $Q(x)$ varies in x for GSIPs while for SIPs it does not. Some important disjunctive aspects of generalized semi-infinite programming are illustrated in the following Example 1.1.1.

Example 1.1.1. *We consider the generalized semi-infinite problem with the feasible set described as above and defined by the function*

$$g(x,y) := \left(x_1 - \left(3 + \frac{1}{2}y\right)\right)^2 + \left(x_2 - \left(3 + \frac{1}{2}y\right)\right)^2 - \left(3 + \frac{1}{2}y\right)^2$$

and

$$Y(x) = \left\{ y \in [0,2] \;\middle|\; v_1(x,y) \geq 0 \right\}$$

with

$$v_1(x,y) := -\left(x_1 - \left(7 + \frac{1}{2}y\right)\right)^2 - \left(x_2 - \left(7 + \frac{1}{2}y\right)\right)^2 + \left(3 + \frac{1}{2}y\right)^2.$$

This is illustrated in Figure 1.1 where the level sets in the x-plane of g for the three different values 0, 1 and 2 of y are the circles in the lower left corner whereas the level sets of v_1 for the same values of y are the circles in the upper right corner.

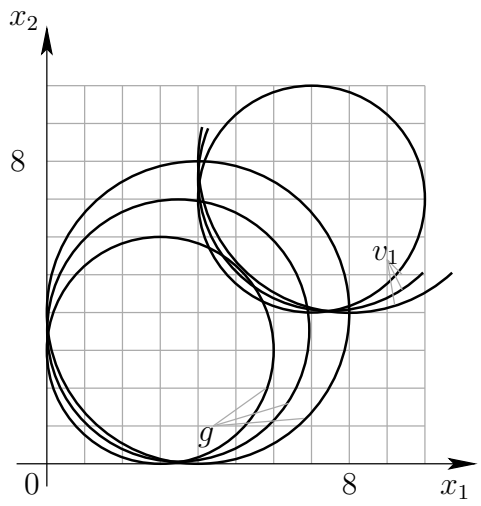

Figure 1.1: Level sets of g and v_1 for different values of y

We now consider an arbitrary point $\tilde{x}$ that is assumed to be feasible for the problem at hand. Then, for a given $y \in [0,2]$ it must not *hold*

$$g(\tilde{x},y) > 0 \qquad \textit{and} \qquad v_1(\tilde{x},y) \geq 0, \tag{1.1}$$

because that means that the supremal value of the lower level problem would be positive. For that reason, for our feasible point $\tilde{x}$ *and for some* $y \in [0,2]$ *the system of inequalities* (1.1) *must be violated and thus we either have*

$$g(\tilde{x}, y) \leq 0 \qquad or \qquad v_1(\tilde{x}, y) < 0,$$

which is illustrated in Figure 1.2 for three distinct values $0, 1$ *and* 2 *of* y *on top of each other. Since this must hold for all values of* $y \in [0,2]$*, with*

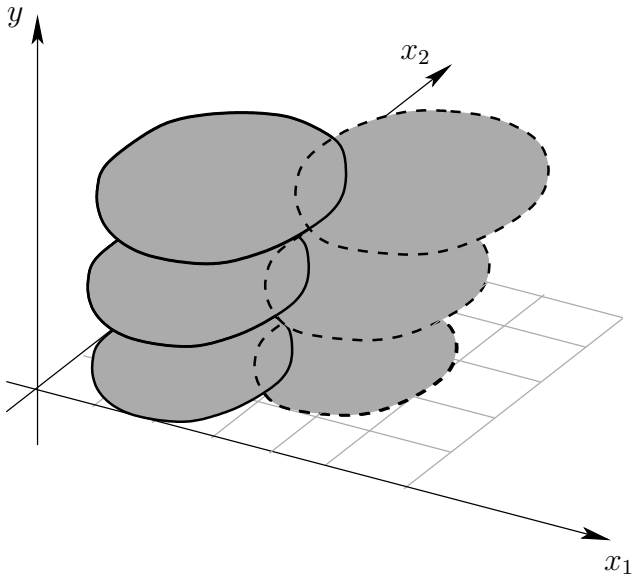

Figure 1.2: Feasible set of generalized semi-infinite program

$$S := \left\{ x \in \mathbb{R}^2 \;\middle|\; \Big(g(x,y) \leq 0\Big) \vee \Big(v_1(x,y) < 0\Big) \quad \forall y \in [0,2] \right\}$$

we obtain for the feasible set of $GSIP$ *the relation* $M_{GSIP} \subset S$*.*

On the other hand, in order to show $M_{GSIP} \supset S$ *we assume a point* $\tilde{x} \in S$*. Then, for the optimal value* $v^*(\tilde{x})$ *of* $Q(\tilde{x})$ *we have* $v^*(\tilde{x}) \leq 0$*, because otherwise there was some* $y \in [0,2]$ *with* $v_1(\tilde{x}, y) \geq 0$ *(thus, feasibility for the lower level problem) and* $g(\tilde{x}, y) > 0$ *(due to the fact that for the optimal value we then had* $v^*(\tilde{x}) > 0$*). So at least one of these two inequalities must be violated and thus it follows*

$$M_{GSIP} = \left\{ x \in \mathbb{R}^2 \;\middle|\; \Big(g(x,y) \leq 0\Big) \vee \Big(v_1(x,y) < 0\Big) \quad \forall y \in [0,2] \right\}.$$

This is depicted in Figure 1.3 on the right. Here we can see that the feasible set contains so-called re-entrant corner points and, moreover, is non-closed. Both aspects typically arise in disjunctive programming.

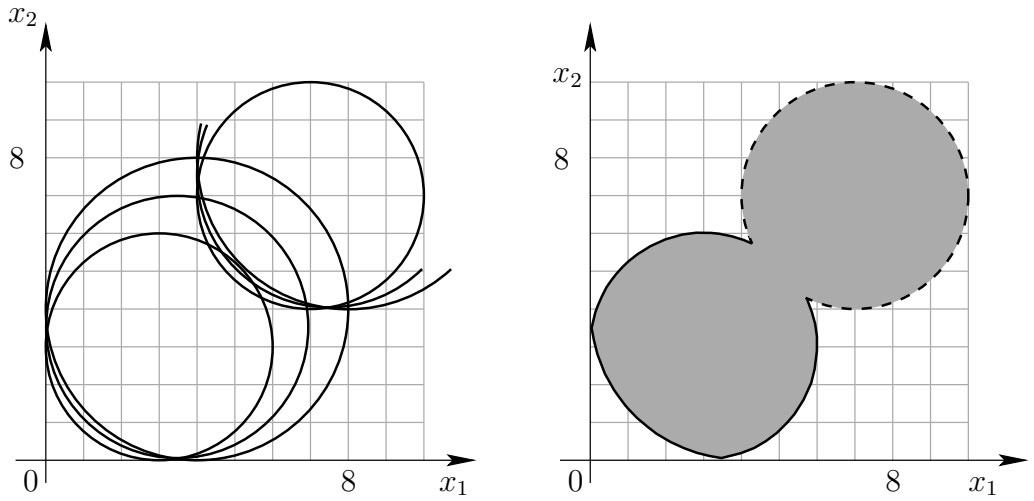

Figure 1.3: Level sets of g and v_1 for different values of y and feasible set of generalized semi-infinite program

The considerations of Example 1.1.1 can be generalized as introduced in [52]. In fact, it is stated that a generalized semi-infinite program can be rewritten as

$$GSIP: \quad \min_{x\in\mathbb{R}^n} f(x) \qquad \text{s.t.} \qquad x \in M$$

with

$$M_{GSIP} = \left\{ x \in \mathbb{R}^n \;\middle|\; \Big(g(x,y) \leq 0\Big) \vee \bigvee_{i\in I} \Big(v_i(x,y) < 0\Big) \quad \forall y \in \mathbb{R}^m \right\}. \tag{1.2}$$

Thus an inherent disjunctive structure becomes apparent that makes it difficult to extend existing methods for standard semi-infinite programs to the generalized semi-infinite case. However, in equation (1.2) the index set does not depend on x, although still being infinite. For that reason we can consider this problem as a kind of disjunctive standard semi-infinite program.

This reformulation is used in [83] to solve GSIPs to global optimality. Another approach for the global solution of generalized semi-infinite problems is proposed in [125] where a binary search of the objective function values is performed. The difficulties of the disjunctive structure is circumvented by using smoothing techniques. The challenges that arise for the generalized case but not for standard semi-infinite problems are also discussed in [117,118] from a theoretical as well as from a computational point of view. A numerical approach to GSIP using the discretization idea under certain limitations is discussed in the recent paper [106].

Under certain convexity assumptions so-called bilevel approaches are also applicable for *GSIP* such as described in [33,114]. Usually, the lower level problem is replaced by an equivalent optimality condition so that a reformulated lifted problem with a finite number of constraints is obtained that can be solved by standard nonlinear programming techniques. Similarly, in [76] a lifting approach is proposed for a special kind of *GSIP* whereas in contrast to other approaches the resulting nonlinear problem is solved to global optimality by means of branch-and-bound.

1.2 Disjunctive Programming

A disjunctive optimization problem is defined by

$$DP: \qquad \min_{x\in\mathbb{R}^n} f(x) \quad \text{s.t.} \quad x \in M_{DP}$$

whose feasible set M_{DP} is described by finitely many constraints of the form $g_i(x) \leq 0$, $i \in I$, with $I = \{1, \ldots, p\}$ and $p \in \mathbb{N}$. In contrast to standard nonlinear programs, there the constraint functions are not only coupled by conjunctions, but also by disjunctions.

All defining functions of DP are assumed to be continuous. Differentiability is not assumed in general, however, it may be imposed later on in this thesis depending on the configuration of algorithms or reformulations. As already mentioned, all defining constraints are connected by finitely many conjunctions and disjunctions. More precisely, we have

$$M_{DP} = \{x \in \mathbb{R}^n \mid \Omega(g_1(x) \leq 0, \ldots, g_p(x) \leq 0) = \text{true}\}$$

with a logical expression $\Omega : \{\text{true}, \text{false}\}^p \to \{\text{true}, \text{false}\}$ consisting of only conjunctions and disjunctions.

If the logical expression has the form

$$\Omega(g_1(x) \leq 0, \ldots, g_p(x) \leq 0) = \bigwedge_{i\in I} \bigvee_{j\in J_i} (g_{ij}(x) \leq 0)$$

with $J_i \subseteq \{1, \ldots, p\}$ for $i \in I$ and a finite index set I, we say that the problem is in conjunctive normal form. Then, the feasible set of DP can be rewritten as

$$\begin{aligned} M_{DP} &= \bigcap_{i\in I} \bigcup_{j\in J_i} \{x \in \mathbb{R}^n : g_{ij}(x) \leq 0\} \\ &= \left\{ x \in \mathbb{R}^n : \max_{i\in I} \min_{j\in J_i} g_{ij}(x) \leq 0 \right\}. \end{aligned}$$

On the other hand, if the logical expression has the form

$$\Omega(g_1(x) \leq 0, \ldots, g_p(x) \leq 0) = \bigvee_{i \in I} \bigwedge_{j \in J_i} (g_{ij}(x) \leq 0)$$

we say that the problem is in disjunctive normal form and the feasible set can be written as

$$\begin{aligned} M_{DP} &= \bigcup_{i \in I} \bigcap_{j \in J_i} \{x \in \mathbb{R}^n : g_{ij}(x) \leq 0\} \\ &= \left\{ x \in \mathbb{R}^n : \min_{i \in I} \max_{j \in J_i} g_{ij}(x) \leq 0 \right\}. \end{aligned}$$

In this thesis, we do not assume Ω to be in any normal form in contrast to most of the papers from the literature. While all logical expressions consisting of only conjunctions and disjunctions can be transformed into disjunctive or conjunctive normal form, the number of terms may increase exponentially. For example, the disjunctive normal form of $(A_1 \vee B_1) \wedge \ldots \wedge (A_n \vee B_n)$ has 2^n terms.

By introducing

$$Y_i(x) := \begin{cases} \text{true} & \text{if } g_i(x) \leq 0 \\ \text{false} & \text{if } g_i(x) > 0 \end{cases}$$

for $i = 1, \ldots, p$ we can equivalently rewrite the feasible set as

$$M_{DP} = \{x \in \mathbb{R}^n \mid \Omega(Y_1(x), \ldots, Y_p(x)) = \text{true}\}.$$

Note that our definition of $Y_i(x)$ differs from a standard one in disjunctive optimization, where $Y_i(x) = \text{false}$ means that the constraint $g_i(x) \leq 0$ is ignored. For the purposes of this thesis, however, our definition is more convenient.

In the following, we briefly comment on the extensive literature available for disjunctive programming. A well-known approach under mild assumptions for modeling disjunctions is the so-called Big–M reformulation. This technique involves the introduction of a large constant as well as binary variables. The problem can then be solved by using techniques from mixed-integer programming. We refer to [85] for a detailed study of appropriate algorithms. It is worth mentioning that with a slight extension [136], also logical expressions that are not given in any normal form can be handled. Unfortunately this approach suffers from mainly two drawbacks: firstly, there may occur numerical instabilities due to the fact that the aforementioned constant has to be chosen rather large in many applications. Secondly, relaxing

the integrality constraints usually results in poor relaxations of the original problem. However, the method is widely used and also reveals the close relationship between disjunctive and mixed-integer optimization. In [12] it is stated that every MILP can be reformulated as a DP, and every bounded DP can be reformulated as a MILP. Moreover, differences and similarities of both kinds of problems are discussed in [19,55] and the advantages as well as disadvantages of the mixed-integer reformulations are examined.

To our knowledge, all the subsequently discussed known solution methods for disjunctive optimization problems either expect a conjunctive or a disjunctive normal form of the constraints. Our approach does without this assumption and may thus avoid an exponential growth in the number of constraint terms during normalization.

A huge amount of literature is devoted to linear disjunctive programming. For the case of linear mixed-integer programming, we mention [31] as one of the very first articles that address this kind of problems. A review of the early work in the field can be found in [11]. One of the main ideas is to minimize the objective function over the convex hull of the feasible set. In case of linear disjunctive programming this yields the same optimal value as a minimization over the original feasible set. Moreover, then the convex hull of the feasible set can be described as the orthogonal projection onto the original space of some set in a higher dimensional space. This approach is described in [12,14,15]. Further methods which explicitly consider logical expressions in linear programming can be found in [19,55]. More generally, logical expressions can always be modeled via linear constraints, together with integer variables ([89,90,136]).

For a review of (generalized) nonlinear disjunctive optimization we refer the reader to [46] and [91] as one of the very first articles in this field. In case of convex disjunctive programming, all occurring nonlinear functions can be approximated by multiple linear underestimators, and thus the techniques from linear disjunctive and linear mixed-integer programming are applicable. This leads to the so-called outer approximation algorithms as described in [37, 41]. A related method can be found in [135]. One of the first implementations of this theory is LogMIP as explained in [128]. In [99], the concept of "basic steps" defined for linear DPs in [12] is extended to nonlinear GDPs, thus, leading to a so-called hierarchy of relaxations.

The above idea of considering the convex hull of disjunctive optimization problems may also be extended to the case of nonlinear and convex problems without linearization of the defining functions, as described in [47,72]. Similar methods that address convex disjunctive problems can be found in [29] and

[119].

A critical point theory for nonlinear disjunctive optimization problems in conjunctive normal form with differentiable, but not necessarily convex defining functions, is developed in [60].

In case of nonconvex problems, in principle all branch-and-bound algorithms from global optimization are applicable via the mixed-integer reformulation. We mention [73] as an example because in this branch-and-bound algorithm, logical expressions are considered explicitly. An extension of this algorithm covering bilinear equality constraints is proposed in [74]. In the more recent articles [100,101] the above mentioned concept of a hierarchy of relaxations is applied to tighten the convex relaxations of the original problem.

Reviews of applications and solution techniques for MINLPs and GDPs, such as branch-and-bound, outer approximation, generalized Benders decomposition and extended cutting plane methods, can be found in [48,49,124].

One of the reasons why this area of research continues to be very active is the long list of applications, in particular in industrial and chemical engineering. These applications include distillation column design [27,49,57,137–139], scheduling problems [28,49,72,91,103], process synthesis and retrofit planning [45,49,72,103,126], strip-packing problems [103,104], pooling problems, waste water network problems, and water usage network problems [74].

Chapter 2

Applications of DP and GSIP in Timber Industry

In applications of optimization in industry, usually different approaches are possible. Actually, many real world problems are tackled using linear or mixed-integer linear programming. However, as these models evolve, often the need for more precise mathematical descriptions arises and thus nonlinear solvers are considered. Often even more improved models are taken into consideration that incorporate, for instance, uncertainty so that robust and semi-infinite approaches, come into question or, moreover, certain issues have to be expressed by disjunctive constraints.

In this section, these considerations are described along examples from timber industry, which is a classical field of applications where the need for optimization is recognized in the meantime. Note, that the applications of this chapter are only to illustrate the benefits of disjunctive and generalized semi-infinite programming on real life instances. However, the derived models are not tackled using the solution techniques described in the remaining chapters of this thesis (see also the explanation at the end of this chapter).

Timber industry is a broad area compromising many different aspects and, thus, there are different requirements regarding planning and optimization techniques. The entire workflow is described in [98] and always starts by harvesting and cutting stems into logs which is considered to be part of forestry. The advantages of optimization techniques are well-known in that area for many years. We briefly sketch some classical models in Section 2.1 and describe some possible applications of disjunctive programming that typically arise.

These logs are then transported to sawmills and cut into lumber. Here

the goal is to produce a certain required demand and, at the same time, exploiting the raw material at its best. Some of these models are described in Section 2.2. The waste is usually processed to wood chips which in turn are used by paper-mills, for instance.

Many products such as furniture or complete houses can be made of wood and also in this step of the workflow, during construction, optimization techniques are applicable. Typical products are trusses that are the basis for roofs as well as other wooden products and thus the whole area of truss topology design can be applied here. This is described in Section 2.3.

2.1 Forest Management

In forestry, operations research models are applied for a very long time. A well-known example is Forplan [62] in the United States. The operations research methods applied in the Chilean forestry are described in [38]. Similar systems are used in several other countries with large forests all over the world.

The goal of forest management is to decide, which areas of the forest are harvested by which crew at what time. This usually leads to mixed-integer linear programs. An example of such a problem taken from [98] is given below. Despite its simplicity in contrast to models used in practice, it covers the most important aspects so that it can be used to illustrate how this models basically work. The following variables and parameters are needed:

x_{itc} ... is 1 if area i is harvested by crew c in period t, and 0 otherwise
y_{ijt} ... flow from area i to industry j in period t
s_{itc} ... supply of area i in period t if harvested by crew c
d_{jt} ... demand of industry j in period t
f_{itc} ... cost if area i is harvested in period t by crew c
c_{ijt} ... transportation cost from area i to industry j in period t per unit

Using this notation the model can be stated as follows. Here the objective function describes the costs that consist of costs for harvesting in the first summand as well as transportation costs in the second term for the whole planning horizon, thus

$$\min \quad \sum_i \sum_t \sum_c f_{itc} x_{itc} + \sum_i \sum_j \sum_t c_{ijt} y_{ijt}.$$

The following restrictions have to be taken into consideration. The first constraint ensures that every area is harvested at most once, thus

$$\sum_t \sum_c x_{itc} \leq 1 \qquad \forall i.$$

Moreover, every crew c can harvest only one area at time t which is represented by the inequality

$$\sum_i x_{itc} \leq 1 \qquad \forall t, c.$$

The transported amount of wood from area i is of course limited by the supply of this area and can be transported only if it is harvested. This is modeled as follows:

$$\sum_j y_{ijt} \leq s_{itc} x_{itc} \qquad \forall i, t, c.$$

Moreover, only the required amount of wood should be transported to a company j at time t which is expressed by

$$\sum_i y_{ijt} \leq d_{jt} \qquad \forall j, t.$$

The variables x_{itc} are binary and the y_{ijt} are continuous but non-negative and so we write

$$x_{itc} \in \{0, 1\} \qquad \forall i, t, c$$

and

$$y_{ijt} \geq 0 \qquad \forall i, j, t.$$

Based on this problem more constraints can be included. In general, disjunctive constraints arise if the spatial structure of forests is taken into consideration. For instance, road constructions give rise to such considerations as described early in literature in [63]. Also one of the first articles addressing this issue is [133]. Some additional aspects such as different timber qualities, market prices, different types of roads and so on are included in the models in [6]. Additional reasons leading to disjunctive programs in forestry are the reduction of erosion and the improvement of water quality [132]. Other difficulties that often need to be included in these models are explained in [79]. More recent articles describe even more sophisticated approaches taking, for example, stochastic aspects into consideration [129] or steep terrains such as the alps where, for instance, road building requires even more constraints [26].

Besides road constructions there are more reasons to take the spatial structure into account that also lead to disjunctions. For example if so-called forest fragmentation should be avoided because of the requirements of certain species. For instance, it might be important not to divide certain habitats that may consist of several possible harvesting areas and so spatial aspects come into play. Thus, so-called habitat fragmentation should be avoided. For instance, elks are known to feed only in regions within two hundred meters away from the forest [6] and thus appropriate areas should be preserved. To achieve this so-called adjacency constraints are included into the model. This is called biodiversity management and is covered in detail in [71] and the references therein. One of the first articles addressing this issue is [120]. Another article that covers disjunctive aspects such as adjacency constraints in forest management is [131]. Special type of these constraints describing certain forbidden regions can be found in [110]. In the review [134] not only forestry resources but agricultural aspects are covered.

Regarding the solution of these problems, in [84,132] different mixed-integer reformulations to express these disjunctions are compared. According to [84] these spatial constraints are the restrictive components of models in forest planning. Consequently, it is important to deal with these disjunctions efficiently. An interesting approach is proposed in [94] where the MILP model is constructed in such a way that the linear relaxations directly yield an integer solution so that the computationally expensive branch-and-cut methods can be circumvented.

Usually, the idea to model adjacency is as follows. If two areas i and j are close to each other, then either area i or area j is to be harvested but not both of them. That means, if area i is harvested at time t by any crew, then area j must not be harvested by any crew from time $t-p$ to $t+p$. The time span p is assumed to be needed for an area to grow up again. This can be modeled (cf. [71]) by adding the disjunctive constraint

$$\left(\bigwedge_c (x_{itc} = 0)\right) \vee \left(\bigwedge_c \left(\bigwedge_{t'=t-p}^{t+p} (x_{jt'c} = 0)\right)\right).$$

As the problem is mixed-integer anyway, this can be rewritten as

$$\sum_c x_{itc} + \sum_c \sum_{t'=t-p}^{t+p} x_{jt'c} \leq 1.$$

The situation becomes even more difficult if two distinct areas are important for a certain species and a corridor between both of them should be preserved.

We consider, for instance, area 1 and area 32 in Figure 2.1 and we assume that a way consisting of neighbouring areas between both of them must not be harvested. Moreover, we assume that the length of this way is restricted and, more precisely, should consist of at most 15 areas.

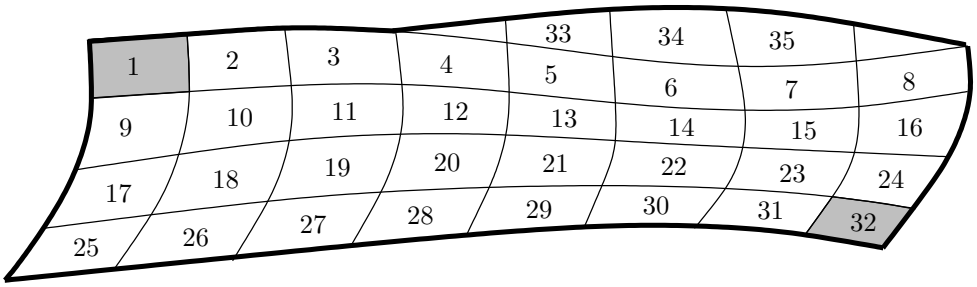

Figure 2.1: Forest divided into different areas

For convenience we introduce the variables w_{ijl} which are 1 if such a way from area i to area j with length at most l areas exists and 0 indicating it does not. So in our model we require $w_{1,32,15} = 1$. By adding more constraints this auxiliary variable has to be connected to the harvesting of areas in order to make this construction work. The existence of such a way is equivalent to the existence of a way of length at most 14 areas from area 1 to area 24 or to area 31 and so we write

$$\Big((w_{1,24,14} = 1) \wedge (a_{24} = 0)\Big) \vee \Big((w_{1,31,14} = 1) \wedge (a_{31} = 0)\Big). \tag{2.1}$$

Area 24 can be reached from area 16 and 23. Area 31 can be reached from area 23 and 30. So we may replace the variables $w_{1,24,14}$ and $w_{1,31,14}$ and rewrite the disjunctive expression (2.1) by

$$\begin{aligned}&\Big[\Big(\big((w_{1,16,13} = 1) \wedge (a_{16} = 0)\big) \vee \big((w_{1,23,13} = 1) \wedge (a_{23} = 0)\big)\Big) \wedge (a_{24} = 0)\Big] \\ \vee&\Big[\Big(\big((w_{1,30,13} = 1) \wedge (a_{30} = 0)\big) \vee \big((w_{1,23,13} = 1) \wedge (a_{23} = 0)\big)\Big) \wedge (a_{31} = 0)\Big].\end{aligned}$$

Continuing recursively leads to a nested logical expression where only variables a_i are involved and all occurrences of w_{ijl} are replaced. Although in theory this disjunctive term can be modeled by using binary variables, in real applications this leads to difficult to solve mixed-integer linear problems that often cannot be handled by standard solvers due to weak relaxations in branch-and-bound methods. Thus it is important to fully understand the disjunctive structure in these models and to handle this appropriately.

2.2 Cutting Patterns for Sawmill Optimization

In a sawmill, first of all, logs are debarked and then cut into lumber. Usually, this is done using either band saws, frame saws or circular saws that all have certain advantages and disadvantages. For every log there are different possibilities to be cut into lumber that yield certain boards and beams in certain dimensions. These possibilities are known as cutting patterns. In a sawmill every log is assigned to a particular cutting pattern so that the resulting boards and beams fit the demand. A good cutting pattern should produce a maximum value of certain types of logs.

There is a large amount of literature devoted to this topic. Assigning a cutting pattern to each log can be modeled as a linear mixed-integer problem as is done in [80]. Taking uncertainty into account leads to robust optimization as applied to the sawmill scheduling problem in [5,127,140].

An approach that considers the entire supply chain from harvesting over transportation to the sawmill and to the market is described in [107]. Usually, cutting the stem into logs is called bucking and is done immediately in the forest. This and the cutting of logs into lumber in a sawmill are considered independently from each other in most cases. However, the integration of both steps can lead to an improvement as proposed in [39]. After the sawmill, the wood is often shortened in length. This subsequent process is called cross cutting. Although usually performed independently from the sawmill and often done in completely different companies, also in this step one can also take advantage of operations research methods as explained in [30,97].

Even though the robust sawmill scheduling problem also gives rise to semi-infinite programs we shall focus on another problem of the sawmill industry here, namely the computation of efficient cutting patterns.

Although in theory there might be more possibilities, commonly logs are cut into parallel slabs, also called flitches, at first which is also known as primary log breakdown. These slabs are then in turn cut into boards and beams which is called secondary log breakdown. Resulting possibilities to cut a log into lumber are depicted in Figure 2.2 on the right. On the left hand side of Figure 2.2 other cuts are sketched that might be possible in theory but are usually not taken into consideration in practice, because this would take too much effort as machines are usually not designed for this purpose. Additional types of cutting patterns that can be taken into consideration in practice but are not considered here are illustrated in Figure 2.3.

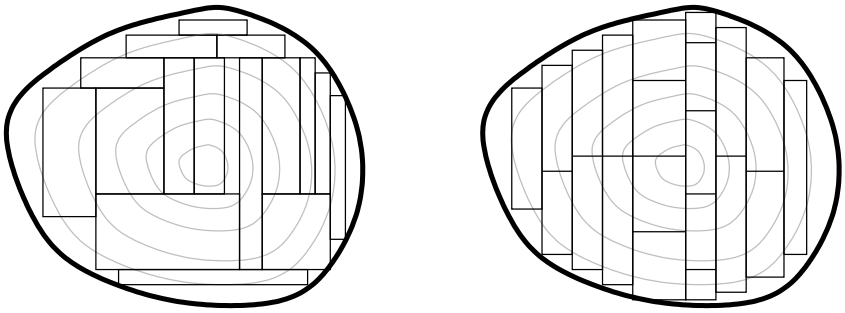

Figure 2.2: Left: Arbitrary cuts that are usually too costly, right: Log with live sawing pattern (based on [122])

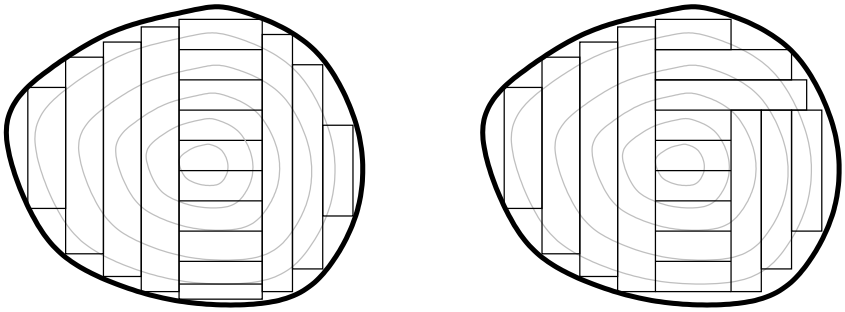

Figure 2.3: Left: Cant sawing, right: Around sawing (based on [122])

Different possibilities to cut a flitch into boards and beams are illustrated in Figure 2.4. This secondary log breakdown is described in detail in [121] whereas in [93,122] the integration of both, primary and secondary log breakdown, are examined. Often, dynamic programming is applied which is already done in [43] where it is also emphasized that this problem can be considered as knapsack problem under suitable assumptions.

Here we assume that the value of lumber in certain quality and certain dimensions does not change over the planning horizon. In real applications this does not necessarily hold as demand for certain pieces in certain quality and dimension may vary so that a cutting pattern might be beneficial although less volume of lumber is obtained. An integrated approach for the computation of cutting patterns together with the sawmill planning problem is proposed in [123].

The computation of optimal cutting patterns can be interpreted as a design centering problem and thus generalized semi-infinite problems are a straightforward possibility to describe this process. We now propose such a model. To achieve this, the data of a log is assumed to be given by a

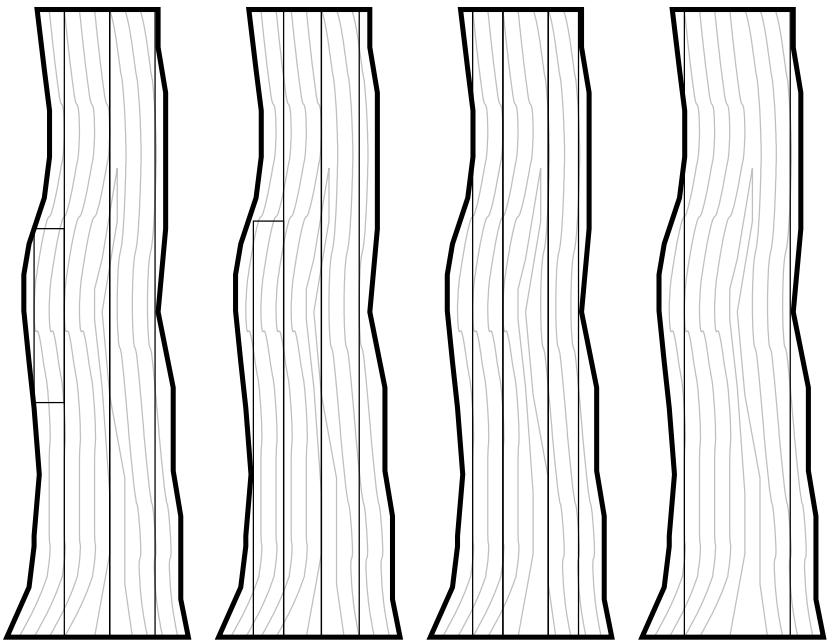

Figure 2.4: Slab that can be sawn using different cutting patterns (based on [122])

function g such that $g(x_1, x_2, x_3) \leq 0$. Although at the first glance, a log can be seen as a cylinder, in reality usually there are much more complex shapes, especially if defects, for instance knotholes that lower quality as well as stability of the wood, are taken into consideration. Thus, writing the log as $G := \{x \in \mathbb{R}^3 | g(x) \leq 0\}$ allows for a great flexibility. Note that detailed information on each log is often available in practice as modern sawmills examine every log by scanners [96].

A certain board or beam that is cut from this log can be modeled by a 3-dimensional box $X := [\underline{x}', \overline{x}'] \times [\underline{x}'', \overline{x}''] \times [\underline{x}''', \overline{x}''']$ with $X \subset G$. The value of this board or beam is denoted by $\vartheta(\overline{x}' - \underline{x}', \overline{x}'' - \underline{x}'', \overline{x}''' - \underline{x}''')$.

A whole flitch can mathematically be written as $G \cap (\mathbb{R}^2 \times [\underline{x}''', \overline{x}'''])$ where $\overline{x}''' - \underline{x}'''$ is the thickness of the considered slab. The maximum value that can be cut from a given flitch can be computed by solving the following generalized semi-infinite program with N and M chosen sufficiently large and where $N \cdot M$ is the maximum number of boards and beams that is assumed to be cut from each slab:

$$GSIP_F(\underline{x}''', \overline{x}''') : \qquad \max \sum_{i=1}^{N} \sum_{j=1}^{M} \vartheta(x'_{i+1} - x'_i, x''_{j+1} - x''_j, \overline{x}''' - \underline{x}''')$$

with the feasible set described by the lower level problem

$$\max_{y \in Y_{ij}(x)} g(y) \leq 0 \quad \text{for} \quad \begin{array}{l} i = 1, \ldots, N, \\ j = 1, \ldots, M \end{array}$$

with

$$Y_{ij}(x) = [x'_i, x'_{i+1}] \times [x''_j, x''_{j+1}] \times [\underline{x}''', \overline{x}'''] \quad \text{for} \quad \begin{array}{l} i = 1, \ldots, N, \\ j = 1, \ldots, M. \end{array}$$

Denoting the optimal value of this problems by $\tilde{v}(\underline{x}''', \overline{x}''')$, an optimal cutting pattern can be computed by solving

$$P: \qquad \max_{x'''_1, \ldots, x'''_L} \sum_{j=1}^{L} \tilde{v}(x'''_j, x'''_{j+1}).$$

Models found in literature are usually more or less simplified, often discretized, versions of this problem. For instance, it is possible to solve $GSIP_F$ for several discrete values of $\underline{x}'''$ and $\overline{x}'''$. Then we can approximate problem P by using these distinct optimal values instead of the continuous version that requires the optimal value functions of $GSIP_F(\underline{x}''', \overline{x}''')$.

2.3 Truss Topology Design

Although trusses can be made of different materials and thus are not necessarily restricted to wood, there is an interesting theory as well as numerical approaches in literature that can be applied in this field. Moreover, trusses made of wood are very common for a very long time, for example in the construction of houses and roofs.

In truss topology design, different trusses that withstand some given external forces are developed. Applying optimization techniques, different goals can be achieved such as structures with minimal weight or costs, for instance. More often the so-called compliance of a truss is minimized so that the resulting structure is as stiff as possible.

