Mixed-Integer Semidefinite Programming with an Application to Truss Topology Design

Gemischt-ganzzahlige semidefinite Programme mit einer Anwendung in der Topologie-Optimierung von Stabwerken

Der Naturwissenschaftlichen Fakultät
der Friedrich-Alexander-Universität Erlangen-Nürnberg
zur
Erlangung des Doktorgrades Dr. rer. nat.

vorgelegt von
Sonja Mars
aus Groß-Gerau

Als Dissertation genehmigt von der Naturwissenschaftlichen Fakultät der
Friedrich-Alexander-Universität Erlangen-Nürnberg

Tag der mündlichen Prüfung:

Vorsitzender der
Promotionskommission: Prof. Dr. Rainer Fink

Erstberichterstatter: Prof. Dr. Alexander Martin

Zweitberichterstatter: Prof. Dr. Marc Pfetsch

Danksagung

Der Weg ist das Ziel – nun, kurz vor Ende meiner Zeit als Promotionsstudentin kann ich diesen Schluss ziehen. Ich hatte eine sehr schöne Zeit mit vielen Herausforderungen unterschiedlichster Art, viel Abwechslung und habe viel Neues gesehen. Ich habe eine neue Uni kennengelernt, war viel unterwegs und bin nun um einige Erfahrungen reicher.

An dieser Stelle möchte ich den vielen Menschen danken, die mich auf dem Weg zu dieser Arbeit unterstützt haben. Zu Beginn sei die DFG und der Sonderforschungsbereich (SFB) 805 erwähnt. Ich danke der DFG, die mich im Rahmen des SFB 805 unter dem Thema *Beherrschung von Unsicherheit in lasttragenden Systemen des Maschinenbaus* finanziert hat und mir so die Möglichkeit zu dieser Arbeit eröffnete. Dieser SFB hat mir viele interessante Einblicke in die Welt des Maschinenbaus gewährt. Ohne die Zusammenarbeit mit den Darmstädter Ingenieuren wäre das Stabwerks-Kapitel nur halb so realitätsnah. Ich möchte mich an dieser Stelle beim gesamten SFB-Team bedanken. Es war eine tolle, produktive Zeit und wir haben viele interessante Projekte angestoßen und durchgeführt.

Ich danke meinen zahlreichen Chefs, Alexander Martin, Stefan Ulbrich und Marc Pfetsch für die Finanzierung und Unterstützung meiner Arbeit. Danke, dass ihr an mich geglaubt habt und mir nie etwas fehlte. Alex, wenn du mich nicht überredet hättest, an einem Freitagabend vor Weihnachten, dem Tag meiner allerletzten Diplomprüfung, wäre ich wohl nie soweit gekommen. Danke auch für die Freiheit, die du mir gelassen hast und für die Möglichkeit auch mal etwas ganz anderes zu machen. Stefan, danke dass ich weiter in Darmstadt angestellt sein durfte und du die Verantwortung übernommen hast, nachdem wir eigentlich in Erlangen waren. Danke, dass du dich um mich gekümmert hast, obwohl ich eigentlich keine deiner Doktorandinnen war und danke vor allem für die gute Arbeitsatmosphäre in Darmstadt. Marc, vielen Dank für das vergangene Jahr mit vielen spannenden Einblicken in SCIP. Danke, dass du mich übernommen hast, obwohl ich auch keine deiner Doktorandinnen war.

Ganz besonders möchte ich meinem eigentlichen Betreuer Lars Schewe danken. Viele endlose Stunden haben wir versucht zu verstehen, wie welcher Löser arbeitet, wir haben zahlreiche Ansätze verfolgt und einige Fehlschläge erlitten. Danke für die Diskussionen und Ideen, die Reflexionen und Herausforderungen. Danke auch dem A-Team für einige lustige Monate im Land der Heuristiken. Ein Dankeschön gebührt auch meinen Kollegen in Darmstadt, die immer für eine schöne Arbeitsatmosphäre und ein lustiges Mittagessen sorgen.

Vielen Dank auch meinen Korrekturlesern (Jan Essert, Sarah Essert, Björn Geißler, Christine Hayn, Silke Horn, Susanne Pape, Madeline Lips und Jakob Schelbert), die so manchen Tippfehler und falsch benannten Index gefunden und einige Ungereimtheiten aufgedeckt haben.

Ein ganz besonders herzlicher Dank gebührt auch meiner Familie, die mich tatkräftig unterstützt hat, sich alle meine Klagen anhörte und ganz tapfer die Stäbchenforschung Korrektur las. Jana, tut mir leid, dass die soziologischen Aspekte zu kurz kommen. Und nun kommt noch ein Danke, dass so groß ist, dass ich es eigentlich nicht in Worte fassen kann. Lieber Andi, danke fürs Zuhören, kritische Fragen stellen, Diskutieren, Verstehen, Überreden, Aufmuntern, wieder auf den Teppich holen, Beruhigen, Kraft geben. Danke für ALLES. Ohne dich hätte ich die Promotion nicht angefangen, ich wäre nicht einmal auf die Idee gekommen, und hätte zwischendurch schon zweimal abgebrochen. Ohne dich sähe mein LaTeX-Code aus wie Kraut und Rüben und ich wäre jetzt vermutlich wahnsinnig. Danke.

Deutsche Zusammenfassung

Viele Anwendungsprobleme, aber auch klassische diskrete Optimierungsprobleme lassen sich als gemischt-ganzzahlige semidefinite Probleme (MISDP) formulieren. So hat neben dem Problem des maximalen Schnittes auch die Topologie-Optimierung von Stabwerken eine semidefinite Formulierung. Im Rahmen eines Projektes im Sonderforschungsbereich 805, das sich mit der optimalen Kombination von aktiven und passiven Bauteilen in Stabwerken befasst, treten zusätzlich binäre Variablen auf, welche die Position der aktiven Elemente modellieren. Damit wird aus dem rein kontinuierlichen SDP ein gemischt-ganzzahliges SDP, das es zu lösen gilt. Die SDP-Formulierung des Problems des maximalen Schnittes besitzt hingegen nur ganzzahlige Variablen.

In dieser Arbeit beschäftigen wir uns mit diesen beiden sehr unterschiedlichen Problemen. Wir formulieren beide als MISDP und zeigen zwei verschiedene Lösungswege auf, die beide auf dem Branch-und-Bound-Algorithmus beruhen. Außerdem vergleichen wir deren Modellierung als MISDP mit einem gemischt-ganzzahligen linearen Modell und zeigen, dass die MISDP-Formulierung deutliche Vorteile bietet.

Software zum Lösen von allgemeinen MISDPs sind der Autorin nicht bekannt. Zwar gibt es einige Ansätze, spezielle Probleme, wie beispielsweise das Problem des maximalen Schnitts, zu lösen, jedoch werden allgemeine MISDPs bisher nicht betrachtet. Ziel dieser Arbeit ist es, allgemeine MISDPs zu lösen. Aus diesem Grund haben wir ein Software-Paket entwickelt, welches zusammen mit dem Branch-und-Bound-Framework SCIP [SCI12] und dem Innere-Punkte-Löser DSDP [BY08], allgemeine MISDPs lösen kann. Dabei werden anstatt der üblichen LP-Relaxierung SDP-Relaxierungen in jedem Knoten des Branch-und-Bound-Baumes gelöst. Zusätzlich gibt es spezielle Presolving-Routinen für SDPs. Außerdem schlagen wir einen zweiten Lösungsweg vor, der ganz ohne SDP-Löser auskommt.

Nach einer kurzen Einleitung im ersten Kapitel werden in zweiten Kapitel die Grundlagen von Semidefinitheit eingeführt. Neben einigen wichtigen Fakten über positiv semidefinite Matrizen steht der Kegel der positiv semidefiniten Matrizen im Vordergrund. Zudem wird die Problemklasse der gemischt-ganzzahligen semidefiniten Probleme eingeführt.

Das dritte Kapitel beschäftigt sich mit dem Lösen der zuvor eingeführten Probleme. Dazu eignen sich vor allem Innere-Punkte-Verfahren. Diese werden ausführlich beschrieben, da sie später im entwickelten Löser eingesetzt werden. Außerdem führen wir zwei weitere Lösungsverfahren ein, Bündel-Methoden und das Augmented Lagrange-Verfahren, und begründen die Entscheidung für Innere-Punkte-Verfahren im MISDP-Löser. Anschließend

führen wir das Branch-und-Bound-Verfahren ein und erweitern es für semidefinite Probleme. Besonderes Augenmerk liegt dabei auf den Unterschieden eines Branch-und-Bound-Algorithmus, der lineare Relaxierungen löst und einem der semidefiniten Relaxierungen löst. Wir stellen diese Unterschiede heraus, identifizieren auftretende Probleme und schlagen Lösungsansätze vor. Zudem zeigen wir eine weitere Variante zur Lösung von MISDPs auf, die keinen SDP-Löser benötigt.

Details des entwickelten Lösers werden im vierten Kapitel betrachtet. Neben Datenstrukturen, Einleseroutine und Solver-Schnittstelle sind vor allem das Lösen eines einzelnen Branch-und-Bound-Knoten und das Behandeln der semidefiniten Nebenbedingungen im Fokus dieses Kapitels.

Im fünften Kapitel widmen wir uns dem Anwendungsproblem der Stabwerks-Topologie Optimierung. Wir präsentieren das nichtlineare Grundmodell, sowie drei äquivalente Modelle: eine gemischt-ganzzahlig lineare Modellierung, eine quadratische Formulierung und das semidefinite Modell. Diese Modelle erweitern wir für diskrete Stabdicken und betrachten verschiedene Modellierung dieses diskreten Problems. Weiterhin zeigen wir zusätzliche Stabilitätsnebenbedingungen, die sich im Rahmen des Projekt im SFB 805 ergeben haben. Aktive Elemente, sogenannte Aktoren, werden innerhalb des Stabwerks platziert, um seine Stabilität zu erhöhen. Eine weitere Maßnahme, um das Stabwerk ausfallsicherer zu machen sind multiple Lastfälle, diese werden ebenfalls in den Modellen berücksichtigt. Schließlich betrachten wir Vibrationsbedingungen, die die Schwingungen im Stabwerk begrenzen sollen.

Für dieses Anwendungsproblem und seine verschiedenen Ausprägungen zeigen wir im sechsten Kapitel Zusamenfassungen unserer ausführlichen Rechenresultate, die sowohl die verschiedenen Modelle für Stabwerke vergleichen, als auch die Vorteile unserer Software aufzeigen. Wir testen verschiedene Parametereinstellungen und vergleichen den MISDP-Ansatz mit einer gemischt-ganzzahlig linearen Problemformulierung.

Ähnliche Rechenresultate präsentieren wir auch im siebten Kapitel für das kombinatorische Optimierungsproblem des maximalen Schnittes. Dieses sehr klassische Problem ist ebenfalls als SDP formulierbar. Schließlich fassen wir im achten Kapitel die Resultate noch einmal zusammen und geben einen kurzen Ausblick über weitere Herausforderungen. Die ausführlichen Rechenresultate zeigen wir im Anhang, um den Lesefluss in den einzelnen Kapiteln nicht zu sehr zu behindern.

Abstract

Mixed-integer semidefinite programming is an evolving field of research. Classical combinatorial optimization problems as well as various different applications can be modeled using semidefinite programs. One such application which can be modeled as semidefinite program (SDP) is the optimization of the topology of trusses. Within a project of the *Collaborative Research Center 805 - Control of Uncertainty in Load-Carrying Structures in Mechanical Engineering at TU Darmstadt* active components need to be positioned in the truss. This positioning is done using binary variables and leads to mixed-integer semidefinite programs (MISDP).

We provide a code package for SCIP [SCI12] that uses the SDP solver DSDP [BY08] and is able to solve general MISDPs. To the knowledge of the author, such a general solver was not available before. To demonstrate the generality of our software, we apply it to the Maximum Cut Problem and Truss Topology Design.

We introduce the required background from linear algebra and present different solving strategies for continuous SDPs. Moreover, we discuss the specific problems when solving SDPs within a branch-and-bound algorithm. Ideas to speed up the branch-and-bound process for MISDPs are also presented. Furthermore, we state a method for approximating the semidefinite cone and comment on the implementation.

Both applications are introduced and different models for numerous extensions are shown. Finally we present very detailed computational results.

Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

Mixed-integer semidefinite programming is an evolving field of research. There are various classical combinatorial optimization problems as well as different applications that can be modeled using semidefinite programs. For example, optimizing the topology of trusses is one such application which can be modeled as semidefinite program (SDP).

To obtain more realistic models or to position special bars within these trusses, binary or integer variables are required. The resulting models are mixed-integer semidefinite programs (MISDP).

Combining the ideas of semidefinite programming and discrete optimization has already been done before, but not yet in the general way we did. There are multiple approaches which use semidefinite programming to compute good relaxations of combinatorial optimization problems. However, most of these approaches only use SDP relaxations to obtain a better lower bound in the root node of a branch-and-bound tree or they only consider one special kind of problems such as the Maximum Cut Problem. The task of this thesis is to consider general discrete semidefinite programs.

We are going to present different solution strategies and useful methods to speed up the solution process. We show that it is possible to solve the MISDP using a pure branch-and-bound algorithm with SDP relaxations instead of linear programming (LP) relaxations. Different algorithms for solving SDPs are discussed and the best one is chosen for our software. Furthermore, the SDP can be approximated using some linear inequalities in a branch-and-cut procedure. Additionally, we introduce some techniques for presolving and heuristics.

We provide a code package for SCIP [SCI12], in which we implemented most of the ideas mentioned above. This package is open source and available on the author's website: `www.opt.tu-darmstadt.de/∼smars/scip_sdp.html`. There is a general interface for SDP solvers and an example implementation for the solver DSDP [BY08]. To illustrate that the ideas of this thesis work for other frameworks than SCIP, we also implemented our ideas in the branch-and-bound framework Acro-Pebbl [EPH01] as well.

In addition to the software, we present two applications: Truss Topology Design and the Maximum Cut Problem. The first one is the main motivation of this thesis and it is discussed in detail. We give an overview of the existing models and extend them to different new features. Moreover, we compare the different models and show their advantages and disadvantages. Many computational results on our own test set as well as on instances from the literature are shown for these models.

To show that our software works for general SDPs, we discuss the Maximum Cut Problem. This problem is validated and analyzed using a randomly generated test set of dense graphs.

This thesis shows that linearizing is not always a good idea. A pure mixed-integer linear formulation is in many cases not able to solve larger instances. For small problems, the linear formulation is faster in most of the cases. However, if the problems are getting closer to real world problems, with more complicated constraints, the linear models fail. It is better to use SDP formulations if they are available. We could speed up the solution process, even though solving SDPs take a very long time in comparison to solving an LP and we did not use the fastest SDP solver available.

Additionally, for complicated problem structure MIP solvers fail due to numerical problems, whereas our solver does not. Of course, we cannot compete with problem specific solvers because we do not exploit the specific problem structure, but our package can easily be extended using problem specific cuts or branching techniques, resulting in improved speed. Also using another SDP solver in every branching node is no problem. We provide an interface that may also be used for other solvers.

SCIP is a general constraint integer programming solver that was recently extended to nonconvex mixed-integer nonlinear problems Using our package it can now also solve mixed-integer semidefinite programming problems, and is therefore the most general mixed-integer solver available at the moment.

An overview

This thesis is structured as follows. First we give a short overview about semidefinite programming and semidefinite matrices. In Chapter 2 we state some important facts about the cone of positive semidefinite matrices and recall the strong duality theorem of semidefinite programming. We observe properties of semidefinite matrices and linear matrix inequalities which will be needed in the following chapters.

Chapter 3 shows different solving strategies for continuous SDPs. An overview is given for interior-point-solvers, the augmented Lagrangian procedure, and the spectral bundle method. With these methods, the lower bounds for the branch-and-bound algorithm are computed. This chapter also provides ideas how the branch-and-bound process can be accelerated. There are comments on heuristics, presolve, and branching strategies. Finally, this chapter presents additional techniques that can be helpful in the context of approximating the semidefinite constraints.

Implementation details of the SCIP package will be shown in Chapter 4. Here we present the implementation of the different callbacks and discuss the data structures we use. We also comment on the difficulties we investigated while implementing the package.

Our main motivation for looking at MISDPs is given by a project in the *Collaborative Research Center 805 - Control of Uncertainty in Load-Carrying Structures in Mechanical Engineering at TU Darmstadt*. This motivation consists of, among others, trusses which are presented in Chapter 5. In this chapter we compare three different models for Truss Topology Design. We extend these models to different constraints. We consider stress constraints and vibrations. Additionally, we show the summary of extensive computational results in Chapter 6. We compare the different approaches for solving MISDPs. Besides we show that the MISDPs models can be solved faster than the MIPs for the same instances.

The second, completely different, application is shown in Chapter 7. There we discuss the Maximum Cut Problem. Within this context we discovered a big difference between the SDPs arising from combinatorial optimization problems in the canonical way and those SDPs, where the semidefinite constraint is rooted in the physics of the problem. This assumption is illustrated by some computational results.

A conclusion and a short outlook are given in Chapter 8. Finally, we present the details of the computational results of the Maximum Cut Problem and the Truss Topology Design problem in the Appendix.

CHAPTER 2

Semidefinite programming

The main focus of this thesis lies on mixed-integer semidefinite programming, so in the beginning we recall some of the most important definitions and results of semidefinite programming, semidefinite matrices, and duality theory.

Semidefinite programs (SDP) are widely discussed in the literature. A very good introduction from a conic point of view is given in [BTN01]. The authors present many examples and introduce SDPs via conic programming. They also give an overview about what kinds of problems are representable as conic quadratic or semidefinite programming problems. In this chapter we show that the set of semidefinite matrices – viewed as a subset of the real vector space of all matrices – is convex. This is the starting point of [Bar02] to look at SDPs. Many examples for semidefinite programming are considered in [VB96]. Moreover, this paper gives an introduction to primal-dual interior-point methods for solving SDPs. Another way of solving SDPs using bundle-methods is presented in [Hel00]. In addition to a detailed introduction to SDPs this work includes the treatment of plenty combinatorial optimization problems and their SDP relaxations. An extensive insight to all kinds of problems and theoretical aspects related to SDPs can be found in two different books called *Handbook of Semidefinite Programming* [LA11] and [WSV00]. Different authors take a look at theory, algorithms, applications, and software.

This thesis also covers these four different topics: theory, algorithms, applications, and software. We begin with some theory in this chapter. After some important facts about semidefinite matrices in the next section, we look at the cone of semidefinite matrices in Section 2.2. In Section 2.3 we introduce a primal and dual semidefinite program and define mixed-integer semidefinite programs. Finally, Section 2.4 introduces duality theory of semidefinite programming. Algorithms for solving SDPs will be introduced in Chapter 3. The software for solving mixed-integer semidefinite programs (MISDP) is described in Chapter 4. In Chapters 5, 6, and 7 we present two very different applications with computational results.

Most of the results and proofs presented in this chapter are well-known from the literature. The idea was to bring all the material together needed for understanding and solving MISDPs. Some of the observations and propositions are very MISDP-specific and therefore we adapted them as well as some of the proofs.

2.1 Linear algebra background

In this section we will first state the definition of positive semidefiniteness and recall some facts about positive semidefinite matrices. Since we are only interested in symmetric matrices, we define positive semidefiniteness to imply symmetry. For the sake of convenience we introduce the following notation we will use throughout this chapter.

Notation 2.1.1. We denote by

- $M_{n,m}$ the set of real $n \times m$ matrices,
- M_n the set of real quadratic $n \times n$ matrices,
- S_n the set of real symmetric $n \times n$ matrices,
- S_n^+ the set of real symmetric positive semidefinite $n \times n$ matrices.

We start with some very basic results for symmetric matrices, which we will state without proof. The Spectral Theorem is presented in a version for general quadratic Hermitian matrices.

Theorem 2.1.2 ([HJ90, Spectral Theorem 2.5.6])**.** *If $A \in M_n$ is Hermitian, then*

(i) all eigenvalues of A are real and

(ii) A is unitarily diagonalizable.

If $A \in S_n$, then A is orthogonally diagonalizable, i.e., there is an orthogonal matrix $P \in M_n$ and a diagonal matrix $\Lambda \in S_n$ with $A = P\Lambda P^T$.

Corollary 2.1.3. *All eigenvalues of a symmetric real matrix A are real.*

Recall the characterization of positive semidefiniteness of a matrix A.

Proposition 2.1.4. *For $A \in S_n$ the following statements are equivalent:*

(i) all eigenvalues of A are nonnegative and

(ii) for all $v \in \mathbb{R}^n$ the value $v^T A v$ is nonnegative.

Proof. Let λ_i be the eigenvalues of A.

(ii) $\Rightarrow$ (i): Let $\lambda_0(A)$ be the smallest eigenvalue of A and let v_0 be the corresponding eigenvector. Then we know that $Av_0 = \lambda_0 v_0$ and by (ii) we have $0 \leq v_0^T A v_0 = v_0^T \lambda_0 v_0 = \lambda_0 v_0^T v_0 = \lambda_0 \|v_0\|^2$. As $\|v_0\|^2 \geq 0$ we have $\lambda_0 \geq 0$. Since λ_0 is the smallest eigenvalue all eigenvalues must be nonnegative.

(i) $\Rightarrow$ (ii): Let $A = P\Lambda P^T$ be the eigenvalue decomposition of A with diagonal matrix Λ. As all eigenvalues of A are nonnegative we can write $\Lambda = \Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}}$, where $\Lambda_{ii}^{\frac{1}{2}} = \sqrt{\Lambda_{ii}}$. Using this we have $v^T A v = v^T P \Lambda^{\frac{1}{2}} \Lambda^{\frac{1}{2}} P^T v = (\Lambda^{\frac{1}{2}} P^T v)^T \Lambda^{\frac{1}{2}} P^T v \geq 0$. □

Definition 2.1.5. A real symmetric $n \times n$ matrix A is called *positive semidefinite*, denoted by $A \succeq 0$ if it satisfies one of the conditions of Proposition 2.1.4. □

There is another important characterization of positive semidefiniteness which we will use in the following.

Proposition 2.1.6. *A matrix A is positive semidefinite ($A \in S_n^+$) if and only if all principal minors of A are positive semidefinite.*

Proof. One direction is trivial because the full matrix is a minor of itself and if every minor of A is positive semidefinite, then so is A.

Conversely, let I be the index set of the rows of A and $J \subseteq I$. By $A_J = \{a_{ij}\}_{i,j \in J}$ we denote the minor of A consisting only of the rows and columns of A whose indices are in J. For every principal minor A_J we consider the vectors $\hat{v} \in \mathbb{R}^n$ with $\hat{v}_j = 0$ for $j \notin J$. Let $\hat{v}_J = \{v_j\}_{j \in J}$ be such a reduced vector in dimension $|J|$ then we have

$$\hat{v}^T A \hat{v} \geq 0 \qquad \Longleftrightarrow \qquad \hat{v}_J^T A_J \hat{v}_J \geq 0.$$

If $v^T A v \geq 0$ for all vectors $v \in \mathbb{R}^n$, it is true in particular for the vectors $\hat{v}$ considered above. This implies that A_J is positive semidefinite. □

Proposition 2.1.7. *If A is positive semidefinite then its determinant is nonnegative, i.e.,* $\det(A) \geq 0$.

Proof. Let A be positive semidefinite, then by Proposition 2.1.4 all eigenvalues of A are nonnegative. But $\det(A) = \prod_i \lambda_i(A)$ and since $\lambda_i(A) \geq 0$ for all $i = 1, \ldots, n$ this product has to be nonnegative. So $\det(A) \geq 0$. □

Note that in Proposition 2.1.6 it is necessary to consider all principal minors not only the leading ones, cf. [Swa73]. In particular all 2×2 principal minors must be positive semidefinite; using this and Proposition 2.1.7 leads to the following quadratic inequality

$$a_{ii}a_{jj} - a_{ij}^2 \geq 0.$$

This inequality shows directly that a_{ij} has to be zero if one of a_{ii} or a_{jj} is zero or equivalently that neither a_{ii} nor a_{jj} may be zero if a_{ij} is not.

Proposition 2.1.8. *For a positive semidefinite matrix A with $a_{ii} = 0$ all entries in row i and (due to symmetry) in column i must be zero.*

We will use this fact later for determining infeasibility of problems within the branch-and-bound algorithm. Using Proposition 2.1.6 we observe the following.