There is a large amount of literature devoted to this topic and thus, not all of the work in this field can be mentioned here. One of the first articles is [34]. A detailed review of this area from optimization point of view can be found in [21]. Several mathematical models are compared in [69]. Moreover,

there are many numerical methods used to solve these problems. A tailored interior point algorithm is presented in [58]. Often, SDP solvers are applied to solve a robust version of the truss topology design problem as explained, for instance, in [22]. The global solution of these problems is aimed in [2] by using a tailored branch-and-bound algorithm. The considered models are explained well in [1].

In the following we consider such a truss topology design problem. The description is rather lengthy due to the explanation of some basics of engineering. Note, however, that our emphasis is on the disjunctive as well as generalized semi-infinite extensions at the end of this section.

The model described in the following is based on the above mentioned articles. According to this, a planar truss consists of a finite number of nodes $i = 1, \ldots, N$ with x-coordinates $\tilde{x}_i, i = 1, \ldots, N$ and y-coordinates $\tilde{y}_i, i = 1, \ldots, N$. The extension to non-planar trusses is straightforward. Two nodes i and j can be connected by bars with cross-sectional area $a_{ij} \geq 0$ and length $\tilde{l}_{ij}$. A value of $a_{ij} = 0$ means that no bar connecting the nodes i and j exists. Here we use tilde over parameters in order to distinguish those from variables of the problem.

Before we continue to describe the mathematical model, some physics have to be introduced. Given a single bar with length $\tilde{l}$ and cross-sectional area a that is stressed by a force f in axial direction, this bar is elongated by δ according to the equation

$$a\frac{\tilde{E}}{\tilde{l}}\delta = f \tag{2.2}$$

where $\tilde{E}$ denotes a material constant called Young's modulus. Although this does only hold for relatively small forces and elongations, this is a common assumption in practice.

In order to apply this principle to the whole truss we take the existence of some external forces into consideration. Thus, if the whole structure is under load, the nodes slightly move resulting in a small displacement u_i in x-direction and v_i in y-direction of every node according to this physical law. This is depicted exemplarily in Figure 2.5.

However, in order to apply equation (2.2) it is important to have the elongation of every bar connecting two nodes i and j depending on the nodal displacements *in axial direction*. Thus, for two nodes i and j with corresponding displacements $(u_i, v_i)^\intercal$ and $(u_j, v_j)^\intercal$ of the original coordinates $(\tilde{x}_i, \tilde{y}_i)^\intercal$ and $(\tilde{x}_j, \tilde{y}_j)^\intercal$, we have to compute the elongation of the bar compared to the original length $\tilde{l}_{ij} = \sqrt{(\tilde{x}_i - \tilde{x}_j)^2 + (\tilde{y}_i - \tilde{y}_j)^2}$.

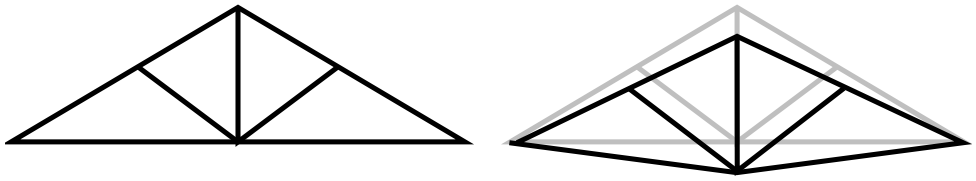

Figure 2.5: Left: Small truss without external forces. Right: Same truss under load.

We start by considering the very simple case where we assume only one displacement $(u_i, 0)^\intercal$ of one node in x-direction. So for a bar between node i and node j with angle $\tilde{\varphi}_{ij}$ between the x-axis and the non-deformed bar as sketched in Figure 2.6 on the left, this can be approximated by $\cos(\tilde{\varphi}_{ij})u_i$.

On the other hand, for a displacement $(0, v_i)^\intercal$ the situation is depicted in Figure 2.6 on the right and the corresponding approximated elongation along the axis can be computed by $\sin(\tilde{\varphi}_{ij})v_i$, again leading to an approximation error. Combining both considerations to the displacement $(u_i, v_i)^\intercal$ the resulting elongation of the bar in axial direction is approximately the sum $\cos(\tilde{\varphi}_{ij})u_i + \sin(\tilde{\varphi}_{ij})v_i$.

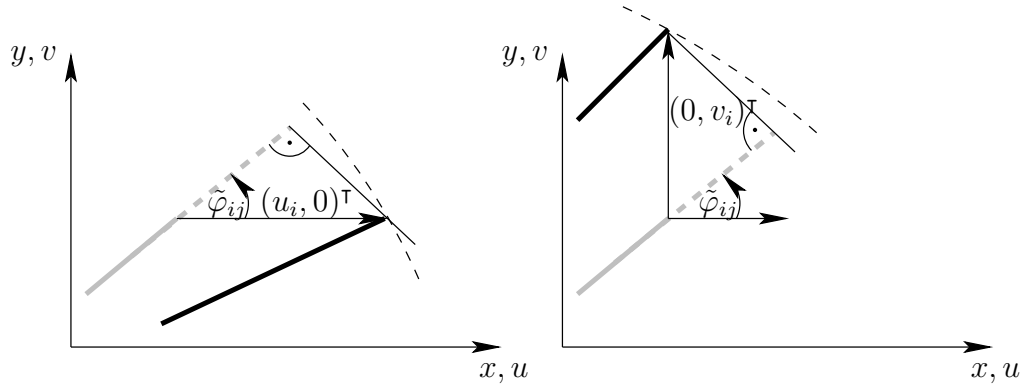

Figure 2.6: Original bars and the corresponding displaced bars under load in a truss

Taking the second node of the bar into consideration and applying the same line of argumentation, letting $\tilde{\gamma}_{ij} := (\cos(\tilde{\varphi}_{ij}), \sin(\tilde{\varphi}_{ij})^\intercal$ yields the elongation

$$\delta_{ij} \approx \tilde{\gamma}_{ij}^\intercal d_i - \tilde{\gamma}_{ij}^\intercal d_j \tag{2.3}$$

in axial direction where d_i is the displacement $(u_i, v_i)^\intercal$ and $d_j = (u_j, v_j)^\intercal$. Similarly to the situation depicted in Figure 2.6 for every bar from node i

to j some $\tilde{\gamma}_{ij}$ can be derived analogously so that the elongation δ_{ij} can be computed as in equation (2.3).

Remark 2.3.1. *Note that the elongations of bars are not computed exactly and, moreover, no approximation error is given. For that reason, trusses that are designed by optimization methods based on these considerations have to be checked carefully.*

According to equation (2.2) this approximated elongation must be obtained by a force with absolute value

$$a_{ij}\frac{\tilde{E}}{\tilde{l}_{ij}}\delta_{ij} \approx a_{ij}\frac{\tilde{E}}{\tilde{l}_{ij}}(\tilde{\gamma}_{ij}^{\intercal}d_i - \tilde{\gamma}_{ij}^{\intercal}d_j)$$

in axial direction. To obtain the corresponding x- and y-components of this force we multiply this value by $\tilde{\gamma}_{ij}$. Given forces in the bar from node i to j in x-direction $f_{i,x}$ and in y-direction $f_{i,y}$, respectively, we then have

$$a_{ij}\frac{\tilde{E}}{\tilde{l}_{ij}}(\tilde{\gamma}_{ij}^{\intercal}d_i - \tilde{\gamma}_{ij}^{\intercal}d_j)\tilde{\gamma}_{ij} = f_{ij}$$

with $f_{ij} = (f_{ij,x}, f_{ij,y})^{\intercal}$.

Here, strictly speaking, the angles of the deformed truss under load should be applied instead of $\tilde{\gamma}_{ij}$. However, it is a standard assumption in the construction of trusses that these angles do not differ too much and thus $\tilde{\varphi}_{ij}$ can be used in the approximation. This procedure as well as the aforementioned approximation of the elongations using the original angles is called *equilibrium on the non-deformed system*. This has to be handled with care as we stress in the following remark. To the best of our knowledge, this aspect is not examined in literature so far in the context of optimization.

Remark 2.3.2. *Although using the equilibrium on the non-deformed system is common in the computation of trusses we point out that it is not clear, if this also works in optimization. First of all, it is inconsistent in some sense to maximize the stiffness and, at the same time, assume the angles of the bars to stay nearly constant if the truss is under load. Moreover, this gap between reality and the mathematical model might be exploited in the solution process so that in a computed optimal point the approximation error might not be negligible.*

We now extend these considerations for one bar to the more complex case of a whole truss. To achieve this, in every node i we assume the existence of so-called external forces $\tilde{f}_i = (\tilde{f}_i^x, \tilde{f}_i^y)^{\intercal}$ with x-component $\tilde{f}_i^x$ and

y-component $\tilde{f}_i^y$ that the truss has to withstand. In practice, usually for many nodes we have $\tilde{f}_i = 0$.

In a truss under load the forces within all nodes have to be balanced so that in a normal node i we have

$$\sum_{j=1}^{N} f_{ij} + \tilde{f}_i = 0.$$

Moreover, some nodes (usually at the basis of the truss) can be assumed to absorb forces, either forces in only one direction or even in x- and y-direction. This can be modeled by introducing an additional force $\bar{f}_i^x$ in x-direction and $\bar{f}_i^y$ in y-direction that may have arbitrary values. For ease of presentation we put $\bar{f}_i = (\bar{f}_i^x, \bar{f}_i^y)^\intercal$.

If, for instance, the first node $i = 1$ may absorb vertical forces but not horizontal forces, we require $\bar{f}_1^x = 0$ and $\bar{f}_1^y \in \mathbb{R}$. Assuming all nodes that may absorb only forces in horizontal direction are contained in the set $A_x \subset \{1, \ldots, N\}$, all nodes that may absorb only vertical forces are in the set $A_y \subset \{1, \ldots, N\}$ and all nodes that may absorb forces in x- as well as y-direction are in the set $A_{xy} \subset \{1, \ldots, N\}$. Thus, we require

$$\begin{array}{lll} \bar{f}_i^x \in \mathbb{R}, \bar{f}_i^y = 0 & \text{for} & i \in A_x \\ \bar{f}_i^x = 0, \bar{f}_i^y \in \mathbb{R} & \text{for} & i \in A_y \\ \bar{f}_i^x \in \mathbb{R}, \bar{f}_i^y \in \mathbb{R} & \text{for} & i \in A_{xy}. \end{array}$$

So in order to balance all these forces, for every node $i = 1, \ldots, N$ we add the constraint

$$\sum_{j=1}^{N} f_{ij} + \tilde{f}_i + \bar{f}_i = 0. \tag{2.4}$$

Note that in practice the size of the problem may be reduced by removing many of these equations together with the artificial variable $\bar{f}_i$.

Usually, in applications only a certain quantity of wood $\tilde{v}$ is available so that the constraint

$$\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \tilde{l}_{ij} a_{ij} \leq \tilde{v}$$

is added to the model.

The goal of truss topology design is often to design structures that are as stiff as possible. To achieve this, it is common that nodes with large external forces are expected to move only slightly whereas other nodes are allowed

to move a little bit more when the truss is under load. Hence, often the so-called compliance as defined by

$$\sum_{i=1}^{N} \tilde{f}_i^{\intercal} d_i$$

is considered to be a suitable objective function (see [1]). Eventually, we arrive at the following optimization problem:

$$\begin{aligned} P: \quad \min_{a,d,f,\bar{f}} \quad & \sum_{i=1}^{N} \tilde{f}_i^{\intercal} d_i \\ \text{s.t.} \quad & a_{ij} \frac{\tilde{E}}{\tilde{l}_{ij}} (\tilde{\gamma}_{ij}^{\intercal} d_i - \tilde{\gamma}_{ij}^{\intercal} d_j) \tilde{\gamma}_{ij} = f_{ij} \qquad i,j = 1,\dots,N \\ & \sum_{j=1}^{N} f_{ij} + \tilde{f}_i + \bar{f}_i = 0 \qquad i = 1,\dots,N \\ & \bar{f}_i^x \in \mathbb{R}, \bar{f}_i^y = 0 \quad \text{for} \quad i \in A_x \\ & \bar{f}_i^x = 0, \bar{f}_i^y \in \mathbb{R} \quad \text{for} \quad i \in A_y \\ & \bar{f}_i^x \in \mathbb{R}, \bar{f}_i^y \in \mathbb{R} \quad \text{for} \quad i \in A_{xy} \\ & \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \tilde{l}_{ij} a_{ij} \leq \tilde{v} \\ & a_{ij} \geq 0, \qquad i,j = 1,\dots,N \text{ with } i \neq j. \end{aligned}$$

This is the classical truss topology design problem which can be used to derive more complex and more realistic models that are even more appropriate for practical applications. For instance, without additional constraints this standard model from literature may lead to trusses that have many bars of very thin cross-sectional areas meaning the truss is dense which is not desirable. A possibility to avoid this is to require that bars have a certain minimum cross-sectional area $\tilde{a}$. To achieve this, the disjunctive constraints

$$(a_{ij} = 0) \quad \vee \quad (a_{ij} \geq \tilde{a})$$

can be added. In [2] a similar truss topology problem is considered where the cross-sectional areas are chosen from a finite set of possibilities.

An additional reason to introduce disjunctive constraints into the model is to avoid that bars intersect as this might lead to torsion and should be avoided. That is, for every two possible bars between nodes i, j and p, q, respectively, that might intersect we require

$$(a_{ij} = 0) \quad \vee \quad (a_{pq} = 0).$$

Thus, the need to tackle disjunctive problems arises.

Another difficulty that is encountered in practice is the problem of different load scenarios. So far, only certain given external forces are assumed. However, it is more realistic, that different external forces from a whole set $\mathcal{F}$ are possible. As long as $\mathcal{F}$ is finite, problem P can be extended to this case in a straightforward manner. In contrast, for an infinite set $\mathcal{F}$ this leads to so-called robust truss topology design. Such difficulties are also examined in literature, for instance in [22], and often solved by SDP techniques.

In the following we derive a model to compute robust trusses by means of generalized semi-infinite programming techniques. Observe, that given the parameters $\tilde{\gamma}_{ij}$, $\tilde{l}_{ij}$, $\tilde{E}$ as well as the angles $\tilde{\varphi}_{ij}$, the truss itself can be described simply by the cross-sectional areas a_{ij} of the bars. As long as we have $a_{ij} \geq 0$ for all $i, j = 1, \ldots, N$ with $i \neq j$ and, moreover, the maximum volume of used material is not exceeded, this can be seen as a valid truss. For convenience we put

$$\mathcal{A} := \Big\{ a \mid \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \tilde{l}_{ij} a_{ij} \leq \tilde{v} \\ a_{ij} \geq 0, \qquad i, j = 1, \ldots, N \text{ with } i \neq j \Big\}.$$

Given cross-sectional areas $a \in \mathcal{A}$ and a certain external load $\tilde{f} \in \mathcal{F}$, then the minimal compliance can be computed by solving

$$\min_{d,f,\bar{f}} \sum_{i=1}^{N} \tilde{f}_i^{\intercal} d_i \qquad \text{s.t.} \qquad (d, f, \bar{f})^{\intercal} \in Y(a, \tilde{f})$$

with

$$Y(a, \tilde{f}) := \Big\{ (d, f, \bar{f})^{\intercal} \mid a_{ij} \frac{\tilde{E}}{\tilde{l}_{ij}} (\tilde{\gamma}_{ij}^{\intercal} d_i - \tilde{\gamma}_{ij}^{\intercal} d_j) \tilde{\gamma}_{ij} = f_{ij} \qquad i, j = 1, \ldots, N \\ \sum_{j=1}^{N} f_{ij} + \tilde{f}_i + \bar{f}_i = 0 \qquad i = 1, \ldots, N \\ \bar{f}_i^x \in \mathbb{R}, \bar{f}_i^y = 0 \quad \text{for} \quad i \in A_x \\ \bar{f}_i^x = 0, \bar{f}_i^y \in \mathbb{R} \quad \text{for} \quad i \in A_y \\ \bar{f}_i^x \in \mathbb{R}, \bar{f}_i^y \in \mathbb{R} \quad \text{for} \quad i \in A_{xy} \Big\}.$$

Now we introduce a whole set of external loads $\mathcal{F}$ where $\tilde{f}$ can be chosen from. Given fixed cross-sectional areas $a \in \mathcal{A}$, in the worst case the compliance that

might occur can be computed by solving the problem

$$\max_{\tilde{f}\in\mathcal{F}} \min_{(d,f,\bar{f})^\intercal\in Y(a,\tilde{f})} \sum_{i=1}^{N} \tilde{f}_i^\intercal d_i.$$

Now, in order to compute the truss described by the variables $a \in \mathcal{A}$ that is prepared for every external load $\tilde{f} \in \mathcal{F}$ we may solve

$$\begin{aligned} \min_{a\in\mathcal{A},z} \quad & z \\ \text{s.t.} \quad & \max_{\tilde{f}\in\mathcal{F}} \min_{(d,f,\bar{f})^\intercal\in Y(a,\tilde{f})} \sum_{i=1}^{N} \tilde{f}_i^\intercal d_i \leq z. \end{aligned}$$

This problem can be seen as a semi-infinite problem with a difficult to solve lower level problem. Thus, in order to avoid this, observe that for fixed $\tilde{f} \in \mathcal{F}$ and $a \in \mathcal{A}$ the problem

$$\min_{(d,f,\bar{f})^\intercal\in Y(a,\tilde{f})} \sum_{i=1}^{N} \tilde{f}_i^\intercal d_i$$

is linear in the remaining variables d, f and $\bar{f}$. Hence, we may replace this minimization problem by its linear dual problem. Denoting the feasible of this problem by $\widetilde{Y}(a,\tilde{f})$, the dual variables by y and the objective function by $b^\intercal y$ we may rewrite the robust truss topology design problem by

$$\begin{aligned} GSIP: \qquad \min_{a\in\mathcal{A},z} \quad & z \\ \text{s.t.} \quad & \max_{\tilde{f}\in\mathcal{F},y\in\widetilde{Y}(a,\tilde{f})} b^\intercal y \leq z \end{aligned}$$

and thus, the generalized semi-infinite structure becomes apparent. Note that linearity is exploited in order to achieve this reformulation. Moreover, a dual problem is used to replace a minimization problem by a maximization problem which is also a standard approach in generalized semi-infinite programming as will be described in more detail in the subsequent chapters and used several times throughout this thesis.

Regarding the robust truss topology design problem we point out that for certain requirements regarding the set $\mathcal{F}$ instead of problem $GSIP$ much simpler problems may be solved, for instance SDPs (see [22]). However, SDPs can be seen as a special kind of semi-infinite programs as well, although there are much more efficient solution methods for SDPs.

In this chapter, we described three different applications and the need to tackle disjunctive and semi-infinite problems is illustrated. The field of timber industry is chosen only for purpose of illustration as this is a very classic

part of industry where the need of optimization is encountered meanwhile. Analogously, applications of DP and GSIP arise in many other fields.

In the remaining chapters of this thesis, different possibilities to solve both kinds of problems, DPs as well as GSIPs, are developed. Although it would be interesting to see how these methods perform on such real world instances, we prefer a different line of research here. Even though a direct application should be possible, at least in theory, this requires even more. So, for example, a thorough preparation of real data is required. Moreover, a refined interface for our implementations to these data or maybe even to modeling languages is necessary. In contrast, we prefer to show the performance of our methods along test problems from literature which also enables us to compare the performance of our algorithms to other approaches. For that reason, the solution of applications of timber industry is beyond the scope of this thesis and left for future research.

Chapter 3

GSIP Reformulations of Disjunctive Problems

In this chapter, which is based on [65], we describe how techniques from generalized semi-infinite optimization may be employed to solve disjunctive optimization problems.

Generalized semi-infinite programs turn out to be much harder to solve than standard semi-infinite programs, although they seem to be only a slight extension of the standard semi-infinite case at first glance. This is mainly due to two geometric properties, which are stable under perturbations of the defining functions in generalized semi-infinite optimization problems, but impossible or unstable, respectively, in finite or standard semi-infinite optimization as already explained in the introduction and illustrated in Example 1.1.1. Firstly, the feasible set of a GSIP need not be closed although all defining functions are continuous. Secondly, the feasible set of a GSIP may possess so-called re-entrant corner points, that is, the feasible set may locally be expressed as the union of finitely many sets with smooth boundaries. The main idea of the present chapter is to exploit this intrinsic disjunctive structure of GSIPs for the solution of disjunctive optimization problems (DPs).

In fact, we shall show how a GSIP, corresponding to any given DP, can be constructed and, by using lower level techniques for the generated semi-infinite programs, we derive purely continuous and conjunctive nonlinear problems without any logical expressions, which can be locally solved by standard nonlinear solvers. Note that we do not impose any convexity assumptions on the defining functions of the disjunctive problem.

This chapter is based mainly on the article [65] and is structured as follows. In Section 3.1, we state more precisely the definitions of the different

kinds of appearing optimization problems. Section 3.2 describes how disjunctive problems can be reformulated as generalized semi-infinite optimization problems. We start by considering problems in a so-called disjunctive or conjunctive normal form. Then, the ideas are extended to handle disjunctive problems with arbitrary logical expressions. The resulting GSIPs are treated by two solution techniques: the lower level duality reformulation in Section 3.3.1, and replacing the lower level problem by its Karush-Kuhn-Tucker optimality conditions combined with smoothing of the resulting mathematical program with complementarity constraints in Section 3.3.2. In any case, we obtain standard nonlinear optimization problems without any logical expressions, which can be solved at least locally by common NLP-solvers. Section 3.3.3 provides a stability analysis for the smoothing approach. In Section 3.4, we perform computational tests on some small examples.

3.1 Definition of the Problem

In this chapter we consider generalized semi-infinite programs with multiple semi-infinite constraints, that is problems of the form

$$GSIP: \qquad \min_{x\in\mathbb{R}^n} f(x) \quad \text{s.t.} \quad x \in M_{GSIP}$$

with the feasible set

$$M_{GSIP} \;:=\; \{x \in \mathbb{R}^n :\; g_i(x, y^i) \leq 0 \text{ for all } y^i \in Y_i(x),\; i \in I\}.$$

Hence, for each index i from the finite index set $I := \{1, \ldots, p\}$ (with $p \in \mathbb{N}$) a set-valued mapping $Y_i : \mathbb{R}^n \rightrightarrows \mathbb{R}^{m_i}$ (with $m_i \in \mathbb{N}$, $i \in I$) describes the index set of inequality constraints. With the so-called lower level optimal value functions

$$\varphi_i(x) \;:=\; \sup_{y^i\in Y_i(x)} g_i(x, y^i),\; i \in I,$$

an obvious reformulation of the feasible set is

$$M_{GSIP} \;=\; \{x \in \mathbb{R}^n :\; \max_{i\in I} \varphi_i(x) \leq 0\}.$$

The functions f and g_i, $i \in I$, are assumed to be at least continuous on their respective domains. In this thesis, we shall assume that also the index set mappings are given in functional form as

$$Y_i(x) \;=\; \{y^i \in \mathbb{R}^{m_i} :\; v^i(x, y^i) \leq 0\}$$

with at least continuous functions $v^i : \mathbb{R}^n \times \mathbb{R}^{m_i} \to \mathbb{R}^{s_i}$ for $s_i \in \mathbb{N}$, $i \in I$.

We recall our definition of a disjunctive optimization problem by which we mean a problem of the type

$$DP: \qquad \min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad x \in M_{DP}\,,$$

whose feasible set M_{DP} is described by finitely many constraints of the form $G_\ell(x) \leq 0$, $1 \leq \ell \leq m$ (with $m \in \mathbb{N}$). In this chapter we assume that these functions are connected by finitely many conjunctions and disjunctions, and that the functions G_ℓ, $1 \leq \ell \leq m$, are at least continuous. More precisely, we put

$$M_{DP} := \{x \in \mathbb{R}^n : \ \Omega(\, G_1(x) \leq 0, \ldots, G_m(x) \leq 0\,) = \text{true}\}\,,$$

with a logical expression $\Omega : \{\text{true}, \text{false}\}^m \to \{\text{true}, \text{false}\}$ consisting of only conjunctions and disjunctions. The structure of the logical expression Ω may be coded in a natural way by means of its expression tree T_Ω. Due to the associativity of both conjunction and disjunction, we may assume that each node of the expression tree T_Ω either corresponds to

- a conjunction $\bigwedge_{j \in J} A_j$, where the A_j, $j \in J$, are either disjunctions or simple terms of the form $G_\ell(x) \leq 0$ for some $\ell \in \{1, \ldots, m\}$, or
- a disjunction $\bigvee_{j \in J} A_j$, where the A_j, $j \in J$, are either conjunctions or simple terms of the form $G_\ell(x) \leq 0$ for some $\ell \in \{1, \ldots, m\}$.

For convenience, conjunctions and disjunctions of length $|J|$ shall be called $|J|$-conjunctions and $|J|$-disjunctions, respectively. For formal reasons, which will become apparent below, we allow 1-conjunctions and 1-disjunctions in Ω, that is, we put $\bigwedge_{j \in \{1\}} A_j = A_1$ and $\bigvee_{j \in \{1\}} A_j = A_1$. Depending on whether the root node of T_Ω is a conjunction or a disjunction, we shall call T_Ω a conjunction tree or a disjunction tree, respectively. Note that the levels of T_Ω alternatingly correspond to conjunctions and disjunctions, and that the simple terms of the form $G_\ell(x) \leq 0$, $\ell \in \{1, \ldots, m\}$, correspond to the leafs of T_Ω. By the height h_Ω of T_Ω, we shall mean the number of edges on the longest downward path between the root and a leaf.

Example 3.1.1. *Consider the problem*

$$DP: \qquad \min_{x \in \mathbb{R}^2} f(x) \quad \textit{s.t.} \quad x \in M_{DP}$$

with

$$M_{DP} = \Big\{ x \in \mathbb{R}^2 : \ \Big(\big(G_1(x) \leq 0 \wedge G_2(x) \leq 0 \big) \vee G_3(x) \leq 0 \vee G_4(x) \leq 0 \Big) \\ \wedge \Big(G_5(x) \leq 0 \vee G_6(x) \leq 0 \big) \Big) \Big\},$$

and

$$\begin{aligned} f(x) &:= -x_1, \\ G_1(x) &:= x_1^2 + (x_2 - \frac{1}{2})^2 - 1, \\ G_2(x) &:= x_1^2 + (x_2 + \frac{1}{2})^2 - 1, \\ G_3(x) &:= (x_1 - 3)^2 + (x_2 - \frac{1}{2})^2 - 1, \\ G_4(x) &:= (x_1 - 3)^2 + (x_2 + \frac{1}{2})^2 - 1, \\ G_5(x) &:= x_1, \\ G_6(x) &:= -x_1 + 3. \end{aligned}$$

The set M_{DP} is illustrated in Figure 3.1. The expression tree T_Ω of Ω, a conjunction tree of height $h_\Omega = 3$, is depicted in Figure 3.2, where k-C stands for a k-conjunction, and k-D for a k-disjunction.

Since our proposal for a GSIP reformulation of DP in Section 3.2 will hinge on the structure of the expression tree for Ω, 'nicely structured' expression trees will result in a clear structure of the GSIP. For this clarity reason, we will assume that the expression tree T_Ω is full in the sense that all its leaf nodes possess the same depth, that is, the number of edges between the root and each leaf node equals the tree height h_Ω. We also assume that T_Ω is leaf disjunctive, by which we mean that the level of leafs corresponds to a disjunction level in T_Ω.

These two assumptions may always be achieved by artificially introducing 1-conjunctions and 1-disjunctions into Ω. We emphasize that, from a computational point of view, this is not desirable, but that our modeling approach is most transparent for full and leaf disjunctive expression trees T_Ω. For computational purposes, it will be clear how to modify the GSIP reformulation of DP when the artificial nodes are removed.

Figure 3.3 shows the modification of the expression tree from Figure 3.2 to a full and leaf disjunctive tree.

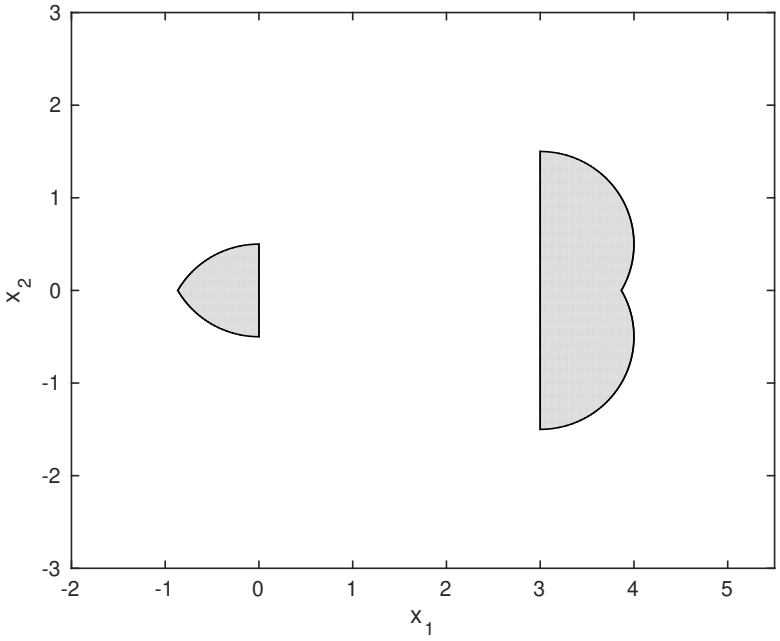

Figure 3.1: Feasible set M_{DP} in Example 3.1.1

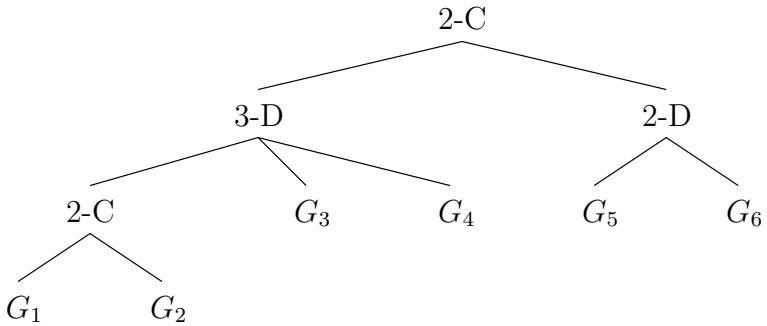

Figure 3.2: Expression tree in Example 3.1.1

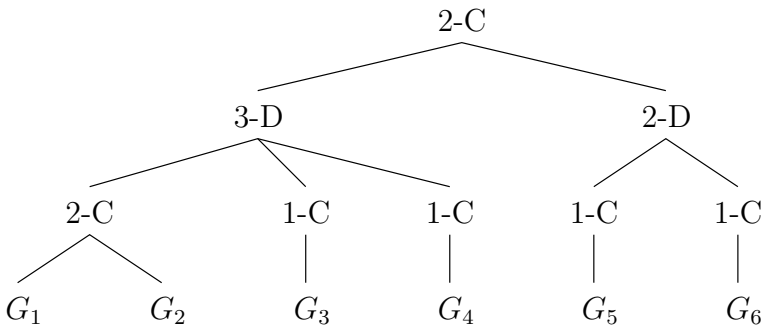

Figure 3.3: Full and leaf disjunctive expression tree in Example 3.1.1

3.2 GSIP Reformulations of DPs

For disjunctive optimization problems in conjunctive normal form, a GSIP reformulation is well-known from [112]. In fact, for

$$M_{DP} \;=\; \bigcap_{i\in C}\bigcup_{j\in D_i}\{x\in\mathbb{R}^n:\; G_{ij}(x)\le 0\}$$

with finite index sets C and D_i, $i\in C$, we put $p=|C|$, $m_i=1$, $g_i(x,y^i)=y^i$, and $v^i(x,y^i)=(y^i-G_{i1}(x),\ldots,y^i-G_{i,|D_i|}(x))^\intercal$, $i\in C$. Then, for each $i\in C$ and $x\in\mathbb{R}^n$ the index set $Y_i(x)$ is one-dimensional and can explicitly be written as

$$Y_i(x) \;=\;]-\infty,\min_{j\in D_i} G_{ij}(x)\,].$$

Hence we obtain $\varphi_i(x)=\min_{j\in D_i} G_{ij}(x)$,

$$M_{GSIP} \;=\; \{x\in\mathbb{R}^n:\; 0\ge\max_{i\in C}\varphi_i(x)=\max_{i\in C}\min_{j\in D_i}G_{ij}(x)\},$$

and thus the following result.

Proposition 3.2.1. *For any problem DP in conjunctive normal form, let GSIP be constructed as described above. Then, we have $M_{GSIP}=M_{DP}$.*

Next, we consider DPs which are not expressed in conjunctive normal form. We describe the main idea to cover this case first for problems in disjunctive normal form, that is, for

$$M_{DP} \;=\; \bigcup_{i\in D}\bigcap_{j\in C_i}\{x\in\mathbb{R}^n:\; G_{ij}(x)\le 0\}$$

with finite index sets D and C_i, $i \in D$. In fact, for each conjunction we introduce an additional variable $z_i \in \mathbb{R}$, $i \in D$, put $p = 1$, $m = 1$, $g(x, z, y) = y$, $v(x, z, y) = (y - z_1, \ldots, y - z_{|D|})^\mathsf{T}$, and we introduce additional constraints $G_{ij}(x) \leq z_i$, $j \in C_i$, $i \in D$. The corresponding GSIP then possesses a lifted feasible set and reads

$$\min_{x,z} f(x) \quad \text{s.t.} \quad (x, z) \in \widetilde{M}_{GSIP}$$

with

$$\begin{aligned} \widetilde{M}_{GSIP} \;=\; \{(x, z) \in \mathbb{R}^n \times \mathbb{R}^{|D|} : y \leq 0 \;\text{ for all }\; y \in Y(z) \\ G_{ij}(x) \leq z_i,\; j \in C_i,\; i \in D\}, \end{aligned}$$

and with the (x-independent) index set

$$Y(z) \;=\; \{y \in \mathbb{R} :\; y \leq z_i,\; i \in D\}.$$

Note that not even the function $\varphi(x, z) = \sup_{y \in Y(z)} y$ depends on x, so that we will denote it as $\varphi(z)$.

In the following, for any $N \in \mathbb{N}$ let $\mathrm{pr}_x : \mathbb{R}^n \times \mathbb{R}^N \to \mathbb{R}^n$ denote the orthogonal projection to the 'x-space' $\mathbb{R}^n$.

Proposition 3.2.2. *For any problem DP in disjunctive normal form, let GSIP be constructed as described above. Then, we have* $M_{DP} = \mathrm{pr}_x \widetilde{M}_{GSIP}$.

Proof. For $(x, z) \in \mathbb{R}^n \times \mathbb{R}^{|D|}$, the index set $Y(z)$ is one-dimensional and can explicitly be written as

$$Y(z) \;=]-\infty, \min_{i \in D} z_i\,].$$

This implies $\varphi(z) = \min_{i \in D} z_i$ and

$$\widetilde{M}_{GSIP} \;=\; \left\{(x, z) \in \mathbb{R}^n \times \mathbb{R}^{|D|} :\; \min_{i \in D} z_i \leq 0,\; \max_{j \in C_i} G_{ij}(x) \leq z_i\,,\; i \in D\right\}.$$

Hence, for each $x \in \mathrm{pr}_x \widetilde{M}_{GSIP}$ there exist some $z \in \mathbb{R}^{|D|}$ and some $i \in D$ with $z_i \leq 0$ and $\max_{j \in C_i} G_{ij}(x) \leq z_i$. This implies $\min_{i \in D} \max_{j \in C_i} G_{ij}(x) \leq 0$, and thus $\mathrm{pr}_x \widetilde{M}_{GSIP} \subseteq M_{DP}$.