Observation 2.1.9. *A block diagonal matrix A is positive semidefinite if and only if all of its block components are.*

Furthermore, a diagonal-dominant matrix with nonnegative diagonal entries is positive semidefinite, i.e.,

$$|a_{ii}| \geq \sum_{j=1, j\neq i}^{n} |a_{ij}| \quad \text{and} \quad a_{ii} \geq 0 \quad \text{for } i = 1, \ldots, n \quad \text{implies} \quad A \succeq 0.$$

Moreover, the definition of positive definite matrices implies more properties on the diagonal entries. The following two observations are taken from [Hel00].

Observation 2.1.10. *Let A be a positive semidefinite matrix, then its diagonal entries are nonnegative and therefore*

$$a_{ii} = |a_{ii}|\,.$$

Moreover, one of the elements of largest absolute value is always a diagonal element:

$$\textit{there exists } i \in \{1, \ldots, n\} \textit{ with } a_{ii} = \max\{|a_{ij}| : i, j = 1, \ldots, n\}$$

Additionally, we may derive a lower bound for the largest eigenvalue $\lambda_{\max}$.

Observation 2.1.11. *Let A be positive semidefinite. Then by the Rayleigh-Ritz-Theorem [HJ90] the largest eigenvalue of A satisfies $\lambda_{\max}(A) = \max_{v\in\mathbb{R}^n, \|v\|=1} v^T A v$. In particular, it follows that*

$$\lambda_{\max}(A) \geq \max\{a_{ii} \colon i = 1, \ldots, n\}.$$

The following proposition is taken from [Hel00] and shows that positive semidefiniteness is invariant under basis transformation with respect to orthogonal matrices.

Proposition 2.1.12. *Let $B \in M_n$ be nonsingular and $A \in S_n$ be symmetric. Then A is positive semidefinite if and only if $B^T AB$ is positive semidefinite.*

Proof. Let $v \in \mathbb{R}^n$ and $w = B^{-1}v$. Then

$$v^T A v = v^T(B^{-T}B^T)A(BB^{-1})v = (v^T B^{-T})B^T AB(B^{-1}v) = w^T B^T ABw.$$

So $v^T A v \geq 0$ if and only if $w^T B^T ABw \geq 0$. □

Using this result together with Observation 2.1.9 we now deduce an important result for reducing the size of the matrix. Before being able to define the so-called Schur complement of a matrix we need another definition.

Definition 2.1.13. A matrix $A \in S_n$ with $v^T A v > 0$ for all $v \in \mathbb{R}^n$ with $v \neq 0$ is called *positive definite.* □

Positive definiteness implies nonsingularity. With this definition we are now able to define the Schur complement:

Definition 2.1.14. Let $X \in \mathbb{R}^{(n+m)\times(n+m)}$ be a matrix with the following structure:

$$X = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix},$$

where A is a positive definite $m \times m$ matrix, $C \in S_n$ is symmetric, and $B \in M_{m,n}$. Then the matrix

$$X_{SC} = C - B^T A^{-1} B \in \mathbb{R}^{n\times n}$$

is called the *Schur complement* of X. □

Note that X_{SC} is of dimension $n \times n$, which can be much smaller than $(n+m)\times(n+m)$. The Schur complement of a matrix can be used to reduce the size of a matrix as the following theorem shows.

Theorem 2.1.15 ([Hel00, Theorem 1.1.9])**.** *The matrix X of Definition 2.1.14 is positive (semi)definite if and only if X_{SC} is that is*

$$X = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix} \succ 0 \qquad \Longleftrightarrow \qquad X_{SC} = C - B^T A^{-1} B \succ 0$$

and

$$X = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix} \succeq 0 \qquad \Longleftrightarrow \qquad X_{SC} = C - B^T A^{-1} B \succeq 0.$$

Proof. In accordance with Definition 2.1.14 the matrix A is positive definite. Therefore we may define a nonsingular matrix Z:

$$Z = \begin{pmatrix} I_m & -A^{-1}B \\ 0 & I_n \end{pmatrix}.$$

Now consider

$$Z^T X Z = \begin{pmatrix} I_m & 0 \\ -B^T A^{-T} & I_n \end{pmatrix} \begin{pmatrix} A & B \\ B^T & C \end{pmatrix} \begin{pmatrix} I_m & -A^{-1}B \\ 0 & I_n \end{pmatrix} = \begin{pmatrix} A & 0 \\ 0 & C - B^T A^{-1} B \end{pmatrix}.$$

Using Observation 2.1.9 the resulting matrix $Z^T X Z$ is positive (semi)definite if and only if the matrices A and $C - B^T A^{-1} B$ are. As A is positive definite by definition, with Proposition 2.1.12 it follows that X is positive (semi)definite if and only if $C - B^T A^{-1} B$ is positive (semi)definite. □

These characterizations are very helpful in the branch-and-bound algorithm for checking the constraints or separating a solution. Later in Section 3.4 we will derive an approximation scheme using (ii) of Proposition 2.1.4 and the eigenvectors v of the matrix A.

We will also use Theorem 2.1.15 for reducing the size of the matrix. This is a crucial fact because the size of the problem formulation and the speed of the solver directly depends on the size of the matrices involved. More properties of semidefinite matrices can be found in [Hel00].

2.2 The semidefinite cone

Our aim is to be able to optimize over the set of positive semidefinite matrices. We will show that this set has the structural properties of a cone and it will be called the semidefinite cone in the following. In this section we want to understand this cone and take a closer look at it. Again we refer to [Hel00], where most of the definitions and results are stated. Another relevant source is [Bar02], where some of the following results are taken from.

First we define a cone and look at some of its properties:

Definition 2.2.1. (i) A set $C \subseteq \mathbb{R}^n$ is a *cone* if $0 \in C$ and it is closed under nonnegative multiplication, i.e.,

$$\lambda \geq 0 \text{ and } x \in C \quad \Longrightarrow \quad \lambda x \in C.$$

(ii) A cone C is *convex* if for any $x, y \in C$ and any $\lambda \in [0, 1]$ the convex combination of x and y is also in the cone, i. e.

$$z = \lambda x + (1 - \lambda) y \in C.$$

(iii) A cone C is *pointed* if $C \cap (-C) = \{0\}$. □

Note that our definition of a cone does not imply that a cone is convex. Nonempty, pointed, convex cones $C \subseteq \mathbb{R}^n$ have some nice properties, a very important one is stated in [BTN01].

Proposition 2.2.2. *Every nonempty, pointed, convex cone K induces a partial ordering $\geq_K$ (or short $\geq$) on $\mathbb{R}^n$, where*

$$A \geq B \quad \Longleftrightarrow \quad A - B \in K.$$

This partial ordering satisfies the following four axioms for $A, B, C, D \in K$, and $\lambda \in \mathbb{R}_0^+$:

(i) reflexivity: $A \geq A$,

(ii) antisymmetry: if $A \geq B$ and $B \geq A$ then $A = B$,

(iii) transitivity: if $A \geq B$ and $B \geq C$ then $A \geq C$,

(iv) compatibility with linear operations:

(a) homogeneity: if $A \geq B$ and $\lambda \geq 0$ then $\lambda A \geq \lambda B$ and

(b) additivity: if $A \geq B$ and $C \geq D$ then $A + C \geq B + D$.

We will prove this proposition for the special case of the semidefinite cone. Therefore we first show that the set of positive semidefinite matrices is such a cone. The following lemma is given in [Hel00], for being self-contained, we will also give a proof.

Lemma 2.2.3. *The set of all positive semidefinite matrices is a cone, it is convex, pointed, and closed.*

Proof. We will first prove that S_n^+ is a cone and then show the additional properties.

(i) S_n^+ is a Cone: Clearly $0 \in S_n^+$. For $A \in S_n^+$ it follows that $\lambda A \succeq 0$ for all $\lambda \geq 0$ because

$$v^T Av \geq 0 \quad \Longrightarrow \quad v^T \lambda Av = \lambda v^T Av \geq 0.$$

(ii) Convexity: let $\lambda \in [0, 1]$ and $A, B \in S_n^+$, then $\lambda A + (1 - \lambda)B \in S_n^+$ because

$$v^T(\lambda A + (1 - \lambda)B)v = v^T \lambda Av + v^T(1 - \lambda)Bv = \underbrace{\lambda v^T Av}_{\geq 0} + \underbrace{(1 - \lambda)v^T Bv}_{\geq 0} \geq 0.$$

(iii) Pointedness: Intersecting the cone of positive semidefinite matrices S_n^+ and negative semidefinite matrices $-S_n^+$ leads to the zero matrix only:
In fact, let $A \in S_n^+ \cap -S_n^+$, then

$$v^T Av \geq 0 \text{ as } A \in S_n^+$$

and

$$v^T Av \leq 0 \text{ as } A \in -S_n^+$$

for all $v \in \mathbb{R}^n$. Therefore $A = 0$.

(iv) Closedness: This is a consequence of the continuity of the eigenvalues.

We may define a function $f : S_n \to \mathbb{R}_+^n$ mapping a matrix to the vector of its eigenvalues in increasing order. It is well-known that this function is continuous because eigenvalues are nothing else than the roots of the characteristic polynomial and these in turn depend continuously on the matrix elements.

Furthermore, note that $[0, \infty)^n$ is closed in $\mathbb{R}^n$ and the preimage $f^{-1}([0, \infty)^n)$ is the set of positive semidefinite matrices by construction. As preimages of closed sets under continuous functions are closed, the cone of semidefinite matrices is closed. □

Note that the set of positive definite matrices is not a cone because zero does not belong to it. Certainly, it is the interior of the cone S_n^+. Also note that in the boundary of the cone of positive semidefinite matrices there are matrices with at least one eigenvalue equal to zero.

Proof of Proposition 2.2.2 for $C = S_n^+$. We denote by $\succeq$ the partial order induced by the cone S_n^+, i.e., $\succeq$ coincides with $\geq_{S_n^+}$. The set of positive semidefinite matrices is nonempty as $0 \in S_n^+$. Let $A, B, C, D \in S_n^+$, then

(i) reflexivity: $A \succeq A \Longleftrightarrow A - A = 0 \succeq 0$.

(ii) antisymmetry: if $A \succeq B$ and $B \succeq A$ then $A - B \succeq 0$ and $B - A \succeq 0$ we conclude that

$$\begin{array}{rrcl} & B - A & \succeq & 0 \\ \Longleftrightarrow & -(A - B) & \succeq & 0 \\ \Longleftrightarrow & A - B & \preceq & 0. \end{array}$$

It follows that $A - B$ is negative and positive semidefinite at the same time. Therefore $A - B = 0$ and hence $A = B$.

(iii) transitivity: note that if $A, B \in S_n^+$ then by definition of positive semidefiniteness $A + B \in S_n^+$. So if $A \succeq B$ and $B \succeq C$ then $A - B \succeq 0$ and $B - C \succeq 0$. Then

$$0 \preceq (A - B) + (B - C) = A - C.$$

(iv) compatibility with linear operations:

(a) homogeneity: if $A \succeq B$ and $\lambda \geq 0$ then $\lambda A - \lambda B = \lambda(A - B) \succeq 0$ because $A - B$ is positive semidefinite by construction and the semidefinite cone is closed under nonnegative multiplication. Hence $\lambda A \succeq \lambda B$.

(b) additivity: if $A \succeq B$ and $C \succeq D$ then $A - B \succeq 0$ and $C - D \succeq 0$, but then by definition of positive semidefiniteness

$$(A - B) + (C - D) \succeq 0$$

and this implies

$$A + C \succeq B + D.$$

□

Now we define the inner product on M_n. Let $C, X \in M_n$, then the inner product is defined by

$$\langle C, X \rangle := \operatorname{tr}(C^T X) = \sum_{i,j=1}^{n} c_{ij} x_{ij}.$$

The following lemma is very helpful. For example it is needed for proving self-duality of the semidefinite cone (Theorem 2.2.6).

Lemma 2.2.4 ([Hel00, Lemma 1.2.3]). *For two positive semidefinite matrices A, B, it follows that $\langle A, B\rangle \geq 0$. Moreover, $\langle A, B\rangle = 0$ if and only if $AB = 0$.*

Proof. Let A, B be two positive semidefinite matrices and let $\operatorname{rank}(A) = k$.

(i) The eigenvalue decomposition of A is given by $A = P\Lambda P^T$, where Λ is a diagonal matrix with the eigenvalues λ_i of A on the diagonal. Then we can rewrite the inner product of two matrices in the following way:

$$\begin{aligned}\langle A, B\rangle &= \operatorname{tr}(P\Lambda P^T B)\\ &= \operatorname{tr}(\Lambda P^T B P)\\ &= \sum_{i=1}^{k} \underbrace{\lambda_i(A)}_{\geq 0} P_{\cdot,i}^T B P_{\cdot,i}.\end{aligned}$$

As B is positive semidefinite and $P_{\cdot,i}$ are vectors, we use the definition of positive semidefiniteness. Thus we have $P_{\cdot,i}^T B P_{\cdot,i}$ is nonnegative for $i = 1, \ldots, k$. Hence for every summand we have

$$\lambda_i(A) P_{\cdot,i}^T B P_{\cdot,i} \geq 0.$$

So we can conclude that the whole sum must be nonnegative and thus $\langle A, B\rangle \geq 0$.

(ii) $\langle A, B\rangle = 0$ implies that the eigenvectors corresponding to positive eigenvalues of A belong to the null space of B and therefore $AB = 0$. □

A very important fact about the cone of semidefinite matrices is its self-duality. We will need this fact for stating the dual form of a semidefinite program. First we define the dual cone and self-duality.

Definition 2.2.5. The *dual cone* C^* of a cone C is the set $\{y : \langle x, y\rangle \geq 0 \text{ for all } x \in C\}$. The dual cone is also called *polar cone*. Cones satisfying $C = C^*$ are called *self-dual* or *self-polar*. □

Theorem 2.2.6 ([Hel00, Lemma 1.2.6]). *The cone of semidefinite matrices is self-dual, i.e.,*

$$S_n^+ = (S_n^+)^*.$$

Proof. Lemma 2.2.4 implies $S_n^+ \subseteq (S_n^+)^*$. So we just need to show $(S_n^+)^* \subseteq S_n^+$. Note that the matrix xx^T is always positive semidefinite for all $x \in \mathbb{R}^n$. Choose $A \in (S_n^+)^*$, then it follows by definition that

$$\langle A, xx^T\rangle \geq 0$$

but

$$\langle A, xx^T\rangle = x^T A x$$

and therefore $x^T A x \geq 0$ for all $x \in \mathbb{R}^n$. So $A \in S_n^+$ and thus $S_n^+ = (S_n^+)^*$. □

This immediately leads to Fejer's Trace Theorem:

Corollary 2.2.7 (Fejer's Trace Theorem)**.** *A matrix A is positive semidefinite if and only if $\langle A, B\rangle \geq 0$ for all $B \in S_n^+$.*

2.3 Semidefinite programming

By now we are able to state the main problem of this thesis, a semidefinite program.

Let $A \in M_{m,n}$, $x \in \mathbb{R}^n$ be the vector of variables and $c \in \mathbb{R}^n$ be the coefficients of the objective function. With a right-hand-side coefficient vector $b \in \mathbb{R}^m$ a linear problem in primal standard form looks as follows:

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0. \end{aligned} \tag{LP}$$

Semidefinite programs can be seen as a generalization of linear programs. Let

$$\begin{aligned} \mathcal{A} = \mathcal{A}(A_1, \dots, A_m) \colon S_n &\to \mathbb{R}^m \\ X &\mapsto \big(\langle A_1, X\rangle, \dots, \langle A_m, X\rangle\big)^T \end{aligned}$$

with real symmetric $n \times n$ matrices A_i. The coefficients for the objective function is given by a matrix $C \in S_n$. Then we can define a semidefinite problem in its primal standard form:

$$\begin{aligned} \min \quad & \langle C, X\rangle \\ \text{s.t.} \quad & \mathcal{A}X = b \\ & X \succeq 0. \end{aligned} \tag{SDP}$$

This shows that the structure of (LP) and (SDP) look quite the same. The only difference is the cone of feasible solutions: $x \in \mathbb{R}^n$, respectively $X \in M_n$. In the linear case the variable vector x is chosen from the set of positive vectors, in the SDP case the vector becomes a matrix X and is taken from the semidefinite cone. The form of an SDP we want to deal with is the dual standard form because the solver we are using only works for this form.

To obtain the dual SDP we follow the transformation given in [Hel00]. Therefore we first need the adjoint operator of $\mathcal{A}$. This is by definition the unique mapping $\mathcal{A}^T : \mathbb{R}^m \to S_n$ obtained by demanding $\langle \mathcal{A}X, y\rangle = \langle X, \mathcal{A}^T y\rangle$ for all $X \in S_n$ and $y \in \mathbb{R}^m$.

Using

$$\langle \mathcal{A}X, y\rangle = \sum_{i=1}^{m} y_i \mathrm{tr}(A_i X) = \mathrm{tr}(X \sum_{i=1}^{m} y_i A_i) = \langle X, \mathcal{A}^T y\rangle,$$

we obtain $\mathcal{A}^T y = \sum_{i=1}^m y_i A_i$. The dual SDP is constructed using the Lagrangian approach. So for all equality constraints we introduce Lagrange multipliers $y \in \mathbb{R}^m$ and lift the equalities into the objective. We obtain the Lagrangian $L(X, y) := \langle C, X\rangle + \langle b - \mathcal{A}X, y\rangle$ which this leads to

$$\sup_y L(X, y) = \begin{cases} \langle C, X\rangle & \text{if } \mathcal{A}X = b, \\ \infty & \text{otherwise.} \end{cases}$$

For the primal problem we use the following inequality

$$\inf_{X \succeq 0} \sup_{y \in \mathbb{R}^m} L(X, y) \geq \sup_{y \in \mathbb{R}^m} \inf_{X \succeq 0} L(X, y).$$

This minimax inequality holds for any real-valued function $L(X, y)$ with X and y taken from an arbitrary but fixed set out of $\mathbb{R}^m$(see [LR05]). Using the properties of the inner product, $L(X, y)$ can also be written as

$$\begin{aligned} L(X, y) &= \langle C, X\rangle + \langle b - \mathcal{A}X, y\rangle \\ &= \langle C, X\rangle + \langle b, y\rangle - \langle \mathcal{A}X, y\rangle \\ &= b^T y + \langle C, X\rangle - \langle \mathcal{A}X, y\rangle \\ &= b^T y + \langle X, C - \mathcal{A}^T y\rangle. \end{aligned}$$

We now want to use the definition of the dual cone $(S_n^+)^*$. Then

$$\inf_{X \succeq 0} L(X, y) = \begin{cases} b^T y & \text{if } C - \mathcal{A}^T y \in (S_n^+)^*, \\ -\infty & \text{otherwise} \end{cases}$$

because if $C - \mathcal{A}^T y \notin (S_n^+)^*$ then there exists $X \succeq 0$ such that $\langle X^T, C - \mathcal{A}y\rangle < 0$. This directly leads to the dual problem over the dual cone of S_n^+:

$$\sup_y \{b^T y : y \in \mathbb{R}^m, C - \mathcal{A}^T y \in (S_n^+)^*\}.$$

As shown in Theorem 2.2.6 we have $S_n^+ = (S_n^+)^*$ and therefore in the above expression we may replace $C - \mathcal{A}^T y \in (S_n^+)^*$ by $C - \mathcal{A}^T y \succeq 0$. Using this we can state the following definition with the convention that $A_0 := C$.

Definition 2.3.1. Let $y \in \mathbb{R}^n$ be a vector of variables and A_i, $i = 0, \ldots, m$, be symmetric $n \times n$ matrices. Then for a coefficient vector $b \in \mathbb{R}^m$ the *dual SDP* is given as follows:

$$\begin{aligned} \max \quad & b^T y \\ \text{s.t.} \quad & A_0 - \sum_{i=1}^m A_i y_i \succeq 0 \\ & y \in \mathbb{R}^m. \end{aligned} \tag{DSDP}$$

□

There is another widely used equivalent formulation of the dual SDP. It can be obtained by using $\tilde{b} := -b$ and $\tilde{A}_i := -A_i$ for all $i = 1, \dots, m$:

$$\begin{aligned} \min \quad & \tilde{b}^T y \\ \text{s.t.} \quad & \sum_{i=1}^{m} \tilde{A}_i y_i - \tilde{A}_0 \succeq 0 \\ & y \in \mathbb{R}^m. \end{aligned} \tag{2.1}$$

Most of the solvers use this definition and thus this is the definition we use for the applications presented in Chapters 5 and 7. Therefore we will define mixed-integer SDPs using this definition and also the following observations and remarks are based on this formulation. However, we first take a closer look at the formulation (2.1).

This problem has a linear objective function and the right-hand-side of the constraint is represented by the matrix A_0. We call the constraint for positive semidefiniteness a *linear matrix inequality* and it can be rewritten as

$$\begin{aligned} & \sum_{i=1}^{m} A_i y_i - A_0 \succeq 0 \\ \iff \qquad & \sum_{i=1}^{m} A_i y_i \succeq A_0. \end{aligned} \tag{2.2}$$

The constraint (2.2) looks like a linear inequality of the form $\sum_i a_i y_i \geq a_0$. The main difference is that the coefficients of the variables are no longer real numbers, they are real $n \times n$ matrices.

Observation 2.3.2. *If the matrix $\sum_{i=1}^m A_i y_i - A_0$ is in block diagonal form, we can split the constraint into two new constraints both of which have to be positive semidefinite. This is due to the fact that a matrix is positive semidefinite if and only if every principal minor of the matrix is (see Proposition 2.1.6):*

$$\begin{aligned} & \sum_{i=0}^{m} \begin{pmatrix} A_{i0} & 0 \\ 0 & A_{i1} \end{pmatrix} y_i - \begin{pmatrix} A_{00} & 0 \\ 0 & A_{01} \end{pmatrix} \succeq 0 \\ \iff \qquad & \sum_{i=0}^{m} A_{i0} y_i - A_{00} \succeq 0 \qquad \textit{and} \qquad \sum_{i=0}^{m} A_{i1} y_i - A_{01} \succeq 0. \end{aligned}$$

Remark 2.3.3. *Constraints of type* (2.2) *can also be used to express linear constraints of the form $\sum_i a_i y_i \geq a_0$ by interpreting every linear constraint as 1×1 SDP block. To simplify notation we are going to summarize all the linear constraints in one LP block. For this block the coefficient matrices and the right-hand-side matrix are diagonal matrices.*

Problems of the form (DSDP) with linear matrix inequalities for continuous variables can be solved using interior-point-solvers, bundle methods or augmented Lagrangian procedures. A detailed description of these algorithms is given in Section 3.1.

2.3.1. Mixed-integer semidefinite programs

While SDPs with continuous variables can be solved easily, there is no direct solution approach available for the case of integer or mixed-integer variables within an SDP. Certainly, this case appears naturally when looking at SDPs.

Definition 2.3.4. We call the following problem a *mixed-integer semidefinite program*:

$$\begin{aligned} \min \quad & \tilde{b}^T y \\ \text{s.t.} \quad & \sum_{i=1}^{m} \tilde{A}_i y_i - \tilde{A}_0 \succeq 0 \\ & y \in \mathbb{Z}^d \times \mathbb{R}^{m-d}. \end{aligned} \tag{MISDP}$$

The $\tilde{A}_i$ are $n \times n$-dimensional, symmetric matrices and can be in block-diagonal structure. The objective coefficients are $\tilde{b} \in \mathbb{R}^m$. □

Note that we are explicitly looking at a transformed dual SDP (2.1) with $\tilde{b} := -b$, $\tilde{A}_i := -A_i$, and an objective function that is to minimize. This is due to the fact that we will use this form for our data structures and the input format in Chapter 4.

Mixed-integer problems arise in various applications. For example in mechanical or electrical engineering there are problems that can be formulated as SDPs due to the physics in the problem. Binary or integer variables often occur in a natural way for example to position or design reasons. Additionally, many classical combinatorial optimization problems like the Maximum Cut Problem can equivalently be stated as MISDP. It is well-known that SDP relaxations produce very good lower bounds in the branch-and-bound tree, usually better than those obtained from the LP relaxations.

So studying this class of problems is not only an analogy to classical linear and mixed-integer programming, it is also desirable from an application point of view. In Chapter 5 we are going to introduce different models for the Truss Topology Design problem and show that the SDP formulation is the only one that can handle some of the occurring aspects – for example constraints on vibrations in trusses.

2.3.2. Note on semidefinite representability

The constraints for controlling vibrations mentioned above can be represented as SDP. This is not completely obvious, therefore we comment on what kind of problems can be represented as SDP. The content of this section is taken from [BTN01].

Definition 2.3.5. A convex set $X \subset \mathbb{R}^n$ is semidefinitely representable if there exists an affine mapping

$$(x,u) \to \mathcal{A}\begin{pmatrix} x \\ u \end{pmatrix} - B \colon \mathbb{R}^n \times \mathbb{R}^k \to S_m$$

such that

$$x \in X \quad \Longleftrightarrow \quad \text{there exists } u \in \mathbb{R}^k : \mathcal{A}\begin{pmatrix} x \\ u \end{pmatrix} - B \succeq 0.$$

Or in other words, X is semidefinitely representable if there exists a linear matrix inequality

$$\mathcal{A}\begin{pmatrix} x \\ u \end{pmatrix} - B \succeq 0$$

such that X is a projection of the solution set of the linear matrix inequality onto the x-components. A linear matrix inequality with this property is called semidefinite representation of the set X. □

Definition 2.3.6. A convex function $f\colon \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is called *semidefinitely representable* if its epigraph

$$\{(x,t) \mid t \geq f(x)\}$$

is a semidefinitely representable set. A semidefinite representation of the epigraph of f is called a semidefinite representation of f. □

In [BTN01] and [Ali95] there are many examples for semidefinitely representable problems and related questions. So for example the sum of the k largest eigenvalues, the determinant of a symmetric positive semidefinite matrix or general quadratic matrix inequalities are semidefinitely representable.

We are interested in one special case, namely the maximum eigenvalue $\lambda_{\max}(A)$ regarded as a function depending on an $m \times m$ symmetric matrix A. The epigraph of this function is

$$\{(A,t) \in S_m \times \mathbb{R} \mid t \geq \lambda_{\max}(A)\}$$

and it is given by the linear matrix inequality

$$tI_m - A \succeq 0.$$

Note that the eigenvalues of $tI_m - A$ are $\{t - \lambda_i(A) \mid \lambda_i(A) \text{ eigenvalue of } A\}$. So this matrix is positive semidefinite if and only if $t \geq \lambda_i(A)$ for all eigenvalues $\lambda_i(A)$ and therefore in particular if $t \geq \lambda_{\max}(A)$. We will use this later in Section 5.6.2.

2.4 Duality

One key idea in the world of linear programming is duality, especially strong duality. In semidefinite programming it cannot be expected that strong duality holds in general. We will give a short overview about duality for SDPs in the following.

First we introduce a slack matrix for the inequality constraint in the dual:

Definition 2.4.1. For the linear matrix inequality

$$C - \mathcal{A}^T y \succeq 0$$

let Z be the positive semidefinite slack matrix in the following way:

$$\mathcal{A}^T y + Z = C, \quad Z \succeq 0.$$ □

Lemma 2.4.2 (Weak duality)**.** *An SDP and its dual satisfy weak duality, i.e.,*

$$\langle C, X\rangle - \langle b, y\rangle \geq 0.$$

Proof. Using the definitions and inserting the equalities leads to

$$\langle C, X\rangle - \langle b, y\rangle = \langle \mathcal{A}^T y + Z, X\rangle - \langle \mathcal{A}X, y\rangle = \langle \mathcal{A}^T y, X\rangle + \langle Z, X\rangle - \langle \mathcal{A}X, y\rangle = \langle Z, X\rangle.$$

By Lemma 2.2.4 we know that $\langle Z, X\rangle \geq 0$ because X and Z are both supposed to be positive semidefinite. This completes the proof. □

Strong duality or $\langle Z, X\rangle = 0$ can only be obtained with some additional conditions and compared to the classical situation it is no longer true that optimality implies strong duality. For illustrating the differences to the linear programming case we present some examples from [WSV00] and [LA11].

Example 2.4.3 (Positive duality gap)**.** *Consider*

$$\begin{aligned} \min \quad & a x_{11} \\ \text{s.t.} \quad & x_{11} + 2x_{23} = 1 \\ & x_{22} = 0 \\ & X \succeq 0 \end{aligned} \tag{P}$$

and

$$\begin{aligned} \max \quad & y_2 \\ \text{s.t.} \quad & \begin{pmatrix} y_2 - a & 0 & 0 \\ 0 & y_1 & y_2 \\ 0 & y_2 & 0 \end{pmatrix} \preceq 0. \end{aligned} \tag{D}$$

The optimal value of (P) *is* $x_{11} = 1$ *and all other equal to zero. It has the objective value* a*. Due to* $x_{22} = 0$ *and Proposition 2.1.8,* x_{23} *must be zero and* $x_{11} = 1$ *is the only feasible solution. With the same argument* y_2 *in* (D) *is zero and therefore the optimal objective value of* (D) *is zero.*

Another example for this can be found in [Hel00] or [VB96]. Positive duality gaps are not the only difficulties that make solving SDPs more complicated, as the next example shows.

Example 2.4.4 (Weak infeasibility)**.** *Again we consider a pair of primal and dual SDPs:*

$$\begin{aligned} \inf \quad & -2x_{12} \\ \text{s.t.} \quad & \begin{pmatrix} 1 & x_{12} \\ x_{12} & 0 \end{pmatrix} \succeq 0 \end{aligned} \tag{P'}$$

and

$$\begin{aligned} \sup \quad & -y_1 \\ \text{s.t.} \quad & \begin{pmatrix} y_1 & -1 \\ -1 & y_2 \end{pmatrix} \succeq 0. \end{aligned} \tag{D'}$$

Now the optimal objective value is zero. It is attained for (P') *with the same argument as in Example 2.4.3, but not for* (D') *because $y_1 = 0$ is not feasible.*

To avoid these difficulties additional conditions are needed. One condition is the following.

Definition 2.4.5. A point X is *strictly feasible* for the primal SDP if it is feasible for the primal SDP and satisfies $X \succ 0$. A pair (y, Z) is *strictly feasible* for the dual SDP if it is feasible for the dual SDP and satisfies $Z \succ 0$. □

This definition claims the existence of an interior point $X \in \operatorname{int}(S_n^+)$ or $Z \in \operatorname{int}(S_n^+)$ such that all constraints are satisfied. This condition is also called *Slater constraint qualification* (see [HUL04]).

On our way to strong duality we will follow [Hel00]. First we need a result that gives bounds on the eigenvalues of a semidefinite matrix.

Lemma 2.4.6. *For $A, B \in S_n^+$ the following inequalities hold:*

$$\begin{aligned} \langle A, B \rangle &\geq \lambda_{\min}(A)\operatorname{tr}(B) \geq \lambda_{\min}(A)\lambda_{\max}(B) \\ \langle A, B \rangle &\leq \lambda_{\max}(A)\operatorname{tr}(B) \leq n\lambda_{\max}(A)\lambda_{\max}(B) \end{aligned}$$

Proof. We start with the first chain of inequalities. For an orthogonal matrix $P \in M_n$ and a diagonal matrix Λ with $A = P\Lambda P^T$ we have

$$\begin{aligned} \langle A, B \rangle = \operatorname{tr}(AB) &= \operatorname{tr}(P\Lambda P^T B) = \operatorname{tr}(\Lambda P^T B P) \\ &\geq \lambda_{\min}(A)\operatorname{tr}(PBP^T) = \lambda_{\min}(A)\operatorname{tr}(B) \\ &\geq \lambda_{\min}(A)\lambda_{\max}(B). \end{aligned}$$

The inequalities in the second row can be shown in a similar way, we also need the eigenvalue decomposition $P\Lambda P^T$ of A:

$$\begin{aligned}
\langle A, B\rangle &= \operatorname{tr}(AB) = \operatorname{tr}(P\Lambda P^T B) = \operatorname{tr}(\Lambda P^T BP) \\
&= \operatorname{tr}(P^T \Lambda BP) = \operatorname{tr}(\Lambda B) \\
&= \operatorname{tr}\left(\begin{pmatrix} \lambda_1(A)b_{11} & 0 & \dots & 0 \\ 0 & \lambda_2(A)b_{22} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & \lambda_n(A)b_{nn} \end{pmatrix}\right) \\
&\leq \operatorname{tr}\left(\begin{pmatrix} \lambda_{\max}(A)b_{11} & 0 & \dots & 0 \\ 0 & \lambda_{\max}(A)b_{22} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & \lambda_{\max}(A)b_{nn} \end{pmatrix}\right) \\
&= \operatorname{tr}\left(\lambda_{\max}(A)\begin{pmatrix} b_{11} & 0 & \dots & 0 \\ 0 & b_{22} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & b_{nn} \end{pmatrix}\right) \\
&= \lambda_{\max}(A)\operatorname{tr}(B) \\
&\leq \lambda_{\max}(A)n\lambda_{\max}(B). \qquad \square
\end{aligned}$$

Lemma 2.4.7. *Let $\mathcal{B} : S_n \to \mathbb{R}^m$ be a linear operator. Assume that there exists a $\hat{y} \in \mathbb{R}^m$ such that $\hat{Z} = \mathcal{B}^T\hat{y} \succ 0$. Then the set $\{\mathcal{B}X : X \succeq 0\}$ is closed.*

Proof. First choose a sequence $(X_i)_{i\in\mathbb{N}} \subset S_n^+$ with $\lim_{i\to\infty} \mathcal{B}X_i = b$. For proving the boundedness of this sequence we use Lemma 2.4.6. This gives a bound on the maximum eigenvalue of every X_i:

$$\langle \hat{y}, \mathcal{B}X_i\rangle = \langle \mathcal{B}^T\hat{y}, X_i\rangle = \langle \hat{Z}, X_i\rangle \geq \lambda_{\min}(\hat{Z})\lambda_{\max}(X_i).$$

We also know

$$\lim_{i\to\infty} \langle \hat{y}, \mathcal{B}X_i\rangle = \langle \hat{y}, b\rangle,$$

so the maximum eigenvalue $\lambda_{\max}(X_i)$ of this sequence is bounded. The boundedness of the elements of X_i is given by Observations 2.1.10 and 2.1.11. The sequence X_i is contained in a compact subset of S_n^+. Therefore we conclude that there exists a convergent subsequence converging to some $\hat{X} \succeq 0$ with $\mathcal{B}\hat{X} = b$. □

Note that $\mathcal{B}$ is a linear operator and therefore the image of a convex set is again convex. Now we want to state an analogue of the Farkas-Lemma. For this we need a separation theorem which can only be applied if the image of the semidefinite cone is closed.

Lemma 2.4.8. *Let $\mathcal{B} : S_n \to \mathbb{R}^m$ be a linear operator. Assume that $\{\mathcal{B}X : X \succeq 0\}$ is closed and let $b \in \mathbb{R}^m$. Then either there exists $X \succeq 0$ such that $\mathcal{B}X = b$ or there is a $y \in \mathbb{R}^m$ such that $\mathcal{B}^T y \succeq 0$ and $b^T y < 0$.*

Proof. We start the proof by assuming that there exists a positive semidefinite $\hat{X}$ such that $\mathcal{B}\hat{X} = b$. If there is $y \in \mathbb{R}^m$ with $b^T y < 0$ then

$$\langle b, y\rangle = \langle \mathcal{B}\hat{X}, y\rangle = \langle \hat{X}, \mathcal{B}^T y\rangle < 0$$

and Corollary 2.2.7 implies that $\mathcal{B}^T y$ cannot be positive semidefinite.

So suppose that there exists no such $X \succeq 0$. Then there must be a hyperplane separating b from the closed convex set $\{\mathcal{B}X : X \succeq 0\}$, i.e.,

$$b^T \hat{y} < 0 \qquad \text{and} \qquad \langle \hat{y}, \mathcal{B}X\rangle = \langle \mathcal{B}^T \hat{y}, X\rangle \geq 0.$$

Again by Corollary 2.2.7 we get the desired result that $\mathcal{B}^T \hat{y}$ is positive semidefinite. □

Equipped with all these tools we can now prove the theorem of strong duality.

Theorem 2.4.9 (Strong duality)**.** *Let $(\hat{y}, \hat{Z})$ be a strictly feasible solution for the dual problem (DSDP) and let*

$$\begin{aligned} p^* &= \inf\{\langle C, X\rangle \mid \mathcal{A}X = b, X \succeq 0\} \quad \textit{and} \\ d^* &= \sup\{\langle b, y\rangle \mid \mathcal{A}^T y + Z = C, Z \succeq 0\}. \end{aligned}$$

Then $p^* = d^*$ and if p^* is finite, it is attained for some element of $\{X \succeq 0 : \mathcal{A}X = b\}$.

Proof. With a strictly feasible solution for the dual problem weak duality gives us $p^* \geq b^T \hat{y}$ and therefore $p^* \neq -\infty$. Use the vector $\begin{pmatrix} 1 \\ -\hat{y} \end{pmatrix}$ to apply Lemma 2.4.7 to the set

$$\left\{ \begin{pmatrix} \langle C, X\rangle \\ \mathcal{A}X \end{pmatrix} : X \succeq 0 \right\}$$

and note that this set is closed.

Now we want to obtain a contradiction and assume that d^* is finite and the inequality $p^* > d^*$ holds. Then the following system, where the primal optimal value p^* which is equal to $\langle C, X\rangle$ should also be equal to the dual optimal value d^*, is infeasible:

$$\begin{aligned} \langle C, X\rangle &= d^* \\ \mathcal{A}X &= b \\ X &\succeq 0. \end{aligned}$$

Using Lemma 2.4.8 we get a vector $\begin{pmatrix} y_0 \\ \bar{y} \end{pmatrix}$ with

$$d^* y_0 + b^T \bar{y} < 0 \qquad \text{and} \qquad y_0 C + \mathcal{A}^T \bar{y} \succeq 0.$$

Now we distinguish three cases regarding y_0:

(i) $y_0 = 0$: Then the two inequalities become

$$b^T \bar{y} < 0 \qquad \text{and} \qquad \mathcal{A}^T \bar{y} \succeq 0.$$

We can choose $\hat{y} + \alpha(-\bar{y})$ with $\alpha \geq 0$ as a dual feasible ray. Along this ray the dual objective function is strictly increasing. So $d^* = \infty$, which is a contradiction to the assumption that d^* is finite.

(ii) $y_0 > 0$: In this case we can divide by y_0 and obtain

$$d^* + \frac{b^T \bar{y}}{y_0} < 0 \qquad \text{and} \qquad C + \frac{\mathcal{A}^T \bar{y}}{y_0} \succeq 0.$$

Choosing $y = \frac{-\bar{y}}{y_0}$ we obtain a dual feasible solution with a better objective value than d^*, as the first inequality shows. This contradicts the assumption that d^* is the optimal dual solution.

(iii) $y_0 < 0$: We can again divide, this time by $-y_0$ and obtain for $\varepsilon_1 > 0$ but small enough and obtain

$$-d^* - \frac{b^T \bar{y}}{y_0} < -\varepsilon_1 \qquad \text{and} \qquad -C - \frac{\mathcal{A}^T \bar{y}}{y_0} \succeq 0.$$

For $0 < \varepsilon_2 < \varepsilon_1$ choose some feasible solution $\tilde{y}$ with

$$d^* - b^T \tilde{y} \leq \varepsilon_2 \qquad \text{and} \qquad C - \mathcal{A}^T \tilde{y} \succeq 0.$$

Summing up these inequalities we obtain

$$b^T(-\tilde{y} - \frac{\bar{y}}{y_0}) < -\varepsilon_1 + \varepsilon_2 < 0 \qquad \text{and} \qquad \mathcal{A}^T(-\tilde{y} - \frac{\bar{y}}{y_0}) \succeq 0.$$

Now we can again find an improving ray $\hat{y} + \alpha(\tilde{y} + \frac{\bar{y}}{y_0})$ with $\alpha \geq 0$ for the dual problem, again contradicting the assumption that d^* is finite.

This proves $p^* = d^*$ and because of the closedness of the set $\left\{ \begin{pmatrix} \langle C, X \rangle \\ \mathcal{A}X \end{pmatrix} : X \succeq 0 \right\}$ it follows that the primal solution is attained for finite p^*. □

The proof can also be found in [Ali95], [NN94], and [Roc70].

CHAPTER 3

Solving strategies

In this chapter we discuss different strategies to solve mixed-integer semidefinite programs (MISDP). Our intention is to solve MISDPs within a branch-and-bound algorithm. Therefore we have to ask four different questions:

1. How can we solve the relaxation of our problem in order to obtain lower bounds?
2. What is the difference between solving SDPs and LPs within a branch-and-bound framework?
3. What can be done to speed up the branch-and-bound algorithm?
4. Is pure branch-and-bound the best way to solve MISDPs?

Within this chapter we will answer these questions, each in a distinct section. First we look at algorithms for solving semidefinite programs. We take a closer look at interior-point-solvers and different variations of them. Additionally, we present the augmented Lagrangian method, cf. [Sti06] and comment on bundle methods, especially the spectral bundle method, cf. [Hel00].

After getting an idea of how to solve the relaxation in every branching node, we take a closer look at the branch-and-bound procedure itself. We examine the problem structure during the search in the branch-and-bound tree One important information is feasibility of the subproblems in the branch-and-bound tree. We also go into details on how to speed up the procedure using heuristics. Furthermore, we present some ideas to approximate the semidefinite cone using linear and quadratic inequalities.

The idea of solving SDPs within a branch-and-bound algorithm is not new at all. There have been different approaches of solving special kinds of MISDPs in this way. To the best of the author's knowledge all of these approaches are problem specific and not able to solve general MISDPs. Therefore most of the concepts presented in Sections 3.2, 3.3, and 3.4 are considered for the first time in the context of a general framework. The idea of approximating an SDP using sparse eigenvector cuts (see Section 3.4) has already been examined for some combinatorial optimization problems, but not in general. Some of the ideas are a result of intensive discussions with Lars Schewe [Sch12].

3.1 Solving SDPs

One algorithm for solving semidefinite programs that is used very often is the interior-point-method, but there also exist other algorithms for solving them. In this section we describe three different well-known methods to solve SDPs. We start with a general description of interior-point-algorithms. There exist numerous variations of these algorithms, we focus on the one the solver which we will introduce later is using. Additionally, we comment on augmented Lagrangian and bundle methods, especially the spectral bundle method. We also explain why we use an interior-point-method instead of the augmented Lagrangian or the spectral bundle algorithm to solve the SDPs within the branch-and-bound tree.

3.1.1. Interior-Point-Method

The idea of an interior-point-solver was introduced by Karmarkar [Kar84] for linear programming. We want to use interior-point-methods for semidefinite programming; therefore we follow [Hel00], [LR05], and [Wri97].

The different variations of interior-point-methods can be characterized using different properties of the method, for example the kind of problem they look at. So an algorithm is called *infeasible interior-point-method* if it starts from an infeasible initial point. Also there exist primal, dual, and primal-dual methods. While primal methods only look at the primal problem formulation, primal-dual methods take both, the primal and the dual, into account.

The main idea is to start with optimality conditions, also called KKT-conditions, and apply some kind of Newton-method in this system of equations where there is a big penalty for reaching the boundary of the feasible set. In the process the step lengths and search directions are modified so that staying near a specific path, the so-called central path, and converging to the optimum is possible. We will now look at the details.

As we have seen in Section 2.4 we need a strictly feasible point for the primal (SDP) and a strictly feasible point for the dual (DSDP) for guaranteeing strong duality (Theorem 2.4.9). Therefore we assume for the rest of this chapter to have such matrices Z and X satisfying:

$$\mathcal{A}X = b \qquad \text{and} \qquad C = \sum_i A_i y_i + Z, \qquad \text{where} \qquad Z \succ 0 \quad \text{and} \quad X \succ 0.$$

As we do not want to leave the interior of the feasible region during the solving process we have to add a barrier parameter to the objective of the problem, which gets very large when a solution is approaching the boundary of the feasible region. The barrier function we choose for solving SDPs is $f(X) = -\ln\det(X)$. This function tends to infinity for X reaching the boundary of the positive semidefinite cone. Also this function is strictly convex and satisfies another very important property.

Definition 3.1.1. A C^3 smooth convex function f defined on a nonempty, convex, open set $Q \subseteq \mathbb{R}^n$ is called self-concordant on Q if it satisfies the following two properties:

(i) Barrier property: For every sequence $(x_i \in Q)_{k\in\mathbb{N}}$ converging to a boundary point of Q we have $f(x_i) \to \infty$.

(ii) Differential inequality of self-concordance: for all $x \in Q$ and $h \in \mathbb{R}^n$ the function f satisfies the differential inequality

$$\left|D^3 f(x)[h,h,h]\right| \leq 2(D^2 f(x)[h,h])^{\frac{3}{2}}. \qquad \square$$

This means that the Hessian of f is Lipschitz continuous with respect to the local Euclidean metric defined by the Hessian itself.

Definition 3.1.2. For a closed convex set $G \subseteq \mathbb{R}^n$ with nonempty interior and $\mu \geq 0$, a function $f\colon \operatorname{int} G \to \mathbb{R}$ is called self-concordant barrier for G with the parameter value μ if

(i) f is self-concordant on the interior of G and

(ii) for all $x \in \operatorname{int} G$ and $h \in \mathbb{R}^n$

$$|Df(x)[h]| \leq \sqrt{\mu}(D^2 f(x)[h,h])^{\frac{1}{2}}. \qquad \square$$

The idea behind this property is to be able to define a family of smooth convex functions which harmonize the Newton method. The Newton method solves the problem using a quadratic approximation of a function, namely the second-order Taylor expansion of f at a point x. In one Newton-step it searches for an iterate of the point by solving the minimization problem of the quadratic approximation and then moving from x along the direction $\bar{x} - x$, where $\bar{x}$ is the optimal solution of the minimization problem.

For a good approximation, i.e., when the function is nearly the quadratic approximation, the resulting direction is good. Moreover, for self-concordant barriers the Hessian is locally bounded by a Lipschitz condition, so the quadratic approximation is of good quality for comparatively large regions.

Proposition 3.1.3. *The function $-\ln\det(X)$ is a self-concordant barrier for the cone of positive semidefinite matrices.*

Proof. See [NN94]. □

The problem we are solving now is the following:

$$\begin{aligned} \min \quad & \langle C, X\rangle - \mu \ln\det(X) \\ \text{s.t.} \quad & \mathcal{A}X = b \\ & X \succeq 0, \end{aligned}$$

Using the barrier parameter μ, the influence of the barrier term on the objective function can be controlled. Following [Hel00] we introduce a Lagrange multiplier y to transform the problem to an unconstrained optimization problem. This leads to:

$$\mathcal{L}_\mu(X, y) = \langle C, X\rangle - \mu \ln\det(X) + \langle y, b - \mathcal{A}X\rangle.$$

Note that the function $\mathcal{L}_\mu(X, \cdot)$ is smooth and strictly convex for a given $y \in \mathbb{R}^m$ on the set of positive semidefinite matrices as it is a sum of a linear and a smooth strictly convex function and it is linear for given $X \succeq 0$. Now we can use the idea of so-called saddle points. As shown in [Roc70] a function is a saddle function if it is convex in one coordinate and concave in the other. This is exactly the situation described above.

Definition 3.1.4. A point $(X_\mu, y_\mu) \in S_n^+ \times \mathbb{R}^m$ is called a *saddle point* if

$$\inf_{X \succeq 0} \mathcal{L}_\mu(X, y_\mu) \geq \mathcal{L}_\mu(X_\mu, y_\mu) \geq \sup_{y \in \mathbb{R}^m} \mathcal{L}_\mu(X_\mu, y). \qquad \square$$

From a saddle point we can get the optimal solution X_μ for the barrier problem and the correct Lagrange multipliers. Using this we can derive necessary conditions for optimality, the so called Karush-Kuhn-Tucker (KKT) conditions.