On the other hand, for each $x \in M_{DP}$ we may set $z_i = \max_{j \in C_i} G_{ij}(x)$, $i \in D$, and immediately obtain $(x, z) \in \widetilde{M}_{GSIP}$ and thus $M_{DP} \subseteq \mathrm{pr}_x \widetilde{M}_{GSIP}$. $\square$

For DPs which are neither given in conjunctive nor in disjunctive normal form, we may combine the above ideas to derive a GSIP reformulation. For the ease of exposition, we will assume the expression tree T_Ω to be full and leaf disjunctive.

In the following, we shall index the nodes V of T_Ω recursively by multi-indices. The root node will be denoted by V_1. Since it either is a $|J|$-conjunction or a $|J|$-disjunction, we may index its child nodes with multi-indices of length 2 by $V_{11}, \ldots, V_{1,|J|}$. Continuing in this way recursively for the child nodes, we index all nodes down to the leaf level $h_\Omega + 1$, where leaf nodes are indexed with multi-indices of length $h_\Omega + 1$. The set of all indices of child nodes of a node V is denoted by $\mathcal{C}(V)$. Furthermore, all indices of nodes in level ℓ of the expression tree form the set $\mathcal{I}(\ell)$, that is, we have $\mathcal{I}(1) = \{1\}$, $\mathcal{I}(2) = \{11, \ldots, 1|J|\}$, and so on.

For a full and leaf disjunctive expression tree of height h_Ω, the second to last level h_Ω consists of conjunctions C_j, $j \in \mathcal{I}(h_\Omega)$. We then introduce additional variables z_j, $j \in \mathcal{I}(h_\Omega)$, as well as the additional constraints

$$G_i(x) \leq z_j, \quad i \in \mathcal{C}(C_j),\ j \in \mathcal{I}(h_\Omega).$$

Furthermore we replace each node C_j, $j \in \mathcal{I}(h_\Omega)$, in Ω by the term $z_j \leq 0$, so that the height of the expression tree decreases by one. Then, the orthogonal projection of the feasible set onto the 'x-space' of this lifted disjunctive program is the feasible set of DP again. This is due to the following lemma which is shown with the ideas from the proof of Proposition 3.2.2.

Lemma 3.2.3. *The conjunction $\bigwedge_{i\in\mathcal{I}}(G_i(x) \leq 0)$ is true if and only if there exists some $z \in \mathbb{R}$ such that $z \leq 0$, and $G_i(x) \leq z$ hold for all $i \in \mathcal{I}$.*

We illustrate this step for the disjunctive program from Example 3.1.1 (Fig. 3.3).

Example 3.2.4. *We have $h_\Omega = 3$, and thus we consider the set $\mathcal{I}(3) = \{111, 112, 113, 121, 122\}$. We start by introducing one new variable z_{111} and rewrite the feasible set of DP as M_2, where M_2 is defined by*

$$\begin{aligned} M_2 := \Big\{(x, z_{111}) \in \mathbb{R}^2 \times \mathbb{R} : \Big(z_{111} \leq 0 \vee G_3(x) \leq 0 \vee G_4(x) \leq 0\Big) \\ \wedge \Big(G_5(x) \leq 0 \vee G_6(x) \leq 0\Big), \\ G_1(x) \leq z_{111}, G_2(x) \leq z_{111}\Big\}. \end{aligned}$$

With the same line of arguments, we introduce further variables z_{112}, z_{113}, z_{121}, z_{122} *for the remaining* 1*-conjunctions. We can then further rewrite* M_2 *as*

$$\begin{aligned} M_3 := \Big\{(x,z) \in \mathbb{R}^2 \times \mathbb{R}^5 : \Big(z_{111} \leq 0 \vee z_{112} \leq 0 \vee z_{113} \leq 0\Big) \\ \wedge \Big(z_{121} \leq 0 \vee z_{122} \leq 0\Big), \\ G_1(x) \ \leq \ z_{111}\,, \quad G_2(x) \ \leq \ z_{111}, \\ G_3(x) \ \leq \ z_{112}\,, \quad G_4(x) \ \leq \ z_{113}, \\ G_5(x) \ \leq \ z_{121}\,, \quad G_6(x) \ \leq \ z_{122}\Big\} \end{aligned}$$

with $z = (z_{111}, z_{112}, z_{113}, z_{121}, z_{122})^\mathsf{T}$.

Let us now continue with level $h_\Omega - 1$ of the remaining expression tree. In the case $h_\Omega - 1 = 0$ we already arrived at the root node, and thus no further reformulation is necessary. Otherwise the level $h_\Omega - 1$ consists of disjunctions D_j, $j \in \mathcal{I}(h_\Omega - 1)$, which we want to remove from the remaining expression tree by using the following lemma, whose proof follows the lines of the proof of Proposition 3.2.1. The height of the expression tree will then again decrease by one.

Lemma 3.2.5. *The disjunction* $\bigvee_{i\in\mathcal{I}}(z_i \leq 0)$ *is true, iff* $\max_{y\in Y(z)} y \leq 0$ *holds, where* $Y(z)$ *is defined by*

$$Y(z) := \{y \in \mathbb{R} : \ y \leq z_i, \ i \in \mathcal{I}\}.$$

Hence, in order to drop the disjunctions D_j, $j \in \mathcal{I}(h_\Omega - 1)$, we introduce new variables y_j and sets

$$Y_j(z) = \{y_j \in \mathbb{R} : \ y_j \leq z_i, \ i \in \mathcal{C}(C_j)\}, \quad j \in \mathcal{I}(h_\Omega - 1).$$

Furthermore we replace each term D_j, $j \in \mathcal{I}(h_\Omega - 1)$, in the remaining expression tree by the term $\max_{y_j \in Y_j(z)} y_j \leq 0$. We continue to demonstrate our reformulation process along Example 3.1.1.

Example 3.2.6. *In the way described above we replace the term*

$$z_{121} \leq 0 \vee z_{122} \leq 0 \tag{3.1}$$

in the description of the set M_3 *(Example 3.2.4) by inserting a new variable* y_{12} *and the semi-infinite constraint* $y_{12} \leq 0$ *for all* $y_{12} \in Y_{12}(z)$ *with*

$$Y_{12}(z) := \{y_{12} \in \mathbb{R} : \ y_{12} \leq z_{121} \text{ and } y_{12} \leq z_{122}\}.$$

According to Lemma 3.2.5, this semi-infinite constraint is equivalent to the inequalities (3.1) *and thus we can modify the description of the feasible set to*

$$\begin{aligned} M_4 := \Big\{(x,z) \in \mathbb{R}^2 \times \mathbb{R}^5 : \Big(z_{111} \leq 0 \vee z_{112} \leq 0 \vee z_{113} \leq 0\Big) \\ \wedge \Big(\max_{y_{12} \in Y_{12}(z)} y_{12} \leq 0\Big), \\ G_1(x) \ \leq \ z_{111}\,, \quad G_2(x) \ \leq \ z_{111}, \\ G_3(x) \ \leq \ z_{112}\,, \quad G_4(x) \ \leq \ z_{113}, \\ G_5(x) \ \leq \ z_{121}\,, \quad G_6(x) \ \leq \ z_{122}\Big\}. \end{aligned}$$

Due to the explanation above we have $M_3 = M_4$. Analogously, we replace the term $z_{111} \leq 0 \vee z_{112} \leq 0 \vee z_{113} \leq 0$ by a semi-infinite constraint and obtain

$$\begin{aligned} M_5 := \Big\{(x,z) \in \mathbb{R}^2 \times \mathbb{R}^5 : \Big(\max_{y_{11} \in Y_{11}(z)} y_{11} \leq 0\Big) \wedge \Big(\max_{y_{12} \in Y_{12}(z)} y_{12} \leq 0\Big), \\ G_1(x) \ \leq \ z_{111}\,, \quad G_2(x) \ \leq \ z_{111}, \\ G_3(x) \ \leq \ z_{112}\,, \quad G_4(x) \ \leq \ z_{113}, \\ G_5(x) \ \leq \ z_{121}\,, \quad G_6(x) \ \leq \ z_{122}\Big\} \end{aligned}$$

with

$$Y_{11}(z) = \{y_{11} \in \mathbb{R} : \ y_{11} \leq z_{11j}\,, \ 1 \leq j \leq 3\}.$$

□

Next, consider level $h_\Omega - 2$ of the remaining expression tree. Again, in the case $h_\Omega - 2 = 0$ we are done. Otherwise we add new variables z_j, $j \in \mathcal{I}(h_\Omega - 2)$, which represent the conjunctions C_j on that level. The idea is the same as the one of Lemma 3.2.3, except that we replace functions of type $\max_{y \in Y(z)} y$ instead of functions of type $G(x)$. So, for every conjunction C_j on level $h_\Omega - 2$ we introduce some z_j with additional constraints

$$\max_{y_i \in Y_i(z)} y_i \leq z_j, \quad i \in \mathcal{C}(C_j), \ j \in \mathcal{I}(h_\Omega - 2).$$

Similarly to the conjunctions above, we replace each C_j in level $h_\Omega - 2$ of the remaining expression tree by the term $z_j \leq 0$. Again, we explain this step along our running example.

Example 3.2.7. *We replace the term*

$$\Big(\max_{y_{11} \in Y_{11}(z)} y_{11} \leq 0\Big) \wedge \Big(\max_{y_{12} \in Y_{12}(z)} y_{12} \leq 0\Big)$$

in the description of the set M_5 (Example 3.2.6) by

$$\max_{y_{11}\in Y_{11}(z)} y_{11} \leq z_1\,, \qquad \max_{y_{12}\in Y_{12}(z)} y_{12} \leq z_1\,, \quad z_1 \leq 0.$$

Altogether, the GSIP reformulation of this DP turns out to be

$$\begin{aligned} GSIP: \quad \min_{x,z} f(x) \quad s.t. \quad & z_1 \leq 0, \max_{y_{11}\in Y_{11}(z)} y_{11} \leq z_1\,, \max_{y_{12}\in Y_{12}(z)} y_{12} \leq z_1 \\ & G_1(x) \leq z_{111}\,, \quad G_2(x) \leq z_{111}, \\ & G_3(x) \leq z_{112}\,, \quad G_4(x) \leq z_{113}, \\ & G_5(x) \leq z_{121}\,, \quad G_6(x) \leq z_{122} \end{aligned}$$

with the vector $z = (z_1, z_{111}, z_{112}, z_{113}, z_{121}, z_{122})^\intercal$ and the sets $Y_{11}(z)$, $Y_{12}(z)$ as defined above.

In this way, for both conjunctive and disjunctive expression trees T_Ω we obtain GSIP reformulations after finitely many steps. We will denote the (lifted) feasible set of the resulting GSIP by $\widetilde{M}_{GSIP}$.

Note that the variable z_1 appearing in the reformulation of a conjunctive tree (as in the above example) is a mere dummy variable, which may be eliminated in computations. As mentioned earlier, also variables which are only introduced to obtain a full and leaf disjunctive expression tree T_Ω may be eliminated. For example, our construction requires a tree of height three (i.e., with four levels), to model a logical expression Ω given in conjunctive normal form. Here, not only the variable corresponding to the root node, but also all variables corresponding to 1-conjunctions in the third level may be eliminated. In the following we shall omit further discussions of these elimination options.

The next theorem is proved by recursively applying Lemmata 3.2.3 and 3.2.5.

Theorem 3.2.8. *For any problem DP, whose expression tree T_Ω is full and leaf disjunctive, let GSIP be constructed as described above. Then, we have*

$$M_{DP} = \mathrm{pr}_x \widetilde{M}_{GSIP}.$$

3.3 GSIP Solution Techniques for DPs

The GSIP reformulation of DPs from Section 3.2 generates generalized semi-infinite constraints, which all share the simple structure

$$\varphi(z) \leq \zeta$$

with

$$\varphi(z) \;=\; \max_{y \in Y(z)} y$$

and

$$Y(z) \;=\; \{y \in \mathbb{R} :\; y \leq z_k,\; 1 \leq k \leq s\},$$

where the variable $\zeta \in \mathbb{R}$ does not belong to the auxiliary variables listed in the vector $z \in \mathbb{R}^s$. Note that the original decision vector $x \in \mathbb{R}^n$ does not explicitly appear in these generalized semi-infinite constraints, but enters via the simple coupling constraints between the entries of z and the functions $G_1, \ldots, G_m$ (see Section 3.2).

In the following, we shall describe different approaches to handle a constraint of the form $\varphi(z) \leq \zeta$ algorithmically. The application of the respective approaches to all appearing generalized semi-infinite constraints then leads to a solution method for the original DP. We will briefly describe the main ideas of the methods, before applying them to the problem at hand. For more details we refer to [116] and the references therein.

All subsequent approaches rely on the bilevel structure of generalized semi-infinite constraints, that is, the function φ is interpreted as the optimal value function of the so-called lower level problem

$$Q(z): \qquad \max_{y \in \mathbb{R}} y \quad \text{s.t.} \quad y \leq z_k\,,\; 1 \leq k \leq s.$$

Since the problem $Q(z)$ is a one-dimensional linear optimization problem, one may expect that sophisticated GSIP techniques collapse to rather obvious approaches. We shall see, however, that this is only partly the case.

3.3.1 The Lower Level Duality Reformulation

For convex lower level problems in generalized semi-infinite optimization, the idea to employ duality arguments goes back at least to [76] (for convex-quadratic problems). Similar approaches have been used in robust optimization and lead to a systematic treatment of GSIPs with smooth and convex lower level problems in [33]. There, the function φ is rewritten as the optimal value function of the corresponding Wolfe dual of the lower level problem, and the additional dual variables are included by lifting the feasible set.

In the present setting of the above problem $Q(z)$, this approach of course collapses to standard linear programming duality. The dual problem is

$$D(z): \qquad \min_{\gamma \in \mathbb{R}^s} \gamma^\intercal z \quad \text{s.t.} \quad \gamma \in \Sigma$$

with the standard simplex

$$\Sigma \;=\; \{\gamma \in \mathbb{R}^s :\; \gamma \geq 0,\; e^\intercal \gamma = 1\}$$

and the all-ones vector e. In view of strong duality we may rewrite the GSIP constraint as

$$\zeta \;\geq\; \varphi(z) \;=\; \min_{\gamma \in \Sigma} \gamma^\intercal z,$$

or, equivalently, as

$$\exists\, \gamma \in \mathbb{R}^s : \quad \gamma \geq 0,\; e^\intercal \gamma = 1,\; \gamma^\intercal z \leq \zeta.$$

Replacing each generalized semi-infinite constraint by this construction and lifting the feasible set to the corresponding set $\widehat{M}_P$ transforms the GSIP reformulation of DP into a finite and purely conjunctive optimization problem.

Example 3.3.1. *The lower level duality reformulation for the GSIP formulation of the DP from Example 3.1.1 in Example 3.2.7 is constructed as follows. Since the index set*

$$Y_{11}(z) \;=\; \{y_{11} \in \mathbb{R} :\; y_{11} \leq z_{11j}\,,\; 1 \leq j \leq 3\}$$

is described by three constraints, we introduce the vector

$$\widehat{\gamma}_{11} = (\gamma_{111}, \gamma_{112}, \gamma_{113})^\intercal \in \mathbb{R}^3$$

and may rewrite the constraint

$$\max_{y_{11} \in Y_{11}(z)} y_{11} \leq z_1$$

as

$$\exists\, \widehat{\gamma}_{11} \in \mathbb{R}^3 : \quad \widehat{\gamma}_{11} \geq 0,\; e^\intercal \widehat{\gamma}_{11} = 1,\; \gamma_{111}\, z_{111} + \gamma_{112}\, z_{112} + \gamma_{113}\, z_{113} \leq z_1.$$

Analogously, we replace the constraint

$$\max_{y_{12} \in Y_{12}(z)} y_{12} \leq z_1$$

by

$$\exists\, \widehat{\gamma}_{12} \in \mathbb{R}^2 : \quad \widehat{\gamma}_{12} \geq 0,\; e^\intercal \widehat{\gamma}_{12} = 1,\; \gamma_{121}\, z_{121} + \gamma_{122}\, z_{122} \leq z_1.$$

This results in the finite conjunctive optimization problem

$$\begin{aligned}
P : \quad \min_{x,z,\gamma} f(x) \quad \textit{s.t.} \quad & z_1 \leq 0, \\
& \widehat{\gamma}_{11} \geq 0,\; e^\intercal \widehat{\gamma}_{11} = 1,\; \gamma_{111}\, z_{111} + \gamma_{112}\, z_{112} + \gamma_{113}\, z_{113} \leq z_1, \\
& \widehat{\gamma}_{12} \geq 0,\; e^\intercal \widehat{\gamma}_{12} = 1,\; \gamma_{121}\, z_{121} + \gamma_{122}\, z_{122} \leq z_1, \\
& G_1(x) \;\leq\; z_{111}\,, \quad G_2(x) \;\leq\; z_{111}, \\
& G_3(x) \;\leq\; z_{112}\,, \quad G_4(x) \;\leq\; z_{113}, \\
& G_5(x) \;\leq\; z_{121}\,, \quad G_6(x) \;\leq\; z_{122}
\end{aligned}$$

with z defined as in Example 3.2.7, and $\gamma = (\gamma_{111}, \gamma_{112}, \gamma_{113}, \gamma_{121}, \gamma_{122})^\intercal$.

The subsequent result follows from our above discussions.

Theorem 3.3.2. *For any problem DP, whose expression tree T_Ω is full and leaf disjunctive, let the purely conjunctive problem P be constructed as described above. Then, we have $M_{DP} = \mathrm{pr}_x \widehat{M}_P$.*

We emphasize that resorting to GSIP techniques would actually not be necessary to come up with this reformulation of DP. In fact, the vertex theorem of linear programming immediately implies that the discrete minimum $\min_{1\leq k\leq s} z_k$ may be rewritten as the continuous minimum $\min_{\gamma\in\Sigma} \gamma^\intercal z$, from which the presented lifting approach follows along the lines sketched above.

Either way, this reformulation approach possesses the drawback that it may produce *spurious Karush-Kuhn-Tucker points*, in which an NLP solver for the problem P may terminate without identifying a locally minimal point of DP. Here, we call a KKT point $(\bar{x}, \bar{z}, \bar{\gamma})$ of P spurious, iff in the original problem DP there exist feasible first order descent directions at $\bar{x}$, so that in terms of any stationarity concept for DPs (e.g., a generalization of the concept from [60]) $\bar{x}$ may only be stationary in a weak sense.

We shall illustrate this effect for a so-called re-entrant corner point of the disjunctive program

$$DP: \qquad \min\ f(x) \quad \text{s.t.} \quad G_1(x) \leq 0 \ \vee\ \ldots\ \vee\ G_n(x) \leq 0$$

with continuously differentiable functions f, G_1, $\ldots$, G_n and $n \geq 2$. In fact, $\bar{x} \in \mathbb{R}^n$ is called a re-entrant corner point of M_{DP}, iff $G_i(\bar{x}) = 0$ holds for all $i = 1, \ldots, n$, and iff the gradients $\nabla G_i(\bar{x})$, $i = 1, \ldots, n$, are linearly independent. Obviously, the feasible set of DP may then locally be expressed as the union of finitely many sets with smooth boundaries. In Example 3.1.1 the point $(3 + \sqrt{3}/2, 0)$ is a re-entrant corner point.

After the deletion of auxiliary variables, the lifted problem P corresponding to DP is

$$P: \qquad \min_{x,\gamma}\ f(x) \quad \text{s.t.} \quad \gamma^\intercal G(x) \leq 0,\ \gamma^\intercal e = 1,\ \gamma \geq 0.$$

Proposition 3.3.3. *For a re-entrant corner point $\bar{x}$ of DP, let there exist multipliers $\bar{\rho}_i > 0$, $i = 1, \ldots, n$, with*

$$\nabla f(\bar{x}) + \sum_{i=1}^{n} \bar{\rho}_i \nabla G_i(\bar{x}) \ = \ 0.$$

Then, there exists a vector $\bar{\gamma} \in \Sigma$ such that $(\bar{x}, \bar{\gamma})$ is a Karush-Kuhn-Tucker point of P, at which the linear independence constraint qualification, as well as strict complementary slackness, are satisfied.

Proof. The vector $\bar{\gamma} := \bar{\rho}/\|\bar{\rho}\|_1$ lies in $\bar{\gamma} \in \Sigma$ and satisfies $\bar{\gamma} > 0$, that is, none of the constraints $\gamma_i \geq 0$, $i = 1, \ldots, n$, in P is active at $(\bar{x}, \bar{\gamma})$. In view of $G(\bar{x}) = 0$, on the other hand, the constraint $\gamma^\intercal G(x) \leq 0$ is active at $(\bar{x}, \bar{\gamma})$.

The gradients, with respect to (x, γ), of the two active constraints are linearly independent as the nontrivial linear combination $\nabla G(\bar{x})\bar{\gamma}$ of the gradients $\nabla G_i(\bar{x})$, $i = 1, \ldots, n$, cannot vanish under the linear independence assumption at re-entrant corner points. Furthermore, the Karush-Kuhn-Tucker condition

$$\begin{pmatrix} \nabla f(\bar{x}) \\ 0 \end{pmatrix} + \lambda \begin{pmatrix} \nabla G(\bar{x})\bar{\gamma} \\ 0 \end{pmatrix} + \mu \begin{pmatrix} 0 \\ e \end{pmatrix} \;=\; \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

of P at $(\bar{x}, \bar{\gamma})$ possesses the unique solution $\bar{\lambda} = \|\bar{\rho}\|_1 > 0$ and $\bar{\mu} = 0$. In particular, also strict complementary slackness holds. □

The following result shows, at least, that algorithms using second order information when solving P should not terminate in a spurious Karush-Kuhn-Tucker point from Proposition 3.3.3, since a second order descent direction always exists.

Proposition 3.3.4. *In addition to the assumptions of Proposition 3.3.3, let the functions f, G_i, $i = 1, \ldots, n$, be twice continuously differentiable. Then, a second order descent direction for P exists at $(\bar{x}, \bar{\gamma})$.*

Proof. With the notation from the proof of Proposition 3.3.3, recall the relations $\bar{\gamma} > 0$ and $\bar{\lambda} > 0$. Let us define the vectors

$$\delta \;:=\; \left(1, \ldots, 1, -\left(\sum_{i=1}^{n-1} \bar{\gamma}_i\right) / \bar{\gamma}_n\right)^\intercal,$$

$$d(c) \;:=\; c(\nabla G(\bar{x}))^{-\intercal}\delta \quad \text{with} \quad c > 0,$$

as well as

$$\eta \;:=\; (-1, \ldots, -1, n-1)^\intercal.$$

Then, for any $c > 0$, the vector $(d(c), \eta)$ lies in the tangent space to the feasible set of P at $(\bar{x}, \bar{\gamma})$, since we have

$$\left\langle \begin{pmatrix} \nabla G(\bar{x})\bar{\gamma} \\ 0 \end{pmatrix}, \begin{pmatrix} d(c) \\ \eta \end{pmatrix} \right\rangle \;=\; c\langle \bar{\gamma}, \delta \rangle \;=\; 0$$

and

$$\left\langle \begin{pmatrix} 0 \\ e \end{pmatrix}, \begin{pmatrix} d(c) \\ \eta \end{pmatrix} \right\rangle \;=\; \langle e, \eta \rangle \;=\; 0.$$

Moreover, the Hessian with respect to (x, γ) of the corresponding Lagrangian of P,

$$L(x, \gamma, \lambda, \mu) \;=\; f(x) + \lambda \gamma^\intercal G(x) + \mu(\gamma^\intercal e - 1)$$

at $(\bar{x}, \bar{\gamma}, \bar{\lambda}, \bar{\mu})$ is

$$D^2_{(x,\gamma)} L(\bar{x}, \bar{\gamma}, \bar{\lambda}, \bar{\mu}) \;=\; \begin{pmatrix} D^2 f(\bar{x}) + \bar{\lambda} \sum_{i=1}^n \bar{\gamma}_i D^2 G_i(\bar{x}) & \bar{\lambda} \nabla G(\bar{x}) \\ \bar{\lambda} \nabla G(\bar{x})^\intercal & 0 \end{pmatrix},$$

and we obtain

$$\begin{aligned} & \begin{pmatrix} d(c) \\ \eta \end{pmatrix}^\intercal D^2_{(x,\gamma)} L(\bar{x}, \bar{\gamma}, \bar{\lambda}, \bar{\mu}) \begin{pmatrix} d(c) \\ \eta \end{pmatrix} \\ &= \; d(c)^\intercal \left(D^2 f(\bar{x}) + \sum_{i=1}^n \bar{\rho}_i D^2 G_i(\bar{x}) \right) d(c) \;+\; 2\bar{\lambda} d(c)^\intercal \nabla G(\bar{x}) \eta \\ &= \; c^2 \delta^\intercal \nabla G(\bar{x})^{-1} \left(D^2 f(\bar{x}) + \sum_{i=1}^n \bar{\rho}_i D^2 G_i(\bar{x}) \right) (\nabla G(\bar{x}))^{-\intercal} \delta \\ &\quad + 2\bar{\lambda} c \delta^\intercal \eta. \end{aligned}$$

Due to $\bar{\lambda} > 0$ and

$$\delta^\intercal \eta \;=\; -(n-1) - \frac{n-1}{\bar{\gamma}_n} \sum_{i=1}^{n-1} \bar{\gamma}_i \;<\; 0,$$

the latter is negative for a sufficiently small value $\bar{c} > 0$, so that $(d(\bar{c}), \eta)$ is the asserted second order descent direction. □

3.3.2 The Outer and Inner Smoothing Reformulations

From an algorithmic point of view it is often useful to know outer and inner approximations of the feasible set of an optimization problem. For the set M_{DP} this section shows how smoothing methods from generalized semi-infinite optimization may be employed to this end.

In fact, for any generalized semi-infinite optimization problem of the general form

$$GSIP: \qquad \min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad x \in M_{GSIP}$$

with the feasible set

$$M_{GSIP} \;=\; \{x \in \mathbb{R}^n : \; g_i(x, y^i) \leq 0 \text{ for all } y^i \in Y_i(x), \; i \in I\}$$

and

$$Y_i(x) \;=\; \{y^i \in \mathbb{R}^{m_i} :\; v^i(x,y^i) \leq 0\},\; i \in I,$$

in [111] the equivalence of $GSIP$ with the problem

$$SG: \qquad \min_{x,y^1,\ldots,y^p} f(x) \quad \text{s.t.} \quad g(x,y^i) \leq 0,\; y^i \text{ is a solution of } Q_i(x),\; i \in I$$

is shown, as long as $Y_i(x)$ is nonempty for all $x \in \mathbb{R}^n$ and $i \in I$. Here, the lower level problem of the constraint indexed with $i \in I$ is denoted by

$$Q_i(x): \qquad \max_{y^i \in \mathbb{R}^{m_i}} g_i(x,y^i) \quad \text{s.t.} \quad v^i(x,y^i) \leq 0.$$

Note that the former index variables y^i, $i \in I$, are treated as additional decision variables in SG, so that already this step of the reformulation is a lifting approach. Since a part of the decision variables is constrained to solve an optimization problem depending on the other decision variables, this problem has the structure of a Stackelberg game ([16,32]).

In a next step, for each $i \in I$ the fact that y^i solves $Q_i(x)$ is equivalently replaced by an optimality condition. If the problems $Q_i(x)$, $i \in I$, are convex, if the Slater condition holds in each set $Y_i(x)$, $x \in \mathbb{R}^n$, $i \in I$, and if all functions g_i, v^i, $i \in I$, are continuously differentiable on their respective domains, then this may be achieved by the Karush-Kuhn-Tucker conditions

$$\begin{aligned} \exists\, \gamma^i \in \mathbb{R}^{s_i}: \quad \nabla_y g_i(x,y^i) - \nabla_y v^i(x,y^i)\,\gamma^i \;&=\; 0, \\ 0 \;\leq\; \gamma^i \;\perp\; -v^i(x,y^i) \;&\geq\; 0 \end{aligned}$$

for each $i \in I$, where the second line stands for a complementarity condition. The introduction of the additional variables γ^i, $i \in I$, leads to another lifting of the feasible set whose description, however, contains complementarity constraints. The resulting problem is thus the mathematical program with complementarity constraints

$$\begin{aligned} MPCC: \quad \min_{x,y^1,\ldots,y^p,\gamma^1,\ldots,\gamma^p} f(x) \quad \text{s.t.} \quad & g(x,y^i) \;\leq\; 0, \\ & \nabla_y g_i(x,y^i) - \nabla_y v^i(x,y^i)\,\gamma^i \;=\; 0, \\ & 0 \;\leq\; \gamma^i \;\perp\; -v^i(x,y^i) \;\geq\; 0,\; i \in I. \end{aligned}$$

MPCCs turn out to be numerically challenging, since the so-called Mangasarian-Fromovitz constraint qualification (MFCQ) is violated everywhere in their feasible set ([105]).

A first numerical approach to the MPCC reformulation of a general GSIP was given in [112,113] by applying the smoothing procedure for MPCCs from

[40]. In fact, each scalar complementarity constraint $0 \le \gamma_k^i \perp -v_k^i(x,y^i) \ge 0$, $1 \le k \le s_i$, $i \in I$, is first replaced by the equation $\psi(\gamma_k^i, -v_k^i(x,y^i)) = 0$ with a complementarity function ψ like the natural residual function

$$\psi^{NR}(a,b) = \min(a,b)$$

or the Fischer-Burmeister function

$$\psi^{FB}(a,b) = a + b - \sqrt{a^2+b^2}.$$

The nonsmooth function ψ is then equipped with a smoothing parameter $\tau > 0$, for example

$$\psi_\tau^{NR}(a,b) \;=\; \frac{1}{2}\left(a + b - \sqrt{(a-b)^2 + 4\tau^2}\right)$$

or

$$\psi_\tau^{FB}(a,b) \;=\; a + b - \sqrt{a^2 + b^2 + 2\tau^2},$$

so that ψ_τ is smooth and ψ_0 coincides with ψ. This gives rise to the family of smoothed problems

$$\begin{aligned} P_\tau: \qquad \min_{x,y^1,\ldots,y^p,\gamma^1,\ldots,\gamma^p} f(x) \quad \text{s.t.} \quad & g(x,y^i) \;\le\; 0, \\ & \nabla_y g_i(x,y^i) - \nabla_y v^i(x,y^i)\,\gamma^i \;=\; 0, \\ & \psi_\tau(\gamma^i, -v^i(x,y^i)) \;=\; 0,\ i \in I \end{aligned}$$

with $\tau > 0$, where ψ_τ is extended to vector arguments componentwise. Under mild assumptions, in [112,113] it is shown that P_τ is numerically tractable, and that stationary points of P_τ tend to a stationary point of $GSIP$ for $\tau \to 0$.

While $MPCC$ still is an equivalent formulation of $GSIP$, the smoothed problem P_τ only is an approximation. In [112] it is shown that its feasible set $\widehat{M}_\tau$ at least satisfies $\mathrm{pr}_x \widehat{M}_\tau \supseteq M_{GSIP}$, that is, it constitutes an *outer* approximation of M_{GSIP} for $\tau > 0$. This means, unfortunately, that the x-parts of optimal points of P_τ must be expected to be infeasible for $GSIP$.

However, in [114] it could be shown that a simple modification of P_τ leads to *inner* approximations of M_{GSIP}, and thus to *feasible* optimal points of the approximating problems. In fact, an error analysis for the approximation of the lower level optimal value proves that the feasible set $\widehat{M}_\tau^\circ$ of

$$\begin{aligned} P_\tau^\circ: \qquad \min_{x,y^1,\ldots,y^p,\gamma^1,\ldots,\gamma^p} f(x) \quad \text{s.t.} \quad & g(x,y^i) + s_i\tau^2 \;\le\; 0, \\ & \nabla_y g_i(x,y^i) - \nabla_y v^i(x,y^i)\,\gamma^i \;=\; 0, \\ & \psi_\tau(\gamma^i, -v^i(x,y^i)) \;=\; 0,\ i \in I \end{aligned}$$

satisfies $\mathrm{pr}_x \widehat{M}_\tau^\circ \subseteq M_{GSIP}$ (where s_i denotes the number of lower level inequality constraints). A combination of the outer and inner smoothing approaches leads to 'sandwiching' procedures for M_{GSIP} ([114]).

For each lower level problem of the form

$$Q(z): \qquad \max_{y\in\mathbb{R}} y \quad \text{s.t.} \quad y \le z_k\,,\ 1 \le k \le s$$

appearing in the GSIP reformulation of DPs, the convexity and differentiability assumptions, as well as the Slater condition are obviously satisfied, so that outer smoothing is achieved by the constraints

$$y \ \le\ 0, \quad e^\intercal\gamma \ =\ 1, \quad \psi_\tau(\gamma_k\,,\, z_k - y) \ =\ 0, \quad 1 \le k \le s,$$

whereas inner smoothing results from

$$y + s\tau^2 \ \le\ 0, \quad e^\intercal\gamma \ =\ 1, \quad \psi_\tau(\gamma_k\,,\, z_k - y) \ =\ 0, \quad 1 \le k \le s.$$

Example 3.3.5. *We construct the outer smoothing reformulation for the GSIP formulation of the DP from Example 3.1.1 in Example 3.2.7 as follows. For the index set*

$$Y_{11}(z) \ =\ \{y_{11} \in \mathbb{R}:\ y_{11} \le z_{11j}\,,\ 1 \le j \le 3\}$$

we again introduce the vector $\widehat{\gamma}_{11} \in \mathbb{R}^3$*, but now subject to the constraints*

$$y_{11} \ \le\ z_1, \quad e^\intercal\widehat{\gamma}_{11} \ =\ 1, \quad \psi_\tau(\gamma_{11j}\,,\, z_{11j} - y_{11}) \ =\ 0, \quad 1 \le j \le 3$$

with some $\tau > 0$*. An analogous construction for the second generalized semi-infinite constraint leads to the finite conjunctive optimization problem*

$$\begin{aligned} P_\tau: \quad \min_{x,y,z,\gamma} \ & f(x) \quad \textit{s.t.} \quad z_1 \le 0, \\ & y_{11} \le z_1,\ e^\intercal\widehat{\gamma}_{11} = 1,\ \psi_\tau(\gamma_{11j}, z_{11j} - y_{11}) \ =\ 0,\ 1 \le j \le 3, \\ & y_{12} \le z_1,\ e^\intercal\widehat{\gamma}_{12} = 1,\ \psi_\tau(\gamma_{12j}, z_{12j} - y_{12}) \ =\ 0,\ 1 \le j \le 2, \\ & G_1(x) \ \le\ z_{111}\,, \quad G_2(x) \ \le\ z_{111}, \\ & G_3(x) \ \le\ z_{112}\,, \quad G_4(x) \ \le\ z_{113}, \\ & G_5(x) \ \le\ z_{121}\,, \quad G_6(x) \ \le\ z_{122} \end{aligned}$$

with $y = (y_{11}, y_{12})^\intercal$ *and the vectors* z *and* γ *defined as in Examples 3.2.7 and 3.3.1.*

The inner smoothing problem, on the other hand, is given by

$$
\begin{aligned}
P_\tau^\circ : \qquad \min_{x,y,z,\gamma} \; f(x) \quad s.t. \quad & z_1 \le 0, \\
& y_{11} + 3\tau^2 \le z_1, \; e^\intercal \widehat{\gamma}_{11} = 1, \\
& \psi_\tau(\gamma_{11j}, z_{11j} - y_{11}) \;=\; 0, \; 1 \le j \le 3, \\
& y_{12} + 2\tau^2 \le z_1, \; e^\intercal \widehat{\gamma}_{12} = 1, \\
& \psi_\tau(\gamma_{12j}, z_{12j} - y_{12}) \;=\; 0, \; 1 \le j \le 2, \\
& G_1(x) \;\le\; z_{111}\,, \quad G_2(x) \;\le\; z_{111}, \\
& G_3(x) \;\le\; z_{112}\,, \quad G_4(x) \;\le\; z_{113}, \\
& G_5(x) \;\le\; z_{121}\,, \quad G_6(x) \;\le\; z_{122}\,.
\end{aligned}
$$

Figure 3.4 shows the projected feasible sets resulting from outer and inner smoothings for the set M_{DP} from Figure 3.1, for several values of τ.