Observation 3.1.5. *In a saddle point (X_μ, y_μ), we know that X_μ minimizes $\mathcal{L}_\mu(\cdot, y_\mu)$ and y_μ maximizes $\mathcal{L}_\mu(X_\mu, \cdot)$. Therefore the partial derivatives of $\mathcal{L}_\mu$ with respect to X and y have to be zero. These first order necessary conditions are also sufficient as proven in [Roc70].*

The above mentioned first derivatives look as follows:

$$\begin{aligned} \nabla_X \mathcal{L}_\mu &= C - \mu X^{-1} - \mathcal{A}^T y = 0 \\ \nabla_y \mathcal{L}_\mu &= b - \mathcal{A}X = 0. \end{aligned}$$

For getting a primal-dual formulation we set $Z = \mu X^{-1}$. Now we rewrite the KKT-system in the following way:

$$\begin{aligned} \mathcal{A}X &= b, & X &\succ 0 & &\text{(primal feasibility)} \\ \mathcal{A}^T y + Z &= C, & Z &\succ 0 & &\text{(dual feasibility)} \\ XZ &= 0 & & & &\text{(complementary slackness).} \end{aligned}$$

The complementary slackness constraint holds because we assumed that there is a strictly feasible point to be able to use the Strong Duality Theorem (Theorem 2.4.9). Therefore the duality gap equals zero:

$$0 = b^T y - \langle C, X\rangle = y^T(\mathcal{A}X) - \langle C, X\rangle = \langle \mathcal{A}^T y - C, X\rangle = \langle Z, X\rangle.$$

Finally, by Lemma 2.2.4 we know that for two positive semidefinite matrices Z, X we have $\langle Z, X\rangle = 0$ if and only if $ZX = 0$.

The system we want to look at is not the system stated above. We want to consider a perturbed version of these KKT conditions:

$$\begin{aligned} \mathcal{A}X &= b, & X &\succ 0 \\ \mathcal{A}^T y + Z &= C, & Z &\succ 0 \\ XZ &= \mu I. \end{aligned} \tag{3.1}$$

Solving the system (3.1) yields a unique solution for X and for Z, but not for y.

Remark 3.1.6. *It is possible to eliminate y from the system because we assumed that there exists an $\hat{X}$ satisfying $\mathcal{A}\hat{X} = b$. The objective function can be reformulated as*

$$\langle b, y\rangle = \langle \mathcal{A}\hat{X}, y\rangle = \langle \hat{X}, \mathcal{A}^T y\rangle = \langle \hat{X}, C - Z\rangle.$$

Now y is still needed to span the feasible set of Z-matrices. This can be avoided by using the range of the space $\mathcal{A}^T$, denoted by $\mathcal{R}(\mathcal{A}^T)$, and the null space of $\mathcal{A}$, denoted by $\mathcal{N}(\mathcal{A})$. These two subspaces are orthogonal, i.e., $\mathcal{R}(\mathcal{A}^T) = \mathcal{N}(\mathcal{A})^\perp$. Now the dual equality constraint $\mathcal{A}^T y + Z = C$ can be expressed as

$$Z \in (S_n^+ \cap \{C + \mathcal{R}(\mathcal{A}^T)\}) = (S_n^+ \cap \{C + \mathcal{N}(\mathcal{A})^\perp\}).$$

The primal equality constraint $\mathcal{A}X = b$ becomes

$$X \in (S_n^+ \cap \{\hat{X} + \mathcal{N}(\mathcal{A})\}).$$

So we can denote the solution of system (3.1) as (X_μ, Z_μ) for a fixed μ. Now X_μ and Z_μ are the unique optimal solutions for the primal and the dual barrier problem. Moreover, they are feasible in the original problem with a gap of

$$\langle C, X\rangle - \langle b, y\rangle = \langle Z, X\rangle = \langle \mu, I\rangle = n\mu.$$

Definition 3.1.7. The set $\{(X_\mu, Z_\mu) \mid \mu > 0\}$ of solutions of (3.1) is called the *central path.* □

Note that this central path is a smooth curve. Still following [Hel00] we now show that for $\mu \to 0$ the central path converges to the optimal solution, i.e., a point (X^*, Z^*) with X^* being the optimal solution of the original primal problem and Z^* being the optimal solution of the original dual formulation. To this end, we first need another lemma:

Lemma 3.1.8. *For primal feasible X', X'' and dual feasible Z', Z'' we have*

$$\langle X' - X'', Z' - Z''\rangle = 0.$$

Proof. For two elements Z' and Z'' of the set $\{C - \mathcal{A}^T y : y \in \mathbb{R}^m\}$ we have

$$Z'' - Z' = (C - \mathcal{A}^T y'') - (C - \mathcal{A}^T y') = C - C - \mathcal{A}^T y'' + \mathcal{A}^T y' = \mathcal{A}^T(y' - y'').$$

Additionally, for two elements X' and X'' of the set $\{X \succeq 0 : \mathcal{A}X = b\}$ we have

$$0 = b - b = \mathcal{A}X' - \mathcal{A}X'' = \mathcal{A}(X' - X'').$$

Hence

$$\begin{aligned} -\langle X' - X'', Z' - Z'' \rangle &= \langle X' - X'', -Z' + Z'' \rangle = \langle X' - X'', \mathcal{A}^T(y' - y'') \rangle \\ &= \langle \mathcal{A}(X' - X''), y' - y'' \rangle = 0. \end{aligned}$$ □

We will use this lemma in the proof of the following theorem.

Theorem 3.1.9. *Let $(\mu_k)_{k\in\mathbb{N}}, \mu_k > 0, k \in \mathbb{N}$ be a sequence with $\mu_k \to 0$. Then the corresponding solutions (X_{μ_k}, Z_{μ_k}) of* (3.1) *converge to a pair of optimal solution (X^*, Z^*) of* (SDP) *and* (DSDP).

Proof. We will use Lemma 3.1.8, the assumption that we have strictly feasible solutions X^0 for the primal and Z^0 for the dual problem and that the sequence (X_{μ_k}, Z_{μ_k}) is contained in a compact set. From Lemma 3.1.8 we know

$$0 = \langle X_{\mu_k} - X^0, Z_{\mu_k} - Z^0 \rangle = \langle X_{\mu_k}, Z_{\mu_k} \rangle - \langle X_{\mu_k}, Z^0 \rangle - \langle X^0, Z_{\mu_k} \rangle + \langle X^0, Z^0 \rangle.$$

Moreover, we know that for strictly feasible X_{μ_k}, Z_{μ_k} the relation $X_{\mu_k} Z_{\mu_k} = \mu_k n$ holds, thus we get

$$\begin{aligned} \langle X_{\mu_k}, Z^0 \rangle + \langle X^0, Z_{\mu_k} \rangle &= \langle X_{\mu_k}, Z_{\mu_k} \rangle + \langle X^0, Z^0 \rangle \\ \langle X_{\mu_k}, Z^0 \rangle + \langle X^0, Z_{\mu_k} \rangle &= \mu_k n + \langle X^0, Z^0 \rangle. \end{aligned}$$

With Lemma 2.4.6 we have

$$\lambda_{\max}(X_{\mu_k})\lambda_{\min}(Z^0) + \lambda_{\min}(X^0)\lambda_{\max}(Z_{\mu_k}) \leq \mu_k n + \langle X^0, Z^0 \rangle.$$

Note that the sequence of iterates is bounded because $\lambda_{\min}(X^0) > 0$ and $\lambda_{\min}(Z^0) < 0$. So there exists a convergent subsequence converging to a point (X^*, Z^*) which must be feasible with $\langle X^*, Z^* \rangle = 0$. Therefore the solution (X^*, Z^*) has to be optimal. □

As we know from linear programming the central path converges to a primal-dual optimal solution $x, z \in \mathbb{R}^n$ where $x_i = 0$ if and only if $z_i \neq 0$ for $i = 1, \ldots, n$, which implies $x^T z = 0$. This property is called *strict complementarity* and the question arises whether this also happens in the context of semidefinite programs and what would be the analogue to strict complementarity for SDPs. In the following we deduce this analogue.

Let X^* and Z^* be the optimal solutions of the primal and the dual SDP. Note that for symmetric matrices X, Z we have $XZ = (ZX)^T$. So we conclude that symmetric matrices commute if their product is symmetric again. Additionally, we know that $\langle X^*, Z^* \rangle = 0$ implies $X^* Z^* = Z^* X^* = 0$ (Lemma 2.2.4). So these two matrices commute.

Now we want to link this property to another property of matrices.

Definition 3.1.10. Let A, B be two $n \times n$ diagonalizable matrices. Then A, B are *simultaneously diagonalizable* if there exists an $n \times n$ matrix S such that $S^{-1}AS$ and $S^{-1}BS$ are diagonal matrices. □

We use the following lemma.

Lemma 3.1.11 ([HJ90, Theorem 1.3.12])**.** *Let A, B be two $n \times n$ diagonalizable matrices. Then $AB = BA$ if and only if A and B are simultaneously diagonalizable.*

Proof. We only proof the if part. The proof for the only if part can be found in [HJ90]. Let A, B be simultaneously diagonalizable: $D_A = S^{-1}AS$ and $D_B = S^{-1}BS$. Then we have $A = SD_AS^{-1}$ and $B = SD_BS^{-1}$. Using this and the fact that diagonal matrices commute, we can write the product of the two matrices in the following way:

$$AB = SD_AS^{-1}SD_BS^{-1} = SD_AD_BS^{-1} = SD_BD_AS^{-1} = SD_BS^{-1}SD_AS^{-1} = BA.$$

So A and B commute. □

As X^* and Z^* commute, they are simultaneously diagonalizable and the nonzero eigenvectors of any optimal primal solution X^* are in the null space of any optimal dual solution Z^* and vice versa. This means that the two minimum faces of the semidefinite cone containing the respective convex sets of primal and dual optimal solutions are spanned by orthogonal subspaces of $\mathbb{R}^n$. So we define strict complementarity for semidefinite matrices according to [dKRT97] in the following way:

Definition 3.1.12. An optimal pair (X^*, Z^*) is *strictly complementary* if $\operatorname{rank}(X^*) + \operatorname{rank}(Z^*) = n$. □

Such a strictly complementary pair of solutions does not always exist in semidefinite programming, but the central path will get as close as possible to such a solution. The point to which the central path converges for $\mu \to 0$ has another property: it is maximally complementary.

Definition 3.1.13. An optimal pair (X^*, Z^*) is *maximally complementary* if X^* and Z^* have maximum rank among all optimal solutions. □

For being able to prove maximum complementarity we need the following lemma taken from [dKRT97]:

Lemma 3.1.14. *Let $(X_{\mu_k})_{k\in\mathbb{N}}$ be a sequence of positive definite matrices converging to a positive semidefinite matrix X^* and let $\bar{X} \in S_n^+$. If there is $K \in \mathbb{R}$ with*

$$\langle X_{\mu_k}^{-1}, \bar{X} \rangle \leq K \qquad \textit{for all} \quad k \in \mathbb{N},$$

then $\operatorname{rank}(X^*) \geq \operatorname{rank}(\bar{X})$.

Proof. Let $K \in \mathbb{R}$ and assume $r = \operatorname{rank}(\bar{X}) > 0$ because for $\operatorname{rank}(\bar{X}) = 0$ the statement is clearly true. Now we look at the eigenvalue decomposition of $\bar{X}$:

$$\bar{X} = \bar{P}\bar{\Lambda}\bar{P}^T,$$

with $\bar{P}^T\bar{P} = I_r$ and some positive definite $r \times r$ diagonal matrix Λ. We can also write the positive definite matrix X_{μ_k} using its eigenvalue decomposition:

$$X_{\mu_k} = P_k \Lambda_k P_k^T,$$

again $P_k^T P_k$ is the identity matrix and Λ is a positive definite diagonal matrix. We want to show that X^* has rank greater or equal to r. Therefore we fix a subsequence of X_{μ_k} such that at least r eigenvalues of Λ_k are bounded away from zero. Then

$$\operatorname{tr}(\bar{P}^T X_{\mu_k}^{-1} \bar{P}) = \operatorname{tr}(\bar{\Lambda}^{-1}\bar{\Lambda}\bar{P}^T X_{\mu_k}^{-1}\bar{P}) \leq \lambda_{\max}(\bar{\Lambda}^{-1})\langle \bar{X}, X_{\mu_k}^{-1}\rangle \leq \frac{1}{\lambda_{\min}(\bar{\Lambda})} K =: \hat{K}.$$

Now we define $Q_k = P_k^T \bar{P}$, then we have $Q_k^T Q_k = I_r$. By $q_{i,\cdot}^k$ we denote the i-th row of the matrix Q_k for $i = 1, \ldots, n$, obtaining

$$\operatorname{tr}(\bar{P}^T X_{\mu_k}^{-1} \bar{P}) = \operatorname{tr}(Q_k^T \Lambda_k^{-1} Q_k) = \sum_{i=1}^{n} \frac{q_{i,\cdot}^k (q_{i,\cdot}^k)^T}{\lambda_i(\Lambda_k)} \leq \hat{K}.$$

For each of the summands we have $0 \leq q_{i,\cdot}^k (q_{i,\cdot}^k)^T \leq 1$ and

$$\sum_{i=1}^{n} q_{i,\cdot}^k (q_{i,\cdot}^k)^T = \operatorname{tr}(Q_k Q_k^T) = \operatorname{tr}(Q_k^T Q_k) = r.$$

So there will be at least r elements of $\{q_{i,\cdot}^k (q_{i,\cdot}^k)^T \mid i \in \{1, \ldots, n\}\}$ that are larger than some constant $\alpha > 0$ independent of k. The r corresponding eigenvalues of Λ_k cannot get smaller than $\frac{\alpha}{\hat{K}}$. As mentioned before, eigenvalues depend continuously on the matrix, therefore at least r eigenvalues of X^* are positive because X_{μ_k} converges to X^*. This proves $\operatorname{rank}(X^*) \geq \operatorname{rank}(\bar{X})$. □

Now we are able to prove the maximum complementarity of the point the central path converges to.

Theorem 3.1.15. *Let $(\mu_k > 0)_{k\in\mathbb{N}}$, be a sequence converging to zero. The corresponding solutions (X_{μ_k}, Z_{μ_k}) of* (3.1) *converge to a maximally complementary optimal pair (X^*, Z^*) of* (SDP) *and* (DSDP).

Proof. With Theorem 3.1.9 we already know that the sequence converges to an optimal solution. So it remains to show that this solution is maximally complementary, i.e., for any solution pair $(\bar{X}, \bar{Z})$

$$\operatorname{rank}(X^*) \geq \operatorname{rank}(\bar{X}) \qquad \text{and} \qquad \operatorname{rank}(Z^*) \geq \operatorname{rank}(\bar{Z}).$$

In the proof of Theorem 3.1.9 we have seen that

$$\langle X_{\mu_k}, Z^0\rangle + \langle X^0, Z_{\mu_k}\rangle = \mu_k n + \langle X^0, Z^0\rangle.$$

Since we know that $\langle \bar{X}, \bar{Z}\rangle = 0$, it follows that

$$\langle X_{\mu_k}, \bar{Z}\rangle + \langle \bar{X}, Z_{\mu_k}\rangle = \mu_k n. \tag{3.2}$$

From the perturbed complementarity condition we get

$$X_{\mu_k} Z_{\mu_k} = \mu_k I \qquad \Longleftrightarrow \qquad X_{\mu_k} = \mu_k Z_{\mu_k}^{-1} \qquad \Longleftrightarrow \qquad Z_{\mu_k} = \mu_k X_{\mu_k}^{-1}.$$

Using this, (3.2) becomes

$$\begin{array}{rlcl} & \langle \mu_k Z_{\mu_k}^{-1}, \bar{Z}\rangle + \langle \bar{X}, \mu_k X_{\mu_k}^{-1}\rangle & = & \mu_k n \\ \Longleftrightarrow & \mu_k(\langle Z_{\mu_k}^{-1}, \bar{Z}\rangle + \langle \bar{X}, X_{\mu_k}^{-1}\rangle) & = & \mu_k n \\ \Longleftrightarrow & \langle Z_{\mu_k}^{-1}, \bar{Z}\rangle + \langle \bar{X}, X_{\mu_k}^{-1}\rangle & = & n. \end{array}$$

However, this implies that $\langle Z_{\mu_k}^{-1}, \bar{Z}\rangle \leq n$ and $\langle \bar{X}, X_{\mu_k}^{-1}\rangle \leq n$. The proof is completed by an application of Lemma 3.1.14, which implies that

$$\begin{array}{c} \operatorname{rank}(Z^*) \geq \operatorname{rank}(\bar{Z}), \\ \operatorname{rank}(X^*) \geq \operatorname{rank}(\bar{X}). \end{array}$$

□

Reformulated, this theorem states that the central path converges to a point in the relative interior of the optimal face. This point is called *analytic center* of the optimal face.

Now having proved all the tools above, we can construct an algorithm for solving the problems. We still want to solve

$$F_\mu(X, y, Z) = \begin{pmatrix} \mathcal{A}X - b \\ \mathcal{A}^T y + Z - C \\ XZ - \mu I \end{pmatrix} = 0.$$

With the Newton method we can compute a step direction $(\Delta X, \Delta y, \Delta Z)$ by solving $F_\mu + \nabla F_\mu(\Delta X, \Delta y, \Delta Z)^T = 0$. Therefore the following linearized system must be solved:

$$\begin{aligned} \mathcal{A}\Delta X &= -(\mathcal{A}X - b) && (3.3) \\ \mathcal{A}^T \Delta y + \Delta Z &= -(\mathcal{A}^T + Z - C) && (3.4) \\ \Delta X Z + X \Delta Z &= \mu I - XZ. && (3.5) \end{aligned}$$

As we have $XZ \neq ZX$, we cannot expect that there exist symmetric ΔX and ΔZ solving the three equations above. With (3.4) we get a symmetric ΔZ when solving the system for square matrices ΔX and ΔZ. We discuss this because we need the next iterate $X + \alpha\Delta X$ to be symmetric and positive definite. To handle this problem there exist various approaches, we are going to discuss only one of them. In [NT97] Nesterov and Todd discuss the idea of self-scaled barrier functions and introduce a scaling point $S = X^{\frac{1}{2}}(X^{\frac{1}{2}}ZX^{\frac{1}{2}})^{-\frac{1}{2}}X^{\frac{1}{2}}$ with

$$S^{-\frac{1}{2}}XS^{-\frac{1}{2}} = S^{\frac{1}{2}}ZS^{\frac{1}{2}}.$$

Using this, the complementarity condition can be reformulated as

$$S^{-1}\Delta XS^{-1} + \Delta Z = \mu X^{-1} - Z.$$

Thus any solution of the system is symmetric. Finally, we can formulate an interior-point-algorithm:

Algorithm 3.1.16. *Interior-point-method for solving SDPs.*
Input: *$\mathcal{A}, b, C$, and a starting point (X^0, y^0, Z^0) with positive definite matrices X^0 and Z^0.*

1. *Set $i = 0$ and $(X^i, y^i, Z^i) := (X^0, y^0, Z^0)$ and choose $\varepsilon_1, \varepsilon_2, \varepsilon_3 > 0$.*
2. *Choose μ.*
3. *Solve the system* (3.3), (3.4), *and a variant of* (3.5) *to compute $(\Delta X, \Delta y, \Delta Z)$.*
4. *Choose $\alpha \in (0, 1]$, such that $X^i + \alpha\Delta X$ and $Z^i + \alpha\Delta Z$ remain positive definite.*
5. *Set $(X^{i+1}, y^{i+1}, Z^{i+1}) := (X^i + \alpha\Delta X, y^i + \alpha\Delta y, Z^i + \alpha\Delta Z)$.*
6. *If $\|\mathcal{A}X^{i+1} - b\| \leq \varepsilon_1$ and $\left\|\mathcal{A}^T y^{i+1} + Z^{i+1} - C\right\| \leq \varepsilon_2$ and $\langle X^{i+1}, Z^{i+1}\rangle \leq \varepsilon_3$ **stop**, otherwise set $i := i + 1$ and go to step 2.*

Remark 3.1.17. *Usually the starting point in Algorithm 3.1.16 has to satisfy additional conditions. For some variants of interior-point-algorithms they need to be feasible.*

More comments on interior-point-methods can be found in [Nem04].

A note on the solver DSDP

For solving the SDPs in our branch-and-bound algorithm, we will use the solver DSDP 5.8 as presented in Chapter 4. We will now briefly present what the solver does (see [BY08]).

Remark 3.1.18. *The solver DSDP uses a dual-scaling algorithm. This dual algorithm does not need a primal solution X for the solving process and therefore no primal solutions are computed. It uses the idea presented above to guarantee symmetric iterates.*

This is the Newton system of the solver DSDP for $\mathcal{A}X = b, \mathcal{A}^T y + Z = C$ and $X = \mu Z^{-1}$:

$$\begin{aligned} \mathcal{A}(X + \Delta X) &= b && (3.6)\\ \mathcal{A}^T\Delta y + \Delta Z &= 0 && (3.7)\\ \mu Z^{-1}\Delta Z Z^{-1} + \Delta X &= \mu Z^{-1} - X. && (3.8) \end{aligned}$$

Now we rewrite this system by

1. expressing ΔX in terms of ΔZ using (3.8),
2. writing ΔZ in terms of Δy with (3.7) and
3. rewriting (3.6) using these definitions.

This yields:

$$\begin{aligned}
\Delta X &= \mu Z^{-1} - X - \mu Z^{-1}\Delta Z Z^{-1} = \mu Z^{-1} - X - X\Delta Z Z^{-1} = -X\Delta Z Z^{-1} \\
\Delta Z &= -\mathcal{A}^T \Delta y \\
\mathcal{A}(X\mathcal{A}^T \Delta y Z^{-1}) &= -(\mathcal{A}X - b) - \mathcal{A}(\mu Z^{-1} - X) = b - \mathcal{A}X.
\end{aligned}$$

The last equation can now be written in terms of Z only and we obtain

$$\begin{aligned}
\mathcal{A}(X\mathcal{A}^T \Delta y Z^{-1}) &= b - \mathcal{A}X \\
\mathcal{A}(\mu Z^{-1}\mathcal{A}^T \Delta y Z^{-1}) &= b - \mathcal{A}\mu Z^{-1} \\
\mu\mathcal{A}(Z^{-1}\mathcal{A}^T \Delta y Z^{-1}) &= b - \mu\mathcal{A}Z^{-1}.
\end{aligned}$$

However, this is nothing else than the Schur complement (see Definition (2.1.14)) of the system (3.6), (3.7), and (3.8):

$$\mu \begin{pmatrix} \langle A_1, Z^{-1}A_1Z^{-1}\rangle & \dots & \langle A_1, Z^{-1}A_mZ^{-1}\rangle \\ \vdots & \ddots & \vdots \\ \langle A_m, Z^{-1}A_1Z^{-1}\rangle & \dots & \langle A_m, Z^{-1}A_mZ^{-1}\rangle \end{pmatrix} \Delta y = b - \mu\mathcal{A}Z^{-1}. \tag{3.9}$$

The matrix on the left-hand side of (3.9) will be called M in the following, using this (3.9) changes to

$$\mu M\Delta y = b - \mu\mathcal{A}Z^{-1}.$$

The matrix M is positive definite if Z is. The DSDP solver now computes step sizes for the dual variables

$$\Delta' y := M^{-1}b \qquad \text{and} \qquad \Delta'' y := M^{-1}\mathcal{A}Z^{-1}.$$

For any $\mu > 0$

$$\Delta_\mu y := \frac{1}{\mu}\Delta' y - \Delta'' y$$

solves (3.9). The value $\Delta_\mu y$ can now be used in (3.8) to obtain

$$X_\mu := \mu(Z^{-1} + Z^{-1}(\mathcal{A}^T\Delta_\mu y)Z^{-1}).$$

Then X_μ satisfies $\mathcal{A}X_\mu = b$. Further we know that X_μ is positive definite if and only if $C - \mathcal{A}^T(y - \Delta_\mu y)$ is positive definite.

For a positive definite X_μ we obtain a new upper bound using the facts that there is a zero duality gap and $\langle X_\mu, Z\rangle = 0$.

$$\begin{aligned}\hat{z} :=& \langle C, Z_\mu\rangle\\ =&b^T y + \langle X_\mu, Z\rangle\\ =& b^T y + \langle \mu(Z^{-1} + Z^{-1}(\mathcal{A}^T\Delta_\mu y)Z^{-1}), Z\rangle\\ =& b^T y + \mu\left(\langle Z^{-1}, Z\rangle + \langle Z^{-1}(\mathcal{A}^T\Delta_\mu y)Z^{-1}, Z\rangle\right)\\ =& b^T y + \mu(n + \Delta_\mu y^T \mathcal{A} Z^{-1}).\end{aligned}$$

However, to compute this bound we do not need to compute X_μ explicitly.

We will not go into details but for the sake of completeness we remark that the solver DSDP uses a potential-reduction-method.

Remark 3.1.19. *The solver DSDP uses the dual potential function:*

$$\varphi(y) := \rho \log(\hat{z} - b^T y) - \ln\det(Z).$$

Here, $\rho \geq 1$ is a constant that weights the duality gap $\hat{z} - b^T y$. The DSDP solver is able to reduce $\varphi(y)$ at every iteration 'enough' to achieve linear convergence.