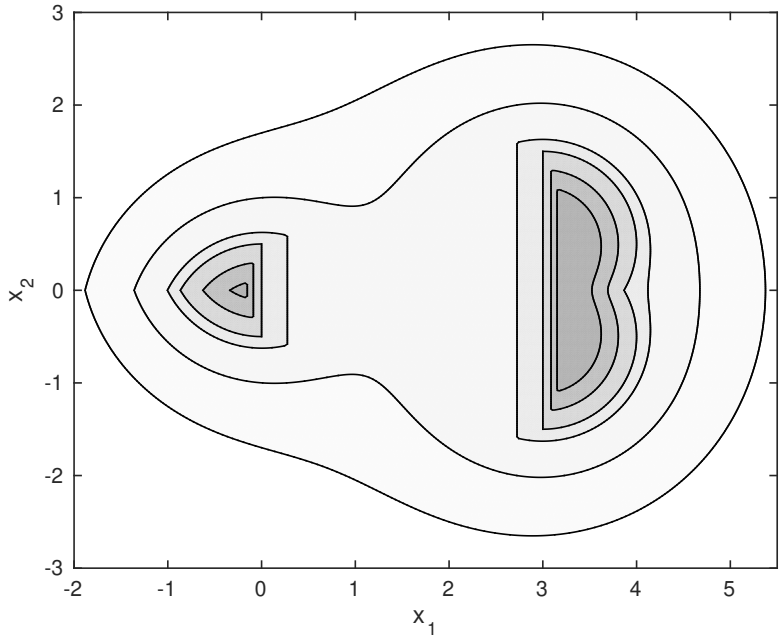

Figure 3.4: Outer and inner smoothings from Example 3.3.5

We summarize our discussion so far in the following result.

Theorem 3.3.6. *For any problem DP, whose expression tree T_Ω is full and leaf disjunctive, let the purely conjunctive smoothing problems P_τ and P_τ° be constructed as described above. Then, we have* $\mathrm{pr}_x \widehat{M}_\tau^\circ \subseteq M_{DP} \subseteq \mathrm{pr}_x \widehat{M}_\tau$.

Unfortunately, smoothing does not prevent the appearance of the spurious Karush-Kuhn-Tucker points from Section 3.3.1. This is a consequence of typical nondegeneracy of spurious Karush-Kuhn-Tucker points in the unperturbed problem P and the implicit function theorem. In fact, these Karush-Kuhn-Tucker points then prevail under small perturbations, and small smoothing parameters $\tau > 0$ may be interpreted as such perturbations. We omit the details of this analysis for space reasons.

Instead, in the following we shall provide explicit estimates for the feasibility and optimality errors occurring in the smoothing reformulation from Theorem 3.3.6. General results for generalized semi-infinite optimization problems which could be applied here are not available in the current literature, so that we shall at least show how to derive such estimates for the outer smoothing reformulation.

3.3.3 Stability Analysis for Outer Smoothing

For $\tau > 0$, let the problem P_τ corresponding to DP be constructed as above. Recall that this construction involves first lifting the original variables x to (x, z), and in a second step to (x, y, z, γ). Subsequently we will not work with the set $\widehat{M}_\tau$ of (x, y, z, γ)-variables, but first consider its orthogonal projection $\widetilde{M}_\tau := \mathrm{pr}_{(x,z)}\widehat{M}_\tau$ to the space of (x, z)-variables, and then deduce stability results in the space $\mathbb{R}^n$ of x-variables.

In the description of the set $\widetilde{M}_{GSIP}$ (Section 3.2), instead of lifting each appearing semi-infinite constraint

$$\varphi(z) \;=\; \max_{y \in Y(z)} y \;\leq\; \zeta \quad \text{with} \quad Y(z) = \{y \in \mathbb{R} : \, y \leq z_k, \, 1 \leq k \leq s\}$$

by employing optimal points and their multipliers, and then smoothing the complementarity constraints, we may equivalently use the logarithmic barrier approach for the lower level problem. It is not hard to see ([112]) that, for any $\tau > 0$, with

$$y_\tau(z) \;:=\; \operatorname*{argmax}_y \left(y + \tau^2 \sum_{k=1}^{s} \log(z_k - y) \right),$$

a functional description of $\widetilde{M}_\tau$ results from the description of $\widetilde{M}_{GSIP}$ by replacing each inequality of the form $\varphi(z) \leq \zeta$ by $y_\tau(z) \leq \zeta$. Note that this reformulation is not suitable for numerical purposes, as checking feasibility for $\widetilde{M}_\tau$ involves the computation of the optimal points $y_\tau(z)$. It is helpful, however, for proving the announced bounds.

In fact, from [18,114] we know the estimates

$$0 \;\leq\; \varphi(z) - y_\tau(z) \;\leq\; s\tau^2$$

for any z. While the lower bound yields the inclusion $\widetilde{M}_{GSIP} \subseteq \widetilde{M}_\tau$, the upper bound allows us to bound the feasibility error, measured as the excess of $\widetilde{M}_\tau$ over $\widetilde{M}_{GSIP}$,

$$\operatorname{ex}(\widetilde{M}_\tau, \widetilde{M}_{GSIP}) \;:=\; \sup_{(x,z)\in\widetilde{M}_\tau} \operatorname{dist}((x,z), \widetilde{M}_{GSIP}),$$

where

$$\operatorname{dist}((x,z), \widetilde{M}_{GSIP}) \;:=\; \inf_{(y,\eta)\in\widetilde{M}_{GSIP}} \|(y,\eta) - (x,z)\|_2$$

denotes the Euclidean distance of (x,z) from $\widetilde{M}_{GSIP}$.

The main assumption for our error estimate will be the validity of a global error bound for the inequality system describing the set $\widetilde{M}_{GSIP}$. To state it, let $\varphi_i(z) \leq \zeta_i$, $i \in I$, denote the semi-infinite constraints in the description of $\widetilde{M}_{GSIP}$, and $g_j(x,z) \leq 0$, $j \in J$, the remaining constraints. Then, the system of inequalities $\varphi_i(z) \leq \zeta_i$, $i \in I$, $g_j(x,z) \leq 0$, $j \in J$, satisfies a global error bound with Hoffman constant $\gamma > 0$ iff

$$\operatorname{dist}((x,z), \widetilde{M}_{GSIP}) \;\leq\; \gamma \max\{(\varphi_i(z) - \zeta_i)^+,\, i \in I,\, g_j^+(x,z),\, j \in J\}$$

holds for all (x,z), where $a^+ := \max(0,a)$ denotes the plus function. For surveys on global error bounds we refer to [8,9,87].

Lemma 3.3.7. *Let the system of inequalities $\varphi_i(z) \leq \zeta_i$, $i \in I$, $g_j(x,z) \leq 0$, $j \in J$, describing $\widetilde{M}_{GSIP}$ satisfy a global error bound with Hoffman constant $\gamma > 0$, and let $\bar{s}$ denote the length of the longest disjunction appearing in the problem DP. Then, for any $\tau > 0$ the feasibility error in the (x,z)-space satisfies*

$$\operatorname{ex}(\widetilde{M}_\tau, \widetilde{M}_{GSIP}) \;\leq\; \gamma\tau^2\bar{s}.$$

Proof. For any $(x,z) \in \widetilde{M}_\tau$ and any $j \in J$ we have $g_j(x,z) \leq 0$ and hence $g_j^+(x,z) = 0$. Moreover, as for any $i \in I$ we have $y_{i,\tau}(z) \leq \zeta_i$, we may conclude

$$\varphi_i(z) - \zeta_i \;\leq\; (\varphi_i(z) - \zeta_i) - (y_{i,\tau}(z) - \zeta_i) \;\leq\; s_i\tau^2.$$

This implies

$$(\varphi_i(z) - \zeta_i)^+ \;\leq\; s_i\tau^2$$

and, altogether, the validity of the error bound results in the estimate

$$\operatorname{dist}((x,z),\widetilde{M}_{GSIP}) \;\leq\; \gamma\tau^2 \max_{i\in I} s_i \;=\; \gamma\tau^2\bar{s}.$$

We arrive at the assertion

$$\operatorname{ex}(\widetilde{M}_\tau,\widetilde{M}_{GSIP}) \;=\; \sup_{(x,z)\in\widetilde{M}_\tau} \operatorname{dist}((x,z),\widetilde{M}_{GSIP}) \;\leq\; \gamma\tau^2\bar{s}.$$

□

By means of the following result, we will be able to transfer the estimate from Lemma 3.3.7 from the space of (x,z)-variables to the corresponding sets in $\mathbb{R}^n$.

Lemma 3.3.8. *Let $A, B \subseteq \mathbb{R}^n \times \mathbb{R}^m$ with $A \supseteq B$, and let $\mathbb{R}^n$ be the space of x-variables. Then, the relations*

$$\operatorname{pr}_x A \supseteq \operatorname{pr}_x B \quad \textit{and} \quad \operatorname{ex}(\operatorname{pr}_x A, \operatorname{pr}_x B) \leq \operatorname{ex}(A,B)$$

hold.

Proof. The inclusion is obvious. Moreover, for any $(x,z) \in A$ we find

$$\begin{aligned}\operatorname{dist}((x,z),B) &= \inf_{(y,\eta)\in B} \|(y,\eta)-(x,z)\|_2 \;\geq\; \inf_{(y,\eta)\in B} \|y-x\|_2 \\ &= \inf_{y\in\operatorname{pr}_x B} \|y-x\|_2 \;=\; \operatorname{dist}(x,\operatorname{pr}_x B).\end{aligned}$$

This implies

$$\begin{aligned}\operatorname{ex}(A,B) &= \sup_{(x,z)\in A} \operatorname{dist}((x,z),B) \;\geq\; \sup_{(x,z)\in A} \operatorname{dist}(x,\operatorname{pr}_x B) \\ &= \sup_{x\in\operatorname{pr}_x A} \operatorname{dist}(x,\operatorname{pr}_x B) \;=\; \operatorname{ex}(\operatorname{pr}_x A,\operatorname{pr}_x B).\end{aligned}$$

□

Combining Lemmata 3.3.7 and 3.3.8 with the identities $\operatorname{pr}_x\widetilde{M}_\tau = \operatorname{pr}_x\widehat{M}_\tau$ and $\operatorname{pr}_x\widetilde{M}_{GSIP} = M_{DP}$ yields the following main stability result for the outer approximation approach.

Theorem 3.3.9. *Let the system of inequalities $\varphi_i(z) \leq \zeta_i$, $i \in I$, $g_j(x,z) \leq 0$, $j \in J$, describing $\widetilde{M}_{GSIP}$ satisfy a global error bound with Hoffman constant $\gamma > 0$, and let $\bar{s}$ denote the length of the longest disjunction appearing in the problem DP. Then, for any $\tau > 0$ the feasibility error satisfies*

$$\operatorname{ex}(\operatorname{pr}_x\widehat{M}_\tau, M_{DP}) \;\leq\; \gamma\tau^2\bar{s}.$$

Corollary 3.3.10. *In addition to the assumptions of Theorem 3.3.9 let the objective function f of DP satisfy a global Lipschitz condition on $\mathrm{pr}_x\widehat{M}_\tau$ with Lipschitz constant $L > 0$ (independently of τ). Assume that f possesses globally minimal points on M_{DP} as well as on $\mathrm{pr}_x\widehat{M}_\tau$, and denote the corresponding optimal values by v and v_τ, respectively. Then, the optimality error satisfies*

$$0 \;\leq\; v - v_\tau \;\leq\; L\gamma\tau^2\bar{s}.$$

Proof. The first asserted inequality is due to the inclusion $M_{DP} \subseteq \mathrm{pr}_x\widehat{M}_\tau$. Moreover, with a globally minimal point $x_\tau^\star$ of f on $\mathrm{pr}_x\widehat{M}_\tau$ the optimality error satisfies

$$v - v_\tau \;\leq\; f(x) - f(x_\tau^\star) \;\leq\; L\|x - x_\tau^\star\|_2$$

for all $x \in M_{DP}$ and, in view of Theorem 3.3.9,

$$\begin{aligned} v - v_\tau \;&\leq\; L \inf_{x\in M_{DP}} \|x - x_\tau^\star\|_2 \;=\; L\,\mathrm{dist}(x_\tau^\star, M_{DP}) \\ &\leq\; L\,\mathrm{ex}(\mathrm{pr}_x\widehat{M}_\tau, M_{DP}) \;\leq\; L\gamma\tau^2\bar{s}. \end{aligned}$$

□

The estimates derived above may not be expected to hold for the outer approximation of any disjunctive optimization problem, since they crucially depend on the global error bound assumption made on the inequality system describing $\widetilde{M}_{GSIP}$. The statement of sufficient conditions for the global error bound is beyond the scope of this thesis, but at least we point out that the nonconvexity of the appearing functions

$$\varphi_i(z) - \zeta_i \;=\; \min_{k=1,\ldots,s_i} z_k - \zeta_i, \quad i \in I,$$

does *not* interfere with the existence of a Hoffman constant. In fact, for a single $i \in I$ define the sublevel set $\Phi_i := \{(z,\zeta_i) : \min_{k=1,\ldots,s} z_k \leq \zeta_i\}$ and choose $(\bar{z},\bar{\zeta}_i)$ from its set complement Φ_i^c. Then, $\bar{z}_k > \bar{\zeta}_i$ and $(\bar{z} - (\bar{z}_k - \bar{\zeta}_i)e_k, \zeta_i) \in \Phi_i$ hold for all $k \in \{1,\ldots,s_i\}$, where e_k denotes the k-th unit vector. As a consequence, $\mathrm{dist}((\bar{z},\bar{\zeta}_i),\Phi_i) \leq \bar{z}_k - \bar{\zeta}_i$ holds for any $k \in \{1,\ldots,s_i\}$, so that we arrive at

$$\mathrm{dist}((\bar{z},\bar{\zeta}_i),\Phi_i) \;\leq\; \min_{k=1,\ldots,s_i} \bar{z}_k - \bar{\zeta}_i \;=\; (\varphi_i(\bar{z}) - \bar{\zeta}_i)^+.$$

As the latter inequality trivially holds for any $(\bar{z},\bar{\zeta}_i) \in \Phi_i$, we have shown that the single nonconvex constraint $\varphi_i(z) - \zeta_i$ satisfies a global error bound with Hoffman constant $\gamma_i = 1$. Unfortunately, a Hoffman constant for the

whole system of inequalities is not easily constructed from this information, but at least our remark shows that the discussed nonconvexity is not harmful.

Finally, let us note that the corresponding stability analysis for the inner approximation $\mathrm{pr}_x \widehat{M}_\tau^\circ$ of M_{DP} is not completely analogous, as an error bound for the τ-dependent set $\mathrm{pr}_{(x,z)} \widehat{M}_\tau^\circ$ has to be assumed. Since the behavior of the corresponding Hoffman constants γ_τ for $\tau \to 0$ is not clear, we leave the analysis of this question for future research.

3.4 Preliminary Numerical Results

In this section, we present our numerical results. Our computational tests are performed on a Intel Pentium CPU 987, 1.5 GHz with 4 GB of RAM running Linux. As a solver for the occurring nonlinear problems we used Ipopt [130], version 3.11.7. The termination tolerance is set to 10^{-6}. The algorithm is implemented in C++ and compiled with GNU g++, version 4.7.

In the largest part of our implementation we make heavy use of operator overloading to construct the conjunctive optimization problem based on the given disjunctive problem. This makes it is rather convenient to model even complex logical expressions since both the lower level duality reformulation as well as the outer and inner smoothing reformulations can be constructed automatically.

However, no automatic or numerical differentiation is implemented, and thus the first derivatives have to be entered manually so far. All Hessians are approximated using a feature of Ipopt.

For the test problem

$$DP: \qquad \min_{x \in \mathbb{R}^2} f(x) \quad \text{s.t.} \quad x \in M_{DP}$$

with

$$M_{DP} = \Big\{ x \in \mathbb{R}^2 : \ \big(G_1(x) \leq 0 \vee G_2(x) \leq 0\big) \wedge G_3(x) \leq 0 \Big\}$$

and

$$\begin{aligned} f(x) &:= -x_2, \\ G_1(x) &:= x_1^2 + x_2^2 - 1, \\ G_2(x) &:= (x_1 - 1)^2 + x_2^2 - 1, \\ G_3(x) &:= -x_2, \end{aligned}$$

the feasible set can be described as the union of two semi-discs, both with radius 1. Their midpoints are the origin and $(1,0)^\intercal$, respectively, and the globally minimal points of the problem are $(0,1)^\intercal$ and $(1,1)^\intercal$. There are no further locally minimal points, while we do have a re-entrant corner point at $(0.5, \frac{\sqrt{3}}{2})^\intercal$ that leads to a spurious Karush-Kuhn-Tucker point in the lower level duality reformulation, as described in Section 3.3.

Using the lower level duality reformulation and starting Ipopt from the point $(x_1^0, x_2^0, 0, \ldots, 0)^\intercal$ with $(x_1^0, x_2^0) = (0,0)$, the algorithm converges to a point with first components $(x_1^*, x_2^*)^\intercal = (0.00000, 1.00000)^\intercal$, which corresponds to one of the globally minimal points of the original problem, within 10 iterations. We obtained very similar results for various different starting points of the form $(x_1^0, x_2^0, 0, \ldots, 0)^\intercal$, where x_1^0 and x_2^0 are chosen from the set $\{-1, 0, 1\}$. The algorithm then always converges to one of the globally minimal points. The results are presented in the upper part of Table 3.1, where (x_1^0, x_2^0) denotes the first two coordinates for the initial point, (x_1^*, x_2^*) denotes the first coordinates of the limit point computed by the nonlinear solver and v^* is the value of the objective function at that point.

There seems to be one notable exception to this observation, that we show in the lower part of Table 3.1. If in the initial point we set $x_1^0 = 0.5$, $x_2^0 \in \mathbb{R}$, and the remaining variables to zero, then the algorithm is likely to converge to the spurious Karush-Kuhn-Tucker point mentioned above. This even holds for more points with $x_1^0 = 0.5$, but not for all, as can be seen from the very last row. This is in accordance with Proposition 3.3.4, which guarantees the existence of a second order descent direction at the spurious KKT point, and thus convergence to this point at most along some lower dimensional manifold.

In principle the numerical observations from the paragraph above remain true if we switch from the lower level duality reformulation to the smoothed mathematical program with complementarity constraints. This can be seen from Table 3.2 and Table 3.3. With $\tau = 0.1$ the smoothing parameter is chosen rather large to demonstrate the effects from the outer and inner approximation. Starting from $(0, 0, 0 \ldots, 0)^\intercal$ the algorithm converges to a point with $(x_1^*, x_2^*)^\intercal = (0.00010, 1.00504)^\intercal$ within 15 iterations. For the inner approximation with the same value of τ it takes 14 iterations to approximate a point with the first two components $(x_1^*, x_2^*)^\intercal = (0.00010, 0.99504)^\intercal$. Thus, in both cases we generate a point close to one of the globally minimal points of the original problem.

There are two test runs in Table 3.2 and Table 3.3, respectively, that do not work properly. The according lines are marked by '-'. For these

Table 3.1: Lower level duality reformulation of the first test problem

(x_1^0, x_2^0)	(x_1^*, x_2^*)	v^*
$(-1.0, -1.0)$	$(0.000000, 1.000000)$	-1.000000
$(-1.0, 0.0)$	$(0.000000, 1.000000)$	-1.000000
$(-1.0, 1.0)$	$(0.000000, 1.000000)$	-1.000000
$(0.0, -1.0)$	$(1.000000, 1.000000)$	-1.000000
$(0.0, 0.0)$	$(0.000001, 1.000000)$	-1.000000
$(0.0, 1.0)$	$(0.000000, 1.000000)$	-1.000000
$(1.0, -1.0)$	$(0.000000, 1.000000)$	-1.000000
$(1.0, 0.0)$	$(0.999999, 1.000000)$	-1.000000
$(1.0, 1.0)$	$(1.000000, 1.000000)$	-1.000000
$(0.5, -2.0)$	$(0.500000, 0.866025)$	-0.866025
$(0.5, -1.0)$	$(0.500000, 0.866025)$	-0.866025
$(0.5, 0.0)$	$(0.500000, 0.866025)$	-0.866025
$(0.5, 1.0)$	$(0.500000, 0.866025)$	-0.866025
$(0.5, 2.0)$	$(0.000000, 1.000000)$	-1.000000

instances Ipopt reports that no feasible point could be found. However, finding a feasible point is usually not an issue for Ipopt, as the remaining results indicate.

We also tested our reformulation approaches for the problem from Example 3.1.1. The numerical results are similar. For the lower level duality approach these are presented in Table 3.4. With $\tau = 10^{-4}$ the parameter for the smoothed mathematical programs is still chosen moderately but already sufficiently small, as can be seen from Table 3.5 and Table 3.6. From certain starting points, also in this example convergence to the spurious Karush-Kuhn-Tucker point $(3 + \sqrt{3}/2, 0)$ occurs in all of our reformulation approaches, as has to be expected from Proposition 3.3.4 and the corresponding remark at the end of Section 3.3.2.

For larger problem instances, one might want to refine the smoothing parameter τ adaptively, if necessary. With respect to outer smoothing, Figure 3.4 suggests to start with a rather large parameter, since the set $\mathrm{pr}_x \widehat{M}_\tau$ then looks 'almost convex'. One may ask under which conditions this can be guaranteed for the outer smoothing of disjunctive optimization problems, how the degree of convexity can be measured appropriately, and when tracing solutions of the smoothed problems for $\tau \to 0$ may approximate a globally minimal point of the original problem. We leave this question for future

Table 3.2: Outer smoothing reformulation of the first test problem with $\tau = 0.1$

(x_1^0, x_2^0)	(x_1^*, x_2^*)	v^*
$(-1.0, -1.0)$	-	-
$(-1.0, 0.0)$	$(0.000100, 1.005037)$	-1.005037
$(-1.0, 1.0)$	$(0.999900, 1.005037)$	-1.005037
$(0.0, -1.0)$	$(0.000100, 1.005037)$	-1.005037
$(0.0, 0.0)$	$(0.000100, 1.005037)$	-1.005037
$(0.0, 1.0)$	$(0.000100, 1.005037)$	-1.005037
$(1.0, -1.0)$	$(0.999900, 1.005037)$	-1.005037
$(1.0, 0.0)$	$(0.999900, 1.005037)$	-1.005037
$(1.0, 1.0)$	$(0.999900, 1.005037)$	-1.005037
$(0.5, -2.0)$	$(0.000100, 1.005037)$	-1.005037
$(0.5, -1.0)$	-	-
$(0.5, 0.0)$	$(0.500000, 0.877496)$	-0.877496
$(0.5, 1.0)$	$(0.500000, 0.877496)$	-0.877496
$(0.5, 2.0)$	$(0.000100, 1.005037)$	-1.005037

research.

Table 3.3: Inner smoothing reformulation of the first test problem with $\tau = 0.1$

(x_1^0, x_2^0)	(x_1^*, x_2^*)	v^*
$(-1.0, -1.0)$	-	-
$(-1.0, 0.0)$	$(0.000100, 0.995038)$	-0.995038
$(-1.0, 1.0)$	-	-
$(0.0, -1.0)$	$(0.999900, 0.995038)$	-0.995038
$(0.0, 0.0)$	$(0.000100, 0.995038)$	-0.995038
$(0.0, 1.0)$	$(0.000100, 0.995038)$	-0.995038
$(1.0, -1.0)$	$(0.000100, 0.995038)$	-0.995038
$(1.0, 0.0)$	$(0.999900, 0.995038)$	-0.995038
$(1.0, 1.0)$	$(0.999900, 0.995038)$	-0.995038
$(0.5, -2.0)$	$(0.000100, 0.995038)$	-0.995038
$(0.5, -1.0)$	$(0.500000, 0.866025)$	-0.866025
$(0.5, 0.0)$	$(0.500000, 0.866025)$	-0.866025
$(0.5, 1.0)$	$(0.500000, 0.866025)$	-0.866025
$(0.5, 2.0)$	$(0.999900, 0.995038)$	-0.995038

Table 3.4: Lower level duality reformulation of the second test problem

(x_1^0, x_2^0)	(x_1^*, x_2^*)	v^*
$(-1.0, -1.0)$	$(0.000000, -0.088129)$	0.000000
$(-1.0, 0.0)$	$(0.000000, 0.000000)$	0.000000
$(-1.0, 1.0)$	$(0.000000, 0.088129)$	0.000000
$(0.0, -1.0)$	$(0.000000, 0.001944)$	0.000000
$(0.0, 0.0)$	$(0.000000, 0.000000)$	0.000000
$(0.0, 1.0)$	$(0.000000, -0.001944)$	0.000000
$(1.0, -1.0)$	$(0.000000, -0.038948)$	0.000000
$(1.0, 0.0)$	$(0.000000, 0.000000)$	0.000000
$(1.0, 1.0)$	$(0.000000, 0.038948)$	0.000000
$(2.0, -1.0)$	$(4.000000, 0.500000)$	-4.000000
$(2.0, 0.0)$	$(3.866026, 0.000001)$	-3.866026
$(2.0, 1.0)$	$(4.000000, -0.500000)$	-4.000000
$(3.0, -1.0)$	$(4.000000, 0.500000)$	-4.000000
$(3.0, 0.0)$	$(4.000000, 0.500000)$	-4.000000
$(3.0, 1.0)$	$(4.000000, -0.500000)$	-4.000000
$(4.0, -1.0)$	$(0.000000, -0.041141)$	0.000000
$(4.0, 0.0)$	$(0.000000, 0.000000)$	0.000000
$(4.0, 1.0)$	$(0.000000, 0.041143)$	0.000000

Table 3.5: Outer smoothing reformulation of the second test problem with $\tau = 10^{-4}$

(x_1^0, x_2^0)	(x_1^*, x_2^*)	v^*
$(-1.0, -1.0)$	$(0.000000, -0.240750)$	0.000000
$(-1.0, 0.0)$	$(0.000000, 0.000000)$	0.000000
$(-1.0, 1.0)$	$(0.000000, 0.240750)$	0.000000
$(0.0, -1.0)$	$(0.000000, 0.195208)$	0.000000
$(0.0, 0.0)$	$(0.000000, 0.000000)$	0.000000
$(0.0, 1.0)$	$(0.000000, -0.195208)$	0.000000
$(1.0, -1.0)$	$(0.000000, 0.020345)$	0.000000
$(1.0, 0.0)$	$(0.000000, 0.000000)$	0.000000
$(1.0, 1.0)$	$(0.000000, -0.020345)$	0.000000
$(2.0, -1.0)$	$(4.000000, -0.500000)$	-4.000000
$(2.0, 0.0)$	$(0.000000, 0.000000)$	0.000000
$(2.0, 1.0)$	$(4.000000, 0.500000)$	-4.000000
$(3.0, -1.0)$	$(4.000000, -0.500000)$	-4.000000
$(3.0, 0.0)$	$(3.866025, 0.000000)$	-3.866025
$(3.0, 1.0)$	$(4.000000, 0.500000)$	-4.000000
$(4.0, -1.0)$	$(4.000000, -0.499999)$	-4.000000
$(4.0, 0.0)$	$(3.866025, 0.000000)$	-3.866025
$(4.0, 1.0)$	$(4.000000, 0.499999)$	-4.000000

Table 3.6: Inner smoothing reformulation of the second test problem with $\tau = 10^{-4}$

(x_1^0, x_2^0)	(x_1^*, x_2^*)	v^*
$(-1.0, -1.0)$	$(0.000000, -0.240750)$	0.000000
$(-1.0, 0.0)$	$(0.000000, 0.000000)$	0.000000
$(-1.0, 1.0)$	$(0.000000, 0.240750)$	0.000000
$(0.0, -1.0)$	$(0.000000, 0.195205)$	0.000000
$(0.0, 0.0)$	$(0.000000, 0.000000)$	0.000000
$(0.0, 1.0)$	$(0.000000, -0.195205)$	0.000000
$(1.0, -1.0)$	$(0.000000, 0.020345)$	0.000000
$(1.0, 0.0)$	$(0.000000, 0.000000)$	0.000000
$(1.0, 1.0)$	$(0.000000, -0.020345)$	0.000000
$(2.0, -1.0)$	$(4.000000, -0.500000)$	-4.000000
$(2.0, 0.0)$	$(0.000000, 0.000000)$	0.000000
$(2.0, 1.0)$	$(4.000000, 0.499999)$	-4.000000
$(3.0, -1.0)$	$(4.000000, -0.500000)$	-4.000000
$(3.0, 0.0)$	$(3.866025, 0.000000)$	-3.866025
$(3.0, 1.0)$	$(4.000000, 0.500000)$	-4.000000
$(4.0, -1.0)$	$(4.000000, 0.500000)$	-4.000000
$(4.0, 0.0)$	$(3.866025, 0.000000)$	-3.866025
$(4.0, 1.0)$	$(4.000000, -0.500000)$	-4.000000

Chapter 4

Global Solution of Disjunctive Programs

In this chapter, which is based on [66], we consider the solution of *box-constrained* disjunctive programs to global optimality, that is problems where the feasible set is described by

$$M_{DP} = \{x \in B \mid \Omega(g_1(x) \leq 0, \ldots, g_p(x) \leq 0) = \text{true}\}$$

with a box $B \subseteq \mathbb{R}^n$ and a logical expression $\Omega : \{\text{true}, \text{false}\}^p \to \{\text{true}, \text{false}\}$ consisting of only conjunctions and disjunctions. As is common in global optimization, we are interested in computing an ε-global optimal point, i.e., a point $x^* \in M_{DP}$ with $f(x) \geq f(x^*) - \varepsilon$ for all $x \in M_{DP}$.

Usually all of the solution methods for DPs as well as GDPs either expect a conjunctive or a disjunctive normal form of the constraints, or they introduce a list of additional binary variables. Our approach in this thesis is to solve GDPs to global optimality without the assumption of a normal form, to avoid an exponential growth in the number of constraints during normalization, and without introducing binary variables, to avoid numerical instabilities and poor relaxations. In the following section we shall present a general branch-and-bound framework for global optimization of disjunctive programs and prove its convergence in Section 4.2. Note that the approach in Chapter 3 only provides locally optimal solutions.

This chapter is mainly based on the article [66], which in turn is based on the master thesis [95] of Fabian Rigterink who also performed the numerical computations explained in Section 4.4 as well as parts of the computational tests in Example 4.1.2. This chapter is structured as follows. In Section 4.1, our branch-and-bound framework, some assumptions, the main ideas as well

as benefits over the mixed-integer reformulation are described. The proof of convergence is given in Section 4.2. More general logical expressions that also involve negations and implications are considered in Section 4.3. We discuss some computational results in Section 4.4.

4.1 The Branch-and-Bound Framework

Our framework is based on standard branch-and-bound algorithms from global optimization, see for example [56]. The main idea is to successively construct a tessellation of the box B and to keep a list of sub-boxes that may contain globally minimal points. We start by dividing the box B into two sub-boxes X_1 and X_2 along the midpoint of a longest edge and compute lower bounds for the globally minimal value on both sub-boxes. The smaller one of these values provides a lower bound for the globally minimal value of the optimization problem. In order to compute the lower bounds, many of the well-known procedures from conjunctive global optimization are applicable. For example, we allow the use of the αBB-relaxation from [3,4,7], or centered and optimal centered forms from [17,70]. The direct application of interval arithmetic is also possible [86], in contrast to its indirect use in the aforementioned lower bounding procedures. There are also lower bounding procedures based on duality, see e.g. [35,36].

In our branch-and-bound framework, we propose how these standard approaches can be used to solve disjunctive programs. The main assumption is that the lower bounding procedure is M-independent, convergent and monotone according to the following definition taken from [64]. All of the aforementioned lower bounding procedures can be used in a straightforward manner such that this assumption is satisfied.

Definition 4.1.1.

a) A function ℓ_f from the set of all sub-boxes X of B to $\mathbb{R}$ is called M-independent lower bounding procedure *for a function f if it satisfies $\ell_f(X) \leq \min_{x \in X} f(x)$ for all sub-boxes $X \subseteq B$.*

b) A lower bounding procedure ℓ_f is called monotone *if $\ell_f(X_1) \geq \ell_f(X_2)$ holds for all boxes $X_1 \subseteq X_2 \subseteq B$.*

c) We call lower bounding procedures convergent *if*

$$\lim_{k \to \infty} \ell_f(X_k) = \lim_{k \to \infty} \min_{x \in X_k} f(x)$$

holds for any exhaustive sequence of boxes $(X_k)_{k\in\mathbb{N}}$ *(i.e., a sequence of boxes* $(X_k)_{k\in\mathbb{N}}$ *with* $X_{k+1} \subseteq X_k$*,* $k \in \mathbb{N}$*, and* $\lim_{k\to\infty} \operatorname{diag}(X_k) = 0$*) and any choice of the function* f*, respectively.*

Using this definition for a box $X \subseteq B$, with a lower bounding procedure ℓ_{g_i} we put

$$\widehat{Y}_i(X) := \begin{cases} \text{true} & \text{if } \ell_{g_i}(X) \leq 0 \\ \text{false} & \text{if } \ell_{g_i}(X) > 0 \end{cases}$$

for $i = 1, \ldots, p$. Observe that $\widehat{Y}_i$ depends on boxes whereas Y_i as defined in the introduction depends on single points $x \in X$. Furthermore, Y_i is linked to a value of the constraint g_i, whereas $\widehat{Y}_i$ is linked to a lower bound for g_i on the considered box. It will make sense to replace the terms $Y_i(x)$ by $\widehat{Y}_i(X)$ in the logical expression Ω, as we will see soon.

After the division of a box into two sub-boxes, we will check if both sub-boxes still have to be kept in the list. There are typically two reasons why boxes may be fathomed: either it can be shown that they do not contain points with sufficiently small objective function values, or it can be shown that they do not contain any feasible points.

In branch-and-bound algorithms, one of the most important questions regarding the feasible set is to decide if it is possible that a given box X contains feasible points, or if the box can be excluded from further considerations, otherwise. We propose to exclude the box X from the list if the expression

$$\Omega(\widehat{Y}_1(X), \ldots, \widehat{Y}_p(X)) \tag{4.1}$$

is false. We will show in Section 4.2 that by applying this rule no box that contains feasible points is excluded. If expression (4.1) yields true, then the box X may or may not contain feasible points. Note that the evaluation of expression (4.1) is just a function evaluation of the logical expression Ω at the vector defined by the lower bounds $\ell_{g_i}(X)$ and, as such, numerically inexpensive.

We are now ready to state our branch-and-bound framework where the main difference to standard branch-and-bound algorithms from global minimization is in Step 3. Here, $\operatorname{mid}(B)$ denotes the geometrical midpoint of the box B.

Algorithm 1: Branch-and-bound framework for disjunctive programs

Input: Tolerance $\varepsilon > 0$, initial lower bound $\widehat{v}_0 = \ell_f(B)$, initial upper bound $u_0 = +\infty$, list $\mathcal{L} = \{(B, \ell(B))\}$, currently best known point $x_0^* = \mathrm{mid}(B)$ and iteration counter $k = 1$.

Output: ε-global optimal point x^*, or certificate for $M_{DP} = \emptyset$.

while $u_{k-1} - \widehat{v}_{k-1} > \varepsilon$ *and* $\mathcal{L} \neq \emptyset$ **do**

Step 1: Choose $(X_k, v_k) \in \mathcal{L}$ with $v_k = \widehat{v}_{k-1}$;

Step 2: Divide X_k along the midpoint of a longest edge into X_k^1 and X_k^2;

Step 3: For $j = 1, 2$, if $\Omega(\widehat{Y}_1(X_k^j), \ldots, \widehat{Y}_p(X_k^j))$ is true, calculate lower bound $v_k^j = \ell_f(X_k^j)$, and add the pair (X_k^j, v_k^j) to the list $\mathcal{L}$;

Step 4: For $j = 1, 2$ choose $x_k^j \in X_k^j$ and define

$$f_k^j := \begin{cases} f(x_k^j) & \text{if } x_k^j \in M_{DP} \cap X_k^j \\ +\infty & \text{else.} \end{cases}$$

Step 5: Put $u_k = \min\{u_{k-1}, f_k^1, f_k^2\}$ and choose $x_k^* \in \{x_{k-1}^*, x_k^1, x_k^2\}$ with $f(x_k^*) = u_k$;

Step 6: Fathoming:

for $(X, v) \in \mathcal{L}$ *with* $v > u_k$ **do**

Remove (X, v) from $\mathcal{L}$;

end

Step 7: Update of lower bound:

if $\mathcal{L} \neq \emptyset$ **then**

$\widehat{v}_k = \min\{v \in \mathbb{R} \mid (X, v) \in \mathcal{L}\}$

end

Step 8: Increment k;

end

In the convergence analysis of Algorithm 1 we will have to address mainly two issues corresponding to Step 3: firstly, boxes that do contain feasible points must not be excluded from the list as long as they may contain sufficiently small objective function values. This is discussed at the beginning of the next section. Secondly, boxes that do not contain feasible points should not affect the algorithm for more than finitely many iterations. This is mainly an immediate consequence of the convergence of the lower bounding procedure as we will also show in the next section.

We emphasize that Algorithm 1 only provides a branch-and-bound *framework* which leaves much room for improvement, at least in Steps 3 and 4. However, our main convergence result in Theorem 4.2.9 below will work under

these minimal assumptions and does not rely on any sophisticated improvements. Thus, a main advantage of the framework formulated in Algorithm 1 is its high flexibility. Clearly, improvements in Step 3 and 4 may lead to a significant acceleration in the performance of the algorithm.

In fact, the lower bounds computed in Step 3 do not explicitly take the constraints into account, but only bound the objective function on the current box. The constraints are only enforced implicitly by fathoming boxes during the branch-and-bound procedure. On the other hand, this leaves room to apply special lower bounding procedures tailored to applications. Moreover, also in the mixed-integer reformulation good lower bounds are generated by branching on the binary variables.

Analogously, since our convergence result does not rely on the particular choice of the points $x_k^j \in X_k^j$ in Step 4 of Algorithm 1, we do not specify them here. Beyond just putting $x_k^j = \mathrm{mid}(X_k^j)$ or using construction heuristics, good upper bounds may result from computing these points by local NLP methods, where employing either standard conjunctive methods for fixed choices of the logical values is possible, or the local disjunctive method from Chapter 3.

We shall comment on further implementation details in Section 4.4.

Before we prove the convergence of Algorithm 1 in the next section, we illustrate the benefits of the proposed framework with the following example.

Example 4.1.2. *Consider the problem*

$$\begin{aligned} C: \qquad \max_x \quad & (1-x_1)(1-x_2) \\ s.t. \quad & x_1^2 + x_2^2 \geq 1, \\ & x \in [0,1]^2. \end{aligned} \tag{4.2}$$

The unique optimal point of C is $(x_1^*, x_2^*) = (1/\sqrt{2}, 1/\sqrt{2})$ with optimal value $f(x_1^*, x_2^*) = 3/2 - \sqrt{2}$. The feasible set M_C of C is shown in Figure 4.1.

Let us approximate the nonlinear constraint (4.2) *by p disjunctive linear constraints of the form*

$$g_i(x) = \tilde{m}_i x_1 - x_2 + \tilde{b}_i \leq 0, \quad i \in I = \{1, \ldots, p\},$$

where with the coordinates of supporting points

$$\begin{aligned} \tilde{x}_i^1 &= \cos(1/(p+1) \cdot \pi/2 \cdot i), \\ \tilde{x}_i^2 &= \sin(1/(p+1) \cdot \pi/2 \cdot i), \end{aligned}$$

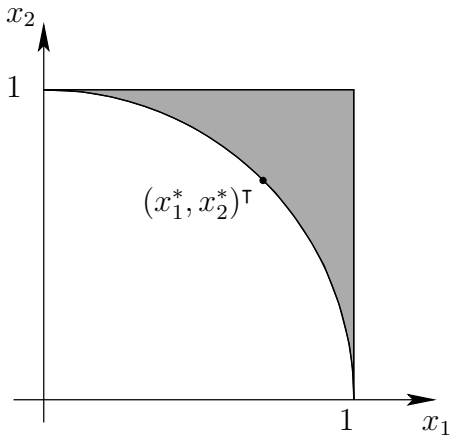

Figure 4.1: Feasible set M_C of problem C

we define $\tilde{m}_i = -\tilde{x}_i^1/\tilde{x}_i^2$ *as well as* $\tilde{b}_i = \tilde{x}_i^2 - \tilde{m}_i \cdot \tilde{x}_i^1$*. This leads to the inner approximation*

$$M_{IA(p)} = \left\{ x \in [0,1]^2 \,|\, \Omega(g_1(x) \leq 0, \ldots, g_p(x) \leq 0) = \bigvee_{i \in I} (g_i(x) \leq 0) = \text{true} \right\}$$

of M_C *for any value of* p*. Figure 4.2 illustrates this construction for the values* $p \in \{3, 9\}$*.*

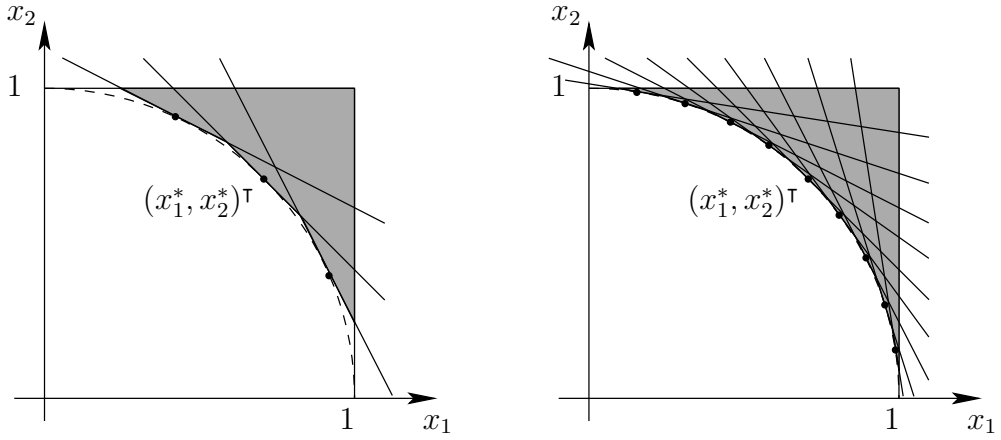

Figure 4.2: The set $M_{IA(p)}$ for $p \in \{3, 9\}$

In fact, for no odd value of p *the constraint*

$$g_{(p+1)/2}(x) = -x_1 - x_2 + \sqrt{2} \leq 0$$

cuts off (x_1^*, x_2^*). *Therefore, for odd* p *the inner-approximation problem*

$$IA(p): \qquad \max_x \quad f(x) \quad s.t. \quad x \in M_{IA(p)}$$

even has the same optimal point and optimal value as problem C.

Obviously $IA(p)$ *is a disjunctive program, and its mixed-integer reformulation is*

$$\begin{aligned} IAR(p): \qquad \max_{x,y} \quad & f(x) \\ s.t. \quad & \tilde{m}_i x_1 - x_2 + \tilde{b}_i y_i \leq 0, \qquad i \in I, \\ & \sum_{i \in I} y_i \geq 1, \\ & x \in [0,1]^2, \\ & y \in \{0,1\}^p. \end{aligned} \tag{4.3}$$

We now investigate how well $IA(p)$ *performs using Algorithm 1 in comparison to* $IAR(p)$ *using an off-the-shelf solver. More specifically, we compare*

- $IA(p)$ *using Algorithm 1 (implemented in C++),*
- $IAR(p)$ *using CPLEX (modelled in AMPL and using CPLEX' "reqconvex 3" option).*

Table 4.1 compares $IA(p)$ *using Algorithm 1 and* $IAR(p)$ *using CPLEX. We see that our implementation of Algorithm 1 clearly outperforms CPLEX. In fact, for* $p \geq 351$, *CPLEX could not solve* $IAR(p)$ *within the time limit of 3,600 seconds. However, even for* $p = 1,000,001$, *Algorithm 1 could solve* $IA(p)$ *in just over 90 seconds.*

4.2 Convergence Analysis

4.2.1 Logical Expressions and Lower Bounds

To guarantee that $\widehat{v}_k$ in iteration k of Algorithm 1 is in fact a lower bound for the globally minimal value, we have to ensure that boxes which contain feasible points are not excluded from the list in Step 3. For simple disjunctive problems this is done in the next results. The ideas are then extended to more general disjunctive problems.

Table 4.1: Comparison of $IA(p)$ using Algorithm 1 and $IAR(p)$ using CPLEX

	$IA(p)$ using Algorithm 1		$IAR(p)$ using CPLEX		
p	# Iters	Time [s]	# MIP iters	# B&B nodes	Time [s]
51	32,738	2.09	687	223	0.41
101	32,826	2.09	1,862	411	2.21
151	33,575	2.19	2,815	611	6.78
201	33,569	2.17	5,394	1,155	34.00
251	33,675	2.19	6,659	1,003	24.58
301	33,621	2.19	161,993	48,712	1,918.66
351	33,645	2.26	-	-	-
401	33,751	2.36	-	-	-
451	33,717	2.37	-	-	-
501	33,746	2.35	-	-	-
1,001	33,798	2.35	-	-	-
10,001	33,804	3.46	-	-	-
100,001	33,804	9.95	-	-	-
1,000,001	33,804	90.04	-	-	-

\- could not be solved within the time limit of 3,600 seconds

From the definition of a lower bound, for all $x \in X$ we have $\ell_{g_i}(X) \leq g_i(x)$ and thus we also have the implication

$$Y_i(x) \Longrightarrow \widehat{Y}_i(X).$$

In case of a standard nonlinear program, all constraints are understood in a conjunctive manner. Thus, if $\ell_{g_i}(X) > 0$ for at least one constraint g_i, the box cannot contain any feasible point. Otherwise, the box X may or may not contain feasible points. In terms of logic, this can be expressed as

$$\bigwedge_{i \in I} Y_i(x) \Longrightarrow \bigwedge_{i \in I} \widehat{Y}_i(X) \tag{4.4}$$

for all $x \in X$. If $g_i(x) > 0$ for some $i \in I$, then implication (4.4) is trivially true, and if $g_i(x) \leq 0$ for all $i \in I$, then the expression is true due to the fact that $\ell_{g_i}(X) \leq g_i(x)$ holds for all $x \in X$. These considerations lead to the following lemma.

Lemma 4.2.1. *Let Ω be a conjunction $\Omega(Y_1(x), \ldots, Y_p(x)) = \bigwedge_{i \in I} Y_i(x)$. Then we have $\Omega(Y_1(x), \ldots, Y_p(x)) \Longrightarrow \Omega(\widehat{Y}_1(X), \ldots, \widehat{Y}_p(X))$ for all boxes $X \subseteq B$ and for all $x \in X$.*

Let us now consider the special disjunctive problem where all constraints are understood in a disjunctive manner:

$$\min_{x\in\mathbb{R}^n} f(x) \quad \text{s.t.} \quad \bigvee_{i\in I} Y_i(x).$$

Then a given box X can be excluded from further considerations if $\ell_{g_i}(X) > 0$ for all $i = 1, \ldots, p$. Otherwise, the box X may or may not contain feasible points. We conclude similarly to the conjunctive case: if $x \in X$ is infeasible, then the implication

$$\bigvee_{i\in I} Y_i(x) \Longrightarrow \bigvee_{i\in I} \widehat{Y}_i(X) \tag{4.5}$$

is trivially true. If $x \in X$ is feasible, then $\Omega(\widehat{Y}_1(X), \ldots, \widehat{Y}_p(X))$ is true due to $\ell_{g_i}(X) \leq g_i(x) \leq 0$ for at least one $i \in \{1, \ldots, p\}$, and thus the implication (4.5) is true again. In summary we have the following result.

Lemma 4.2.2. *Let Ω be a disjunction $\Omega(Y_1(x), \ldots, Y_p(x)) = \bigvee_{i\in I} Y_i(x)$. Then we have $\Omega(Y_1(x), \ldots, Y_p(x)) \Longrightarrow \Omega(\widehat{Y}_1(X), \ldots, \widehat{Y}_p(X))$ for all boxes $X \subseteq B$ and for all $x \in X$.*

Let us now consider more general disjunctive expressions consisting of only conjunctions and disjunctions for which we derive an analogous result. We illustrate this in the following where a logical expression is given which is neither a simple conjunction or disjunction nor in any normal form. The problem considered in the example below is a slight modification of the one in Example 3.1.1 in Chapter 3.

Example 4.2.3.

$$\min_{x\in\mathbb{R}^2} f(x) := x_1 \quad s.t. \quad x \in M_{DP}$$

with

$$M_{DP} = \{x \in B \mid \Omega(Y_1(x), \ldots, Y_6(x)) = \text{true}\}$$

and

$$\Omega(Y_1, \ldots, Y_6) = \Big(\big(Y_1 \wedge Y_2\big) \vee Y_3 \vee Y_4\Big) \wedge \Big(Y_5 \vee Y_6\Big),$$

$B = [-1, 1] \times [-1, 1]$ *and some constraints g_i, $i = 1, \ldots, 6$.*

Adopting the notation from Chapter 3 and [65], for every logical expression Ω there is a corresponding expression tree T_Ω. Again, we denote its height by h_Ω. The expression tree corresponding to the logical expression in Example 4.2.3 is already shown in Figure 3.2 in the previous chapter and has height three.

For convenience we recall the most important considerations regarding expression trees of logical expressions. Due to the associativity of both conjunction and disjunction, we can assume that each node of the expression tree T_Ω either corresponds to

- a conjunction $\bigwedge_{j\in J} A_j$ where the A_j, $j \in J$, are either disjunctions or simple terms $Y_i(x)$ for some $i \in I$, or
- a disjunction $\bigvee_{j\in J} A_j$ where the A_j, $j \in J$, are either conjunctions or simple terms $Y_i(x)$ for some $i \in I$.

Conjunctions and disjunctions of length $|J|$ will be called $|J|$-conjunctions and $|J|$-disjunctions, respectively. In the following, we index the nodes V of T_Ω recursively by multi-indices. The root node will be denoted by V_1. Since the root node is either a $|J|$-conjunction or a $|J|$-disjunction, we index its child nodes with multi-indices of length two by $V_{11}, \ldots, V_{1|J|}$. Continuing this way recursively for the child nodes, we index all nodes down to the leaf level $h_\Omega + 1$, where leaf nodes are indexed with multi-indices of length $h_\Omega + 1$. The set of all indices of child nodes of a node V is denoted by $\mathcal{C}(V)$. Furthermore, all indices of nodes on level ℓ of the expression tree form the set $\mathcal{I}(\ell)$, that is, we have $\mathcal{I}(1) = \{1\}$, $\mathcal{I}(2) = \{11, \ldots, 1|J|\}$, and so on.

We will now use Lemmas 4.2.1 and 4.2.2 to show that an analogous result also holds for general logical expressions. Consider a node on level h_Ω of the tree. For some $j \in \mathcal{I}(h_\Omega)$ this is either a conjunction of $Y_i(x)$, $i \in \mathcal{C}(j)$, or a disjunction of $Y_i(x)$, $i \in \mathcal{C}(j)$. In the first case, we have for a box $X \subseteq B$ and all $x \in X$ the implication

$$\bigwedge_{i\in\mathcal{C}(j)} Y_i(x) \Longrightarrow \bigwedge_{i\in\mathcal{C}(j)} \widehat{Y}_i(X)$$

due to Lemma 4.2.1. In the second case, we have

$$\bigvee_{i\in\mathcal{C}(j)} Y_i(x) \Longrightarrow \bigvee_{i\in\mathcal{C}(j)} \widehat{Y}_i(X)$$

due to Lemma 4.2.2. In both cases we can simplify the logical expression by introducing a new symbol $Y_j(x)$, $j \notin I$, and by replacing all occurrences of $\bigwedge_{i\in\mathcal{C}(j)} Y_i(x)$ or $\bigvee_{i\in\mathcal{C}(j)} Y_i(x)$ by $Y_j(x)$ together with the additional constraint

$$Y_j(x) = \bigwedge_{i\in\mathcal{C}(j)} Y_i(x) \quad \text{or} \quad Y_j(x) = \bigvee_{i\in\mathcal{C}(j)} Y_i(x),$$

respectively. Using this reformulation, we continue to replace all disjunctions or conjunctions on level h_Ω by introducing $Y_j(x)$ for all $j \in \mathcal{C}(h_\Omega)$. Furthermore, we define

$$\widehat{Y}_j(X) := \bigwedge_{i \in \mathcal{C}(j)} \widehat{Y}_i(X) \quad \text{resp.} \quad \widehat{Y}_j(X) := \bigvee_{i \in \mathcal{C}(j)} \widehat{Y}_i(X).$$

Note that in both cases we have $Y_j(x) \Longrightarrow \widehat{Y}_j(X)$, $j \in \mathcal{C}(i)$, for all $x \in X$. For the purpose of illustration, we show this reformulation for the expression tree in Example 4.2.3.

Example 4.2.4. *We have* $\mathcal{I}(h_\Omega) = \{111\}$ *and* $\mathcal{C}(111) = \{1, 2\}$. *We introduce* $Y_{111} := Y_1(x) \wedge Y_2(x)$ *and rewrite the feasible set as*

$$\begin{aligned} M_{DP} = \Big\{ x \in B \mid\ & \text{true} = \Omega_2(Y_{111}(x), Y_1(x), \ldots, Y_6(x)) = \\ & \Big(\big(Y_{111}(x) \vee Y_3(x) \vee Y_4(x) \big) \wedge \big(Y_5(x) \vee Y_6(x) \big) \Big), \\ & Y_{111}(x) = Y_1(x) \wedge Y_2(x) \Big\}. \end{aligned}$$

Note that after this reformulation step, the height of the expression tree decreases by one and hence the height of the new logical expression tree for Ω_2 is $h_{\Omega_2} = h_\Omega - 1$. While $h_{\Omega_2} \geq 2$, we continue to replace nodes on level $h_{\Omega_2} - 1$ again by introducing new $Y_i(x)$, $i \in \mathcal{C}(j)$, for all $j \in \mathcal{I}(h_{\Omega_2})$. We will illustrate this in our running example.

Example 4.2.5. *The feasible set of the disjunctive problem from Example 4.2.3 can be rewritten as*

$$\begin{aligned} & M_{DP} = \\ & \Big\{ x \in B \mid\ \text{true} = \\ & \qquad \Omega_3(Y_{11}(x), Y_{12}(x), Y_{111}(x), Y_1(x), \ldots, Y_6(x)) = Y_{11}(x) \vee Y_{12}(x), \\ & \qquad Y_{11}(x) = Y_{111}(x) \vee Y_2(x) \vee Y_3(x), \\ & \qquad Y_{12}(x) = Y_5(x) \vee Y_6(x), \\ & \qquad Y_{111}(x) = Y_1(x) \wedge Y_2(x) \Big\}. \end{aligned}$$

In the last step of the recursion, the feasible set is

$$\begin{aligned} M_{DP} = \Big\{ x \in B \mid \text{ true} = \\ & \Omega_4(Y_1(x), Y_{11}(x), Y_{12}(x), Y_{111}(x), Y_1(x), \ldots, Y_6(x)) = Y_0(x), \\ & Y_0(x) = Y_{11}(x) \vee Y_{12}(x), \\ & Y_{11}(x) = Y_{111}(x) \vee Y_2(x) \vee Y_3(x), \\ & Y_{12}(x) = Y_5(x) \vee Y_6(x), \\ & Y_{111}(x) = Y_1(x) \wedge Y_2(x) \Big\}. \end{aligned}$$

For a box $X \subseteq B$ and for all $x \in X$, we can now conclude $Y_{111}(x) \Longrightarrow \widehat{Y}_{111}(X)$ due to the definition of Y_{111}, the definition of $\widehat{Y}_{111}$ and Lemma 4.2.1. Using this, the definition of $Y_{11}, Y_{12}, \widehat{Y}_{11}$ and $\widehat{Y}_{12}$ and Lemma 4.2.2, we obtain

$$Y_{11}(x) \Longrightarrow \widehat{Y}_{11}(X) \quad \text{and} \quad Y_{12}(x) \Longrightarrow \widehat{Y}_{12}(X).$$

With the same line of arguments we have

$$Y_0(x) \Longrightarrow \widehat{Y}_0(X)$$

and equivalently

$$\Omega(Y_1(x), \ldots, Y_6(x)) \Longrightarrow \Omega(\widehat{Y}_1(X), \ldots, \widehat{Y}_6(X)).$$

For the general case, we conclude analogously to the arguments at the end of Example 4.2.5. So, with $x \in X$, from $Y_i(x) \Longrightarrow \widehat{Y}_i(X)$, $i = 1, \ldots, p$, the implications $Y_j(x) \Longrightarrow \widehat{Y}_j(X)$, $j \in \mathcal{C}(h_\Omega)$ follow. By applying Lemmas 4.2.1 and 4.2.2 recursively, we obtain

$$\begin{aligned} & \Big(Y_i(x) \Longrightarrow \widehat{Y}_i(X),\ i \in \{1, \ldots, p\} \Big) \\ \Longrightarrow & \Big(Y_j(x) \Longrightarrow \widehat{Y}_j(X),\ j \in \mathcal{C}(h_\Omega) \Big) \end{aligned}$$

and continuing recursively yields

$$\begin{aligned} & \Big(Y_i(x) \Longrightarrow \widehat{Y}_i(X),\ i \in \{1, \ldots, p\} \cup \mathcal{C}(h_\Omega) \Big) \\ \Longrightarrow & \Big(Y_j(x) \Longrightarrow \widehat{Y}_j(X),\ j \in \mathcal{C}(h_\Omega - 1) \Big) \end{aligned}$$

and so on until we arrive at the root node of the expression tree with

$$\begin{aligned} & \Big(Y_i(x) \Longrightarrow \widehat{Y}_i(X),\ i \in \{1, \ldots, p\} \cup \mathcal{C}(h_\Omega) \cup \ldots \cup \mathcal{C}(2) \Big) \\ \Longrightarrow & \Big(Y_j(x) \Longrightarrow \widehat{Y}_j(X),\ j \in \mathcal{C}(1) \Big). \end{aligned}$$

Putting all the aforementioned implications together we have

$$\begin{aligned} &\Big(Y_i(x) \Longrightarrow \widehat{Y}_i(X),\ i = 1, \ldots, p\Big) \\ \Longrightarrow &\Big(Y_j(x) \Longrightarrow \widehat{Y}_j(X),\ j \in \mathcal{C}(1) = \{1\}\Big) \end{aligned}$$

and we arrive at the following result.

Proposition 4.2.6. *Let Ω be a general logical expression consisting of only conjunctions and disjunctions. Then we have*

$$\Omega(Y_1(x), \ldots, Y_p(x)) \Longrightarrow \Omega(\widehat{Y}_1(X), \ldots, \widehat{Y}_p(X))$$

for all boxes $X \subseteq B$ and for all $x \in X$.

In particular, this implies that a box containing feasible points is not excluded from the list in Step 3 of Algorithm 1.

4.2.2 Convergence of the Branch-and-Bound Framework

To show the convergence of Algorithm 1, we additionally have to prove that boxes which do not contain any feasible point affect the algorithm for at most a finite number of iterations. This is done in the following lemma.

Lemma 4.2.7. *Let the box $X \subseteq B \setminus M_{DP}$ be contained in the list $\mathcal{L}$. Furthermore, assume that some convergent lower bounding procedures ℓ_{g_i}, $i = 1, \ldots, p$, are used in Algorithm 1. Then in Step 1 of the algorithm a box $\overline{X} \subseteq X$ is selected at most in finitely many iterations.*

Proof. Suppose not. Then an exhaustive sequence of boxes is created by the algorithm, i.e., we have a sequence of boxes with X_{k_j}, $j \in \mathbb{N}$, $X_{k_{j+1}} \subseteq X_{k_j} \subseteq X$, $j \in \mathbb{N}$, and $\lim_{j\to\infty} \operatorname{diag}(X_{k_j}) = 0$. Because of the assumption none of these boxes contain any feasible point. From property c) in Definition 4.1.1, we know that for j sufficiently large we have

$$g_i(x) > 0 \iff \ell_{g_i}(X_{k_j}) > 0$$

for all $x \in X_{k_j}$ and $i = 1, \ldots, p$. Then for all $x \in X_{k_j}$ and $i = 1, \ldots, p$ we have $Y(x) \iff \widehat{Y}_i(X_{k_j})$ and hence

$$\Omega(Y_1(x), \ldots, Y_p(x)) \iff \Omega(\widehat{Y}_1(X_{k_j}), \ldots, \widehat{Y}_p(X_{k_j})).$$

As x cannot be feasible, $\Omega(Y_1(x), \ldots, Y_p(x))$ is false, and therefore

$$\Omega(\widehat{Y}_1(X_{k_j}), \ldots, \widehat{Y}_p(X_{k_j})) = \text{false}\,.$$

However, boxes with this property are excluded in Step 3, and thus we have derived a contradiction. □

Using the previous lemma we can show the convergence of the lower bounds in the branch-and-bound algorithm.

Lemma 4.2.8. *Assume that in Algorithm 1 some convergent and monotone lower bounding procedures ℓ_f and ℓ_{g_i}, $i = 1, \ldots, p$, are used. If the infinite branch-and-bound procedure corresponding to $\varepsilon = 0$ does not terminate, then the feasible set M_{DP} is nonempty and we have $\lim_{k\to\infty} \widehat{v}_k = v^*$ where v^* denotes the globally minimal value of DP.*

Proof. First observe that, by the assumption, an exhaustive sequence of boxes X_{k_j}, $j \in \mathbb{N}$, is generated by the algorithm. Consider the unique element $\overline{x}$ which is contained in all sets X_{k_j}, $j \in \mathbb{N}$. If $\overline{x}$ was infeasible then we had

$$\text{false} = \Omega(Y_1(\overline{x}), \ldots, Y_p(\overline{x})) = \Omega(\widehat{Y}_1(X_{k_j}), \ldots, \widehat{Y}_p(X_{k_j}))$$

for all sufficiently large j due to the same line of arguments as in the proof of Lemma 4.2.7. Thus X_{k_j} does not contain any feasible point and, by Lemma 4.2.7, does not affect the algorithm for all sufficiently large $j \in \mathbb{N}$. This, however, contradicts the construction of the boxes X_{k_j}, $j \in \mathbb{N}$. Hence $\overline{x}$ is feasible and, as asserted, the set M_{DP} is nonempty.

Next, by the monotonicity of ℓ_f, the sequence $(\widehat{v}_k)_{k\in\mathbb{N}}$, is monotonically increasing, bounded above by the finite value v^* and thus convergent. Its limit is then also bounded above by v^*, so that the inequality $\lim_{k\to\infty} \widehat{v}_k \geq v^*$ remains to be shown. In fact, as the subsequence $(\ell_f(X_{k_j}))_{j\in\mathbb{N}}$ of $(\widehat{v}_k)_{k\in\mathbb{N}}$ inherits its limit, the convergence of the lower bounding procedure ℓ_f and the feasibility of $\overline{x}$ lead to

$$\lim_{k\to\infty} \widehat{v}_k = \lim_{j\to\infty} \ell_f(X_{k_j}) = \lim_{j\to\infty} \min_{x\in X_{k_j}} f(x) = f(\overline{x}) \geq v^*.$$

□

After this analysis of an approximation procedure for the optimal value, we state an approximation procedure for an optimal point. It is shown along the same lines as [64, Proposition 4.4].

Theorem 4.2.9. *Assume that in Algorithm 1 some convergent and monotone lower bounding procedures ℓ_f and ℓ_{g_i}, $i = 1, \ldots, p$, are used, and that the infinite branch-and-bound procedure does not terminate. Furthermore, let $(X_k)_{k\in\mathbb{N}}$ be a subsequence of boxes chosen in Step 2, and let $(x_k)_{k\in\mathbb{N}}$ be a sequence of points with $x_k \in X_k$, $k \in \mathbb{N}$. Then $(x_k)_{k\in\mathbb{N}}$ possesses a cluster point, and any such cluster point is a globally minimal point of DP.*

By Theorem 4.2.9, Algorithm 1 with convergent and monotone lower bounding procedures ℓ_f and ℓ_{g_i}, $i = 1, \ldots, p$, terminates after finitely many iterations for any prescribed tolerance $\varepsilon > 0$. Note that Lemma 4.2.7 with $X = B$ implies that an empty feasible set M_{DP} is detected after finitely many steps. In this case, the algorithm terminates with $\mathcal{L} = \emptyset$ and $u_{k-1} = +\infty$.

4.3 Integration of Negations into the Logical Expression

In this section an enhancement for disjunctive programs whose logical expressions also involve negations is considered. Using this we can also handle implications like $Y_1(x) \Longrightarrow Y_2(x)$ via the equivalent reformulation $\neg Y_1(x) \vee Y_2(x)$.

Formally, negations $\neg(g_i(x) \leq 0)$ can be included by considering $g_i(x) > 0$ instead. We define additional symbols

$$Y_{p+i}(x) := \begin{cases} \text{true} & \text{if } g_i(x) > 0 \\ \text{false} & \text{if } g_i(x) \leq 0 \end{cases}, \ i \in I,$$

and replace all occurrences of $\neg Y_i(x)$ in Ω by $Y_{p+i}(x)$ for $i \in I$. The resulting logical expression is denoted by $\widehat{\Omega}(Y_1(x), \ldots, Y_{2p}(x))$ and the feasible set can be written as

$$M_{DP} = \left\{ x \in B \mid \widehat{\Omega}(Y_1(x), \ldots, Y_{2p}(x)) = \text{true} \right\}.$$

Unfortunately, while this feasible set is bounded, due to the strict inequalities it may not be closed. Thus, solvability is no longer guaranteed and numerical difficulties may occur. In the following, we will present two remedies for this problem, namely outer and inner approximation of the feasible set.

4.3.1 Outer Approximation

This approach relaxes the feasible set by considering $g_i(x) \geq 0$ instead of $\neg(g_i(x) \leq 0)$. We define additional symbols

$$\widehat{Y}_{p+i}(x) := \begin{cases} \text{true} & \text{if } g_i(x) \geq 0 \\ \text{false} & \text{if } g_i(x) < 0 \end{cases}, \ i \in I,$$

and the feasible set is now

$$\widehat{M}_{DP} := \{x \in B \mid \widehat{\Omega}(Y_1(x), \ldots, Y_p(x), \widehat{Y}_{p+1}(x), \ldots, \widehat{Y}_{2p}(x)) = \text{true}\}.$$

Observe that the closed set $\widehat{M}_{DP}$ is a relaxation of the possibly open set M_{DP} due to Proposition 4.2.6.

If the enlargement of the original feasible set is small in some sense, we may have

$$\inf_{x \in M_{DP}} f(x) = \min_{x \in \widehat{M}_{DP}} f(x).$$

Unfortunately, this does not hold in general, as we will illustrate in Example 4.3.1.

Example 4.3.1. *Consider the following disjunctive problem that is defined by*

$$\min_{x \in \mathbb{R}} f(x) = -x \quad s.t. \quad x \in M_{DP}$$

together with the constraint function

$$g_1(x) := -x(x-1)^2$$

and the logical expression $\Omega(Y_1(x)) := \neg Y_1(x)$ *so that we have the feasible set*

$$\begin{aligned} M_{DP} &= \{x \in \mathbb{R} \mid \Omega(Y_1(x)) = \neg Y_1(x) = \text{true}\} \\ &= \{x \in \mathbb{R} \mid x(x-1)^2 < 0\} \ = \ \{x \in \mathbb{R} \mid x < 0\}. \end{aligned}$$

It is easy to see that we have $\inf_{x \in M_{DP}} f(x) = 0$. *The relaxed problem can be written as*

$$\widehat{DP}: \quad \min_{x \in \mathbb{R}} f(x) = -x \quad s.t. \quad x(x-1)^2 \leq 0$$

and here, in contrast to the original problem, not only the origin but also the point $\widehat{x}^* = 1$ *becomes feasible. In fact, we have* $\min_{x \in \widehat{M}_{DP}} f(x) = -1$ *and thus the relaxed problem cannot be seen as a good approximation of the original one.*

Unfortunately, after solving $\widehat{DP}$ we cannot estimate errors as they appear in Example 4.3.1. For that reason we will introduce a constraint qualification for relaxed (i.e., "negation-free") DPs which ensures that the enlargement of the feasible set is not too big. Observe that the optimal point of problem $\widehat{DP}$ in Example 4.3.1 is degenerated since the gradient of the constraint vanishes at the optimal point. We say that a point $\overline{x}$ is not degenerated if the following constraint qualification holds.

Definition 4.3.2. *Let DP be a disjunctive optimization problem whose logical expression Ω only contains conjunctions and disjunctions. Then, at $\overline{x} \in M_{DP}$ the* Mangasarian-Fromovitz constraint qualification for disjunctive programs (MFCQDP) *is said to hold if and only if there exists a set of indices $\overline{I} \subseteq I$ such that*

$$y_i := \begin{cases} \text{true} & \text{if } i \in \overline{I} \\ \text{false} & \text{if } i \notin \overline{I} \end{cases}$$

yields $\Omega(y_1, \ldots, y_p) = \text{true}$ as well as $y_i \Longrightarrow Y_i(\overline{x})$, $i \in \overline{I}$, and there is a direction $d \in \mathbb{R}^n$ such that $d^\intercal \nabla g_i(\overline{x}) < 0$, $i \in \overline{I}_0(\overline{x})$, holds where $\overline{I}_0(\overline{x}) = \{i \in \overline{I} \mid g_i(\overline{x}) = 0\}$ is the set of active indices.

Remark 4.3.3. *Note that in case of a standard conjunctive nonlinear problem, the choice $\overline{I} = \{1, \ldots, p\}$ is mandatory and thus the definition of MFCQDP coincides with the definition of the standard MFCQ from nonlinear programming for inequality constrained problems.*

To illustrate that MFCQDP is a natural extension of the standard MFCQ for our purposes, consider the following example.

Example 4.3.4. *Consider the feasible set $M_{DP} \subseteq \mathbb{R}^2$ defined by the logical expression $\Omega(Y_1(x), Y_2(x)) = Y_1(x) \vee Y_2(x)$ and the constraints*

$$\begin{aligned} g_1(x) &:= x_1^2 + x_2^2 - 1, \\ g_2(x) &:= (x_1 - 1)^2. \end{aligned}$$

The feasible set is described by $M_{DP} = M_1 \cup M_2$ where $M_1 = \{x \in \mathbb{R}^2 \mid g_1(x) \leq 0\}$ is the unit disk and $M_2 = \{x \in \mathbb{R}^2 \mid g_2(x) \leq 0\}$ contains all points from $\mathbb{R}^2$ with $x_1 = 1$.