Remark 3.1.20. *The DSDP solver does not compute X_μ during the solving process. This is a special characteristic of the dual-scaling algorithm and it is responsible for the good performance. After the solving is completed and the solver converged, X^* can be computed.*

Convergence and feasibility of the solver DSDP

The algorithm used by DSDP only converges if the primal and the dual problem have a feasible region and the current solution are elements of the interior. For satisfying these assumptions, DSDP solves a slightly modified problem, where the variables y are bounded by some lower bounds $l \in \mathbb{R}^m$ and some upper bounds $u \in \mathbb{R}^m$, in the default settings $l_i = 10^{-7}$ and $u_i = 10^7$. Additionally, it bounds the trace of the matrix X by some large value $\Gamma = 10^{10}$. This leads to a slightly modified pair of problems with Lagrange variables $x^l \in \mathbb{R}^m$, $x^u \in \mathbb{R}^m$ and r for the bounds:

$$\begin{aligned}\min \quad & \langle C, X\rangle + u^T x^u - l^T x^l\\ \text{s.t.} \quad & \mathcal{A}X + x^u - x^l = b\\ & \langle I, X\rangle \leq \Gamma\\ & X \succeq 0\\ & x^u, x^l \geq 0,\end{aligned} \tag{3.10}$$

$$\begin{aligned} \max \quad & b^T y - \Gamma r \\ \text{s.t.} \quad & A_0 - \sum_{i=1}^{m} A_i y_i + Ir \succeq 0 \\ & y \in [l, u]^m \\ & r \geq 0. \end{aligned} \tag{3.11}$$

These two problems are bounded and feasible, their optimal objective values are identical. Additionally, these two problems can be expressed in the form of primal and dual pair of SDPs as we have already seen in Chapter 2. If the objective values of the primal (SDP) and the dual SDP (DSDP) exist and are equal then the solutions of the modified problems (3.10) and (3.11) are also solutions to (SDP) and (DSDP). So DSDP is able to detect infeasibility or unboundedness of (SDP) and (DSDP) by examining the solutions of (3.10) and (3.11) as follows. Let ε_D, ε_P be two parameters, then

- if $r \leq \varepsilon_D$, $\frac{\|\mathcal{A}X - b\|_\infty}{\langle I, X \rangle} > \varepsilon_P$ and $b^T y > 0$, (DSDP) is unbounded and (SDP) is infeasible and
- if $r > \varepsilon_D$, $\frac{\|\mathcal{A}X - b\|_\infty}{\langle I, X \rangle} \leq \varepsilon_P$, (DSDP) is infeasible and (SDP) is unbounded.

Remark 3.1.21. *This is a very important fact because this allows us to use DSDP in our branch-and-bound framework as described in Section 3.2. This resolves various difficulties arising in the branch-and-bound algorithm and makes DSDP the solver of choice. Without using these modified problems DSDP would have the same difficulties as the algorithms we describe in the next two subsections.*

3.1.2. Augmented Lagrangian methods

Closely related to interior-point-methods are the penalty-barrier-multiplier methods (PBM) for solving convex programming problems. The PBM methods were introduced in [BTZ95] and they combine the ideas of penalty and barrier methods with the augmented Lagrangian method. The nonquadratic augmented Lagrangians considered here have a penalty parameter which is a function of the multipliers and satisfies some properties such that the Newton method is guaranteed to behave well.

The generalized augmented Lagrangian algorithm, which is based on the modified penalty-barrier-multiplier method was introduced by [KS03] and can be used to solve semidefinite programs. A detailed description can also be found in [Sti06]. The idea of the code PENNON is to solve nonlinear problems as well as semidefinite and second-order-cone problems and to be able to exploit different types of sparsity in the problem data. One very important thing is the choice of the penalty function for matrix inequalities. We will follow [KS03] in this section.

The problem we want to solve within this section is of the following form:

$$\min_{y \in \mathbb{R}^n} \{ b^T y : \mathcal{A}^T y \preceq 0 \}, \tag{3.12}$$

where b is a n-dimensional real vector and $\mathcal{A}^T y := \sum_{i=1}^n A_i y_i + A_0$. The method for solving this problem is based on a specific choice of a one-dimensional penalty/barrier function φ that penalizes the inequality constraint. Then we construct another function Φ that penalizes the matrix inequality constraint.

Let $\varphi\colon (-\infty, b) \to \mathbb{R}$, with $0 < b \leq \infty$, be a penalty function with the following properties:

(i) φ is strictly convex, strictly increasing, and two times continuously differentiable,

(ii) $\varphi(0) = 0$,

(iii) $\varphi'(0) = 1$,

(iv) $\lim_{t\to b} \varphi'(t) = \infty$ and

(v) $\lim_{t\to -\infty} \varphi'(t) = 0$.

The standard choice of φ for nonconvex nonlinear programs is a quadratic logarithmic function. Indeed as described in [KS03] this is not a good choice for semidefinite programs because the resulting Φ might be nonconvex and the computation of the gradient and the Hessian might be very inefficient because dense matrices would appear. Therefore we will choose a different φ, namely the hyperbolic barrier function:

$$\varphi^{\text{hyp}}(t) = \frac{1}{1-t} - 1. \tag{3.13}$$

Again let $A = P^T \Lambda P$ be the eigenvalue decomposition of A with eigenvalues $\lambda_1, \dots, \lambda_d$. Using this and φ we can define the penalty function Φ.

Definition 3.1.22. Let $p > 0$ be a given real number and $\Phi_p\colon S_m \to S_m$ with

$$\Phi_p\colon A \mapsto P^T \begin{pmatrix} p\varphi(\frac{\lambda_1}{p}) & 0 & \dots & 0 \\ 0 & p\varphi(\frac{\lambda_2}{p}) & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & p\varphi(\frac{\lambda_d}{p}) \end{pmatrix} P. \tag{3.14}$$

We call Φ_p a *penalty function.* □

Using the construction of φ we have for any $p > 0$

$$\mathcal{A}^T y \preceq 0 \text{ is equivalent to } \Phi_p(\mathcal{A}^T y) \preceq 0.$$

So we know that for any $p > 0$ the problem (3.12) and the following 'augmented' problem have the same solution

$$\min_{y\in\mathbb{R}^n} \{ b^T y : \Phi_p(\mathcal{A}^T y) \preceq 0 \}.$$

Now we introduce the Lagrangian of this problem:

$$F(y,U,p) = b^T y + \langle U, \Phi_p(\mathcal{A}^T y)\rangle. \tag{3.15}$$

This function can be interpreted as a generalized augmented Lagrangian of (3.12).

In general, for calculating Φ_p, its gradient and its Hessian we explicitly need the eigenvalues of A. Certainly, for the specific choice of φ^{hyp} in the definition of Φ_p, which we denote by Φ_p^{hyp}, we can avoid this. As shown in [KS03] we can compute Φ_p^{hyp}, its gradient and its Hessian without knowing the eigenvalue decomposition of $\mathcal{A}^T y$.

Theorem 3.1.23 ([KS03, Theorem 2.2]). *Let $\mathcal{A}^T : \mathbb{R}^n \to S_m$ be a convex operator and let Φ_p^{hyp} be a function defined by* (3.14) *using φ^{hyp}. Then for any $y \in \mathbb{R}^n$ there exists $p > 0$ such that*

$$\begin{aligned}
\Phi_p^{\text{hyp}}(\mathcal{A}^T y) &= p^2 \mathcal{Z}(y) - pI \\
\frac{\partial}{\partial y_i}\Phi_p^{\text{hyp}}(\mathcal{A}^T y) &= p^2 \mathcal{Z}(y) \frac{\partial \mathcal{A}^T y}{\partial y_i} \mathcal{Z}(y) \\
\frac{\partial^2}{\partial y_i \partial y_j}\Phi_p^{\text{hyp}}(\mathcal{A}^T y) &= p^2 \mathcal{Z}(y) \left(\frac{\partial \mathcal{A}^T y}{\partial y_i} \mathcal{Z}(y) \frac{\partial \mathcal{A}^T y}{\partial y_j} - \frac{\partial^2 \mathcal{A}^T y}{\partial y_i \partial y_j} + \frac{\partial \mathcal{A}^T y}{\partial y_j} \mathcal{Z}(y) \frac{\partial \mathcal{A}^T y}{\partial y_i} \right) \mathcal{Z}(y)
\end{aligned}$$

with

$$\mathcal{Z}(y) = (\mathcal{A}^T y - pI)^{-1}.$$

Furthermore, $\Phi_p^{\text{hyp}}(\mathcal{A}^T y)$ is monotone and convex in y.

For the specific choice of $\mathcal{A}^T y = \sum_{i=1}^n A_i y_i + A_0$, with $A_i \in S_m, i = 0, 1, \ldots, m$, we have

$$\frac{\partial \mathcal{A}^T y}{\partial y_i} = A_i \text{ and } \frac{\partial^2 \mathcal{A}^T y}{\partial y_i \partial y_j} = 0.$$

So the gradient and the Hessian simplify to:

$$\begin{aligned}
\frac{\partial}{\partial y_i}\Phi_p^{\text{hyp}}(\mathcal{A}^T y) &= p^2 \mathcal{Z}(y) A_i \mathcal{Z}(y) \\
\frac{\partial^2}{\partial y_i \partial y_j}\Phi_p^{\text{hyp}}(\mathcal{A}^T y) &= p^2 \mathcal{Z}(y) \left(A_i \mathcal{Z}(y) A_j + A_j \mathcal{Z}(y) A_i \right) \mathcal{Z}(y).
\end{aligned}$$

By now we can state the basic algorithm in a very general from. We are going to present the single steps for the specific choice of the penalty function of Φ_p^{hyp} in the following. We denote the derivative of Φ with respect to A in direction U by $D\Phi(A)[U]$.

Algorithm 3.1.24. *Generalized augmented Lagrangian method for solving semidefinite programs.*
Input: *Starting points* y^1 *and* D^1, $p^1 > 0$

1. *Calculate* $y^{k+1} = \arg\min_{y \in \mathbb{R}^n} F(y, U^k, p^k)$.
2. *Update the multiplier* $U^{k+1} = D\Phi_p(\mathcal{A}^T y)[U^k]$.
3. *Choose a new penalty parameter* $p^{k+1} < p^k$.
4. *If a stopping criterion is satisfied, then* ***stop****,*
   otherwise set $k := k + 1$ *and go to step 1.*

By Theorem 3.1.23 we may use any $y \in \mathbb{R}^n$, so for starting our algorithm we choose $y^0 = 0$. For the multipliers the following start values are used:

$$\begin{aligned} U_j^0 &= \mu_j^s I_{m_j}, \qquad j = 1, \dots, k_s, \\ u_i^0 &= \mu_i^l, \qquad i = 1, \dots, k_l, \end{aligned}$$

where I_{m_j} denotes the identity matrix of dimension $m_j \times m_j$ and

$$\mu_j^s = m_j \max_{1 \le s \le n} \frac{1 + |b_j|}{1 + \left\| \frac{\partial \mathcal{A}^T y}{\partial y_s} \right\|}$$

$$\mu_i^l = \max_{1 \le l \le n} \frac{1 + |b_i|}{1 + \left\| \frac{\partial \mathcal{A}^T y}{\partial y_l} \right\|}$$

Additionally, $\pi > 0$ is chosen such that

$$\lambda_{\max}(\mathcal{A}^T y) < \pi, \qquad j = 1, \dots, k.$$

We therefore use $p^0 = \pi e$, where $e \in \mathbb{R}^{k_s + k_l}$ is the vector of all ones.

For calculating y^{k+1} in step 1 a modified Newton method combined with a cubic line-search is used. The search direction d is computed by solving the Newton equation and finding $\alpha_{\max}$ with $0 < \alpha < \alpha_{\max}$ and

$$\lambda_{\max}(\mathcal{A}_j^T(y^k + \alpha d)) < p_j^k, \qquad j =, \dots, k.$$

The update formula for U^{k+1} and the special choice of the penalty function Φ_p^{hyp} can be rewritten as:

$$U^{k+1} = (p^k)^2 \mathcal{Z}(y) U^k \mathcal{Z}(y).$$

Algorithm 3.1.24 stops if one of the following two inequalities holds for some $\varepsilon > 0$:

$$\frac{\left| b^T y^k - F(y^k, U^k, p) \right|}{|b^T y^k|} < \varepsilon, \qquad \frac{\left| b^T y^k - b^T y^{k-1} \right|}{|b^T y^k|} < \varepsilon.$$

Additionally, there is another feature in the generalized augmented Lagrangian method for SDPs. During the first three iterations the penalty vector p is not updated. More details on the updating and stopping strategies can be found in [KS03].

Remark 3.1.25. *The algorithm presented in this section seems to have many advantages. However, we are not going to use it for our framework. The data handling for using this algorithm in a general branch-and-bound framework would be very complex. So for example $\mathcal{A}^T y - pI$ needs to be nonsingular for the different choices of p. We cannot guarantee this because when fixing some variables within the branch-and-bound algorithm it may occur that a whole row and column of the matrix $\mathcal{A}^T \hat{y}$ for this specific choice of $\hat{y}$ only has zero entries. The algorithm and the resulting software PENSDP are not able to deal with this. Additionally, pure constant rows in the matrix $\mathcal{A}^T \hat{y}$, i.e., rows without any variable entries, cause difficulties.*

Furthermore, there is no handling for infeasible or unbounded problems. Especially the first kind appears very often within a branch-and-bound algorithm and PENSDP would need numerous iterations and therefore a lot of time to detect this infeasibility. We will comment on this problem in Section 3.2.

3.1.3. Bundle methods

The method presented in Section 3.1.1 for solving SDPs has one very big disadvantage. The $m \times m$ matrix M needed for computing the search direction Δy is dense in general. Storing M and computing its Cholesky factorization for large m is very expensive and time consuming. Therefore the development of other algorithms that can exploit the problem structure and are still applicable for large m is motivated. One idea is introduced by Helmberg [Hel00]: the spectral bundle method. In this section we give a rough idea how it works.

The spectral bundle method is not able so solve general SDPs, it can only solve problems that can be written as eigenvalue optimization problems of the following form:

$$\min_{y \in \mathbb{R}^m} \lambda_{\max}(C - \mathcal{A}^T y) + b^T y. \tag{3.16}$$

An extended version, which we are not going to discuss, can solve problems like (3.16) with additional bounds on the variable y. It can be shown that problems of type (3.16) are equivalent to the duals of semidefinite programs with a bounded feasible set. We will now consider a slightly modified version of the primal and dual SDPs:

$$\begin{aligned} \max \quad & \langle C, X \rangle \\ \text{s.t.} \quad & \mathcal{A}X = b \\ & X \succeq 0, \end{aligned} \tag{P}$$

and

$$\begin{aligned} \min \quad & b^T y \\ \text{s.t.} \quad & \mathcal{A}^T y - C = Z \\ & Z \succeq 0, \end{aligned} \tag{D}$$

Additionally, we make the following assumption.

Assumption 3.1.26. *There exists $\bar{y} \in \mathbb{R}^m$ with $I = \mathcal{A}^T \bar{y}$.*

Using this assumption we can reformulate the dual SDP (D) in the form of (3.16) because $Z \succeq 0$ is equivalent to $\lambda_{\min}(Z) \geq 0$. Furthermore, we have

$$0 \geq -\lambda_{\min}(Z) = \lambda_{\max}(-Z).$$

A Lagrange multiplier κ can now be used to put this constraint into the objective function and obtain

$$\min_y \kappa\lambda_{\max}(C - \mathcal{A}^T y) + b^T y. \tag{3.17}$$

We summarize what is known in the following proposition [Hel00].

Proposition 3.1.27. *Let $\mathcal{A}$ satisfy Assumption 3.1.26. Then* (D) *is equivalent to* (3.16) *for $\kappa = \max\{0, b^T\bar{y}\}$. If* (P) *is feasible then all its feasible solutions X satisfy* $\operatorname{tr}(X) = \kappa$. *Additionally, the primal optimum is attained and is equal to the infimum of the dual problem.*

Proof. We start the proof with $\bar{y}$ satisfying $\mathcal{A}^T\bar{y} = I$. Then the ray $\{y + \lambda\bar{y} : \lambda \geq \lambda_{max}(C - \mathcal{A}^T y)\}$ is feasible for (D) and $y \in \mathbb{R}^m$ due to the following inequality

$$\mathcal{A}^T(y + \lambda\bar{y}) - C = \lambda\mathcal{A}^T\bar{y} + \mathcal{A}^T y - C = \lambda I - (C - \mathcal{A}^T y) \succeq 0.$$

Now two cases are distinguished, $b^T\bar{y} < 0$ and $b^T\bar{y} \geq 0$.

(i) $b^T\bar{y} < 0$: then $\kappa = 0$ and the objective values of (D) and (3.17) are exactly the same and minus infinity along any ray $\{y + \lambda\bar{y}\}$ for $\lambda \to \infty$.

(ii) $b^T\bar{y} \geq 0$: then $\kappa = b^T\bar{y}$ and (3.17) is constant along directions $\lambda\bar{y}$ for $\lambda \in \mathbb{R}$ because

$$\begin{aligned} \kappa\lambda_{\max}(C - \mathcal{A}^T(y + \lambda\bar{y}) + b^T(y + \lambda\bar{y}) &= \kappa\lambda_{\max}(C - \mathcal{A}^T y - \lambda I) + b^T y + \lambda\kappa \\ &= \kappa\lambda_{\max}(C - \mathcal{A}^T y) - \kappa\lambda + b^T y + \lambda\kappa \\ &= \kappa\lambda_{\max}(C - \mathcal{A}^T y) + b^T y \end{aligned}$$

If we choose $\hat{\lambda} = \lambda_{\max}(C - \mathcal{A}^T y)$ we know that $\lambda_{\max}(C - \mathcal{A}^T(y + \hat{\lambda}\bar{y})) = 0$. So $y + \hat{\lambda}\bar{y}$ is feasible for (D) and has the same objective value as (3.17). Conversely we have for any feasible solution y of (D) that

$$y + \mathcal{A}^T(y + \lambda y)\bar{y}$$

is also a feasible solution of (D) and the objective values of the two problems are identical. Because of $\lambda_{\max}(C - \mathcal{A}^T y) = \lambda_{\max}(-Z) \leq 0$ the objective value is not greater than $b^T y$. This proves the equivalence of (D) and (3.17).

The last part of the proposition can be proved using the Strong Duality Theorem 2.4.9. As (D) is strictly feasible, the supremum of (P) is attained if (D) is finite. Additionally, every feasible solution X of (P) satisfies $\mathcal{A}X - b = 0$ and therefore we have

$$0 = \langle \mathcal{A}X - b, \bar{y} \rangle = \langle X, \mathcal{A}^T \bar{y} \rangle - b^T \bar{y} = \langle X, I \rangle - \kappa = \operatorname{tr}(X) - \kappa,$$

which completes the proof. □

Now that we know which kind of problems can be modeled as eigenvalue problem we come back to the solving techniques and some properties of the maximum eigenvalue function. The maximum eigenvalue of a symmetric matrix can be characterized in the following way:

$$\lambda_{\max}(X) = \max_{\|v\|=1} v^T X v.$$

This is called the Rayleigh-Ritz ratio. Now recall that $v^T X v = \langle X, vv^T \rangle$. As the set

$$\mathcal{W} = \{W \succeq 0 : \operatorname{tr}(W) = 1\}$$

is the convex hull of $\{vv^T : \|v\| = 1\}$ we can rewrite the maximum eigenvalue problem as an SDP:

$$\lambda_{\max}(X) = \max\{\langle X, W \rangle : W \in \mathcal{W}\}. \tag{3.18}$$

The function $\lambda_{\max}(\cdot)$ is convex and Lipschitz continuous and $\mathcal{W}$ is bounded. As the maximum eigenvalue function $\lambda_{\max}(\cdot)$ is also nonsmooth, the idea is to use techniques from nonsmooth optimization.

Since nonsmooth functions do not have gradients we are going to use the more general concept of subgradients instead.

Definition 3.1.28. A vector $s \in \mathbb{R}^n$ is called a *subgradient* of a convex function $f \colon D \to \mathbb{R} \cup \{+\infty, -\infty\}$ with $D \subseteq \mathbb{R}^n$ at a point $x \in D$ if it satisfies the following inequality:

$$f(y) \geq f(x) + \langle s, y - x \rangle \qquad \text{for all } y \in D.$$

The set of all subgradients of f at x is called *subdifferential* at x and is denoted by $\delta f(x)$. □

Thus a subgradient together with the function value $f(x)$ is a supporting hyperplane of the function in the point x.

Theorem 3.1.29 ([HUL04, Theorem 2.2.1])**.** *For a convex function $f \colon \mathbb{R}^n \to \mathbb{R}$ the following three properties are equivalent*

1. *f is minimized at x over $\mathbb{R}^n$, i.e., $f(y) \geq f(x)$ for all $y \in \mathbb{R}^n$,*

2. *$0 \in \delta f(x)$ and*

3. *$f'(x, d) \geq 0$ for all $d \in \mathbb{R}^n$.*

Proof. For the proof of this theorem we refer the reader to [HUL04]. □

In our case the subgradients are the eigenvectors to the maximum eigenvalues. These subgradients for the maximum eigenvalue function at X are the linear forms satisfying the subgradient inequality for all symmetric $n \times n$ matrices Y

$$\lambda_{\max}(Y) \geq \lambda_{\max}(X) + \langle W, Y - X \rangle.$$

This set of subgradients, the subdifferential, can be characterized as the following matrices

$$\begin{aligned} W \in \text{Argmax}\{\langle X, W \rangle : W \in \mathcal{W}\} &= \{W \in \mathcal{W} \langle X, W \rangle = \lambda_{\max}(X)\} \\ &= \{PVP^T : \text{tr}(V) = 1, V \succeq 0\}, \end{aligned}$$

where the columns of P are an orthogonal basis of the eigenspace of the maximum eigenvalue of X.

Coming back to problem (3.17) for simplicity we will now assume that $\kappa = 1$. So we consider the function

$$f(y) = \lambda_{\max}(C - \mathcal{A}^T y) + b^T y;$$

by (3.18) this can be reformulated to

$$f(y) = \max_{W \in \mathcal{W}} \langle C - \mathcal{A}^T y, W \rangle + b^T y = \max_{W \in \mathcal{W}} \langle C, W \rangle + \langle b - \mathcal{A}W, y \rangle. \tag{3.19}$$

Using this we can formulate the subdifferential of f at y

$$\begin{aligned} \delta f(y) &= \{b - \mathcal{A}W : W \in \mathcal{W}, \langle C - \mathcal{A}^T y, W \rangle = \lambda_{\max}(C - \mathcal{A}^T y)\} \\ &= \{\nabla f_W : W \in \delta \lambda_{\max}(C - \mathcal{A}^T y)\}. \end{aligned}$$

As $\mathcal{W}$ is bounded, the subdifferential is bounded. From Theorem 3.1.29 we know that $\bar{y}$ is a minimizer of the function f if and only if $0 \in \delta f(\bar{y})$ or equivalently:

$$\begin{aligned} \bar{y} \in \text{Argmin} f \quad \iff \quad & \text{there exists} \quad \overline{W} \in \mathcal{W} : \mathcal{A}\overline{W} = b \\ & \text{and} \quad \langle C - \mathcal{A}^T \bar{y}, \overline{W} \rangle = \lambda_{\max}(C - \mathcal{A}^T \bar{y}). \end{aligned}$$

The last step to obtain a minorizing function for f is to replace the set $\mathcal{W}$ by some subset $\widehat{\mathcal{W}} \subseteq \mathcal{W}$ in (3.19):

$$f_{\widehat{\mathcal{W}}}(y) := \max_{W \in \widehat{\mathcal{W}}} f_W(y) \leq f_{\mathcal{W}}(y) = f(y) \qquad \text{for all } \widehat{\mathcal{W}} \subseteq \mathcal{W}, y \in \mathbb{R}^m. \tag{3.20}$$

Note that extremal eigenvalues and eigenvectors of structured matrices can be computed using an iterative method. For example one could use the Lanczos method.