The standard MFCQ from conjunctive programming is satisfied at every point in M_1, but violated at every point in M_2. Then MFCQDP should also be satisfied at every point $x \in M_1 \setminus M_2$ and violated at every point $x \in M_2 \setminus M_1$. This can easily be checked by choosing $\overline{I} = \{1\}$ resp. $\overline{I} = \{2\}$. The most

interesting case is the point $\overline{x} = (0,1)^\intercal \in M_1 \cap M_2$. *As there is a direction* $d \in \mathbb{R}^2$ *pointing into the interior of the feasible set, or more precisely, into the part of* M_{DP} *that is described by* M_1, *we would naturally expect MFCQDP to hold at* $\overline{x}$. *This turns out to be true since we may ignore the malign activity of* g_2 *at* $\overline{x}$ *by choosing* $\overline{I} = \{1\}$, *which results in the active index set* $\overline{I}_0(\overline{x}) = \{1\}$.

In fact, by assuming MFCQDP we can show the equality of the infimum of the original problem DP and the minimal value of the relaxed problem $\widehat{DP}$. In the following, for better readability it is convenient to have a definition of functions g_i, $i = p+1, \ldots, 2p$, and so we put $g_{p+i}(x) := -g_i(x)$, $i \in I$.

Proposition 4.3.5. *If MFCQDP holds at some globally minimal point of* $\widehat{DP}$, *then* M_{DP} *is nonempty, and we have*

$$\inf_{x \in M_{DP}} f(x) = \min_{x \in M_{\widehat{DP}}} f(x).$$

Proof. Due to the inclusion $M_{DP} \subseteq M_{\widehat{DP}}$ we have

$$\inf_{x \in M_{DP}} f(x) \geq \min_{x \in M_{\widehat{DP}}} f(x).$$

We now show that the strict inequality cannot hold.

Let $\widehat{x}^*$ be a globally minimal point of $\widehat{DP}$ at which MFCQDP holds in $\widehat{M}_{DP}$. By Definition 4.3.2, there are y_i, $i = 1, \ldots, 2p$, together with a corresponding set $\overline{I}$ such that $\widehat{\Omega}(y_1, \ldots, y_{2p}) = \text{true}$ and $y_i \Longrightarrow Y_i(\widehat{x}^*)$, $i \in \overline{I}$, as well as a direction $d \in \mathbb{R}^n$ with $d^\intercal \nabla g_i(\widehat{x}^*) < 0$, $i \in \overline{I}_0(\widehat{x}^*)$.

By a Taylor expansion (as is common in conjunctive nonlinear programming under MFCQ), for all $i \in \overline{I}_0(\widehat{x}^*)$ and all sufficiently small $\lambda > 0$ we obtain $g_i(\widehat{x}^* + \lambda d) < 0$. Furthermore, for all $i \in \overline{I} \setminus \overline{I}_0(\widehat{x}^*)$ we have $g_i(\widehat{x}^*) < 0$ and thus $g_i(\widehat{x}^* + \lambda d) < 0$ due to continuity arguments. In terms of logical expressions, for all sufficiently small $\lambda > 0$ and all $i \in \overline{I}$ we have shown $Y_i(\widehat{x}^* + \lambda d) = \text{true}$ and hence

$$y_i \Longrightarrow Y_i(\widehat{x}^* + \lambda d)$$

for these i and λ. Due to $y_i = \text{false}$, $i \in \{1, \ldots, 2p\} \setminus \overline{I}$, the latter trivially holds also for all $i \in \{1, \ldots, 2p\} \setminus \overline{I}$, and with Proposition 4.2.6 we arrive at

$$\text{true} = \widehat{\Omega}(y_1, \ldots, y_{2p}) \Longrightarrow \widehat{\Omega}(Y_1(\widehat{x}^* + \lambda d), \ldots, Y_{2p}(\widehat{x}^* + \lambda d)).$$

This shows that $\widehat{x}^* + \lambda d \in M_{DP}$ for sufficiently small $\lambda > 0$ and, in particular, that M_{DP} is nonempty. Moreover, due to $\inf_{x \in M_{DP}} f(x) \leq f(\widehat{x}^* + \lambda d)$, the continuity of f for $\lambda \searrow 0$ yields $\inf_{x \in M_{DP}} f(x) \leq f(\widehat{x}^*) = \min_{x \in M_{\widehat{DP}}} f(x)$, as desired. □

Remark 4.3.6. *We conjecture that the assumption of MFCQDP at all feasible points of some DP is mild in the following sense: in standard conjunctive optimization, MFCQ is weaker than the so-called linear independence constraint qualification (LICQ), and in [59] it is shown that generically even LICQ holds at every feasible point. Since the feasible set of DP may be written as the union of finitely many sets with conjunctive description (via the disjunctive normal form), this genericity result should hold for DPs as well. A detailed analysis, however, is beyond the scope of this thesis.*

4.3.2 Inner Approximation

While the relaxation of the feasible set yields an outer approximation method, we now propose an idea which will result in an inner approximation of the feasible set. To this end, with some parameter $\delta > 0$ we put

$$Y_{p+i}^{\delta}(x) := \begin{cases} \text{true} & \text{if } g_i(x) \geq \delta \\ \text{false} & \text{if } g_i(x) < \delta \end{cases}$$

and with

$$M_{DP}^{\delta} := \{x \in B \mid \widehat{\Omega}(Y_1(x), \ldots, Y_p(x), Y_{p+1}^{\delta}(x), \ldots, Y_{2p}^{\delta}(x)) = \text{true}\}$$

we obtain another closed set that can be considered instead of the original one. Observe that for all sub-boxes $X \subseteq B$ and $\delta > 0$ we have the implications

$$\begin{aligned} & \widehat{\Omega}(Y_1(X), \ldots, Y_p(X), Y_{p+1}^{\delta}(X), \ldots, Y_{2p}^{\delta}(X)) \\ \Longrightarrow\ & \widehat{\Omega}(Y_1(X), \ldots, Y_p(X), Y_{p+1}(X), \ldots, Y_{2p}(X)) \quad (\text{i.e., } M_{DP}^{\delta} \subseteq M_{DP}) \\ \Longrightarrow\ & \widehat{\Omega}(Y_1(X), \ldots, Y_p(X), \widehat{Y}_{p+1}(X), \ldots, \widehat{Y}_{2p}(X)) \quad (\text{i.e., } M_{DP} \subseteq \widehat{M}_{DP}), \end{aligned}$$

and thus the chain of inclusions $M_{DP}^{\delta} \subseteq M_{DP} \subseteq \widehat{M}_{DP}$ holds. In particular, for any $\delta > 0$ the set M_{DP}^{δ} is compact, but possibly empty. The latter sandwiching result implies that for any $\delta > 0$, lower and upper bounds for the infimum of DP are

$$\min_{x \in M_{\widehat{DP}}} f(x) \leq \inf_{x \in M_{DP}} f(x) \leq \inf_{x \in M_{DP}^{\delta}} f(x).$$

Here, the lower bound is tight under MFCQDP at some optimal point of $\widehat{DP}$ (cf. Proposition 4.3.5) and the infimum in the upper bound is attained if M_{DP}^{δ} is nonempty. In the following, we will show that for sufficiently small $\delta > 0$ this is the case, and that for $\delta \searrow 0$ and without the assumption of any constraint qualification, the upper bound also becomes arbitrarily tight.

Proposition 4.3.7. *If M_{DP} is nonempty, then the identity*

$$\inf_{x \in M_{DP}} f(x) = \lim_{\delta \searrow 0} \min_{x \in M_{DP}^{\delta}} f(x)$$

holds.

Proof. For $\delta \searrow 0$ the values $\inf_{x \in M_{DP}^{\delta}} f(x)$ are monotonically decreasing, and due to $M_{DP}^{\delta} \subseteq M_{DP}$ for all $\delta > 0$, they are also bounded below by $\inf_{x \in M_{DP}} f(x)$. Hence, the asserted limit exists and satisfies

$$\inf_{x \in M_{DP}} f(x) \leq \lim_{\delta \searrow 0} \inf_{x \in M_{DP}^{\delta}} f(x).$$

To see the reverse inequality, for arbitrary $\varepsilon > 0$ we choose some ε-global optimal point x_{ε}^{*} of DP and put $\delta_{\varepsilon}^{*} = \min_{i \in I}\{g_i(x_{\varepsilon}^{*}) \mid Y_{p+i}(x_{\varepsilon}^{*}) = \text{true}\}$. Then

$$Y_i(x_{\varepsilon}^{*}) \Longrightarrow Y_i^{\delta_{\varepsilon}^{*}}(x_{\varepsilon}^{*})$$

for all $i = p+1, \ldots, 2p$, and thus with Proposition 4.2.6 it is ensured that x_{ε}^{*} lies in $M_{DP}^{\delta_{\varepsilon}^{*}}$. This also implies $x_{\varepsilon}^{*} \in M_{DP}^{\delta}$ for any $0 < \delta < \delta_{\varepsilon}^{*}$ and hence

$$\inf_{x \in M_{DP}} f(x) + \varepsilon \geq f(x_{\varepsilon}^{*}) \geq \min_{x \in M_{DP}^{\delta}} f(x)$$

for all such δ. In particular, the sets M_{DP}^{δ} are nonempty for all sufficiently small $\delta > 0$ so that the infima of f over these sets are attained. Taking the limit in this inequality for $\delta \searrow 0$ yields

$$\inf_{x \in M_{DP}} f(x) + \varepsilon \geq \lim_{\delta \searrow 0} \min_{x \in M_{DP}^{\delta}} f(x)$$

and, as $\varepsilon > 0$ was chosen arbitrarily, the assertion. □

Note that while this inner approximation procedure does not rely on a constraint qualification, numerically one may only use small positive parameters $\delta > 0$ instead of taking the limit, so that the optimal value is only approximated rather than computed exactly. Furthermore, it may not always be clear how small to choose $\delta > 0$ for M_{DP}^{δ} to be nonempty.

4.4 Computational Results

A detailed computational study of Algorithm 1 including running times for different test problems is performed in [66]. In this section, we briefly discuss the computational results presented in this article. The method was implemented by Fabian Rigterink as part of his master thesis [95].

We emphasize, that the implementation from [66] is still preliminary and, moreover, is not meant to be competitive to other approaches like solving the mixed-integer reformulation with additional binary variables by Baron and CPLEX. In fact, in Step 3 lower bounds are computed using a simple αBB implementation which, for example, does not exploit convex hulls of special functions. Moreover, in Step 4 $x_k^j = \hat{x}_k^j$ for $j = 1, 2$ is chosen where $\hat{x}_k^j$ is the minimizer corresponding to the lower bound computed in Step 3. Already in these lower and upper bounding procedures, there is much room for improvement, as discussed earlier at the end of Section 4.1. Furthermore, we neither use presolves nor bound tightening and the implementation is in Matlab with a Java interface for the evaluation of logical expressions. Matlab's "fmincon" routine is used to determine the αBB lower bounds (see Section 4.4.2 for details).

4.4.1 Instances

In [66] Algorithm 1 is tested on a total of 14 problem instances from the literature [37,47,49,72,73,99,100,103]. Most of these instances are illustrative examples (some test problems from [47,49,72,73,99,100]). The remaining instances have applications in industrial and chemical engineering, for example

- optimal positioning of a new product in a multiattribute space from [37] – a convex problem,
- a job shop scheduling problem from [49] – a linear problem,
- strip-packing problems from [49,103] – also linear problems.

The problem instances differ significantly in size: from illustrative examples with only two variables and three constraints (taken from [99]) to much more challenging instances. Some of the instances' logical expressions involve negations, for which, however, the MFCQDP holds. Negations are handled as explained in Section 4.3.1.

Note that the test problems also vary in the type of problem. While many instances only have linear or convex constraints, there are also several instances that involve nonconvexities. While there are certainly better suited algorithms to solve the linear and convex instances, in [66] these test problems are solved in order to demonstrate the versatility of the proposed method.

4.4.2 Computational Results

It is compared how Algorithm 1 performs on the original logical expression Ω versus how the algorithm performs if Ω is required to be in disjunctive or conjunctive normal form (DNF or CNF). One method which, given a logical expression Ω, finds an equivalent logical expression Ω' in minimal DNF or CNF, is the Quine-McCluskey algorithm that is also called method of prime implicants and is used in [66] to transform the logical expressions into normalform. The time limit for converting Ω to an equivalent Ω' in minimal DNF or CNF is set to two hours.

Moreover, Algorithm 1 is implemented in Matlab (release R2011b, version 7.13.0.564). The tolerance for the stopping criterion is set to $\varepsilon = 0.001$. The lower bounds in Step 3 of the algorithm computed using the αBB-relaxation from [3,4,7] and "fmincon" from the Matlab Optimization Toolbox. The INTLAB Toolbox (version 6) is used for interval arithmetic. Again, the time limit for the overall algorithm is set to two hours.

It is worth noting that for some instances from [37,49,103], Ω could not be converted to a minimal normal form within the time limit. Those are the more challenging instances.

It turns out that running the algorithm on the original logical expression without requiring any normal form is computationally advantageous. If the "wrong" minimal normal form is chosen, the algorithm performs much worse. In general, it can be concluded that the less literals, the better the algorithm's performance (see [66]).

Unfortunately, many of the instances from the literature are already in some minimal normal form, and only choosing the other minimal normal form demonstrates how poorly classical GDP algorithms can perform if one requires the conversion of Ω to a specific normal form. However, one can easily think of applications with nested logical expressions in which both, the minimal disjunctive and conjunctive normal form, will perform poorly.

Chapter 5

Global Solution of GSIPs

In this chapter, which is based on [67], we consider the global minimization of *box-constrained* generalized semi-infinite programs, that is, problems of the type

$$GSIP: \qquad \min_{x\in\mathbb{R}^n} f(x) \qquad \text{s.t.} \qquad x \in M$$

where M is defined by infinitely many constraints

$$M = \left\{ x \in \overline{X} \;\middle|\; g(x,y) \leq 0 \text{ for } y \in Y(x) \right\}$$

with

$$Y(x) = \left\{ y \in \overline{Y} \;\middle|\; v_i(x,y) \geq 0 \text{ for } i \in I \right\}$$

and boxes $\overline{X} \subset \mathbb{R}^n$ and $\overline{Y} \subset \mathbb{R}^m$. All defining functions of *GSIP* are assumed to be at least continuous on their respective domains.

As already mentioned in the introduction of this thesis, a well-known numerical approach for dealing with standard semi-infinite programs is to consider a relaxed problem by choosing a finite subset of the infinitely many constraints. The resulting problem can be solved by applying standard NLP techniques. Then for the computed optimal point $\overline{x}$ the feasibility problem

$$Q(\overline{x}): \qquad \max_{y\in\mathbb{R}^m} g(\overline{x},y) \qquad \text{s.t.} \qquad v_i(y) \geq 0,\ i \in I,\ y \in \overline{Y}$$

can be solved in order to obtain an index $\overline{y}$ of a constraint that cuts off $\overline{x}$. Adding the corresponding restriction $g(x,\overline{y}) \leq 0$ to the relaxed problems yields a new optimization problem that provides a tighter approximation of the original problem.

Unfortunately, this approach cannot be applied to GSIPs without further considerations. Nevertheless, in this chapter we propose different possibilities

to achieve this by using certain reformulations and develop a tailored branch-and-bound algorithm together with appropriate lower bounding procedures to solve these problems.

This chapter is based on the article [67] and is organized as follows. In the next section we illustrate the difficulties that occur when trying to solve GSIPs by the aforementioned discretization approach in a straightforward manner and sketch our main idea to circumvent this issue by using disjunctive problems in conjunctive normalform. In Section 5.2 we describe some lower bounding procedures for solving this special kind of disjunctive optimization problems. In Section 5.3 these will be used to state a branch-and-bound algorithm for disjunctive programs which in turn will be applied to solve the subproblems that arise in the discretization method for GSIPs. In Section 5.4 we give computational results as a proof of concept.

5.1 Discretizations of GSIPs

Unfortunately, the discretization method sketched in the introduction cannot be applied to GSIPs without further considerations as already mentioned. This can be seen from the test problems in [75,83] and is illustrated in the following example.

Example 5.1.1. *The following test instance is taken from [75] (see also [83]). Let us consider the problem*

$$GSIP: \qquad \min_{x\in\mathbb{R}^2} f(x) = \left(x_1 - \frac{1}{4}\right)^2 + x_2^2 \qquad s.t. \qquad x \in M$$

where M is defined as

$$M = \left\{ x \in [-1,1]^2 \;\middle|\; g(x,y) = y + x_2 \leq 0 \text{ for } y \in Y(x) \right\}$$

with

$$Y(x) = \left\{ y \in [-1,1] \;\middle|\; v(x,y) = -y^2 + x_1 \geq 0 \right\}.$$

It is not hard to see that the unique optimal point of GSIP is $x^* = (0,0)^\intercal$.

Assume that we want to solve this problem by a simple discretization approach for standard semi-infinite programs. We might start by solving

$$P: \qquad \min_{x\in\mathbb{R}^2} f(x) = \left(x_1 - \frac{1}{4}\right)^2 + x_2^2 \qquad s.t. \qquad x \in [-1,1]^2$$

and obtain $x^* = (\frac{1}{4}, 0)^\intercal$. *Inserting* x^* *into the feasibility problem yields*

$$Q(x^*): \quad \max_{y \in \mathbb{R}} y \qquad s.t. \qquad v(x^*, y) = -y^2 + \frac{1}{4} \geq 0, \ y \in [-1, 1]$$

with the unique optimal point $y^* = \frac{1}{2}$. *Unfortunately, adding the corresponding constraint* $g(x, \frac{1}{2}) = \frac{1}{2} + x_2 \leq 0$ *cuts off the optimal point* $x^* = (0, 0)^\intercal$ *of GSIP.*

However, as introduced in [52] and already described in the introduction, a generalized semi-infinite program can be rewritten as

$$GSIP: \quad \min_{x \in \mathbb{R}^n} f(x) \qquad \text{s.t.} \qquad x \in M$$

with

$$M = \left\{ x \in \overline{X} \;\middle|\; \Big(g(x, y) \leq 0 \Big) \vee \bigvee_{i \in I} \Big(v_i(x, y) < 0 \Big) \quad \forall y \in \overline{Y} \right\}. \tag{5.1}$$

In equation (5.1) the index set $\overline{Y}$ does not depend on x, although still being infinite. For that reason we can consider this problem as a kind of disjunctive standard semi-infinite program.

The aggregation of the single inequality constraints into one constraint function seems nontrivial, since one of them is nonstrict while the others are strict. However, with the definition $v_0(x, y) = g(x, y)$ and $I_0 = I \cup \{0\}$ at least the relaxation

$$\widehat{M} = \left\{ x \in \overline{X} \;\middle|\; \bigvee_{i \in I_0} \Big(v_i(x, y) \leq 0 \Big) \quad \forall y \in \overline{Y} \right\}$$

of the feasible set M of *GSIP* can be written in the aggregated form

$$\widehat{M} = \left\{ x \in \overline{X} \;\middle|\; \Big(\min_{i \in I_0} v_i(x, y) \Big) \leq 0 \quad \forall y \in \overline{Y} \right\}.$$

Note that $\widehat{M}$ has the format of a standard semi-infinite feasible set with the constraint function $G(x, y) = \min_{i \in I_0} v_i(x, y)$. Due to the relaxation property, the minimization of f over $\widehat{M}$ will generate a lower bound for the optimal value of *GSIP*.

In [51,52] it is shown that generically $\widehat{M}$ is not much larger than the feasible set of *GSIP*, but just describes the topological closure of the, in

general, nonclosed set M. In this case the minimal value of f over $\widehat{M}$ actually coincides with the minimal value of *GSIP* and, from the numerical point of view, it thus makes sense to use the solution of the relaxation as the appropriate solution concept for *GSIP*.

Let us stress, however, that the relaxation $\widehat{M}$ of M may be much coarser than the topological closure of M, if the genericity assumptions from [51,52] are not met. This happens, for example, for the (degenerate) reformulation of an equality constraint $u(x,y) = 0$ in the description of the index set $Y(x)$ by two inequalities $\pm u(x,y) \geq 0$. On the other hand, even in such degenerate cases, the minimization of f over $\widehat{M}$ provides a lower bound for the minimal value of *GSIP*.

As consequences, firstly the minimization of f over $\widehat{M}$ motivates another feasibility problem for *GSIP*, whose own feasible set no longer depends on x, namely

$$\widehat{Q}(x): \qquad \max_{y \in \mathbb{R}^m} \min_{i \in I_0} v_i(x,y) \quad \text{s.t.} \quad y \in \overline{Y}.$$

Secondly, the application of the usual discretization approach for standard semi-infinite programs to the above description of $\widehat{M}$, that is, replacing the infinite set $\overline{Y}$ by a finite subset $Y \subset \overline{Y}$, yields a further relaxation of *GSIP* by the problem

$$DP(Y): \qquad \min_{x \in \mathbb{R}^n} f(x) \qquad \text{s.t.} \qquad x \in M_{DP}(Y)$$

with

$$M_{DP}(Y) = \left\{ x \in \overline{X} \;\middle|\; \bigwedge_{y \in Y} \bigvee_{i \in I_0} \Big(v_i(x,y) \leq 0 \Big) \right\}.$$

This disjunctive optimization problem is in conjunctive normal form. The main idea of our approach in the present chapter is to determine a lower bound at the globally optimal value of *GSIP* by solving an iterated sequence of such disjunctive optimization problems and feasibility problems.

Our focus is on the development of efficient branch-and-bound algorithms tailored to the occurring disjunctive problems $DP(Y)$. This complements the related approach in [83] where also f is minimized over the relaxed feasible set $\widehat{M}$, but auxiliary integer variables are introduced to reformulate the problems $DP(Y)$ as mixed integer NLPs. For the latter problems [83] computes upper and lower bounds by state-of-the-art solvers in order to take advantage of their sophisticated implementation. Our approaches, on the other hand, neither necessarily need to increase the problem dimension nor leave the setting of purely continuous optimization.

5.2 Lower Bounding Procedures for DPs

In this section we propose some lower bounding procedures for disjunctive optimization problems $DP(Y)$ that arise during the solution process of our branch-and-bound algorithm. Although the following considerations will be applicable to other kinds of disjunctive optimization problems, we will restrict the descriptions to problems in so-called conjunctive normal form for simplicity.

As seen in Section 5.1 of this chapter, the appearing disjunctive programs are of the form

$$DP(Y): \qquad \min_{x\in\mathbb{R}^n} f(x) \qquad \text{s.t.} \qquad \bigwedge_{y\in Y}\bigvee_{i\in I_0}\Big(v_i(x,y)\leq 0\Big),\ x\in\overline{X}.$$

They can be solved to global optimality within a branch-and-bound framework. To achieve this, for a given box $X\subset\overline{X}$ lower bounds at the globally minimal value of $DP(Y)$ restricted to the box X have to be computed. Therefore, we define the problem

$$DP(X,Y): \qquad \min_{x\in\mathbb{R}^n} f(x) \qquad \text{s.t.} \qquad x\in M_{DP}(X,Y)$$

where we put

$$M_{DP}(X,Y):=\left\{x\in X\;\middle|\;\bigwedge_{y\in Y}\bigvee_{i\in I_0}\Big(v_i(x,y)\leq 0\Big)\right\}.$$

There are different approaches to compute lower bounds for problems of the type $DP(X,Y)$. In this chapter we shall examine three different possibilities. The first one is based on the previous chapter and allows to take advantage of existing lower bounding procedures of standard conjunctive global optimization. The main ideas are sketched in Section 5.2.1. Secondly, in Section 5.2.2 we use standard ideas to further relax the problem $DP(X,Y)$ to a linear disjunctive problem $LDP(X,Y)$. We study two alternatives for its algorithmic treatment, namely the widely-used mixed-integer reformulation technique as described in, for example, [46]. Our third way to tackle $DP(X,Y)$ is based on linear disjunctive programming for the relaxation $LDP(X,Y)$.

5.2.1 Lower Bounds Based on Standard Procedures in Global Optimization

Our first approach is to use the solution method for disjunctive programs from Chapter 4. To make this work, we need an M-independent and monotone

lower bounding procedure. As already mentioned, most of the lower bounding procedures commonly used in global optimization can be used here. In this chapter we propose to apply optimal centered forms [17,70] although other methods like, for example, αBB relaxations are also applicable.

In Algorithm 2 we can now state our first lower bounding procedure, which essentially is a part of the branch-and-bound algorithm of the previous chapter tailored to the special type of disjunctive problem $DP(X, Y)$.

Algorithm 2: Lower bounding procedure for $DP(X, Y)$ based on a logical expression

---

Input: Box $X \subset \overline{X}$, finite set $Y \subset \overline{Y}$.
Output: Lower bound $\ell(X)$ to the minimal value of problem $DP(X, Y)$.
Compute M-independent lower bounds $\ell_{v_i(\cdot,y)}(X)$ for $i \in I_0$, $y \in Y$;
For $i \in I_0$ and $y \in Y$ put

$$\widehat{Y}_{i,y}(X) := \begin{cases} \text{true} & \text{if } \ell_{v_i(\cdot,y)}(X) \leq 0 \\ \text{false} & \text{else.} \end{cases}$$

if $\bigwedge_{y \in Y} \bigvee_{i \in I_0} \widehat{Y}_{i,y}(X)$ *is true* **then**
  calculate lower bound $\ell(X) = \ell_f(X)$;
else
  put $\ell(X) = +\infty$;
end

Insertion of this lower bounding procedure into a generic branch-and-bound framework as in Algorithm 5 below leads to a convergent solution method for DPs. A more detailed description as well as a formal proof is given in Chapter 4 as well as in [66].

5.2.2 Lower Bounds Based on Linearization

Our second approach is to use the lower bounding procedure for disjunctive programs described in the following. Our method is related to ideas presented in [20]. In order to explain the method we start by quickly reviewing the standard procedure for the conjunctive program

$$P: \qquad \min_{x \in \mathbb{R}^n} f(x) \qquad \text{s.t.} \qquad g_i(x) \leq 0,\ i = 1, \ldots, p,$$

where all defining functions are assumed to be factorable, meaning that they can be evaluated in a finite number of steps. The calculation of the lower

bound is divided into two steps:

- a symbolic reformulation step
- and a linearization step.

During the reformulation step the problem is transformed into a lifted equivalent nonlinear problem. To achieve this we consider the expression trees of the defining functions as done in [108,109]. All leaves are described either by constants or variables. In contrast, the nodes are described by operators such as $+$, $-$, $\cdot$, $/$, powers or sin, cos, exp and log. These are the only elementary functions that we consider here although the extension to others is straightforward.

We describe the main ideas along Example 5.2.1. The method is formally explained in [81,102,108,109].

Example 5.2.1. *The conjunctive nonlinear problem*

$$P: \quad \min_{x\in\mathbb{R}^2} f(x) := -\exp(x_1) \qquad s.t. \qquad g_1(x) := \sin(x_1x_2) \leq 0, x \in [-1,1]^2$$

can be reformulated as follows. We first introduce new variables x_f, x_{g_1} so that instead of P we can equivalently solve the problem

$$\begin{aligned} P_1: \quad \min_{x\in\mathbb{R}^4} x_f \qquad s.t. \qquad & x_{g_1} \leq 0 \\ & x_{g_1} = \sin(x_1x_2) \\ & x_f = -\exp(x_1) \\ & x \in [-1,1]^2 \times \mathbb{R}^2. \end{aligned}$$

With the same technique we continue to lift the problem by introducing a new variable x_3 together with the constraint $x_3 = x_1x_2$. Analogously we proceed with the objective function and thus we obtain

$$\begin{aligned} P_2: \quad \min_{x\in\mathbb{R}^6} x_f \qquad s.t. \qquad & x_{g_1} \leq 0 \\ & x_3 = x_1x_2 \\ & x_{g_1} = \sin(x_3) \\ & x_4 = \exp(x_1) \\ & x_f = -x_4 \\ & x \in [-1,1]^2 \times \mathbb{R}^4. \end{aligned}$$

We have thus replaced the constraint $g_1(x) \leq 0$ by an equality constraint system $\tilde{g}_1(x) = 0$ and an inequality constraint $x_{g_1} \leq 0$ in a lifted space, as

well as the objective function $f(x)$ *by an equality constraint* $\tilde{f}(x) = 0$ *and an additional variable* x_f*. Appropriate box constraints at the new variables can be determined by interval arithmetic, see [86] for example. Given the original box constraints* $x \in \overline{X}$ *these new ones are denoted by* $B(\overline{X})$ *so that in summary we have* $x \in \overline{X} \times B(\overline{X}) \subset \mathbb{R}^{\overline{n}}$*.*

So far we have derived an equivalent optimization problem P_2 *that is still nonconvex and thus still difficult to solve to global optimality. However, in a next step we may convexify and even linearize the equality constraints of* P_2*, to obtain an easier to solve relaxation of the problem that provides at least a lower bound at the globally minimal value of* P*.*

The first nonlinear constraint $x_3 = x_1 x_2$ *can be linearized by its convex envelope, see [81].*

Similarly, the equality constraint $x_4 = \exp(x_1)$ *can be relaxed by some convex constraints, such as* $x_4 \geq \exp(x_1)$*. Due to monotonicity of exponentiation and due to the box constraints on* x_1 *we also have* $\exp(-1) \leq x_4 \leq \exp(1)$*. Some additional calculations yield* $x_4 \leq ax_1 + b$ *with* $a = 0.5(\exp(1) - \exp(-1))$ *and* $b = \exp(1) - a$*.*

The constraint $x_{g_1} = \sin(x_3)$ *is handled in different ways in the literature. We simply propose to use interval arithmetic [86] to compute a lower bound* ℓ_{g_1} *and an upper bound* u_{g_1} *at* x_{g_1} *and insert the constraints* $\ell_{g_1} \leq x_{g_1} \leq u_{g_1}$*.*

In summary we can compute a lower bound at the optimal value of P *over* $[-1,1] \times [-1,1]$ *by solving a convex relaxation, or we can even further relax these convex constraints and obtain a linear program where* $A_{\exp} x \leq b_{\exp}$ *denotes the linear relaxation of the exponential function and* $A_{MC} x \leq b_{MC}$ *stands for the McCormick relaxation.*

$$
\begin{aligned}
\widehat{P}: \qquad \min_{x \in \mathbb{R}^6} x_f \qquad s.t. \qquad & x_{g_1} \leq 0 \\
& A_{MC} x \leq b_{MC} \\
& \ell_{g_1} \leq x_{g_1} \leq u_{g_1} \\
& A_{\exp} x \leq b_{\exp} \\
& x_f = -x_4 \\
& x \in [-1,1]^2 \times B([-1,1]^2).
\end{aligned}
$$

Aggregating all these linear relaxations of the constraint g_1 *to* $A_{g_1} \overline{x} \leq b_{g_1}$ *and*

the linear relaxations regarding the objective function to $A_f\overline{x} \leq b_f$ yields

$$\begin{aligned} \widehat{P}: \qquad \min_{\overline{x}\in\mathbb{R}^6} \overline{x}_f \qquad s.t. \qquad & \overline{x}_{g_1} \leq 0 \\ & A_{g_1}\overline{x} \leq b_{g_1} \\ & A_f\overline{x} \leq b_f \\ & \overline{x} \in [-1,1]^2 \times B([-1,1]^2) \end{aligned}$$

where $\overline{x}$ consists of the original variables x as well as auxiliary variables x' introduced during the reformulation, that is,

$$\overline{x} = \begin{pmatrix} x \\ x' \end{pmatrix} \quad \textit{with} \quad x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad \textit{and} \quad x' = \begin{pmatrix} x_3 \\ x_4 \\ x_f \\ x_{g_1} \end{pmatrix}.$$

By applying these ideas every factorable inequality constraint $g(x) \leq 0$ can be equivalently reformulated in a lifted space by $\overline{x}_g \leq 0$ together with some additional equality constraints $\tilde{g}(\overline{x}) = 0$ and appropriate box constraints. Those can then be relaxed by the linear constraints $A_g\overline{x} \leq b_g$ where $\overline{x}$ consists of the original variables x as well as the new variables x', that is,

$$\overline{x} = \begin{pmatrix} x \\ x' \end{pmatrix},$$

where the variables x_g and x_f are components of x'. Note that the matrices A_g and the vector b_g differ for different boxes X which will only be expressed explicitly by $A_g(X)$ and $b_g(X)$ if needed. We assume these linearizations of a constraint to be convergent in the following sense.

Definition 5.2.2. *Linearizations of a constraint $g(x) \leq 0$ are said to be convergent if for any exhaustive sequence of boxes $(X_k)_{k\in\mathbb{N}}$ and $\tilde{x} \in X_k$, $k \in \mathbb{N}$, the following correspondences hold for all k sufficiently large:*

$$\begin{aligned} & & g(\tilde{x}) \leq 0 \\ \iff & \exists\, \overline{x} \in X_k \times B(X_k): & \overline{x}_g \leq 0, \tilde{g}(\overline{x}) = 0 \\ \iff & \exists\, \overline{x} \in X_k \times B(X_k): & A_g(X_k)\,\overline{x} \leq b_g(X_k). \end{aligned}$$

This convergence is a mild assumption as it holds for all commonly used linearization techniques in global optimization and, additionally, can be achieved by using interval arithmetic. In fact, the first equivalence can be seen easily by considering points $(\tilde{x}, x') \in X_k \times B(X_k)$ and sequentially solving the equality system $\tilde{g}(\tilde{x}, x') = 0$.

Next we explain how to extend this lower bounding procedure to the special case of the disjunctive programs

$$DP(Y): \quad \min_{x\in\mathbb{R}^n} f(x) \qquad \text{s.t.} \qquad \bigwedge_{y\in Y}\bigvee_{i\in I_0}\Big(v_i(x,y)\leq 0\Big),\ x\in\overline{X}$$

that arise when solving a GSIP.

The lifted and reformulated constraints of $v_i(x)\leq 0$ are denoted by $x_{v_i}\leq 0, \tilde{v}_i(x)=0$ and the relaxed linear approximations of $v_i(x,y)$ by $A_{i,y}x\leq b_{i,y}$. Similarly, the reformulation of the objective function and its relaxation are $\tilde{f}(\overline{x})=0$ and $Ax\leq b$, respectively, so that we obtain the relaxed subproblem

$$LDP(Y): \quad \min_{\overline{x}\in\mathbb{R}^{\overline{n}}} \overline{x}_f \quad \text{s.t.} \quad \Big(A\overline{x}\leq b\Big)\wedge\bigwedge_{y\in Y}\bigvee_{i\in I_0}\Big(A_{i,y}\overline{x}\leq b_{i,y}\Big), \quad \overline{x}\in\overline{X}\times B(\overline{X}),$$

where $\overline{x}_f\in\mathbb{R}$ is the $\overline{x}$-component that represents the objective function in the reformulation described above. Note that due to the innermost conjunctive system of linear inequalities, $LDP(Y)$ is no longer stated in conjunctive normal form.

As before, for the branch-and-bound framework we also consider restrictions of the problems $DP(Y)$ and $LDP(Y)$ to certain boxes $X\subset\overline{X}$, that is, the problems

$$DP(X,Y): \quad \min_{x\in\mathbb{R}^n} f(x) \qquad \text{s.t.} \qquad \bigwedge_{y\in Y}\bigvee_{i\in I_0}\Big(v_i(x,y)\leq 0\Big),\ x\in X$$

and

$$LDP(X,Y): \quad \min_{\overline{x}\in\mathbb{R}^{\overline{n}}} \overline{x}_f \quad \text{s.t.} \quad \Big(A\overline{x}\leq b\Big)\wedge\bigwedge_{y\in Y}\bigvee_{i\in I_0}\Big(A_{i,y}\overline{x}\leq b_{i,y}\Big), \quad \overline{x}\in X\times B(X),$$

respectively. Note again that the matrices A and $A_{i,y}$ as well as the right-hand sides b and $b_{i,y}$ may differ for different boxes X, but that in our notation we omit this dependency for the ease of presentation.