General subgradient methods construct a polyhedral cutting plane model of the cost function using accumulated subgradient information. One such method is the proximal bundle method [Kiw90]. This method starts with the following cutting plane model:

$$\hat{f}^k(y) = \max_{i=1,\dots,k} f(y^i) + \langle g^i, y - y^i \rangle.$$

This linear approximation is of good quality in the neighborhood of y^i. So the next trial point y^{k+1} is the minimizer of

$$f^k(y) = \hat{f}^k(y) + \frac{u}{2}\left\|y - \hat{y}^k\right\|^2.$$

This is an augmented model with a finite and unique minimum. The distance of y^{k+1} to $\hat{y}^k$ (which is also called the stability center) can be controlled using the weight $u > 0$.

At this point we have to distinguish two different cases. The first one is if we can determine some progress in the function value $f(y^{k+1})$ relative to the decrease predicted by the model value $\hat{f}(y^{k+1})$. This is the case if for some $\kappa \in (0, 1)$

$$f(\hat{y}^k) - f(y^{k+1}) \geq \kappa\left[f(\hat{y}^k) - \hat{f}^k(y^{k+1})\right].$$

Then we set $\hat{y}^{k+1} = y^{k+1}$ and call this a *descent step.*

Otherwise we need to improve the model by adding another cutting plane but keeping the old stability center $\hat{y}^{k+1} = \hat{y}^k$. This is called a *null step.* These steps will be repeated until the predicted decrease of the cutting plane model is small related to the function value.

The spectral bundle method is a specialization of the proximal bundle method and improves it by additionally exploiting the semidefinite structure of the maximum eigenvalue function. So it chooses a specific cutting plane model:

$$\widehat{\mathcal{W}}^k = \{P_k V P_k^T + \alpha \overline{W}_k : \operatorname{tr}(V) + \alpha = 1, V \in S_{r_k}^+, \alpha \geq 0\}, \tag{3.21}$$

with a $n \times r_k$ matrix P_k and $\overline{W}_k \in \mathcal{W}$. The corresponding cutting plane model is $f_{\widehat{\mathcal{W}}^k}$ as in (3.20).

The augmented model for the spectral bundle method can be expressed using the augmented Lagrangian

$$L^k(y, W) := f_W(y) + \frac{u}{2}\left\|y - \hat{y}^k\right\|^2 = \langle C - \mathcal{A}^T y, W\rangle + b^T y + \frac{u}{2}\left\|y - \hat{y}^k\right\|^2. \tag{3.22}$$

Now set

$$f^k(y) := \max_{W \in \widehat{\mathcal{W}}^k} L^k(y, W) = f_{\widehat{\mathcal{W}}^k}(y) + \frac{u}{2}\left\|y - \hat{y}^k\right\|^2.$$

For optimizing the dual problem, we have to solve $\min_y f^k(y)$ and by the next lemma this is equivalent to

$$\max_{W \in \widehat{\mathcal{W}}^k} \min_y L^k(y, W).$$

Lemma 3.1.30 ([Hel00, Lemma 5.2.2])**.** *Let L^k be as defined in* (3.22)*. Then*

$$\min_y \max_{W \in \widehat{\mathcal{W}}^k} L^k(y, w) = L^k(y^{k+1}, W^{k+1}) = \max_{W \in \widehat{\mathcal{W}}^k} \min_y L^k(y, w)$$

where $y^{k+1} = y^k_{\min}(W^{k+1})$ is unique, and W^{k+1} an optimal solution of

$$\begin{aligned} \min \quad & \frac{1}{2u}\|b - \mathcal{A}W\|^2 - \langle W, C - \mathcal{A}^T\hat{y}^k\rangle - b^T\hat{y}^k \\ \text{s.t.} \quad & P_kVP_k^T + \alpha\overline{W}_k = W \\ & \operatorname{tr}(V) + \alpha = 1 \\ & V \succeq 0 \\ & \alpha \geq 0. \end{aligned} \tag{3.23}$$

For fixed $W \in \widehat{\mathcal{W}}^k$ the corresponding optimal y can be expressed as

$$y^k_{\min}(W) := \hat{y}^k + \frac{1}{u}(\mathcal{A}W - b) = \hat{y}^k - \frac{1}{u}\nabla f_W.$$

The problem (3.23) solved in Lemma 3.1.30 is a quadratic semidefinite program. For small r_k in (3.21) the quadratic SDP has small dimensions. Therefore it could be solved efficiently with interior-point-methods. Additionally, the optimal solution W^{k+1} can be used to compute a new candidate point:

$$y^{k+1} = y^k_{\min}(W^{k+1}).$$

Now we need to update the model by adding the new subgradient information $W_S = vv^T$ to the model. This is done by adding the new eigenvector v as orthonormalized column of P. Thus the number of columns r increases by one, note that it is important to keep r bounded as it is responsible for the efficiency of solving (3.23). We are not going into details here.

The algorithm should stop if the objective value is close to the optimal value of $\min_y f(y)$. A lower bound for this minimum is not available, but as in the augmented model the quadratic term can be interpreted as a trust region constraint for the cutting plane model, we may view y^{k+1} as the minimizer of $f_{\widehat{W}^k}$. A lower bound on f over a ball can be obtained from $f_{\widehat{W}^k}(y^{k+1}) = f_{W^{k+1}}(y^{k+1})$ for a small weight u. So if the gap between $f(\hat{y}^k)$ and $f_{W^{k+1}}(y^{k+1})$ is small enough, good progress of the algorithm within the trust region cannot be expected. Therefore we stop if

$$f(\hat{y}^k) - f_{W^{k+1}}(y^{k+1}) \leq \varepsilon(|f(\hat{y}^k)| + 1).$$

Now we can formulate the spectral bundle method.

Algorithm 3.1.31. *Spectral Bundle Method.*
Input: $y^0 \in \mathbb{R}^m, \varepsilon > 0, \kappa \in (0,1)$ *and a weight* $u > 0$.

1. *Set $k = 0, \hat{y}^0 = y^0$, compute $f(y^0)$ and $\widehat{\mathcal{W}}^0$.*

2. *(Trial point finding): Compute W^{k+1} and $y^{k+1} = y^k_{\min}(W^{k+1)}$ using Lemma 3.1.30.*

3. *(Stopping criterion): If $f(\hat{y}^k) - f_{W^{k+1}}(y^{k+1}) \leq \varepsilon(|f(\hat{y}^k)| + 1)$ then **stop**.*

4. *(Evaluation): Find* $W_S^{k+1} \in \operatorname{Argmax}_{W \in \mathcal{W}} \langle C - \mathcal{A}^T y^{k+1}, W \rangle$ *and determine* $f(y^{k+1})$.

5. *(Descent test): If* $f(\hat{y}^k) - f(y^{k+1}) \geq \kappa[f(\hat{y}^k) - f_{\mathcal{W}^{k+1}}(y^{k+1})]$ *then set* $\hat{y}^{k+1} = y^{k+1}$ *(descent step); otherwise set* $\hat{y}^{k+1} = \hat{y}^k$ *(null step).*

6. *(Model updating): Choose a* $\widehat{\mathcal{W}}^{k+1} \supset \{W^{k+1}, W_S^{k+1}\}$ *of the form* (3.21).

7. *Increase* $k := k + 1$ *and go to step* 2.

Observation 3.1.32. *As the spectral bundle method is nothing else than a cutting plane method it also shows the typical behavior of those methods. After a lot of improvement in the beginning, the method stagnates if it comes close to the optimal solution.*

Remark 3.1.33. *We are not going to use the spectral bundle method for our branch-and-bound algorithm because it is not able to solve general SDPs. Assumption 3.1.26 together with Proposition 3.1.27 limits the solvable problems to those with constant trace. We want to solve general SDPs and especially problems from Truss Topology Design. Trusses do not satisfy this condition in general. So using the spectral bundle method and the code conic bundle for solving is no possibility.*

3.2 Solving MISDPs using a branch-and-bound framework

Solving mixed-integer linear problems is normally done using the branch-and-bound algorithm or one of its extensions. However, this algorithm is not restricted to the linear problems, it can be extended to general mixed-integer problems, where the relaxation is still convex. In the following we describe how to use the branch-and-bound algorithm to solve general convex mixed-integer semidefinite programming problems. The branch-and-bound algorithm was introduced by Land and Doig [LD60].

Although, the generalization for MISDPs seems straightforward we face many difficulties due to the facts that we are using interior-point-methods within a branch-and-bound algorithm and that strong duality is not always satisfied.

3.2.1. The general idea

The basic concept of the branch-and-bound algorithm is to partition the feasible set into smaller subsets and to solve relaxations of minimization problems to obtain a lower bound on the problems objective value. Certainly, this is only possible if the corresponding relaxed problem is convex. Additionally, the algorithm searches for feasible solutions to obtain an upper bound on the objective value. Using the upper and lower bound the algorithm can finally prove optimality of one of the feasible solutions found. A more formal description of the algorithms for binary problems is given in the following. The algorithm works for integer variables but for the sake of simplicity we only present the binary version.

Let P be a convex minimization problem in variables $x \in \mathbb{R}^n$ with linear objective function $c^T x$ for some coefficient vector $c \in \mathbb{R}^n$ and over a convex set C:

$$P_C := \min_{x \in \mathbb{R}^n} \{c^T x \mid x \in C\}.$$

Furthermore, let $P_{\{0,1\},d}$ be the binary version of P_C, i.e., $x \in \{0,1\}^d \times \mathbb{R}^{n-d}$:

$$P_{\{0,1\},d} := \min_{x \in \{0,1\}^d \times \mathbb{R}^{n-d}} \{c^T x : x \in C\}.$$

As $P_{\{0,1\},d}$ is nonconvex and we therefore do not have a description of C as convex hull of its integer points, solving the problem with a simplex solver will not lead to an integer solution. This is the reason why we need the branch-and-bound algorithm. This algorithm partitions the feasible set into smaller subsets until these subsets are convex again, normally until all integer or binary variables are fixed to a certain value. This technique will span a tree of subproblems, the so-called branch-and-bound tree.

Observation 3.2.1. *Note that for a partition of the feasible set C into two convex subsets C_1 and C_2 we have for $x \in \{0,1\}^d \times \mathbb{R}^{n-d}$:*

$$\min\{c^T x : x \in C\} = \min\left\{\min\{c^T x : x \in C_1\}, \min\{c^T x : x \in C_2\}\right\}.$$

Obviously the relaxed problem P_C always gives a lower bound for the binary problem $P_{\{0,1\},d}$, as P_C is a minimization problem over a larger feasible set than $P_{\{0,1\},d}$ is. Without loss of generality choose one of the subsets C_1, C_2. Then P_C gives a lower bound for the relaxation P_{C_1} over the convex subset C_1. Moreover, P_{C_1} is again a lower bound for $\min_{x \in \{0,1\}^d \times \mathbb{R}^{n-d}} \{c^T x : x \in C_1\}$. Now C_1 can be partitioned as well and P_{C_1} gives a lower bound of each of the relaxations over the convex subsets of C_1 and so on.

Using these ideas we can state the branch-and-bound algorithm.

Remark 3.2.2. *Note that these lower bounds computed by the relaxation are only locally valid for the problem they are computed for. Furthermore, they are valid for all the problems over subsets of the feasible set of the relaxation. In other words if a local bound is valid for a specific node in the tree, then this bound is also valid for all children of this node in the tree, but not necessarily for other nodes. So for example the lower bound obtained from $\min_{x \in \mathbb{R}^n} \{c^T x : x \in C_1\}$ is not necessarily valid for $\min_{x \in \mathbb{R}^n} \{c^T x : x \in C_i \subset C_2\}$. Only the lower bound obtained from P_C, which we also call the root relaxation, is valid for all subproblems. However, a globally valid lower bound can be obtained from the subproblems at the leaves of the current branch-and-bound tree. This lower bound is the minimum over all the objective values of the subproblems in the current leaves.*

Algorithm 3.2.3. *Branch-and-bound algorithm for convex mixed-binary minimization problems.*
Input: *Problem $P_{\{0,1\},d}$, its relaxation P_C and $\varepsilon > 0$.*

1. *Initialize lower bound $l = -\infty$, upper bound $u = \infty$ and problem pool $Q = \{P_C\}$.*

2. *If Q is empty **stop**, otherwise choose $P \in Q$ and set $Q := Q \setminus \{P\}$.*

3. *Solve P, denote its solution by $\bar{x}$.*

   If $c^T\bar{x} > u$ or P is infeasible go to step 2.

   Otherwise let S_{leaf} be the set of the relaxation solutions of all the current leaves of the branch-and-bound tree and update the global lower bound $l := \min\{c^T\bar{x}, S_{\text{leaf}}\}$.

4. *If $\bar{x}$ is feasible for $P_{\{0,1\},d}$ and $c^T\bar{x} < u$ set $u := c^T\bar{x}$.*

5. *If $u - l < \varepsilon$, **stop**, otherwise go to the next step.*

6. *Choose a variable x_i that is supposed to be binary but has a nonbinary value in $\bar{x}$.*

7. *Partition the feasible set C of P into two subsets by fixing variable x_i to zero respectively one:*
$$C = C_1 \cup C_2 = (C \cup \{x_i = 1\}) \cup (C \cup \{x_i = 0\}).$$

8. *Create new subproblems P_{C_1} and P_{C_2} for each of these sets, add them to the solution pool*
$$Q := Q \cup \{P_{C_1}, P_{C_2}\}$$
   and go to step 2.

This algorithm shows the general basic concept. Note that we did not define which solver we use, this depends on the problem class. Standard software where this algorithm is implemented, such as CPLEX [CPL12], GUROBI [GUR11] or SCIP [SCI12], normally use the dual simplex method for solving linear programs. However, it is also possible to use other solvers, for example an interior-point-method or an nonlinear programming solver.

Additionally, there are numerous different ideas how to extend Algorithm 3.2.3, for example heuristics can be applied to obtain better feasible solutions, or cutting planes can be used to obtain a better lower bound, then the algorithm would be called branch-and-cut. A very important technique is so-called presolving, which tries to clean up the data, detect and exploit structures in the constraints and variables.

Moreover, there exist many different branching strategies that decide on which variable the algorithm is going to branch (steps 6 to 8). For these branching decisions the feasibility of the different constraints describing the set C are not necessarily considered. Also the node selection (step 2) is very important for the performance of the algorithm.

3.2.2. Solving MISDPs

As mentioned before, using SDP relaxations within a branch-and-bound algorithm is not as easy as it seems. With our definition of MISDPs this kind of problems has a convex SDP relaxation and therefore we can use Algorithm 3.2.3 in the form we stated it. However, we have to take care of the solving process by the interior-point-method because the Slater constraint qualification, i.e., the existence of a strictly feasible point for the primal and the dual problem, might not be satisfied. So the critical point we have to focus on is step 3.

Moreover, there is more bad news: most of the extensions for speeding up the branch-and-bound algorithm described above will not work for SDPs. This is because for example some of them need information about the reduced costs or the slack variables of the linear constraints. For SDPs we do not have this information. We will discuss this problem in the following section. First we go into detail with step 3 of Algorithm 3.2.3.

For this purpose we again state the problem we want to solve, as presented in Section 2.3.1. For $m \times m$-dimensional, symmetric matrices A_i and objective coefficients $b \in \mathbb{R}^n$, we want to solve the following problem:

$$\begin{aligned} \min \quad & b^T y \\ \text{s.t.} \quad & \sum_{i=1}^{n} A_i y_i - A_0 \succeq 0 \\ & y \in \{0,1\}^d \times \mathbb{R}^{n-d}. \end{aligned} \tag{MISDP}$$

To simplify notation, we introduce the matrix $Z := \sum_{i=1}^{n} A_i y_i - A_0$. The question we need to ask is, what could go wrong when solving SDPs within a branch-and-bound algorithm. Before being able to answer this question we need to choose a method for solving SDPs. As mentioned in Section 3.1 we are going to use a dual-scaling interior-point-algorithm.

Now we go through the difficulties the branch-and-bound algorithm might produce for interior-point-methods. Solving these difficulties can be done by choosing the right solver and additional work in the branch-and-bound framework. We describe our ideas later.

(1) A variable is fixed to some value, which means that there exists an equality constraint.

(2) Due to variable fixings some rows and columns of the matrix Z are independent of the variables y_i.

(3) As a special case of (2): Due to variable fixings some rows and columns of the matrix Z only have zero entries.

(4) Due to variable fixings the subproblem is infeasible.

(5) All integer variables are fixed, but there are some continuous variables.

(6) All variables are fixed.

(7) The subproblem is unbounded.

The different cases presented above naturally appear in a branch-and-bound algorithm and we cannot prevent them. Any of them can cause serious trouble in the solving process of SDPs. So it would be a good idea to eliminate these difficulties before starting the SDP solver, so that the SDP solver would terminate normally. For some of the cases, namely (1), (2), (3), and (6), this is exactly what we do. However, in some cases this is not possible or too costly, therefore we need the solver to be able to handle difficulties (4), (5), and (7).

Difficulties (1), (2), and (3) are all of the same kind and can be solved similarly, which we describe in the following.

(1): If a variable y_k for some k is fixed to the value c, we can rewrite $\sum_{i=1}^{n} A_i y_i - A_0 \succeq 0$ in the following way:

$$\sum_{i=1, i\neq k}^{n} A_i y_i - \tilde{A}_0 \succeq 0$$

where $\tilde{A}_0 = A_0 - A_k c$. For the solver the number of variables is reduced by one and it does not realize that there is one fixed variable.

(2): If there is a row in the matrix Z where no variables appear, we can use Theorem 2.1.15 to eliminate this constant row and reduce the number of rows and columns of the matrix by one. Let E be the index set of rows with nonconstant entries, Z_E be the corresponding minor of Z and k be the row with only constant entries z_{ki}, then

$$Z = \begin{pmatrix} z_{kk} & z_{ki} \\ z_{ki}^T & Z_E \end{pmatrix} \succeq 0 \qquad \Longleftrightarrow \qquad Z_E - z_{ki}^T z_{kk}^{-1} z_{ki} = Z_E - \frac{1}{z_{kk}} z_{ki}^T z_{ki} \succeq 0.$$

The value of z_{kk} is nonzero because otherwise, the whole row would be zero and we would be in case (3). Let A_{iE} be the minor corresponding to the index set E of the i-th constraint matrix. Transformed into the original description of the constraint matrix we have:

$$\sum_{i=1}^{n} A_{iE} y_i - A_{0E} - z_{ki}^T z_{kk}^{-1} z_{ki} = \sum_{i=1}^{n} A_{iE} y_i - \tilde{A}_{0E} \succeq 0,$$

where $\tilde{A}_{0E} = A_{0E} + z_{ki}^T z_{kk}^{-1} z_{ki}$.

(3): Rows with only zero entries do not affect the semidefiniteness of a matrix, so they can just be deleted and the matrix size is reduced by the number of deleted rows. This can be shown by decomposing the matrix into its block components and using Observation 2.1.9. Let P be the index set of rows with nonzero entries and Z_P be the corresponding minor, then

$$Z = \begin{pmatrix} 0 & 0 \\ 0 & Z_P \end{pmatrix} \succeq 0 \qquad \Longleftrightarrow \qquad Z_P \succeq 0,$$

as zero is always positive semidefinite.

Notes on the implementation of these strategies can be found in Section 4.2.

The most dangerous problem from a solver point of view is (6) because there are no interior points at all. In this case we cannot start an interior-point-algorithm. However, we do not need to start the solver because we already have a solution in this case. The only thing left to do is checking the solution for feasibility.

Checking feasibility or rather detecting infeasibility is one major part of solving semidefinite programs from a constraint programming point of view. Therefore an additional tool is needed that can efficiently check the positive semidefiniteness of a matrix. We discuss this in detail in Section 4.4.

The solver we choose for solving the SDP relaxations based on a dual-scaling interior-point-algorithm is able to detect infeasibility and unboundedness efficiently. This is one reason why we do not use the augmented Lagrangian based solver PENSDP. It is not able to detect unboundedness and for detecting that the problem it is currently solving is infeasible, PENSDP needs a huge number of iterations and therefore a lot of time. So the difficulties (4) and (7) are treated by the correct choice of the solver. Additionally, for speeding up the solving process there are some strategies for detecting infeasibility in case (4) without explicitly solving the SDP. We discuss this later in Section 3.3.

The solver DSDP is sometimes even able to handle problem (5) where all integer variables are fixed. In some cases we cannot be sure that the solver behaves correctly, therefore we explicitly need to check the feasibility of these problems. In the following section we are going to describe a penalty approach for this.

3.2.3. Checking feasibility

Due to numerical difficulties it may happen that a branch-and-bound subproblem with some fixed variables cannot be solved correctly, so the solver does not converge. This does not mean that there does not exist a feasible solution, the solver was just not able to find one. In this case cutting off the node and consequently the whole subtree below that node is wrong, this may only be done if the solver reports real infeasibility. So we need to find out if the subproblem really is infeasible.

One way of doing this is solving a pure feasibility formulation of the current problem without the objective function. If this feasibility formulation is not able to find a feasible point then there is no feasible point. Hence the subproblem is infeasible and the whole subtree which would be developed out of this problem can be cut off. This feasibility formulation is similar to the penalty formulation we presented at the end of Section 3.1.1.

We are looking at the following relaxed model:

$$
\begin{aligned}
\min \quad & b^T y \\
\text{s.t.} \quad & \sum_{i=1}^{n} A_i y_i - A_0 \succeq 0 \\
& y \in \mathbb{R}^n.
\end{aligned}
$$

Now we introduce a new variable α and modify the problem:

$$\begin{aligned} \min \quad & \alpha \\ \text{s.t.} \quad & \sum_{i=1}^{n} A_i y_i - A_0 + \alpha I \succeq 0 \\ & y \in \mathbb{R}^n. \end{aligned}$$

Solving this modified problem tells us whether there is a feasible solution to the original problem. If $\alpha > 0$ there is no feasible point in the original problem because a positive α means that we need to add something to the matrix $\sum_{i=1}^{n} A_i y_i - A_0$ independent of the values of the variables y_i to make it positive semidefinite. This in turn means that the semidefinite constraint of the original problem can never be satisfied.

This modification completely changes the structure of the problem, therefore it is more efficient for detecting real infeasibility than the formulation DSDP uses, as described at the end of Section 3.1.1 and in Remark 3.1.21. Certainly, DSDP only needs to solve the problem once because it is able to transform the objective value and the optimal solution of its modified version back to the original problem. Using the reformulation presented here we need to solve the problem twice because we cannot be sure that we found the optimal solution of the original problem.

For the performance this does not matter that much because we only need to check for feasibility if the DSDP approach fails in solving. As our experiments showed this does not occur very often.

Remark 3.2.4. *The modification presented above could also be applied to the solver PENSDP and of course this would resolve some of the difficulties we mentioned before. However, as just remarked this is very expensive because we need to solve every problem in the branch-and-bound tree twice. This is why we decided not to use the solver PENSDP.*

3.2.4. Restrictions to the models

Obviously not everything can be modeled as a semidefinite problem. Furthermore, even if a semidefinite reformulation is available not every kind of linear constraint is allowed. So for example equality constraints will always cause trouble for SDP solvers. Therefore the data format we use can only represent inequalities. Thus we should always try to reformulate the model into one with inequality constraints only. Note that even implicit equalities given through two inequalities will cause trouble, depending on the SDP solver used. We will explain this format in detail in Section 4.1.

There are even more difficulties in MISDPs that are not caused by the branch-and-bound algorithm, but by the model itself. In the MIP world it is a widely used technique to make choices within the model from a set of possibilities via binary variables and a constraint that ensures that only one of these variables is nonzero. For example this is used when modeling a network design problem where we need to decide which kind of edge is used to transport some flow through the network. One possibility of modeling such a situation is to define variables y_{ij} that are equal to one if edge i is of kind j. For the capacity of the network every kind of edge has an influence of d_j and so the capacity restriction with minimum capacity $\mathrm{Cap}_{\min}$ would look like:

$$\sum_{i=1}^{n}\sum_{j=1}^{p} d_j y_{ij} \geq \mathrm{Cap}_{\min}.$$

Then, of course, every edge can only be of at most one kind, but it is also possible that the edge does not exist in the final network. This leads to the following constraint:

$$\sum_{j=1}^{p} y_{ij} \leq 1 \quad \text{for all } i = 1, \ldots, n. \tag{3.24}$$

Transporting this idea to an SDP model, where such a choice is also possible, would lead to the same constraint as before (3.24) and the following semidefinite constraint:

$$\sum_{i=1}^{n}\sum_{j=1}^{p} d_j A_i y_{ij} - A_0 \succeq 0.$$

In this constraint the different matrices $d_k A_i$ for a fixed i and all k are linearly dependent. This can cause undefined behavior in some solvers because the solver does not know that in the solution only one of these matrices $d_k A_i$ is present. In the solver we use this problem does not occur.