Lower Bounds Based on a Mixed-Integer Reformulation

The first and straightforward idea to tackle $LDP(X,Y)$ is to reformulate it as a mixed integer linear program and thus to shift the difficulties of the

disjunctions to the linear solver. This can be achieved by applying the well-known big-M reformulation, see [136] for example. We rewrite $LDP(X,Y)$ as

$$
\begin{aligned}
MILP(X,Y): \qquad \min_{\overline{x},z} \ \overline{x}_w \qquad \text{s.t.} \qquad & A\overline{x} \leq b \\
& A_{i,y}\overline{x} \leq b_{i,y} + (1 - z_{i,y})M, \ y \in Y \\
& \sum_{i \in I_0} z_{i,y} \geq 1, \ y \in Y \\
& \overline{x} \in X \times B(X), \ z \in \{0,1\}
\end{aligned}
$$

with M sufficiently large. Note that in this formulation $y \in Y$ serves as an index variable. Also observe that, due to the compactness of the feasible set, there are appropriate values for M. Problem $MILP(X,Y)$ is equivalent to $LDP(X,Y)$ in the sense that the optimal values of both problems coincide and, moreover, the orthogonal projection of any optimal point (x^*, z^*) of $MILP(X,Y)$ onto the x-space is an optimal point of $LDP(X,Y)$. Thus a straightforward idea to compute a lower bound at the globally minimal value of $DP(X,Y)$ on a given box $X \subset \overline{X}$ is to solve the mixed-integer reformulation $MILP(X,Y)$ of its relaxation $LDP(X,Y)$ instead. Formally this is described in Algorithm 3.

Algorithm 3: Lower bounding procedure for $DP(X,Y)$ based on a mixed-integer reformulation

Input: Box $X \subset \overline{X}$, finite set $Y \subset \overline{Y}$
Output: Lower bound $\ell(X)$ to the minimal value of problem $DP(X,Y)$.
Construct problem $MILP(X,Y)$;
Compute the optimal value $\ell(X)$ of $MILP(X,Y)$.

We now show that the aforementioned ideas really lead to an applicable lower bounding procedure.

Lemma 5.2.3. *For any box $X \subset \overline{X}$ and finite set $Y \subset \overline{Y}$, the value $\ell(X)$ computed by Algorithm 3 satisfies $\ell(X) \leq \min_{x \in M(X,Y)} f(x)$.*

Proof. In case that $M(X,Y)$ is empty the result is true due to the usual setting $\min_{x \in M(X,Y)} f(x) = +\infty$.

Otherwise let $x^* \in M(X,Y)$ be a global minimal point of $DP(X,Y)$. We shall construct a feasible point of $MILP(X,Y)$ with objective function value $f(x^*)$. This will imply $\ell(X) \leq f(x^*)$ and, thus, the assertion.

Due to the construction of the lifted and reformulated problem, there is a point $\overline{x} = (x^*, x') \in X \times B(X)$ with

$$\tilde{v}_i(\overline{x}) = 0,\ \overline{x}_{v_i} \leq 0 \text{ for } i \in I_0 \tag{5.2}$$

as well as

$$\tilde{f}(\overline{x}) = 0,\ \overline{x}_f = f(x^*). \tag{5.3}$$

Equation (5.2) ensures the requirements regarding the constraints of the original problem whereas equation (5.3) is important for the objective function value. As the linear constraints $A\overline{x} \leq b, A_{i,y}\overline{x} \leq b_{i,y}$ are relaxations of these constraints, the point $\overline{x}$ is also feasible to $LDP(X, Y)$. As already mentioned, the problems $LDP(X, Y)$ and $MILP(X, Y)$ are equivalent, and thus there is a point $(\overline{x}, z)$ which is feasible for $MILP(X, Y)$ with objective function value $f(x^*)$. □

The convergence properties of this method in the continuous case can be extended to show that this lower bounding procedure is also convergent for our disjunctive problems, as we state in the following result.

Theorem 5.2.4. *For a finite set $Y \subset \overline{Y}$ let Algorithm 3 use convergent linearizations. Then for any exhaustive sequence of boxes $(X_k)_{k\in\mathbb{N}}$ and $\tilde{x} \in X_k$, $k \in \mathbb{N}$, and for lower bounds $\ell(X_k)$ computed by Algorithm 3 we have*

$$\lim_{k\to\infty} \ell(X_k) = f(\tilde{x})$$

if $\tilde{x}$ is feasible for $DP(Y)$, and

$$\lim_{k\to\infty} \ell(X_k) = \infty$$

otherwise.

Proof. In case that $\tilde{x}$ is infeasible for $DP(Y)$ we have

$$\bigwedge_{y\in Y} \bigvee_{i\in I_0} \Big(v_i(\tilde{x}, y) \leq 0\Big) = \text{false}$$

and, due to Definition 5.2.2, we also have

$$v_i(\tilde{x}, y) \leq 0 \Longleftrightarrow A_{i,y}\overline{x} \leq b_{i,y}, i \in I_0, y \in Y$$

for every $\overline{x} \in X_k \times B(X_k)$ and k sufficiently large. This means that we also have

$$\bigwedge_{y\in Y} \bigvee_{i\in I_0} \Big(A_{i,y}\overline{x} \leq b_{i,y}\Big) = \text{false}$$

for all $\overline{x} \in X_k \times B(X_k)$, k sufficiently large, and thus the problems $LDP(X_k, Y)$ and $MILP(X_k, Y)$ are inconsistent, meaning that $\ell(X_k) = \infty$.

We now consider the case of $\tilde{x}$ being feasible for $DP(Y)$. The values $\ell(X_k)$ are computed as the optimal values of $MILP(X_k, Y)$ or $LDP(X_k, Y)$, respectively, and in view of Lemma 5.2.3 we have $\ell(X_k) \leq \min_{x \in M(X_k, Y)} f(x) \leq f(\tilde{x})$ for all $k \in \mathbb{N}$. Let us assume the existence of some $d > 0$ such that $\ell(X_k) \leq f(\tilde{x}) - d$ holds for all sufficiently large k. Then for these k there is a point $\overline{x} \in X_k \times B(X_k)$ such that

$$\begin{aligned} &\overline{x}_f \leq f(\tilde{x}) - d \\ &A\overline{x} \leq b \\ &\bigwedge_{y \in Y} \bigvee_{i \in I_0} \Big(A_{i,y} \overline{x} \leq b_{i,y} \Big) = \text{true}\,. \end{aligned}$$

Due to Definition 5.2.2 for k sufficiently large this is equivalent to the existence of a point $\overline{x} \in X_k \times B(X_k)$ that fulfills

$$\begin{aligned} &\overline{x}_f \leq f(\tilde{x}) - d \\ &\tilde{f}(\overline{x}) = 0 \\ &\bigwedge_{y \in Y} \bigvee_{i \in I_0} \Big(\tilde{v}_i(\overline{x}, y) = 0, \overline{x}_{v_i} \leq 0 \Big) = \text{true} \end{aligned}$$

and this again is equivalent to the existence of some point $x \in X_k$ with

$$\begin{aligned} &f(x) \leq f(\tilde{x}) - d \\ &\bigwedge_{y \in Y} \bigvee_{i \in I_0} \Big(v_i(x, y) \leq 0 \Big) = \text{true}\,. \end{aligned}$$

However, since the sets X_k, $k \in \mathbb{N}$, shrink to the singleton $\{\tilde{x}\}$, the continuity of f rules out the first inequality, and thus we derived a contradiction. □

Lower Bounds Based on Linear Disjunctive Programming

The mixed-integer reformulation of $LDP(X, Y)$ might be a practical possibility to solve the problem and, at the same time, it enables us to take advantage of existing, well developed software packages for linear programming. On the other hand, in case of large parameters M numerical instabilities may occur and, moreover, the corresponding continuous relaxations needed to solve the problem are known to be not too tight in general. Fortunately, a large amount of literature is devoted to disjunctive problems, especially for the case where

all defining functions are linear and, thus, there are algorithms tailored to this kind of problem. The present section discusses one such approach for lower bounding the problem $LDP(X, Y)$.

Many algorithms in the literature on disjunctive programming actually aim at solving problems of the form

$$DP: \qquad \min_{x \in X} c^\intercal x \qquad \text{s.t.} \qquad \bigvee_{i \in I} \Big(A_i x \leq b_i \Big)$$

whose feasible set that can be regarded as the union of the polyhedra $\{x \in \mathbb{R}^n | A_i x \leq b_i\}$, $i \in I$. Often this is achieved by constructing the convex hull of the latter union. It is well-known that the optimal value of DP coincides with the optimal value of

$$P: \qquad \min_{x \in \mathbb{R}^n} c^\intercal x \qquad \text{s.t.} \qquad x \in \operatorname{conv} \left(\bigcup_{i \in I} \Big\{ x \in X \;\Big|\; A_i x \leq b_i \Big\} \right)$$

due to the linear objective function ([12]). For our solution method presented below we propose to adapt the cutting plane algorithm from [13] in order to iteratively compute the convex hull of the union of the polyhedra. For the sake of completeness we briefly review this procedure. A main concept will be that a linear constraint $a^\intercal x - b \leq 0$ with parameters $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$ is said to be implied by the constraints $\bigvee_{i \in I} (A_i x \leq b_i)$ if it is fulfilled for all the corresponding feasible points of DP.

The idea of the cutting plane algorithm is to successively cut away a point $x^* \in \mathbb{R}^n$ by introducing a new constraint $a^\intercal x - b \leq 0$ that is implied by $\bigvee_{i \in I} (A_i x \leq b_i)$ and satisfies $a^\intercal x^* > b$. In order to find appropriate parameters a and b the so-called cut generating linear program has to be solved which we will derive in the following.

For $x \in \operatorname{conv} \left(\bigcup_{i \in I} \{ x \in X \mid A_i x \leq b_i \} \right)$ we can compute a valid cut by solving

$$\begin{aligned} LP_1(x^*): \qquad \max_{a,b} \quad & a^\intercal x^* - b \\ \text{s.t.} \quad & a^\intercal v \leq b \text{ for all } v \in V \\ & V := \left\{ v \in \mathbb{R}^n \,\middle|\, \bigvee_{i \in I} (A_i v \leq b_i) \right\} \\ & -1 \leq b \leq 1, \; a \in [-1, 1]^n \end{aligned}$$

which, from a semi-infinite point of view, can be rewritten as

$$\begin{aligned} LP_2(x^*): \quad & \max_{a,b} \quad a^\intercal x^* - b \\ & \text{s.t.} \quad \max_{v \in V_i} a^\intercal v \leq b, \ i \in I \\ & \qquad -1 \leq b \leq 1, \ a \in [-1,1]^n \end{aligned}$$

with $V_i := \left\{ v \in \mathbb{R}^n \middle| A_i v \leq b_i \right\}$, $i \in I$. Replacing the lower level problems by their duals immediately yields

$$\begin{aligned} LP_3(x^*): \quad & \max_{a,b} \quad a^\intercal x^* - b \\ & \text{s.t.} \quad \min_{w_i \in W_i} b_i^\intercal w_i \leq b, \ i \in I \\ & \qquad -1 \leq b \leq 1, \ a \in [-1,1]^n \end{aligned}$$

with $W_i := \{w_i | A_i^\intercal w_i = a, w_i \geq 0\}$, $i \in I$. This is equivalent to the final cut generating problem

$$\begin{aligned} LP(x^*): \quad & \max_{a,b,w} \quad a^\intercal x^* - b \\ & \text{s.t.} \quad b_i^\intercal w_i \leq b, i \in I \\ & \qquad A_i^\intercal w_i = a, i \in I \\ & \qquad w_i \geq 0, i \in I \\ & \qquad -1 \leq b \leq 1, \ a \in [-1,1]^n. \end{aligned}$$

Although not presented in a semi-infinite context, this result is well-known from the literature on disjunctive programming [13]. In fact, the technique itself already appeared in 1977 in a slightly different context [10].

These observations lead to cutting plane methods for DPs as outlined in the following. Formally they are described in [13].

1. Solve a relaxation of the original problem DP.
2. If the computed optimal point x^* of the relaxation is feasible for DP, then Stop.
3. Otherwise: Solve the cut generating linear program $LP(x^*)$ in order to find a new constraint $a^\intercal x \leq b$ and add this to the relaxation of DP.
4. Go to Step 1.

Unfortunately, like many algorithms in the field of disjunctive programming, this method does not address problems like $LDP(X,Y)$ since these are not stated in conjunctive normal form. At least the feasible set of $LDP(X,Y)$ can be described as the intersection of the union of polyhedra. The corresponding logical expression is then said to be in *regular form* [12].

For $LDP(X,Y)$ the effort to compute the convex hull of the feasible set is much higher, but if we use the techniques from literature to compute the convex hull of the union of polyhedra we have at least

$$\begin{aligned} &\operatorname{conv}\left(\bigcap_{y\in Y}\bigcup_{i\in I}\{\overline{x}\in X\times B(X)|A_{i,y}\overline{x}\leq b_{i,y}\}\right)\\ \subset&\bigcap_{y\in Y}\operatorname{conv}\left(\bigcup_{i\in I}\{\overline{x}\in X\times B(X)|A_{i,y}\overline{x}\leq b_{i,y}\}\right) \end{aligned} \tag{5.4}$$

although equality does not necessarily hold [12].

Figure 5.1 illustrates this fact for the depicted example sets $M_{i,y}=\{\overline{x}\in\mathbb{R}^2|A_{i,y}\overline{x}\leq b_{i,y}\}$ with $i\in I=\{1,2,3\}$ and $y\in Y=\{1,2\}$. For instance the point $\widetilde{x}$ does not lie in the original feasible set but is contained in its superset $\bigcap_{y\in Y}\operatorname{conv}\left(\bigcup_{i\in I}\{\overline{x}\in X\times B(X)|A_{i,y}\overline{x}\leq b_{i,y}\}\right)$.

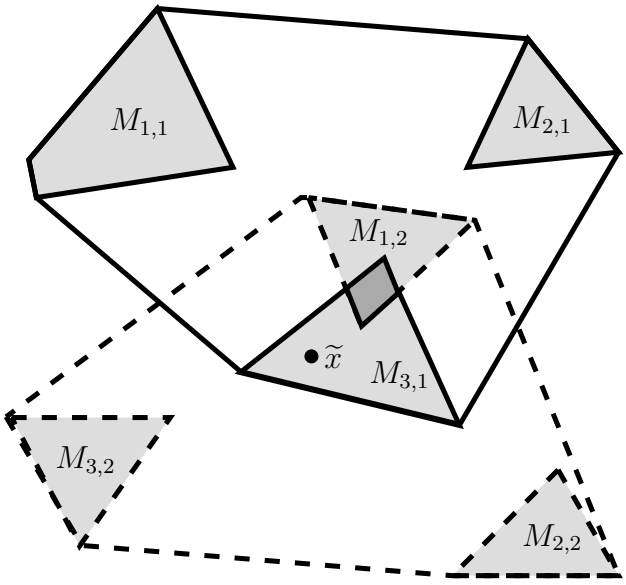

Figure 5.1: Convex hull of feasible set of a regular disjunctive program and its relaxation

These observations lead to a possibility to at least determine a lower bound at the optimal value of $LDP(X,Y)$ by computing the optimal value of the relaxed problem

$$\begin{aligned}\widehat{LDP}(X,Y): \quad & \min_{\overline{x}\in\mathbb{R}^{\overline{n}}} \overline{x}_f \\ \text{s.t.} \quad & A\overline{x} \leq b \\ & \overline{x} \in \bigcap_{y\in Y} \operatorname{conv}\left(\bigcup_{i\in I}\{\overline{x}\in X\times B(X) | A_{i,y}\overline{x}\leq b_{i,y}\}\right).\end{aligned}$$

In order to solve $\widehat{LDP}(X,Y)$ we extend the cutting plane method described above. We start by computing an initial lower bound by omitting the constraints

$$\bigwedge_{y\in Y}\bigvee_{i\in I_0}(A_{i,y}\overline{x}\leq b_{i,y})$$

of $LDP(X,Y)$ and solving the *conjunctive* problem

$$\widehat{LDP}_1(X,Y): \quad \min \overline{x}_f \quad \text{s.t.} \quad A\overline{x}\leq b, \quad \overline{x}\in X\times B(X).$$

Since $\widehat{LDP}_1(X,Y)$ relaxes the problem $DP(X,Y)$, the minimal value of $\widehat{LDP}_1(X,Y)$ is a lower bound for the minimal value of $DP(X,Y)$. Now let us consider the computed optimal point $\overline{x}_1^*$ of $\widehat{LDP}_1(X,Y)$. If for every $y\in Y$ we have $\overline{x}_1^*\in \operatorname{conv}\left(\bigcup_{i\in I}\{\overline{x}\in X\times B(X)|A_{i,y}\overline{x}\leq b_{i,y}\}\right)$ then $\overline{x}_1^*$ is feasible for $\widehat{LDP}(X,Y)$, so that we are not able to further improve the lower bound by adding more cutting planes of this type. Otherwise, for every $y\in Y$ for which the latter is not the case, we may add a new constraint and, thus, improve the approximation of the feasible set by solving the cut generating problem

$$\begin{aligned}LP(\overline{x}_1^*,y): \quad \max_{a,b,w} \quad & a^\intercal\overline{x}_1^* - b \\ \text{s.t.} \quad & b_{i,y}^\intercal w_i \leq b, i\in I \\ & A_{i,y}^\intercal w_i = a, i\in I \\ & w_i\geq 0, i\in I \\ & -1\leq b\leq 1,\ a\in[-1,1]^{\overline{n}}.\end{aligned}$$

Then, for every such y, we add the new cutting plane to the conjunctive linear lower bounding problem. Formally this is described in Algorithm 4.

Algorithm 4: Lower bounding procedure for $DP(X, Y)$ based on linear disjunctive programming

Input: Box $X \subset \overline{X}$, finite set $Y \subset \overline{Y}$.
Output: Lower bound $\ell(X)$ to the minimal value of problem $DP(X, Y)$.
Step 1: Set iteration counter $i = 1$;
Step 2:
if $\widehat{LDP}_i(X, Y)$ *is consistent* **then**
  Compute an optimal point $\overline{x}_i^*$ of $\widehat{LDP}_i(X, Y)$;
else
  Stop: put $\ell(X) = \infty$;
end
Step 3: Set feasible_bool = true;
For every $y \in Y$ improve convex hull approximations:
if *optimal value of cut generating problem* $LP(\overline{x}_i^*, y)$ *is positive* **then**
  add corresponding cutting plane $a^\intercal \overline{x} \leq b$ to conjunctive linear lower bounding problem;
  feasible_bool = false;
end
if *feasible_bool* **then**
  No improvement is possible, Stop: define $\ell(X)$ as the optimal value of $\widehat{LDP}_i(X, Y)$;
end
Step 4: Increment iteration counter i and go to Step 2;

Despite the obvious discrepancy between the problems $LDP(X, Y)$ and their relaxations $\widehat{LDP}(X, Y)$, we will be able to present a convergence result on this lower bounding procedure. We start by confirming that $\ell(X)$ computed as shown above actually is a lower bound for $DP(X, Y)$.

Lemma 5.2.5. *For any box $X \subset \overline{X}$ and finite set $Y \subset \overline{Y}$, the value $\ell(X)$ computed by Algorithm 4 satisfies $\ell(X) \leq \min_{x \in M(X,Y)} f(x)$.*

Proof. The result is due to the following chain of relaxations: as seen above, $LDP(X, Y)$ is a relaxation of $DP(X, Y)$, by (5.4) the problem $\widehat{LDP}(X, Y)$ is a relaxation of $LDP(X, Y)$ and, by construction of the cutting planes, for any i the problem $\widehat{LDP}_i(X, Y)$ is a relaxation of $\widehat{LDP}(X, Y)$. □

Theorem 5.2.6. *For a finite set $Y \subset \overline{Y}$ let Algorithm 4 use convergent linearizations. Then for any exhaustive sequence of boxes $(X_k)_{k \in \mathbb{N}}$ and $\tilde{x} \in$*

X_k, $k \in \mathbb{N}$, and for lower bounds $\ell(X_k)$ computed by Algorithm 4 we have

$$\lim_{k\to\infty} \ell(X_k) = f(\tilde{x})$$

if $\tilde{x}$ is feasible for $DP(Y)$, and

$$\lim_{k\to\infty} \ell(X_k) = \infty$$

otherwise.

Proof. In case that $\tilde{x}$ is infeasible for $DP(Y)$, with the same lines of arguments as in the proof of Theorem 5.2.4 we obtain

$$\bigwedge_{y\in Y} \bigvee_{i\in I_0} \left(A_{i,y}\overline{x} \leq b_{i,y} \right) = \text{false}$$

for all $\overline{x} \in X_k \times B(X_k)$ and k sufficiently large. This means that there is some $\tilde{y} \in Y$ such that

$$\bigvee_{i\in I_0} \left(A_{i,\tilde{y}}\overline{x} \leq b_{i,\tilde{y}} \right) = \text{false}$$

holds and, thus,

$$\bigcup_{i\in I_0} \left\{ \overline{x} \in X_k \times B(X_k) \mid A_{i,\tilde{y}}\overline{x} \leq b_{i,\tilde{y}} \right\} = \emptyset,$$

$$\text{conv}\left(\bigcup_{i\in I_0} \left\{ \overline{x} \in X_k \times B(X_k) \mid A_{i,\tilde{y}}\overline{x} \leq b_{i,\tilde{y}} \right\} \right) = \emptyset,$$

$$\text{and} \quad \bigcap_{y\in Y} \text{conv}\left(\bigcup_{i\in I_0} \left\{ \overline{x} \in X_k \times B(X_k) \mid A_{i,y}\overline{x} \leq b_{i,y} \right\} \right) = \emptyset.$$

For this reason the feasible set of $\widehat{LDP}(X_k, Y)$ is empty. Thus there is an $i \in \mathbb{N}$ such that also the feasible set of $\widehat{LDP}_i(X_k, Y)$ is empty after sufficiently many cutting planes have been added ([13]). It follows that for all sufficiently large k Algorithm 4 terminates with $\ell(X_k) = \infty$.

We now consider the case of $\tilde{x}$ being feasible for $DP(Y)$. Then all problems $\widehat{LDP}(X_k, Y)$, $k \in \mathbb{N}$, are consistent, and Algorithm 4 computes the lower bound $\ell(X_k)$ as the optimal value of $\widehat{LDP}_i(X_k, Y)$ for some $i \in \mathbb{N}$. Moreover, in view of Lemma 5.2.5 the inequalities $\ell(X_k) \leq \min_{x\in M(X_k,Y)} f(x) \leq f(\tilde{x})$ are satisfied for all $k \in \mathbb{N}$.

Let us assume that the sequence $\ell(X_k)$ does not converge to $f(\tilde{x})$. Then there is some $d > 0$ so that for every $k \in \mathbb{N}$ it holds $\ell(X_k) \leq f(\tilde{x}) - d$. This means that there is some feasible point $\overline{x} \in X_k \times B(X_k)$ of $\widehat{LDP}_i(X_k, Y)$ with $\overline{x}_f \leq f(\tilde{x}) - d$.

The feasibility of $\overline{x}$ for $\widehat{LDP}_i(X_k, Y)$ particularly implies $A\overline{x} \leq b$. According to Definition 5.2.2 this is equivalent to the existence of some $x \in X_k$ with $f(x) \leq f(\tilde{x}) - d$ for all sufficiently large k. However, due to $\lim_{k\to\infty} \operatorname{diag}(X_k) = 0$ and the continuity of f this is not possible and, thus, we have derived a contradiction. □

5.3 A Branch-and-Bound Framework for GSIPs

Roughly, our method for solving GSIPs works as follows:

1. Solve a problem $DP(Y)$ with $Y \subset \overline{Y}$. If the problem is not solvable, then terminate.
2. Check if the resulting optimal point x^* is feasible by solving $\widehat{Q}(x^*)$. Note that, despite its non-smooth objective function, $\widehat{Q}(x^*)$ can be solved by tailored branch-and-bound algorithms in a straightforward manner.
3. If x^* lies in $\widehat{M}$, then terminate. Else improve Y and continue.

To make this more specific, the present section is divided into two parts. In Subsection 5.3.1 we start by stating a branch-and-bound algorithm for solving nonlinear programs to global optimality. Due to the use of the lower bounding procedures from Section 5.2 our method is able to cope with disjunctive programs. An additional important property of this method is its ability to perform warm starts. This allows us to add constraints to a currently solved problem and then continue to solve the updated problem without the need to start at the root node of the branch-and-bound tree. In Subsection 5.3.2 we will take advantage of this algorithm to state our new discretization method for GSIPs.

5.3.1 Global Solution of DPs with Warm Starts

The algorithm for generalized semi-infinite problems solves a sequence of discretizations, such as $DP(Y)$ with $Y \subset \overline{Y}$. In fact, for given $Y \subset \overline{Y}$ and

the resulting optimal point $\overline{x}$, the corresponding feasibility problem $\widehat{Q}(\overline{x})$ is solved to global optimality. If the optimal value is nonpositive, then $\overline{x}$ is feasible and thus a globally minimal point of f over $\widehat{M}$. Otherwise, the obtained optimal point y^* of the feasibility problem is added to Y, and an improved discretized problem $DP(Y \cup \{y^*\})$ has to be solved.

In order to implement this procedure, a repeated sequence of similar problems has to be solved, and our goal in this section is to exploit this similarity in order to develop an efficient solver. In this regard our approach significantly differs from the one in [83] because there the occurring problems are solved independently from each other, in order to take advantage of state of the art solvers in global optimization.

Basically two types of similarity can be addressed, one between successive disjunctive problems, and another between successive feasibility problems. Especially when the discretization procedure eventually converges, we may hope that many points $\overline{x}_1$ and $\overline{x}_2$ are generated that are close to each other and thus, due to the continuity of the objective function, it is a reasonable idea to try to take advantage of information gained before. However, although the constraints might differ only slightly in the parameter x, a global branch-and-bound solver has to consider all constructed boxes again to compute the correct lower bounds and to ensure that the slight change does not lead to significant effects regarding the new feasible set.

However, there is a second similarity in the proposed solution method for GSIPs that can be exploited much better and this is the one between the problems $DP(Y_1)$ and $DP(Y_2)$ with $Y_2 = Y_1 \cup \{y^*\}$. Instead of solving problem $DP(Y_2)$ from scratch, we propose to continue the branch-and-bound procedure from the solution of $DP(Y_1)$ to obtain a globally minimal point of $DP(Y_2)$.

For this warm start ability it is important that during the solution of $DP(Y_1)$ boxes that contain a globally minimal point of $DP(Y_2)$ are not fathomed due to bad objective function values from the list $\mathcal{L}$ of candidate boxes for globally minimal points of $DP(Y_1)$. In fact, adding a new constraint in $DP(Y_2)$ may cut off the optimal point of $DP(Y_1)$ and, at the same time, the optimal value of $DP(Y_2)$ becomes larger than the one of $DP(Y_1)$. A fathoming procedure for $DP(Y_1)$ may thus delete boxes which do not contain optimal points of $DP(Y_1)$, but very well optimal points of $DP(Y_2)$.

To rectify this, we propose instead to remove such boxes from the list $\mathcal{L}$, but save them to a second list $\mathcal{M}$ which is not further considered during the algorithm, as long as no additional constraints are added. Especially the current best lower bound is only determined by considering boxes in the list

$\mathcal{L}$. Thus for the current cycle boxes in the list $\mathcal{M}$ are treated as if they were fathomed. If we add new constraints to the optimization problem we can join the boxes from the list $\mathcal{M}$ to $\mathcal{L}$ and continue to solve the problem. In this way the proposed algorithm is able to cope with warm starts.

Observe that, on the other hand, boxes can always be removed due to infeasibility in $DP(Y_1)$ because adding a new constraint cannot make them feasible again in $DP(Y_2)$. If the algorithm terminates with $\mathcal{L} = \emptyset$ the problem $DP(Y)$ is inconsistent.

In principle all methods from Section 5.2 are applicable as a lower bounding procedure. We are now ready to state our branch-and-bound framework. The term $\mathrm{mid}(\overline{X})$ will stand for the geometrical midpoint of the box $\overline{X}$.

Algorithm 5: Branch-and-bound framework for $DP(Y)$

Input: Finite set $Y \subset \overline{Y}$, initial lower bound $\widehat{v}_0 = \ell_f(\overline{X})$, initial upper bound $u_0 = +\infty$, list $\mathcal{L} = \{(\overline{X}, \ell(\overline{X}))\}$, best known point so far $x_0^* = \mathrm{mid}(\overline{X})$ and iteration counter $k = 1$.

Output: Global optimal point x^* of $DP(Y)$, or certificate for $M_{DP}(Y) = \emptyset$.

while $u_{k-1} - \widehat{v}_{k-1} > 0$ *and* $\mathcal{L} \neq \emptyset$ **do**

Step 1: Choose $(X_k, v_k) \in \mathcal{L}$ with $v_k = \widehat{v}_{k-1}$;

Step 2: Divide X_k along the midpoint of a longest edge into X_k^1 and X_k^2;

Step 3: For $j = 1, 2$ calculate lower bounds $v_k^j = \ell(X_k^j)$ by applying Algorithm 2, 3 or 4;

For $j = 1, 2$, if $v_k^j < \infty$, add the pair (X_k^j, v_k^j) to the list $\mathcal{L}$;

Step 4: For $j = 1, 2$ choose $x_k^j \in X_k^j$ and define

$$f_k^j := \begin{cases} f(x_k^j) & \text{if } x_k^j \in M_{DP}(X_k^j, Y) \\ +\infty & \text{else.} \end{cases}$$

Step 5: Put $u_k = \min\{u_{k-1}, f_k^1, f_k^2\}$ and choose $x_k^* \in \{x_{k-1}^*, x_k^1, x_k^2\}$ with $f(x_k^*) = u_k$;

Step 6: Fathoming:

for $(X, v) \in \mathcal{L}$ *with* $v > u_k$ **do**

Remove (X, v) from $\mathcal{L}$ and put it into $\mathcal{M}$;

end

Step 7: Update of lower bound:

if $\mathcal{L} \neq \emptyset$ **then**

$\widehat{v}_k = \min\{v \in \mathbb{R} \mid (X, v) \in \mathcal{L}\}$

end

Step 8: Increment k;

end

5.3.2 The Discretization Method for GSIPs

In this section we will use Algorithm 5 for disjunctive optimization problems to lower bound GSIPs by means of a discretization method.

In the first iteration we obtain a lower bound of *GSIP* by setting $Y_1 = \emptyset$ and solving $DP(Y_1)$. According to the theorem of Weierstrass it is clear that an optimal point x_1^* is obtained. We then continue to solve $\widehat{Q}(x_1^*)$ which is solvable for the same reason. In case that its optimal value is nonpositive,

x_1^* is already feasible for the semi-infinite program and thus we are done. Otherwise we set $Y_2 = Y_1 \cup \{y_1^*\}$ where y_1^* denotes the obtained optimal point of $\widehat{Q}(x_1^*)$ and solve the new discretization $DP(Y_2)$. Formally, this is described in Algorithm 6.

Algorithm 6: Discretization method for GSIP

Input: Initial index set $Y_1 = \emptyset$ and iteration counter $k = 1$.
Output: global optimal point x^* of f over $\widehat{M}$, or certificate for $\widehat{M} = \emptyset$.
while true **do**
  Step 1: Apply Algorithm 5 to $DP(Y_k)$;
  if $DP(Y_k)$ *is not solvable* **then**
    $\widehat{M}$ is empty, terminate;
  end
  Step 2: Let x_k^* be an optimal point of problem $DP(Y_k)$;
  Step 3: Determine an optimal point y_k^* and the optimal value v_k^* of $\widehat{Q}(x_k^*)$;
  if $v_k^* \leq 0$ **then**
    x_k^* is an optimal point, terminate;
  end
  Step 4: Put $Y_{k+1} = Y_k \cup \{y_k^*\}$
  Step 5: Increment k;
end

Theorem 5.3.1. *Assume that in Algorithm 6 the discretization method does not terminate. Then, the sequence $(x_k^*)_{k\in\mathbb{N}}$ possesses a cluster point, and any such cluster point is a globally minimal point of GSIP.*

Proof. The convergence of this method immediately follows from established results on the numerics of standard semi-infinite programming [53]. □

Step 3 of Algorithm 6 can possibly be accelerated by avoiding the effort of computing a globally optimal y_k^* of $\widehat{Q}(x_k^*)$. In fact, as soon as any feasible point y_k of $\widehat{Q}(x_k^*)$ with positive objective function value has been determined, the infeasibility of x_k^* is clear, and y_k may be used to refine the discretization Y_k in Step 4. Such a point y_k may occur, for example, as an iterate in a branch-and-bound procedure for the solution of $\widehat{Q}(x_k^*)$, or as the output of a local NLP solver. To still achieve convergence of this modified discretization method, the objective function values of the used feasible points y_k have to tend to globally maximal values while the algorithm proceeds.

5.4 Numerical Results

In this section we present our numerical results. Algorithms 2–6 are implemented in C++ and compiled with GNU g++, version 6.3. To integrate interval arithmetic, the library PROFIL/BIAS V2, see [68], is linked in. All occurring linear optimization problems are solved with GNU Linear Programming Kit 4.61, see [78]. All computations are performed on a laptop computer with Intel Core i5-2540M (2.60GHz, 3M cache) running Linux 4.9.0.

The termination tolerance of the overall semi-infinite solver is set to $\varepsilon = 0.01$. A point x is accepted as a feasible point with a feasibility tolerance $\varepsilon_f = 0.01$. All occurring global optimization problems are solved up to a tolerance of 0.001.

5.4.1 The Main Approach

All 16 test problems from [75] (cf. [83]) are solved. An overview is given in Table 5.1. The columns of the following tables are denoted as follows. The first one named "Problem" contains the name of the test problem. By n the number of x-variables is given in column two and m stands for the number of y-variables in column three. The term $|I|$ describes the number of lower level constraints. For every instance there is only one upper level constraint. The last column $f(x^*)$ contains the optimal value as reported in literature.

For test problem "L02" we could even correct the usually reported optimal point $(0,0)^\intercal$ with objective function value 0 to $(-1,-1)^\intercal$ with optimal value -1. Note that the point $(0,0)^\intercal$ actually is optimal for the original problem statement from [61]. However, for the test set in [75] the lower level constraint $y \geq -1$ and the upper level constraints $x \in [-1,1]^2$ were added to the problem statement which result in the extension of the feasible set $\{x \in \mathbb{R}^2 |\, x_1^2 < 2x_2\} \cup \{0\}$ by the line segment $[-1,1] \times \{-1\}$, which consists of optimal points with value -1. The fact that previous numerical tests in the literature still identified $(0,0)^\intercal$ as an optimal point may be due to the degenerate geometry of this feasible set. This also shows that, on the contrary, our approach does not run into troubles here.

Table 5.2 shows the results of Algorithm 6 with Algorithm 2 as lower bounding procedure in the global branch-and-bound framework. Besides the name of the problem, lower and upper bounds as well as number of iterations and the time needed to solve the problem are reported. Time is wall clock time given in seconds. In case a problem could not be solved within a time frame of two hours, the algorithm is terminated. In the following tables this

Table 5.1: Overview of the test problems

Problem	n	m	$\lvert I\rvert$	$f(x^*)$
L01	2	1	1	0.0625
L02	2	1	1	-1.0
L03	2	3	1	-0.5
L04	1	1	1	0.0
L05	2	2	2	-5.0
L06	2	3	3	-6.0
L07	2	1	1	-0.5
L08	2	1	1	-1.0
L09	1	1	1	0.043264
L10	2	1	2	-1.0
L11	3	1	1	0.5
L12	1	1	1	0.5
L13	3	1	1	2.935930275
L14	3	2	1	0.381924
L15	2	1	1	-3.710448
L16	6	2	1	-10.666

is indicated by '-'.

Analogously, the results of Algorithm 6 with lower bounding procedures from Algorithm 3 and Algorithm 4 in the global solver are presented in Table 5.3 and Table 5.4, respectively.

A comparison of the lower and upper bounds with the optimal values from Table 5.1 shows the effect of our several relaxation steps for the feasible set: the computed lower bounds are all valid as expected because lower bounds at a relaxation are also lower bounds for the original problem. On the other hand, upper bounds for a relaxed problem are not necessarily valid for the original problem and, in fact, several of the upper bounds from Tables 5.2, 5.3 and 5.4 are smaller than the optimal values from Table 5.1. In order to obtain valid values a combination with other techniques such as restriction of the right hand side from [83] is possible which might be incorporated into the global branch-and-bound solver so that the upper and the lower bounding problem can be solved simultaneously. This idea is left for future research.

It can be seen that the lower bounding procedure from Algorithm 2 is much faster compared to the other two algorithms for most of the problems. However, in applications often problems with many linear functions arise. For such problems, lower bounding procedures based on linearizations can

Table 5.2: Results with lower bounding procedure Algorithm 2

Problem	lower bound	upper bound	iterations	time
L01	0.055176	0.061111	8	0.003621
L02	-1.000000	-1.000000	1	0.000545
L03	-0.539062	-0.531193	20	99.414400
L04	-0.000000	0.000000	1	0.000674
L05	-5.000000	-5.000000	5	0.001807
L06	-	-	-	-
L07	-0.503906	-0.500977	6	0.005659
L08	-1.000000	-1.000000	1	0.000403
L09	0.041260	0.044495	2	0.009682
L10	-1.023438	-1.015625	7	0.003129
L11	0.491142	0.501084	2	0.017399
L12	0.494385	0.499893	2	0.000319
L13	2.925935	2.935930	2	0.024932
L14	0.371338	0.380920	4	0.005458
L15	-3.717773	-3.709412	6	0.010374
L16	-10.666667	-10.666667	2	0.000887

Table 5.3: Results with lower bounding procedure Algorithm 3

Problem	lower bound	upper bound	iterations	time
L01	0.060469	0.064758	9	0.304246
L02	-1.000000	-0.992188	1	0.011050
L03	-0.536792	-0.531005	17	4.040980
L04	-	-	-	-
L05	-5.000000	-4.990234	1	0.016379
L06	-6.000004	-5.993896	1	0.026348
L07	-0.507170	-0.503906	6	0.217583
L08	-1.000000	-0.992188	1	0.011842
L09	0.041618	0.047852	2	0.012854
L10	-1.013671	-1.003906	7	0.220980
L11	0.495301	0.497818	2	0.193993
L12	0.499268	0.505432	2	0.014410
L13	2.934678	2.942978	2	0.067032
L14	0.381277	0.381409	4	0.222761
L15	-3.715931	-3.709412	3	0.143179
L16	-	-	-	-

Table 5.4: Results with lower bounding procedure Algorithm 4

Problem	lower bound	upper bound	iterations	time
L01	0.060350	0.063553	8	1.203560
L02	-1.000000	-0.992188	1	0.013772
L03	-0.533370	-0.531005	19	93.45110
L04	-	-	-	-
L05	-5.000000	-4.990234	1	0.028632
L06	-6.000004	-5.993896	1	0.052105
L07	-0.507170	-0.503906	6	10.84950
L08	-1.000000	-0.992188	1	0.016547
L09	0.041618	0.047852	2	0.181970
L10	-	-	-	-
L11	0.495301	0.497818	2	3.584430
L12	0.499268	0.505432	2	0.074808
L13	2.934678	2.942978	2	1.378540
L14	0.381277	0.381409	4	10.039500
L15	-3.715931	-3.709412	3	21.333300
L16	-	-	-	-

be expected to be beneficial.

Numerical results for a similar algorithm using standard mixed-integer reformulations for the disjunctions and state-of-the-art solvers such as Baron and LindoGlobal can be found in [83]. It can be seen that our tailored lower bounding procedures lead to significant savings in time. The results are developed under similar conditions so that a comparison of the solution times is valid to a certain degree. However, we point out that the algorithm in [83] involves an additional upper bounding procedure leading to additional effort.

5.4.2 Additional Approaches

In the following some additional experiments are described which, however, did not lead to the expected improvement of the solver. For example, in each iteration of the semi-infinite solver there are two nonlinear problems, that have to be solved to global optimality, namely the discretization of the original GSIP and the feasibility problem. However, especially at the beginning of the solution process one might try to proceed faster without solving the subproblem to the highest possible accuracy.

Consider for instance the solution of the feasibility problem. It is well-known that many global optimization solvers often find the global optimal point early in the solution process and then spend most of the time to improve the lower bound in order to have a certificate of optimality. In our setting, this optimal point corresponds to the new constraint that is added to the discretization and, of course, we need to have a global optimal point in order to ensure convergence of our method. However, especially at the beginning of the solution process we might accept any such point and add the corresponding cutting plane, without the need to have accurate lower bounds that ensure global optimality of this point. Similarly, there is no need to solve every discretization of the generalized semi-infinite problem up to the highest accuracy, as the optimal point of this problem might be cut off later in the same iteration anyway. However, especially towards the end of the solution process global optimality is crucial to ensure convergence of the overall method.

As it turns out, our experiments in this direction are not very promising. For instance if we start with a termination tolerance of the global optimization solver of $\varepsilon_1 = 2.0$ and improve this value in every further iteration k by setting

$$\varepsilon_k = \max\left\{\frac{\varepsilon_{k-1}}{2}, 0.001\right\}$$

only few solution times strictly improve. Moreover, most of the test problems even take more time to solve. The same effect is observed for different sequences of tolerances $(\varepsilon_k)_{k\in\mathbb{N}}$.

Analogously, one might try to accelerate the semi-infinite solver by restricting the number of iterations of the global nonlinear solver at the beginning of the solution process. The motivation behind this approach is the same because higher accuracy comes at the cost of more iterations which is needed only later in the solution process of the semi-infinite framework. However, also here the test problems take more time to solve.

A possible explanation for the failure of these attempts is the increased number of iterations that lead to more discretization points and thus more disjunctions that have to be handled in later iterations of the semi-infinite solver. Although possible in principle, this seems to be very expensive from a computational point of view.

Chapter 6

Conclusions and Future Research

In this thesis, the close relationship between disjunctive problems and generalized semi-infinite programs is examined. We close this thesis by summarizing the main contributions and point out some additional questions that are left for future research.

6.1 Conclusions

In Chapter 3, we presented a reformulation of disjunctive programs with relatively general logical expressions into standard nonlinear programs. This opens the possibility to solve DPs locally by applying existing semi-infinite solution techniques. In contrast, the remaining part of this document covers the global solution of both kinds of problems.

In Chapter 4, we proposed a new branch-and-bound framework for global minimization of disjunctive programs with general logical expressions. We proved the convergence of our algorithm and discussed how to integrate negations and implications into logical expressions. The main novelty of our algorithm is that it neither requires the logical expressions to be in any normal form, nor introduces additional binary variables. Our computational results indicate how poorly classical algorithms can perform if one requires a normal form, and that running our algorithm on the original logical expression (which does not require any normal form) is computationally highly advantageous on those instances.

Based on these results, in Chapter 5 we presented an algorithm for the

global solution of generalized semi-infinite optimization problems by means of three different lower bounding procedures.

6.2 Outlook

However, there are still some issues which have to be addressed. For example, many applications involve logical expressions, which also make use of negations and implications. Since it is well-known that implications can be transformed into conjunctions or disjunctions involving negations, this raises the need to extend our reformulation method to cope with negated constraints of the type $\neg(G(x) \leq 0)$ or, in other words, constraints of the type $-G(x) < 0$. Unfortunately, for continuous functions G the feasible set then does not need to be closed, and one needs to take special care when showing solvability of such a problem. Moreover, relaxing the feasible set by considering the constraint $-G(x) \leq 0$ instead of $-G(x) < 0$ may heavily enlarge the feasible (at least in the absence of constraint qualifications), so that a computed solution of the nonlinear program may have little to do with a solution of the original disjunctive problem. For the branch-and-bound algorithm in Chapter 4 a possibility for inner and outer approximations of these types of DPs is discussed in Section 4.3. These considerations could be applied to the reformulation techniques in Chapter 3 analogously.

Similarly, it may also be beneficial to integrate even more operators into logical expressions, such as biconditional ("if and only if", $Y_i \leftrightarrow Y_j$), NAND ("not both", $Y_i \uparrow Y_j$), NOR ("neither... nor", $Y_i \downarrow Y_j$), XOR ("either... or", $Y_i \oplus Y_j$), etc. These issues will be subject of future research.

Moreover, our GSIP solver described in Chapter 5 provides lower bounds at the globally minimal value due to the relaxations of the feasible set. Additional techniques such as restriction on the right-hand-side as proposed in [83] are also applicable to compute valid upper bounds for the objective function on the original feasible set.

Additionally, current state-of-the-art solvers in global optimization are much more developed than our simple branch-and-bound solvers and involve much more sophisticated techniques to problems, for example bound tightening and presolve strategies, that help to reduce the size of the problem in advance. In contrast, our simple branch-and-bound algorithm is tailored to the disjunctive problems that arise in the solution process of generalized semi-infinite problems and thus it has its own advantages. A perfect solution would, of course, have to combine all these ideas in one solver.

Finally, beside the test problems solved in this document, there are many real applications with nested logical expressions or generalized semi-infinite constraints. A few examples are reviewed in Chapter 2. Thus, it would be interesting to see, how the different reformulations and solution methods of this thesis perform on these real-world instances. A thorough preparation of real data provided, a direct application should be possible, at least in theory. Moreover, if necessary, the described algorithms could be adapted to these problems. Again, this is left for future research.

Acknowledgements. This research was partially supported by the DFG (Deutsche Forschungsgemeinschaft) under grant STE 772/14-1.

Bibliography

[1] W. Achtziger, *On simultaneous optimization of truss geometry and topology*, Structural and Multidisciplinary Optimization, 33 (2007), 285-304

[2] W. Achtziger, M. Stolpe, *Truss topology optimization with discrete design variables — Guaranteed global optimality and benchmark examples*, Structural and Multidisciplinary Optimization, 34 (2007), 1-20

[3] C.S. Adjiman, S. Dallwig, C.A. Floudas, A. Neumaier, *A global optimization method, αBB, for general twice-differentiable constrained NLPs – I. Theoretical advances*, Computers & Chemical Engineering, 22 (1998), 1137-1158

[4] C.S. Adjiman, I.P. Androulakis, C.A. Floudas, *A global optimization method, αBB, for general twice-differentiable constrained NLPs – II. Implementation and computational results*, Computers & Chemical Engineering, 22 (1998), 1159-1179

[5] P.P. Alvarez, J.R. Vera, *Application of robust optimization to the sawmill planning problem*, Annals of Operations Research, 219 (2014), 457-475

[6] N. Andalaft, P. Andalaft, M. Guignard, A. Magendzo, A. Wainer, A. Weintraub, *A problem of forest harvesting and road building solved through model strengthening and lagrangean relaxation*, Operations Research, 51 (2003), 613-628

[7] I.P. Androulakis, C.D. Maranas, C.A. Floudas, *αBB: A global optimization method for general constrained nonconvex problems*, Journal of Global Optimization, 7 (1995), 337-363

[8] A. Auslender, M. Teboulle, *Asymptotic Cones and Functions in Optimization and Variational Inequalities*, Springer, New York (2003)

[9] D. Azé, *A survey on error bounds for lower semicontinuous functions*, In: Proceedings of 2003 MODE-SMAI Conference, EDP Sci., Les Ulis, 2003, 1-17

[10] E. Balas, *A note on duality in disjunctive programming*, Journal of Optimization Theory and Applications, 21 (1977), 523-528

[11] E. Balas, *Disjunctive programming*, Annals of Discrete Mathematics, 5 (1979), 3-51

[12] E. Balas, *Disjunctive programming and a hierarchy of relaxations for discrete optimization problems*, SIAM Journal on Algebraic Discrete Methods, 6 (1985), 466-486

[13] E. Balas, S. Ceria, G. Cornuéjols, *A lift-and-project cutting plane algorithm for mixed 0–1 programs*, Mathematical Programming, 58 (1993), 295-324

[14] E. Balas, *Disjunctive programming: properties of the convex hull of feasible points*, Discrete Applied Mathematics, 89 (1998), 3-44

[15] E. Balas, M. Perregaard, *Lift-and-project for mixed 0-1 programming: recent progress*, Discrete Applied Mathematics, 123 (2002), 129-154

[16] J.F. Bard, *Practical Bilevel Optimization*, Kluwer, Dordrecht (1998)

[17] E. Baumann, *Optimal centered forms*, BIT Numerical Mathematics, 28 (1988), 80-87

[18] M.S. Bazaraa, H.D. Sherali, C.M. Shetty, *Nonlinear Programming*, John Wiley & Sons (2013)

[19] N. Beaumont, *An algorithm for disjunctive programs*, European Journal of Operational Research, 48 (1990), 362-371

[20] P. Belotti, *Disjunctive cuts for nonconvex MINLP*, Mixed Integer Nonlinear Programming, Springer New York, 2012, 117-144

[21] M.P. Bendsøe, A. Ben-Tal, J. Zowe, *Optimization methods for truss geometry and topology design*, Structural Optimization, 7 (1994), 141-159

[22] A. Ben-Tal, A. Nemirovski, *Robust truss topology design via semidefinite programming*, SIAM Journal on Optimization, 7 (1997), 991-1016

[23] B. Bhattacharjee, W.H. Green, P. Barton, *Interval methods for semi-infinite programs*, Computational Optimization and Applications, 30 (2005), 63-93

[24] B. Bhattacharjee, P. Lemonidis, W.H. Green, P. Barton, *Global solution of semi-infinite programs*, Mathematical Programming, 103 (2005), 283-307

[25] J.W. Blankenship, J.W. Falk, *Infinitely constrained optimization problems*, Journal of Optimization Theory and Applications, 19 (1976), 261-281

[26] L.G. Bont, H.R. Heinimann, R.L. Church, *Concurrent optimization of harvesting and road network layouts under steep terrain*, Annals of Operations Research, 232 (2015), 41-64

[27] J.A. Caballero, I.E. Grossmann, *Generalized Disjunctive Programming Model for the Optimal Synthesis of Thermally Linked Distillation Columns*, Industrial & Engineering Chemistry Research, 40 (2001), 2260-2274

[28] P.M. Castro, I.E. Grossmann, *Generalized Disjunctive Programming as a Systematic Modeling Framework to Derive Scheduling Formulations*, Industrial & Engineering Chemistry Research, 51 (2012), 5781-5792

[29] S. Ceria, J. Soares, *Convex programming for disjunctive convex optimization*, Mathematical Programming, 86 (1999), 595-614

[30] D.F. Cook, M.L. Wolfe, *Genetic algorithm approach to a lumber cutting optimization problem*, Cybernetics and Systems, 22 (1991), 357-365

[31] R.J. Dakin, *A tree-search algorithm for mixed integer programming problems*, The Computer Journal, 8 (1965), 250-255

[32] S. Dempe, *Foundations of Bilevel Programming*, Kluwer (2002)

[33] M. Diehl, B. Houska, O. Stein, S. Steuermann, *A lifting method for generalized semi-infinite programs based on lower level Wolfe duality*, Computational Optimization and Applications, 54 (2013), 189-210.

[34] W.S. Dorn, R.E. Gomory, H.J. Greenberg, *Automatic design of optimal structures*, Journal de Mécanique, 3 (1964), 25-52

[35] M. Dür, *Dual bounding procedures lead to convergent branch-and-bound algorithms*, Mathematical Programming, 91 (2001), 117-125

[36] M. Dür, *A class of problems where dual bounds beat underestimation bounds*, Journal of Global Optimization, 22 (2002), 49-57

[37] M.A. Duran, I.E. Grossmann, *An outer-approximation algorithm for a class of mixed-integer nonlinear programs*, Mathematical Programming, 36 (1986), 307-339

[38] R. Epstein, R. Morales, J. Serón, A. Weintraub, *Use of OR systems in the Chilean forest industries*, Interfaces, 29 (1999), 7-29

[39] B. Faaland, D. Briggs, *Log bucking and lumber manufacturing using dynamic programming*, Management Science, 30 (1984), 245-257

[40] F. Facchinei, H. Jiang, L. Qi, *A smoothing method for mathematical programs with equilibrium constraints*, Mathematical Programming, 85 (1999), 107-134

[41] R. Fletcher, S. Leyffer, *Solving mixed integer nonlinear programs by outer approximation*, Mathematical Programming, 66 (1994), 327-349

[42] C.A. Floudas, O. Stein, *The adaptive convexification algorithm: a feasible point method for semi-infinite programming*, SIAM Journal on Optimization, 18 (2007), 1187-1208

[43] J.M.P. Geerts, *Mathematical solution for optimising the sawing pattern of a log given its dimensions and its defect core*, New Zealand Journal of Forestry Science, 14 (1984), 124-134

[44] M.A. Goberna, M.A. López, *Linear Semi-infinite Optimization*, John Wiley & Sons, Chichester (1998)

[45] I.E. Grossmann, H. Yeomans, Z. Kravanja, *A rigorous disjunctive optimization model for simultaneous flowsheet optimization and heat integration*, Computers & Chemical Engineering, 22 (1998), S157-S164

[46] I.E. Grossmann, *Review of nonlinear mixed-integer and disjunctive programming techniques*, Optimization and Engineering, 3 (2002), 227-252

[47] I.E. Grossmann, S. Lee, *Generalized convex disjunctive programming: nonlinear convex hull relaxation*, Computational Optimization and Applications, 26 (2003), 83-100

[48] I.E. Grossmann, J.P. Ruiz, *Generalized disjunctive programming: a framework for formulation and alternative algorithms for MINLP optimization*, In: J. Lee, S. Leyffer, *Mixed Integer Nonlinear Programming*,

The IMA Volumes in Mathematics and its Applications, 154 (2012), 93-115

[49] I.E. Grossmann, F. Trespalacios, *Systematic modeling of discrete-continuous optimization models through generalized disjunctive programming*, AIChE Journal, 59 (2013), 3276-3295

[50] F. Guerra Vázquez, J.J. Rückmann, O. Stein, G. Still, *Generalized semi-infinite programming: a tutorial*, Journal of Computational and Applied Mathematics, 217 (2008), 394-419

[51] F. Guerra Vázquez, H.T. Jongen, V. Shikhman, *General semi-infinite programming: symmetric Mangasarian-Fromovitz constraint qualification and the closure of the feasible set*, SIAM Journal on Optimization, 20 (2010), 2487-2503

[52] H. Günzel, H.T. Jongen, O. Stein, *On the closure of the feasible set in generalized semi-infinite programming*, Central European Journal of Operations Research, 15 (2007), 271-280

[53] R. Hettich, K.O. Kortanek, *Semi-infinite programming: theory, methods, and applications*, SIAM Review, 35 (1993), 380-429

[54] R. Hettich, P. Zencke, *Numerische Methoden der Approximation und semi-infiniten Optimierung*, Teubner, Stuttgart (1982)

[55] J.N. Hooker, M.A. Osorio, *Mixed logical-linear programming*, Discrete Applied Mathematics, 96-97 (1999), 395-442

[56] R. Horst, H. Tuy, *Global Optimization: Deterministic Approaches*, Springer, Berlin (1996)

[57] J.R. Jackson, I.E. Grossmann, *A disjunctive programming approach for the optimal design of reactive distillation columns*, Computers & Chemical Engineering, 25 (2001), 1661-1673

[58] F. Jarre, M. Kočvara, J. Zowe, *Optimal truss design by interior point methods*, SIAM Journal on Optimization, 8 (1998), 1084-1107

[59] H.T. Jongen, P. Jonker, F. Twilt, *Critical sets in parametric optimization*, Mathematical Programming, 34 (1986), 333-353

[60] H.T. Jongen, J.J. Rückmann, O. Stein, *Disjunctive optimization: critical point theory*, Journal of Optimization Theory and Applications, 93 (1997), 321-336

[61] H.T. Jongen, J.J. Rückmann, O. Stein, *Generalized semi-infinite optimization: a first order optimality condition and examples*, Mathematical Programming, 83 (1998), 145-158

[62] B. Kent, B.B. Bare, R.C. Field, G.A. Bradley, *Natural resource land management planning using large-scale linear programs: the USDA Forest Service experience with Forplan*, Operations Research, 39 (1991), 13-27

[63] M.W. Kirby, W.A. Hager, P. Wong, *Simultaneous planning of wildland management and transportation alternatives*, TIMS Studies in the Management Sciences, 21 (1986), 371-387

[64] P. Kirst, O. Stein, P. Steuermann, *Deterministic upper bounds for spatial branch-and-bound methods in global minimization with nonconvex constraints*, TOP, 23 (2015), 591-616

[65] P. Kirst, O. Stein, *Solving disjunctive optimization problems by generalized semi-infinite optimization techniques*, Journal of Optimization Theory and Applications, 169 (2016), 1079-1109

[66] P. Kirst, F. Rigterink, O. Stein, *Global optimization of disjunctive programs*, Journal of Global Optimization, 69 (2017), 283-307

[67] P. Kirst, O. Stein, *Global optimization of generalized semi-infinite programs using disjunctive programming*, Journal of Global Optimization, 73 (2018), 1-25

[68] O. Knüppel, *PROFIL/BIAS-a fast interval library*, Computing, 53 (1994), 277-287

[69] M. Kočvara, J. Zowe, *How mathematics can help in design of mechanical structures*, In: *Pitman Research Notes in Mathematics Series* (1996), 76-93

[70] R. Krawczyk, K. Nickel, *Die zentrische Form in der Intervallarithmetik, ihre quadratische Konvergenz und ihre Inklusionsisotonie*, Computing, 28 (1982), 117-137

[71] M. Kurttila, *The spatial structure of forests in the optimization calculations of forest planning — a landscape ecological perspective*, Forest Ecology and Management, 2001 (142), 129-142

[72] S. Lee, I.E. Grossmann, *New algorithms for nonlinear generalized disjunctive programming*, Computers & Chemical Engineering, 24 (2000), 2125-2141

[73] S. Lee, I.E. Grossmann, *A global optimization algorithm for nonconvex generalized disjunctive programming and applications to process systems*, Computers & Chemical Engineering, 25 (2001), 1675-1697

[74] S. Lee, I.E. Grossmann, *Global optimization of nonlinear generalized disjunctive programming with bilinear equality constraints: applications to process networks*, Computers & Chemical Engineering, 27 (2003), 1557-1575

[75] P. Lemonidis, *Global optimization algorithms for semi-infinite and generalized semi-infinite programs*, PhD thesis, Massachusetts Institute of Technology (2008)

[76] E. Levitin, R. Tichatschke, *A branch-and-bound approach for solving a class of generalized semi-infinite programming problems*, Journal of Global Optimization, 13 (1998), 299-315

[77] M. López, G. Still, *Semi-infinite programming*, European Journal of Operational Research, 180 (2007), 491-518

[78] A. Makhorin, *GNU Linear Programming Kit*, Department for Applied Informatics, Moscow Aviation Institute, 2010

[79] D.L. Martell, E.A. Gunn, A. Weintraub, *Forest management challenges for operational researchers*, European Journal of Operational Research, 104 (1998), 1-17

[80] S. Maturana, E. Pizani, J. Vera, *Scheduling production for a sawmill: A comparison of a mathematical model versus a heuristic*, Computers & Industrial Engineering, 59 (2019), 667-674

[81] G.P. McCormick, *Computability of global solutions to factorable nonconvex programs: Part I – Convex underestimating problems*, Mathematical Programming, 10 (1976), 145-175

[82] A. Mitsos, *Global optimization of semi-infinite programs via restriction of the right-hand side*, Optimization 60 (2011), 1291-1308

[83] A. Mitsos, A. Tsoukalas, *Global optimization of generalized semi-infinite programs via restriction of the right hand side*, Journal of Global Optimization, 61 (2015), 1-17

[84] A.T. Murray, R.L. Church, *Measuring the efficacy of adjaceny constraint structure in forest planning models*, Canadian Journal of Forest Research, 25 (1995), 1416-1424

[85] G.L. Nemhauser, L.A. Wolsey, *Integer and Combinatorial Optimization*, John Wiley & Sons, New York (1988)

[86] A. Neumaier, *Interval Methods for Systems of Equations*, Cambridge University Press, Cambridge (1990)

[87] J.S. Pang, *Error bounds in mathematical programming*, Mathematical Programming, 79 (1997), 299-332

[88] E. Polak, *Optimization. Algorithms and Consistent Approximations*, Springer, Berlin (1997)

[89] R. Raman, I.E. Grossmann, *Relations between MILP modelling and logical inference for chemical process synthesis*, Computers & Chemical Engineering, 15 (1991), 73-84

[90] R. Raman, I.E. Grossmann, *Symbolic integration of logic in mixed-integer linear programming techniques for process synthesis*, Computers & Chemical Engineering, 17 (1993), 909-927

[91] R.A. Raman, I.E. Grossmann, *Modelling and computational techniques for logic based integer programming*, Computers & Chemical Engineering, 18 (1994), 563-578

[92] R. Reemtsen, S. Görner, *Numerical methods for semi-infinite programming: a survey*, In: R. Reemtsen, J.J. Rückmann (eds.): *Semi-Infinite Programming*, 195-275, Kluwer, Boston (1998)

[93] M.P. Reinders, Th.H.B. Hendriks, *Lumberproduction optimization*, European Journal of Operational Research, 42 (1989), 243-253

[94] C. ReVelle, S. Snyder, *A shortest path model for the optimal timing of forest harvest decisions*, Enviroment and Planning B: Planning and Design, 23 (1996), 165-175

[95] F. Rigterink, *Ein Branch-and-Bound-Verfahren für disjunktive Optimierungsprobleme*, Master thesis, Karlsruhe Institute of Technology, Institute of Operations Research, 2014

[96] A. Rinnhofer, A. Petutschnigg, J.-P. Andreu, *Internal log scanning for optimizing breakdown*, Computers and Electronics in Agriculture, 41 (2003), 7-21

[97] E.M. Rönnqvist, E. Åstrand, *Integrated defect detection and optimization for cross cutting of wooden boards*, European Journal of Operational Research, 108 (1998), 490-508

[98] E.M. Rönnqvist, *Optimization in forestry*, Mathematical Programming, 97 (2003), 267-284

[99] J.P. Ruiz, I.E. Grossmann, *A hierarchy of relaxations for nonlinear convex generalized disjunctive programming*, European Journal of Operational Research, 218 (2012), 38-47

[100] J.P. Ruiz, I.E. Grossmann, *Using convex nonlinear relaxations in the global optimization of nonconvex generalized disjunctive programs*, Computers & Chemical Engineering, 49 (2013), 70-84

[101] J.P. Ruiz, I.E. Grossmann, *Global optimization of non-convex generalized disjunctive programs: a review on reformulations and relaxation techniques*, Journal of Global Optimization, 67 (2017), 43-58

[102] M. Tawarmalani, N.V. Sahinidis, *Convexification and Global Optimization in Continous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software and Applications*, Springer Science and Business Media, (2002)

[103] N.W. Sawaya, I.E. Grossmann, *A cutting plane method for solving linear generalized disjunctive programming problems*, Computers & Chemical Engineering, 29 (2005), 1891-1913

[104] N.W. Sawaya, I.E. Grossmann, *A hierarchy of relaxations for linear generalized disjunctive programming*, European Journal of Operational Research, 216 (2012), 70-82

[105] H. Scheel, S. Scholtes, *Mathematical programs with complementarity constraints: Stationarity, optimality, and sensitivity*, Mathematics of Operations Research, 25 (2000), 1-22

[106] J. Schwientek, T. Seidel, K.-H. Küfer, *A transformation-based discretization method for solving general semi-infinite optimization problems*, Optimization Online Preprint-ID 2017-12-6380 (2017)

[107] M. Singer, P. Donoso, *Internal supply chain management in the Chilean sawmill industry*, International Journal of Operations & Production Management, 27 (2007), 524-541

[108] E.M. Smith, C.C. Pantelides, *Global optimisation of nonconvex MINLPs*, Computers & Chemical Engineering, 21 (1997), S791-S796

[109] E.M. Smith, C.C. Pantelides, *A symbolic reformulation/spatial branch-and-bound algorithm for the global optimization of nonconvex MINLPs*, Computers & Chemical Engineering, 23 (1999), 457-478

[110] S. Snyder, C. ReVelle, *The grid packing problem: selecting a harvesting pattern in an area with forbidden regions*, Forest Science, 42 (1996), 27-34

[111] O. Stein, G. Still, *On generalized semi-infinite optimization and bilevel optimization*, European Journal of Operational Research, 142 (2002), 444-462

[112] O. Stein, *Bi-level Strategies in Semi-infinite Programming*, Kluwer, Boston (2003)

[113] O. Stein, G. Still, *Solving semi-infinite optimization problems with interior point techniques*, SIAM Journal on Control and Optimization, 42 (2003), 769-788

[114] O. Stein, A. Winterfeld, *Feasible method for generalized semi-infinite programming*, Journal of Optimization Theory and Applications, 146 (2010), 419-443

[115] O. Stein, P. Steuermann, *The adaptive convexification algorithm for semi-infinite programming with arbitrary index sets*, Mathematical Programming, 136 (2012), 183-207

[116] O. Stein, *How to solve a semi-infinite optimization problem*, European Journal of Operational Research, 223 (2012), 312-320

[117] G. Still, *Generalized semi-infinite programming: Theory and methods*, European Journal of Operational Research, 119 (1999), 301-313

[118] G. Still, *Generalized semi-infinite programming: numerical aspects*, Optimization, 49 (2001), 223-242

[119] R.A. Stubbs, S. Mehrotra, *A branch-and-cut method for 0-1 mixed convex programming*, Mathematical Programming, 86 (1999), 515-532

[120] E.F. Thompson, B.G. Haltermann, T.J. Lyon, R.L. Miller, *Integrating timber and wildlife management planning*, The Forestry Chronicle, 49 (1973), 247-250

[121] C.L. Todoroki, E.M. Rönnqvist, *Secondary log breakdown optimization with dynamic programming*, Journal of the Operational Research Society, 48 (1997), 471-478

[122] C.L. Todoroki, E.M. Rönnqvist, *Combined primary and secondary log breakdown optimisation*, Journal of the Operational Research Society, 50 (1999), 219-229

[123] C.L. Todoroki, E.M. Rönnqvist, *Dynamic control of timber production at a sawmill with log sawing optimization*, Scandinavian Journal of Forest Research, 17 (2002), 79-89

[124] F. Trespalacios, I.E. Grossmann, *Review of Mixed-Integer Nonlinear and Generalized Disjunctive Programming Methods*, Chemie Ingenieur Technik, 86 (2014), 991-1012

[125] A. Tsoulakas, B. Rustem, E.N. Pistikopoulos, *A global optimization algorithm for generalized semi-infinite, continuous minimax with coupled constraints and bi-level problems*, Journal of Global Optimization, 44 (2009), 235-250

[126] M. Türkay, I.E. Grossmann, *Disjunctive Programming Techniques for the Optimization of Process Systems with Discontinuous Investment Costs-Multiple Size Regions*, Industrial & Engineering Chemistry Research, 35 (1996), 2611-2623

[127] M. Varas, S. Maturana, R. Pascual, I. Vargas, J. Vera, *Scheduling production for a sawmill: A robust optimization approach*, International Journal of Production Economics, 150 (2014), 37-51

[128] A. Vecchietti, I.E. Grossmann, *LOGMIP: a disjunctive 0–1 nonlinear optimizer for process systems models*, Computers & Chemical Engineering, 21 (1997), S427-S432

[129] F.B. Veliz, J.-P. Watson, A. Weintraub, R.J.-B. Wets, D.L. Woodruff, *Stochastic optimization models in forest planning: a progressive hedging solution approach*, Annals of Operations Research, 232 (2015), 259-274

[130] A. Wächter, L.T. Biegler, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, Mathematical Programming, 106 (2006), 25-57

[131] A. Weintraub, R.L. Church, A.T. Murray, M. Guignard, *Forest management models and combinatorial algorithms: analysis of state of the art*, Annals of Operations Research, 96 (2000), 271-285

[132] W. Weintraub, A.T. Murray, *Review of combinatorial problems induced by spatial forest harvesting planning*, Discrete Applied Mathematics, 154 (2006), 867-879

[133] A. Weintraub, D. Navon, *A forest management planning model integrating silvicultural and transportation acitivities*, Management Science, 22 (1976), 1299-1309

[134] A. Weintraub, C. Romero, *Operations research models and the management of agriculture and forestry resources: a review and comparison*, Interfaces, 36 (2006), 446-457

[135] T. Westerlund, F. Pettersson, *An extended cutting plane method for solving convex MINLP problems*, Computers & Chemical Engineering, 19 (1995), 131-136

[136] H.P. Williams, *Model Building in Mathematical Programming*, John Wiley & Sons (2013)

[137] H. Yeomans, I.E. Grossmann, *Nonlinear disjunctive programming models for the synthesis of heat integrated distillation sequences*, Computers & Chemical Engineering, 23 (1999), 1135-1151

[138] H. Yeomans, I.E. Grossmann, *Disjunctive Programming Models for the Optimal Design of Distillation Columns and Separation Sequences*, Industrial & Engineering Chemistry Research, 39 (2000), 1637-1648

[139] H. Yeomans, I.E. Grossmann, *Optimal Design of Complex Distillation Columns Using Rigorous Tray-by-Tray Disjunctive Programming Models*, Industrial & Engineering Chemistry Research, 39 (2000), 4326-4335

[140] M.K. Zanjani, D. Ait-Kadi, M. Nourelfath, *Robust production planning in a manufacturing environment with random yield: A case in sawmill production planning*, European Journal of Operational Research, 201 (2010), 882-891

Eidesstattliche Versicherung

Eidesstattliche Versicherung gemäß §13 Absatz 2 Ziffer 3 der Promotionsordnung des Karlsruher Instituts für Technologie für die KIT-Fakultät für Wirtschaftswissenschaften

1. Bei der eingereichten Dissertation zu dem Thema "*Disjunctive Aspects in Generalized Semi-infinite Programming*" handelt es sich um meine eigenständig erbrachte Leistung.

2. Ich habe nur die angegebenen Quellen und Hilfsmittel benutzt und mich keiner unzulässigen Hilfe Dritter bedient. Insbesondere habe ich wörtlich oder sinngemäß aus anderen Werken übernommene Inhalte als solche kenntlich gemacht.

3. Die Arbeit oder Teile davon habe ich bislang nicht an einer Hochschule des In- oder Auslands als Bestandteil einer Prüfungs- oder Qualifikationsleistung vorgelegt.

4. Die Richtigkeit der vorstehenden Erklärungen bestätige ich.

5. Die Bedeutung der eidesstattlichen Versicherung und die strafrechtlichen Folgen einer unrichtigen oder unvollständigen eidesstattlichen Versicherung sind mir bekannt.

Ich versichere an Eides statt, dass ich nach bestem Wissen die reine Wahrheit erklärt und nichts verschwiegen habe.

Ort und Datum Unterschrift