Summing up we conclude that using MISDPs with additional constraints should always be done very carefully. Equality constraints or dependencies coming from discrete decisions can cause trouble for interior-point-methods.

3.3 Speeding up the branch-and-bound algorithm

One bad news for solving MISDPs within a branch-and-bound framework is that numerous nice features provided by these frameworks do not work for MISDPs. We already mentioned this and here we want to take a closer look which of the features can be used and what else can be done to speed up the solving process. We will not go into details how the different features are implemented, thus the reader is referred to [Ach07].

In the following we will call things like reduced costs, row activities or slack values and other things coming from the linear programming world and which are available when solving problems with the simplex method *LP information*. This LP information is what most of the features need and we cannot provide when solving SDP relaxations.

Note that it is possible that our MISDP consists of semidefinite and linear constraints. On the linear constraints all features can be used that do not explicitly need an LP to be solved. So for example most of the presolving components are available.

3.3.1. Heuristics, separation routines, node selection and branching rules

Most of the ideas in heuristics, separation routines and node selection rules are based on linear programming and will not work for MISDPs. For example strong branching has to solve some LPs in order to decide how to branch. This is a very successful strategy in the MIP context because LPs can be solved very fast. Solving SDPs is expensive, therefore this strategy is not promising in our context. The same idea is used in many different diving heuristics. They solve LPs to find feasible solutions, which again is too expensive for SDPs.

Separation routines also use problem specific information and most of them only work for LPs. Additionally, we want to remark that SDP relaxations already produce very good lower bounds such that most of the separation routines do not cut off anything.

There are also good news: some heuristics, separation routines, node selection and branching rules do not use LP information. Thus they will be utilized in our solving process. As we use the framework SCIP, we may only apply the routines provided by SCIP. It will be one future task to generalize the routines for SDPs.

For example different rounding strategies also work for SDPs, as they normally do not use LP information. We describe one of these heuristics in the following. Given a solution this heuristic rounds the values that are supposed to be binary but are continuous up or down. The following algorithm describes the process in detail:

Algorithm 3.3.1. *Simple rounding algorithm for mixed-binary semidefinite programs.*
Input: *A relaxation solution $\hat{y} \in \mathbb{R}^n$, information about the binary variables*

```
1. Initialize $k = 1$.
2. If $\hat{y}$ is not binary but is supposed to be if $\hat{y} > 0.5$ set $\hat{y} := 1$ otherwise $\hat{y} := 0$.
3. If $k < n$ set $k := k + 1$ and go to step 2.
4. Check the solution $\hat{y}$ for feasibility and stop.
```

This algorithm is helps finding feasible solutions as well as reducing the difficulties of the SDP solver within the branch-and-bound algorithm. As presented in the previous section interior-point-methods have an issue with problems with a small set of interior points. Using this heuristic we do not need to start the SDP solver for problems with many fixed variables because this heuristic is very good in finding feasible solution when a lot of variables are fixed. So we do not save numerous solving calls, but we do not need to start those that might be problematic for the solver.

The implementation of this heuristic is done by Denis Aßmann. He also had the idea for a non-deterministic version of Algorithm 3.3.1, where up- or down-rounding is randomly chosen. We will elaborate on the impact of this simple heuristic in Chapter 6.

3.3.2. Presolve

One very efficient tool in the MIP context is presolving. There exist various different techniques to get some additional information out of the problem data. This information is used to make the solving process more efficient, for example if there is a variable $x_i \in [0,1]$ which is declared as integer. Then for sure this variable can be declared as binary. This is a very obvious fact for a human but it is important for the solver to recognize this. Sometimes even new constraints can be created during presolving.

As the branch-and-bound framework does not understand a semidefinite constraint, it cannot extract any useful information out of it. To speed up the solving process, we need to convert information encoded in the semidefinite constraint to linear or quadratic information, as this is what the framework can work with. This happens in the presolving routines.

All the ideas are based on the facts we know from Chapter 2, some of them we have already seen in Section 3.2.2. The problem we are looking at is still (MISDP):

(1) Using the fact that diagonal entries of semidefinite matrices need to be nonzero, we construct the following linear inequalities, where we denote by a^i_{jj} the diagonal-entry of row j of the i-th coefficient matrix:

$$\sum_{i=1}^{n} a^i_{jj} y_i - a^0_{jj} \geq 0 \qquad \text{for } j = 1, \ldots, m$$

(2) Using the fact that principal minors of a semidefinite matrix must also be positive semidefinite, we construct the following quadratic inequalities for $j, k = 1, \ldots, n$:

$$\left(\sum_{i=1}^{n} a^i_{jj} y_i - a^0_{jj}\right)\left(\sum_{i=1}^{n} a^i_{kk} y_i - a^0_{kk}\right) - \left(\sum_{i=1}^{n} a^i_{jk} y_i - a^0_{jk}\right)^2 \geq 0 \qquad \text{for } j \neq k$$

(3) If there are rows in the matrix which are independent of all variables, we can eliminate these constant rows as described in item (2) of Section 3.2.2 using the Schur complement.

(4) We search for a block structure in the matrix and generate an own semidefinite constraint for each block, then the original semidefinite constraint can be removed. This is an important idea because it reduces the size of the SDP blocks and therefore has a direct influence on the solving time.

(5) A special case of item (4) is a block of size one. This block can be transformed into a linear constraint. Let k be the index of the constant block in the constraint matrix, then

$$\sum_{i=1}^{n} a^i_{kk} y_i - a^0_{kk} \geq 0.$$

(6) Variables that were already fixed by other presolve routines are eliminated using the idea of item (1) of Section (3.2.2).

(7) Using Proposition 2.1.8 we can also construct linear inequalities using the constant matrix A_0. Let B be a positive semidefinite matrix, $j \neq k$ and $b_{jk} \neq 0$, then we know that the corresponding diagonal entries b_{jj} and b_{kk} cannot be zero. We can use this for our problem and formulate linear inequalities for every nonzero entry a^0_{jk} in the constant matrix A_0:

$$\sum_{i=1}^{n} a^i_{jj} y_i \geq 0 \qquad \text{and} \qquad \sum_{i=1}^{n} a^i_{kk} y_i \geq 0.$$

This idea can also be employed to detect infeasible variable fixings within the branch-and-bound framework.

Note that most of these inequalities cannot replace the semidefinite constraint because they do not include all the information but they can give a hint for the framework, which strategy might be good. It turn out that (1) is a good idea and speeds up the solving process, whereas (2) is not a good idea and slows down the solving process. This might be because (1) adds m linear constraints and (2) adds $\binom{n}{2} = \frac{n(n-1)}{2}$ quadratic constraints.

The presolving ideas (3), (4), (5), and (6) are some kind of data cleaning and really helpful because they reduce the problem size and therefore speed up the solving process. As (4) is already available in some of the SDP solvers it is not implemented yet and therefore one future task.

Typically the constant matrix A_0 is sparse, therefore using (7) does not lead to many new constraints. This idea produces helpful information out of the semidefinite constraint and thus it should be used.

One extension of this block splitting method is the following. A matrix $B \in \mathbb{R}^{n \times n}$ can always be regarded as an incidence matrix of a graph with n vertices. For every diagonal entry b_{ii} of the matrix B there exists a vertex in the graph. The off-diagonal entries b_{ij} correspond with the edges connecting vertex i with vertex j. In this graph we can look for articulation nodes, i.e., nodes for which the graph would decompose into several connected components if we deleted this node. Graphs with an articulation node i have the following structure in their incidence matrix:

$$B = \begin{pmatrix} * & * & 0 & 0 \\ * & b_{ii} & * & * \\ 0 & * & * & * \\ 0 & * & * & * \end{pmatrix}.$$

Roughly speaking two blocks of this matrix overlap each other only by one row and column. We want to get rid of this overlapping and being able to decompose the matrix into two blocks. Therefore we duplicate the node in the graph and then look at the two new different components. In terms of the incidence matrix what we do looks as follows:

$$\tilde{B} = \begin{pmatrix} * & * & 0 & 0 & 0 \\ * & \frac{b_{ii}}{2} & 0 & 0 & 0 \\ 0 & 0 & \frac{b_{ii}}{2} & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & * & * & * \end{pmatrix}.$$

We will now prove that this decomposition is valid.

Theorem 3.3.2. *Let B be an arbitrary matrix defined as above and let $\tilde{B}$ be the decomposition of this matrix as stated above. Then B is positive semidefinite if $\tilde{B}$ is.*

Proof. If $\tilde{B}$ is positive semidefinite, then by Observation 2.1.9 both blocks of $\tilde{B}$ are positive semidefinite as well. Additionally, using Proposition 2.1.6 all principal minors are positive semidefinite. Obviously every minor of $\tilde{B}$ without the entry $\frac{b_{ii}}{2}$, i.e., $\tilde{B}_K$ for all $K \subset I$ with $i \notin K$, is also a minor of B.

Now look at the minor of $\tilde{B}$ with exactly one entry $\frac{b_{ii}}{2}$, i.e., $\tilde{B}_K$ for all $K \subset I$ with $i \in K$ or $i+1 \in K$ but not both. Without loss of generality these minors can be rearranged to have the following form:

$$\tilde{C} := \begin{pmatrix} * & * \\ * & \frac{b_{ii}}{2} \end{pmatrix}.$$

Let

$$C := \begin{pmatrix} * & * \\ * & b_{ii} \end{pmatrix}$$

be the corresponding minor of the matrix B, and let m be the dimension of the matrices C and $\tilde{C}$, then $m \leq n$. As $\tilde{C}$ is positive semidefinite we have for all $v \in \mathbb{R}^m$:

$$\begin{aligned} 0 &\leq v^T \tilde{C} v \\ &= \sum_{k=1,k\neq i}^{m} c_{kk} v_k^2 + \sum_{k=1}^{m} \sum_{1\leq j<k} 2c_{kj} v_k v_j + \frac{b_{ii}}{2} v_i^2 \\ &\leq \sum_{k=1,k\neq i}^{m} c_{kk} v_k^2 + \sum_{k=1}^{m} \sum_{1\leq j<k} 2c_{kj} v_k v_j + b_{ii} v_i^2 \\ &= v^T C v. \end{aligned}$$

Therefore $v^T C v \geq 0$ and C is positive semidefinite.

The only thing left to show is that the above is true for minors of $\tilde{B}$ with two entries $\frac{b_{ii}}{2}$, i.e., $\tilde{B}_K$ for all $K \subset I$ with $i \in K$ and $i+1 \in K$. All these minors have the following form:

$$\tilde{D} = \begin{pmatrix} * & * & 0 & 0 \\ * & \frac{b_{ii}}{2} & 0 & 0 \\ 0 & 0 & \frac{b_{ii}}{2} & * \\ 0 & 0 & * & * \end{pmatrix}.$$

Again let

$$D = \begin{pmatrix} * & * & 0 \\ * & b_{ii} & * \\ 0 & * & * \end{pmatrix}$$

be the corresponding minor of matrix B, and let l be the dimension of the matrices D and $\tilde{D}$, then $m \le l \le n$.

As $\tilde{D}$ is positive semidefinite we have for all $v \in \mathbb{R}^l$:

$$\begin{aligned} 0 &\le v^T \tilde{D} v \\ &= \sum_{k=1, k \ne i, k \ne i+1}^{l+1} d_{kk} v_k^2 + \sum_{k=1}^{l+1} \sum_{1 \le j < k} 2 d_{kj} v_k v_j + \frac{b_{ii}}{2} v_i^2 + \frac{b_{ii}}{2} v_{i+1}^2 . \end{aligned}$$

This is in particular true if $v_i = v_{i+1}$, so

$$\frac{b_{ii}}{2} v_i^2 + \frac{b_{ii}}{2} v_{i+1}^2 = \frac{b_{ii}}{2} v_i^2 + \frac{b_{ii}}{2} v_i^2 = b_{ii} v_i^2 .$$

This implies that

$$0 \le \sum_{k=1, k \ne i}^{l} d_{kk} v_k^2 + \sum_{k=1}^{l} \sum_{1 \le j < k} 2 d_{kj} v_k v_j + b_{ii} v_i^2 = v^T D v .$$

Therefore $v^T D v \ge 0$ and D is positive semidefinite. This implies that B is positive semidefinite and completes the proof. □

This idea can be used to reduce the block size of the semidefinite constraint matrix, but it is also has its costs. Articulation nodes must be found and the whole data needs to be reorganized, but the advantages of smaller blocks prevail. Moreover, the number of semidefinite constraints increases. Additionally, there is no equivalence between the two formulations. Here is an example for which B is positive semidefinite, but $\tilde{B}$ is not.

Example 3.3.3. *Let B be the following positive semidefinite matrix*

$$B = \begin{pmatrix} 1 & -2 & 0 \\ -2 & 6 & 1 \\ 0 & 1 & 2 \end{pmatrix}$$

with approximate eigenvalues 6.88448370*,* 0.23126569*, and* 1.88425060*. For the matrix*

$$\tilde{B} = \begin{pmatrix} 1 & -2 & 0 & 0 \\ -2 & 3 & 0 & 0 \\ 0 & 0 & 3 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix}$$

the first block $\begin{pmatrix} 1 & -2 \\ -2 & 3 \end{pmatrix}$ *is not positive semidefinite (the eigenvalues are* -0.23606798 *and* 4.23606798*) and therefore $\tilde{B}$ is not positive semidefinite.*

This is why we need to be very careful because it might happen that there is no solution for the decomposition matrix $\tilde{B}$, but this does not imply that the problem formulated in terms of B is infeasible. Because of this difficulty this decomposition idea is not yet implemented for our branch-and-bound code.

Similar ideas can be found in [FKMN01] and [NFF$^+$03]. In these two papers the authors use more advanced graph-theory ideas and use the block splitting for solving continuous SDPs efficiently. They implemented their ideas within the code SDPA [FFK$^+$08] and could speed up the solving process of SDPs.

As there is no advantage for the branch-and-bound framework of have two instead of one semidefinite constraint, the block-splitting should be performed within the SDP solver. This would speed up the solving of one relaxation and so the whole solving process could be done in less time. We will comment on this in Chapter 8.

3.3.3. Detecting infeasibility

As mentioned before infeasible subproblems occur very often within the branch-and-bound algorithm and cause trouble for some of the solvers. Hence it would be preferable to have a tool that detects infeasibility without explicitly solving an SDP. This tool combines some of the ideas presented above. We use Proposition 2.1.8, property (7) of Section 3.3.2, and item (1) of Section 3.2.2.

This idea is used for every node in the branch-and-bound tree just before the SDP solving starts. So let us assume we are somewhere in the tree and the subproblem we want to solve has variables y_i fixed to some values c_i.

Let I be the index set of all variables. By $F \subset I$ we denote the set of indices of the fixed variables and $N \subset I$ with $F \cap N = \emptyset$ is the set of indices of the nonfixed variables. Then using item (1) of Section 3.2.2 we can rewrite the semidefinite constraint:

$$\sum_{i \in I} A_i y_i - A_0 = \sum_{i \in N} A_i y_i + \sum_{i \in F} A_i c_i - A_0 \succeq 0.$$

Additionally, we can now use Proposition 2.1.8 and item (7) of Section 3.3.2 to detect infeasibility of this variable fixing.

Therefore we consider the new constant matrix

$$\tilde{A}_0 := \sum_{i \in F} A_i c_i - A_0$$

and its entries. If there is a nonzero entry $\tilde{a}^0_{jk}$ then the corresponding diagonal entries for the rows j and k in $\sum_{i \in N} A_i y_i$ has to be nonzero. If this is not possible, the variable fixing is invalid and we can cut off the node and the whole subtree.

3.4 Approximation using eigenvalue cuts

Trying to solve an SDP in every node of a branch-and-bound tree is very time consuming. We presented various ideas to make this process faster but still there is an SDP to solve in almost every branching node. Solving SDPs is expensive, we may ask if there is another way to describe the feasible set where SDPs are not needed so that the corresponding models can be solved with a standard MIP solver? The answer to this question is yes, an approximation of the SDP cone exists but it is not efficient. In this section we describe a method for approximating the SDP cone that uses eigenvectors to perform a cutting plane approach.

The idea presented in this chapter is a cutting plane approach that tries to approximate the semidefinite cone using linear inequalities. These inequalities are iteratively constructed using the eigenvectors of a solution.

We solve an LP consisting of all the linear constraints and variables of the MISDP to get a starting solution $\bar{x}$. In most of the cases this solution is not positive semidefinite. Therefore we want to find a linear inequality that cuts off this solution. If the solution $\bar{x}$ is not feasible this means that there exists a negative eigenvalue with eigenvector $\bar{v}$, i.e.,

$$\bar{v}^T(\sum_i A_i\bar{x}_i - A_0)\bar{v} < 0$$

for this solution $\bar{x}$. Since positive semidefinite implies that

$$v^T(\sum_i A_i x_i - A_0)v \geq 0$$

for all v, we can find the following cutting plane:

$$\begin{aligned} &\bar{v}^T(\sum_i A_i x_i - A_0)\bar{v} \geq 0 \\ \iff &\sum_i \bar{v}^T A_i \bar{v} x_i - \bar{v}^T A_0 \bar{v} \geq 0. \end{aligned}$$

This inequality is a linear inequality:

$$a_i x_i - a_0 \geq 0,$$

where $a_i := \bar{v}^T A_i \bar{v}$ and $a_0 := \bar{v}^T A_0 \bar{v}$ and it cuts off the current solution. Now we solve the LP again and find a new solution which can also be separated unless it is feasible. We describe this procedure in the following algorithm.

Algorithm 3.4.1. *Cutting plane approximation scheme for MISDPs.*
Input: *SDP constraint data A_i for $i = 1, \dots, n$ and start solution x^0*

1. *Initialize $k = 1$, $x^k := x^0$.*
2. *Calculate the eigenvector $\bar{v}$ corresponding to the smallest eigenvalue of $\sum_i A_i x_i^k - A_0$.*
3. *Generate a cut that separates the solution x^k: $\sum_i \bar{v}^T A_i \bar{v} x_i - \bar{v}^T A_0 \bar{v} \geq 0$.*
4. *Add this cut to the LP formulation.*
5. *Solve the LP using the simplex method.*
6. *Check if the new solution $\hat{x}$ is feasible for the SDP constraint, then* ***stop****, otherwise set $k = k + 1$ and $x^k := \hat{x}$ and go to step 2.*

If the LP solution does not change very much in two different iterations, we interrupt this procedure, start branching and add standard LP cuts. In every node this procedure is resumed. Our test-runs showed that this approximation scheme helps to find feasible solutions quickly.

Additionally, for small examples it was able to solve them in just a fraction of the time the SDP branch-and-bound algorithm needed. For bigger examples the method made a lot of progress in reducing the gap in the beginning but in the end the progress was very slow. So we decided to use this method together with the SDP branch-and-bound algorithm as a hybrid approach. A detailed analysis of the behavior is given in Chapters 6 and 7.

The idea of eigenvector cuts and different sparsification techniques are also presented in [QBM12]. One sparsification algorithm calculates $w_{\max} := v^T A v$ for the eigenvector v corresponding to the smallest eigenvalue, then it tries to set different entries of the eigenvectors equal to zero.

If the corresponding product in terms of the new sparser eigenvector v_S is still negative and $v_S^T A v_S < w_{\max} p$, with a manually chosen parameter $p \in [0, 1]$, this new v_S is taken instead of v. Additionally, the number of nonzero entries in v_S can be controlled.

We also tried to use this technique for the problem instances we discuss in Chapter 5 and found out that for Truss Topology Design these techniques are quite useless. This technique will only work for SDP problems with sparse coefficient matrices A_i. We will show this in a small example.

Example 3.4.2. *Look at the following semidefinite constraint:*

$$\begin{pmatrix} 3 & 0 & 2 \\ 0 & 0 & 0 \\ 2 & 0 & 4 \end{pmatrix} x_1 + \begin{pmatrix} 2 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 2 \end{pmatrix} x_2 + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} x_3 \succeq 0$$

or equivalently

$$C = \begin{pmatrix} 3x_1 + 2x_2 & x_2 & 2x_1 \\ x_2 & 4x_2 & x_2 \\ 2x_1 & x_2 & 4x_1 + 2x_2 + x_3 \end{pmatrix} \succeq 0.$$

One of the cuts presented above for this matrix looks as follows:

$$
\begin{aligned}
v^T C v &= v^T \begin{pmatrix} (3x_1 + 2x_2)v_1 + x_2 v_2 + 2x_1 v_3 \\ x_2 v_1 + 4x_2 v_2 + x_2 v_3 \\ 2x_1 v_1 + x_2 v_2 + (4x_1 + 2x_2 + x_3)v_3 \end{pmatrix} \\
&= (3x_1 + 2x_2)v_1^2 + 4x_2 v_2^2 + (4x_1 + 2x_2 + x_3)v_3^2 + 2x_2 v_1 v_2 + 4x_1 v_1 v_3 + 2x_2 v_2 v_3 \geq 0.
\end{aligned}
$$

As some of the variables have entries in more than one row, using a sparser vector v_S would not necessarily eliminate one of the variables from the cut. For example if $v_1 = 0$, then x_1, x_2 and x_2 remain in the cut, the same is true for $v_2 = 0$. For $v_3 = 0$ the variable x_3 can be eliminated from the cut. This shows that a sparsification of the cut is not always a good idea. For examples with sparser coefficient matrices A_i this idea would be helpful.

CHAPTER 4

Implementation details

After a detailed description of the ideas and algorithms for solving MISDPs in Chapter 3 we now take a closer look at the implementation. Within this chapter we present different components of our software package for solving MISDPs [MS12] and describe their interaction with the general MIP solver SCIP (Solving Constraint Integer Programs). The whole package is implemented using SCIP (see [Ach09] and [SCI12]) as a branch-and-bound framework. The ideas also work for other frameworks, so for example we also implemented a very basic MISDP solver in the framework Acro-PEBBL [EPH01]. This solver is rather rudimentary due to the fact that there are no heuristics or separators available in Acro-PEBBL, and also the branching decisions and the node selection rules are not as advanced as they are in SCIP.

The programming language used for our software is C++, this is an object-oriented programming language. We assume that the reader is familiar with basics of such languages, for an introduction we refer to [Str97]. In our code we provide a number of different data structures and a general interface to SDP solvers. For the SDP solver DSDP [BY08] we already implemented the necessary functions and this is the solver we use for our computational results in Chapters 6 and 7. Including other solvers will be a future task.

In the first section of this chapter we describe the new extended SDPA data format for MISDPs and how the file reader works. In Section 4.2 we take a closer look at the different data structures. These data structures are a very important part of our software and they are necessary for the implementation of some of the ideas presented in Section 3.2. Another important part is the interface to SDP solvers. This general interface gives the opportunity to use other SDP solvers for solving the relaxations in every node. We present this interface in Section 4.3.

As already stated in Section 3.2 we need a handling procedure that is able to understand SDP constraints and can check them for feasibility. Therefore we provide an SDP constraint handler, this one is described in Section 4.4. The core of our package is of course the part that solves the branching nodes. In SCIP this is called a relaxator and we present our implementation for SDPs in Section 4.5. Additionally, we want to comment on different parameter settings in Section 4.6. There are various differences to default linear MIP settings and therefore we need to be careful with the different parameters available in SCIP.

The software is joint work with Lars Schewe [MS12], Jakob Schelbert, and Denis Aßmann. Jakob Schelbert implemented most of the parts of the SDPA reader (see Section 4.1). The rounding heuristic for SDPs was implemented by Denis Aßmann (see Section 4.5). Some implementation for the data structures (see Section 4.2), the memory management, and the presolve routines was done by Lars Schewe. Finally, there were intensive discussions with Lars Schewe of how to implement the different methods.

The software is available through the following website:

`www.opt.tu-darmstadt.de/∼smars/scip_sdp.html`

As this SDP package is implemented as a plugin for the branch-and-bound framework SCIP, it needs SCIP to run. The SCIP version should be 3.0 or higher because some of the features for relaxation solutions were not implemented in older SCIP versions.

4.1 Extended SDPA format

The structure of semidefinite programs is quite similar to those of linear programs. Linear programs are stored in rows and columns, representing the inequalities and the variables, and in every row each column has a real-valued coefficient. In the branch-and-bound framework SCIP, which is the one we are using, there are even more details stored within a linear constraint, a row or a column. Additionally, SCIP needs lower and upper bounds for variables and linear constraints. Therefore if we have a linear inequality of the form

$$a^T x - b \geq 0$$

we would give the following linear constraint to SCIP:

$$0 \leq a^T x - b \leq \infty.$$

For semidefinite programs and semidefinite constraints we do not need an upper bound, but for each column we need a coefficient matrix, not only a real number. Therefore this similarity does not suffice to use the same file format for both kinds of programs.

Most of the SDP solvers understand the so called SDPA format, named after the solver SDPA [FFK$^+$08]. This is a sparse format to store SDPs and it uses the suffix `dat-s`. A format for storing MISDPs does not exist so far, therefore we extend the SDPA format to be able to store mixed-integer SDPs, we describe this extension in the following, but first we give a short introduction to the standard SDPA format.

As already mentioned in the previous chapter a semidefinite constraint can be split into smaller semidefinite constraints using block diagonal structures. Additionally, linear constraints can also be represented as semidefinite constraint with diagonal entries only (see Remark 2.3.3). Unfortunately standard SDP solvers are not able to detect this structure on their own. Therefore the different blocks must be labeled within the data-file and this is exactly what the SDPA format provides.

The problem decomposed into m diagonal blocks and one LP block in the $m+1$-th matrix is of the following form:

$$\begin{aligned} \min \quad & b^T y \\ \text{s.t.} \quad & \sum_{i=1}^{n} D_{ij} y_i - D_{0j} \succeq 0 \qquad \text{for } j = 1, \ldots, m \\ & \sum_{i=1}^{n} d_{ir}^{m+1} y_i - d_{0r}^{m+1} \geq 0 \qquad \text{for } r = 1, \ldots, p \\ & y \in \{0,1\}^n. \end{aligned}$$

This problem has n variables, $m+1$ different blocks, p linear inequalities and the objective coefficients are $b_1, \ldots, b_n$. Let s_j $(j = 1, \ldots, m)$ be the sizes of the different blocks. This information is now put into the first four lines of an SDPA file:

```
n
m+1
s_1 ... s_m −p
b_1 ... b_n
⋮
```

where $-p$ is the marker for indicating the linear block. Of course its size is p rather than $-p$.

The following lines describe the nonzero matrix entries in the different blocks. We assume that all matrices appearing in SDPs are symmetric, therefore it suffices to provide one half of the matrix, the upper or the lower triangular matrix. Each nonzero has its own line of the following format:

```
i j k l d_kl
```

where i is the number of the variable the matrix block belongs to. The number of the block is j and k, l are the positions within the matrix where the entry d_{kl} is located. Note that for the constant terms we have $i = 0$. Comments in the SDPA format need a separate line that must begin with the symbol `*`.

For our extension we use the fact that `*` indicates a comment line. For the information which variable is integer we add lines which are treated as comments in the standard SDPA format at the end of the file in the following way:

```
*INTEGER*
*1
*2
*3
```

First we add the key word `*INTEGER*`, then the index of each integer variable is put into a separate line, again beginning with `*`. For modeling binary variables, we can use integer variables with lower bound equal to zero and upper bound equal to one. This extension has the advantage that the file is still readable for SDP solvers, so they are able to solve the relaxation of the problem from the same file as the SDP package. We demonstrate the format in the following example:

Example 4.1.1. *We consider the following problem with three variables, two of them are binary, two SDP blocks and three linear inequalities.*

$$\begin{aligned}
\min\quad & y_1 + y_2 + y_3 \\
\text{s.t.}\quad & \begin{pmatrix} 1 & 0.25 & 0 \\ 0.25 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} y_1 + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0.5 & 2 \\ 0 & 2 & 0.5 \end{pmatrix} y_2 + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} y_3 \succeq 0 \\
& \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} y_1 + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 3 \\ 0 & 3 & 1 \end{pmatrix} y_2 + \begin{pmatrix} 1 & 0.2 & 0 \\ 0.2 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} y_3 \succeq 0 \\
& -y_1 - y_2 \geq -1 \\
& y_1 + y_3 \geq 1 \\
& y_3 \geq 0 \\
& y_1, y_2 \in \{0, 1\}.
\end{aligned}$$

The extended SDPA file would look as follows:

```
3
3
3 3 −7
1 1 1
0 3 1 1 −1
0 3 2 2 1
0 3 6 6 −1
0 3 7 7 −1
1 1 1 1 1
1 1 1 2 0.25
1 1 2 2 1
1 2 1 1 1
1 3 1 1 −1
```

```
1 3 2 2 1
1 3 3 3 1
1 3 6 6 −1
2 1 2 2 0.5
2 1 2 3 2
2 1 3 3 0.5
2 2 2 2 1
2 2 2 3 3
2 2 3 3 1
2 3 1 1 −1
2 3 4 4 1
2 3 7 7 −1
3 1 3 3 1
3 2 1 1 1
3 2 1 2 0.2
3 2 2 2 1
3 3 2 2 1
3 3 5 5 1
*INTEGER*
*1
*2
```

For using this data format we provide a file reader for those `dat-s` files. This reader is implemented using the callbacks and methods SCIP provides for file readers. First of all it checks the data for consistency. It does not matter whether the upper or the lower triangle part of a symmetric matrix is given because the reader is able to handle both. Additionally, the reader stores the SDP constraints in some special data structure. We introduce this structure in Section 4.2. Using this data structure it generates an SDP constraint for each of the blocks, except the linear block. The linear constraints are handled differently. They are given to SCIP as standard linear constraints, SCIP saves them and takes care of them. Finally, the objective function and the variables are generated for SCIP.

Note that the file reader exactly saves the data, as it is given in the SDPA file. It does not detect hidden block-structure or LP rows within the SDP blocks. The whole data handling is done by the presolve routines of the constraint handler (see Section 4.4) and the data structure we use (see Section 4.2).

The reader is also able to write SDPA files from SCIP data. This is a very helpful tool, for example it can be used to write difficult subproblems, occurring within the branch-and-bound tree, into files. It allows the user to try to figure out what makes them difficult or solve them with a different SDP solver. This is very helpful for detecting and understanding the problems we presented in Section 3.2.

4.2 The data structures

As already mentioned SDP constraints need other data structures than rows and columns because they have matrices as variable coefficients. Furthermore, we explained in detail in Section 3.2.2 what kind of problems might occur during the branch-and-bound algorithm and which data handling is needed.

One thing we did not mention so far is a small communication problem between the branch-and-bound framework and the SDP solver. In the branch-and-bound framework every variable has a name and if this name does not occur in any constraint or the objective function, this variable will be ignored or deleted. For the SDP solver the variables do not have names, they are numbered consecutively and if one number is missing the SDP solver will get into serious trouble. So we need something that keeps account on the variables and their indices.

For being able to handle these problems, we implemented three different data structures, each has its own class. The first class, called `varmapper`, takes care of the indices, the second one is called `problemdata` and brings together the SDP and the LP data and finally there is a class called `sdpcone`, this one does the SDP data handling. Without any of these classes the whole solving process would fail. We introduce the three classes in detail in the following.

Variable indices

Due to the fact that branch-and-bound frameworks do presolving and reorder the variables during this process, we need to take care, which variable of the branch-and-bound framework is a specific variable for the SDP solver. This becomes even more important when branching starts and some of the variables are fixed to specific values. Then we apply the elimination technique presented in (1) of Section 3.2.2.

This fixed variable does not exist for the SDP solver and therefore its index is assigned to some other variable. This whole work is done in a specific class: the `varmapper` class. It maps a variable to a specific index. This is done in every node of the branch-and-bound tree anew. Additionally, this class counts the number of fixed variables and the number of fixed integer variables. As mentioned in Section 3.2.2 this information is needed if we do not want to get into trouble with the SDP solver.

SDP data and its cone

We store the data of each semidefinite constraint in one distinct instance of the class `sdpcone`. This is a very powerful class with many important tools. The data is stored in different arrays. Note that we think of the constant matrix as coefficient matrix of the non-existing variable zero. In our data structures the data of the constant matrix is saved in its own arrays.

The `sdpcone` class does most of the data handling presented in Section 3.2.2.

(1) First of all it knows the whole data of a semidefinite constraint and has methods for returning
   - the block size,
   - the number of variables,
   - the number of nonzeros in the constant matrix A_0 and
   - the maximum entry of the constant matrix.

(2) Additionally, it can return
   - the constant matrix A_0,
   - the coefficient matrix A_i for a specific variable y_i and
   - the coefficient matrix A_i for a specific variable y_i without the zero rows and columns.

(3) There also exists a method for calculating the product of the coefficient matrix A_i for a specific variable y_i with a vector. This method is needed for the approximation scheme described in Section 3.4.

(4) Moreover, we implemented a method that computes the following matrix for a specific solution $\bar{y}$:

$$A = \sum_{i=1}^{n} A_i \bar{y}_i - A_0.$$

This method is called each time a solution is checked for positive semidefiniteness because this is exactly the matrix that needs to be positive semidefinite.

(5) Internally the `sdpcone` stores the data in a compressed format, this is done once in the beginning of the whole solving process. Therefore methods are needed that return the uncompressed information.

(6) After SCIP finished presolving, variables might be fixed, negated or aggregated. These different states are treated in a special method, which eliminates fixed variables. Negated variables get a negative coefficient and aggregated variables are replaced by the appropriate other variables.

(7) The SDP constraint data is only accessible through two different iterators, one for the matrices A_i and one for the constant matrix A_0. These iterators take care of fixed variables and empty rows as described in Section 3.2. Using these two iterators we can walk through the data and only get the adjusted values (where the fixed variables and empty rows are eliminated and deleted).

(8) Furthermore, there are methods for sorting the data using bucket sort (see [CLRS01]) and some SCIP specific methods.

Especially (7) is very important because this is the only method for returning the matrix entries. Additionally, it solves many difficulties we discussed in Section 3.2.2. The convenient property using these iterators is that the data cleaning is done automatically without additionally calling a method. If we need the data, we can only get the cleaned version of it.

Subproblems in the branch-and-bound algorithm

The class `problemdata` takes care of the data of the subproblems. In every node of the branch-and-bound tree we gather all `sdpcones` and all linear constraints in an object of the class `problemdata`. Additionally, this class takes care of the variable bounds.

The `sdpcones` are stored completely, the LP data in contrast is converted to some SDP-specific structure. With the LP data the same techniques must be applied as described for the `sdpcone`. Linear constraints in which all variables are fixed to a specific value cannot be added because fixed variables are not allowed in linear constraints. So if a variable y_k is fixed to a value c we need to eliminate y_k from the linear constraint. Therefore we use the same technique as described for SDP constraints:

$$\sum_{i=1, i\neq k}^{n} a_i y_i - a_0 + a_k c \geq 0.$$

Then $\tilde{a_0} := a_0 - a_k c$ becomes the new constant term.

The LP data is stored in three arrays: one for the coefficient values a_i and a_0, one for the variable number i the coefficient belongs to and one for the number of the constraint in which this pair of variable and coefficient occurs. This is one point where we need the class `varmapper` because the variable number i already needs to be converted for the SDP solver. Again we treat the constant terms as coefficients for variable number zero.

Note that we have to be careful regarding the bounds of the constraints. We already mentioned at the beginning of this section that SCIP stores the linear constraints with upper and lower bounds in the following form:

$$l_j \leq \sum_{i=1}^{n} a_{ij} y_i \leq u_j \qquad \text{for } j = 1, \ldots, p.$$

Normally when modeling a problem as a semidefinite program we only set one of these bounds (the lower bound) to a specific value, the upper bound is infinity. We cannot use an infinity-bound in the SDP solver due to numerical troubles. Therefore we only hand over the inequality where no infinity occurs. However, during the branch-and-bound algorithm these bounds can change to something different from infinity. Therefore we have to add one constraint for every bound $l_i \neq -\infty$ and $u_i \neq \infty$.

For the constraints of type $\sum_{i=1}^{n} a_{ij} y_i \leq u_j$, we additionally need to switch the signs and transform the constraint into SDP standard form:

$$\sum_{i=1}^{n} -a_{ij} y_i \geq -u_j.$$

As we use SCIP as our branch-and-bound framework there is a bunch of methods used for tightening the bounds of a variable. Thus for continuous and integer variables upper and lower bound can change in every node, due to fixings or a new global lower bound. Tightening the bounds helps the solving process a lot and therefore it is important to additionally use these variable bounds for computing our relaxations. Just as described in the SDPA format we add these bounds as linear constraints to our SDP solver.

4.3 The SDP solver interface

We provide a general interface for SDP solvers, so users of our software package may use a different SDP solver if they implement the necessary callback methods.

First of all the data needs to be converted from the `sdpcone` which is saved in the `problemdata` to the specific structure the solver needs. Within this transformation again the `varmapper` class is needed. Additionally, all the linear constraints of `problemdata` need to be converted, depending on the solver we are using. Then the data should be given to the solver and everything should be prepared for solving. Depending on the solver there are different create and setup methods.

If data is passed to the solver, solving can start. Then the solving parameters can be modified and an initial solution can be given to the solver. The callback function in which all this happens is called `sdp_solve`. After solving is complete this callback function should read the status and the solution type and the solution. We explain the different types in Section 4.5. Furthermore, the solution and the objective value should be handed over to SCIP.

Additionally, there is one more callback for handling difficult nodes. This function uses a penalty formulation of the problem and tries to determine whether there exists a feasible point in the current subproblem or not, as described in Section 3.2.3. If the subproblem is feasible this function should hand over the lower bound of the parent node to SCIP, as no real new bound is available.

We mentioned in Section 3.2.3 that it is also possible to check the feasibility of a subproblem before solving it. If the solver needs such a check, then the function for checking feasibility should be a part of the data conversion. Solving the whole problem should only start if the feasibility check returns feasible.

4.4 Handling SDP constraints

Understanding SDP constraints is essential for solving MISDPs. Therefore we implemented an SDP constraint handler that uses the standard callback methods for constraint handlers in SCIP and takes care of all semidefinite constraints. We have one SDP constraint for each semidefinite block in the problem formulation and each of these SDP constraints has its own `sdpcone`. Within this `sdpcone` the data for the constraint is saved and this `sdpcone` is available through a special get-method.

Additionally, there are some important callbacks for

- the initialization of the solving process,
- presolving the constraint,
- checking the constraint for feasibility,
- separating a solution and
- enforcing a solution.

Initialization, checking and enforcing are necessary for the correctness of the solving process, presolving and separation are optional. We explain the different methods in the following:

(1) Initialization: First of all we need to transform all the variables appearing in the constraint. This is SCIP-specific because SCIP does not work using the original problem, it only uses a transformed or copied problem for the solving process. Additionally, we need to prevent SCIP from multi-aggregating variables because we do not know how to handle them within an SDP context.

(2) Presolving: All the ideas described in Section 3.3.2 are implemented in this callback. This is where we try to get as much linear information out of the SDP as possible. Presolving is also where the block splitting should be performed in future. As already mentioned this block splitting is not implemented yet. Linear blocks can already be detected.

(3) Checking: For being able to deal with SDP constraints the most important thing is to know whether the constraint is satisfied by a given solution or not. We need to do that many times: for example after each solving and each call of a heuristic we need to check feasibility, so this must be implemented very efficiently. Checking positive semidefiniteness is not as easy as showing that a linear constraint is satisfied. We always need to compute the eigenvalues and check their sign. Therefore we implemented a method that only computes the smallest eigenvalue and checks this one.

For the computation we use some of the methods described in Section 4.2 and the Lapack-function `DSYEVR` (see [DMSR04], [DP04a], [DP04b], [Dhi97]). This function computes the i-th eigenvalue of a symmetric matrix. For determining if the matrix is positive semidefinite we do not just compare this computed eigenvalue to zero, we also look at the DIMACS error-norms [Mit03]. This is a standard tool when working with SDPs. As we only want to check feasibility of the dual SDP we use the following error norm:

$$\text{err} = \max\left\{0, \frac{-\lambda_{\min,k}(y)}{1 + \|A_0\|_1}\right\},$$

where $\|A_0\|_1$ denotes the maximum of the absolute values of the entries of the matrix A_0, which is the constant matrix or the right-hand-side of the problem. If err is positive the matrix is not positive semidefinite. Using this norm helps to avoid numerical problems because the minimum eigenvalue is related to the maximum entry of the constant matrix.

(4) Separation and enforcing: Here we try to add cutting planes, separating an infeasible solution. The cutting planes we use are generated using the eigenvector of the smallest eigenvalue as described in Section 3.4. We can also try to make these cuts sparser as described in the end of Section 3.4.

4.5 Solving nodes in the branch-and-bound tree

Combining all the methods and classes described above we are able to solve the SDP relaxations in the branch-and-bound nodes. Using `problemdata`, `varmapper`, and SDP interface we put all the data into the SDP solver and solve the subproblem. As mentioned before when using an interior-point-method we have to take care of the integrity of the data. The different data structures provide capabilities for doing so. We have to do the variable-index-mapping and the setting up of the problem data in each node.

Because of all the different problems that might occur for SDPs within a branch-and-bound algorithm as described in Section 3.2, the solving procedure is not just solving. We have to take care of different cases, which are described below. Additionally, the problem size decreases when going down in the branch-and-bound tree. This decrease of problem sizes speeds up the whole solving process. Note that the interior-point-solver we use is not able to do a warmstart, so starting from scratch at each node yields no loss in performance.

As already introduced in Section 3.2 we do not call the SDP solver if there are no more variables left, i.e., all variables are fixed. In this case we only check feasibility using the methods described in Section 4.4. If there are enough variables unfixed, we start solving. After every successful solving the following procedure is applied.

Algorithm 4.5.1. *Node-procedure for solving MISDPs.*

Input: *A solution of an SDP relaxation.*

```
1. Hand over the objective value to SCIP.
2. Take the solution and check the bounds.
        - If the bound-check was successful:
            3. separate the solution and
            4. choose branching candidates.
        - If the bound-check failed, but the solution was feasible and converged:
            3. use a penalty function to determine whether there is a feasible point or not.
```

To determine if a solving procedure is successful we need to distinguish numerous different cases depending on the return code of the SDP solver. This return code has two dimensions: on the one hand the convergence status and on the other hand the result status. As we implemented an interface to the solver DSDP 5.8 [BY05] we now describe the different states and characteristics for this specific solver.

There are nine different convergence states or stopping reasons for the dual-scaling algorithm, as implemented in DSDP.

(1) `DSDP_CONVERGED`: The algorithm converged.

(2) `DSDP_UPPERBOUND`: The algorithm terminated because the dual objective value exceeded its bound.

(3) `DSDP_SMALL_STEPS`: During the solving process the step size became so small that no progress could be achieved.

(4) `DSDP_MAX_IT`: The maximum iteration limit, set by the user, is exceeded.

(5) `DSDP_INFEASIBLE_START`: The given starting point was infeasible.

(6) `DSDP_INDEFINITE_SCHUR_MATRIX`: During the solving process due to numerical issues the schur complement matrix as described at the end of Section 3.1.1 was indefinite.

(7) `DSDP_NUMERICAL_ERROR`: Some numerical error occurred and it is not clear if the solution is feasible.

(8) `DSDP_USER_TERMINATION`: The user aborted the solving process.

(9) `DSDP_FINISHED`: The algorithm stopped solving, but the reason for this is not clear.

The only status where everything is fine and we can continue without any further analysis is (1). In all other cases the solver has not converged and we need to be careful. Other SDP solvers like PENSDP are also able to report similar states. Additionally, DSDP reports a result status, these results are from the dual point of view, as DSDP uses a dual-scaling algorithm:

- `feasible`: The primal and the dual problem have feasible solutions.
- `infeasible`: The dual problem is infeasible and the primal problem is unbounded.
- `unbounded`: The dual problem in unbounded and the primal SDP is infeasible.
- `unknown`: It is not clear whether the solution is feasible. DSDP was not able to check it.

Now we take a closer look to the different cases where these states occur. After solving there are two possibilities: the solver has converged or not. For each of these two cases the solutions can be feasible, infeasible, unbounded or unknown. The following listing shows what we do in the different cases:

- SDP solver converged
    - result is feasible or unknown
        - ⇒ follow the node-procedure (Algorithm 4.5.1)
    - result is infeasible
        - ⇒ cut off the node
    - result is unbounded
        - ⇒ set the bound to infinity and go on with branch-and-bound algorithm
- SDP solver has **not** converged
    - result is unknown
        - if all integer variables are fixed
            - ⇒ follow the node-procedure (Algorithm 4.5.1)
        - otherwise
            - ⇒ use a penalty function to determine whether there is a feasible point or not (Section 3.2.3)
    - result is infeasible or unbounded
        - if all integer variables are fixed
            - ⇒ cut off the node
        - otherwise
            - ⇒ return `didnotrun`
    - result is feasible
        - ⇒ follow the node-procedure (Algorithm 4.5.1)

In some cases above we use a penalty function to check whether the problem is infeasible. The penalty function uses a method in DSDP. We described this method in Section 3.2.3. It returns whether there is a feasible point for the problem or not.

A rounding heuristic

Solving the relaxation at every branching node is one thing, the other is finding feasible solutions. For most of the examples we are looking at in the next few chapters it is hard to find such feasible solutions. Therefore we implemented a rounding heuristic that can be used for SDP solutions. This heuristic is already described in Algorithm 3.3.1. It is called after the solving of each node because it needs the solution of the relaxation.

The heuristic can be called after each node because its running time is very small. In most of the cases this heuristic is not successful because just rounding variables up and down can also lead to infeasibilities in other constraints. However, if we are located in a node of the branch-and-bound tree that is very deep and near one of the leaves, this heuristic is likely to find a feasible solution.

4.6 Parameter settings – Differences to the MIP-world

After all these explanations it is obvious that solving mixed-integer semidefinite programs is different from solving mixed-integer linear programs. Due to this fact the standard parameter settings in SCIP will not be suitable for our purpose. As there are a lot of parameters in SCIP it is not possible to find the optimal setting but in this section we want to give some hints which parameter setting might be better than the standard one.

The first and most obvious difference between MIP and MISDP is the relaxation used for computing the lower bounds and its solution time. Where LPs can be solved fast and even for a huge number of constraints and variables solving times are still acceptable, solving SDPs takes much time and even small examples cannot be solved fast. So solving one node in an MISDP branch-and-bound tree takes much longer than for a MIP. It is not possible to give such an information to SCIP, but we can change the parameters for those plugins that act on the assumption that solving a node can be done quickly.

For example diving, plunging or strong branching do not make sense for SDPs. The underlying idea of all these things is the following: from the node we are currently located we start a sub-branch-and-bound algorithm, where we fix variables and solve relaxations for different variable configurations. Using the information gathered from this sub-solving process we try to make better decisions for branching in the original branch-and-bound algorithm or obtain feasible solutions. All these methods solve relaxations for getting more information. For MISDPs this is too expensive and it would slow down the solving process, so we are not going to use any of these methods.

The node selection rule we use is called hybrid estimate, this is a combination of best bound search and best estimate search. We use it as pure best bound search without plunging. This search strategy chooses the node with the best lower bound to be chosen next.

Additionally, we do not want to use any pseudo-cost branching, as pseudo-costs also need LP information for being reasonable. The branching rule also works without LP information but the decisions are just as good as branching on a random variable.

The second difference is that we do not have any LP information like reduced costs and we do not have an LP. As mentioned before many heuristics do not work without these information. Fortunately SCIP checks this on its own and we do not have to take care of which plugin is able to run or not. The plugins that need LPs will not be called if no LP is available.

The third difference is the accuracy of the solving process. Our experiments showed that the feasibility tolerances of SDP solvers are not that precise as for LP solvers. Therefore we need to adjust the different tolerances of SCIP. The feasibility tolerance is 10^{-6} by default, but we choose $\varepsilon_{\text{feas}} = 10^{-4}$. Additionally, the tolerance SCIP uses to compare numbers or to check integrality is 10^{-9} and we had to choose $\varepsilon = 10^{-6}$.

Furthermore, if we want to use the approximation procedure (see Section 3.4) additional parameter settings are needed for the generated cutting planes. For example the cuts generated in the root node should not be deleted. Using this approximation many cuts are generated. In order not to have too many of those cuts around, the age of the cuts, as SCIP calls this, must be bounded by some smaller value than the default. Finally, we increase the number of separation rounds per node from five to 20.

CHAPTER 5

Truss Topology Design

Mixed-integer semidefinite programs arise from many applications. One application where they naturally appear as convex reformulation of a nonconvex problem is Truss Topology Design. The motivation underlying this chapter arises from the *Collaborative Research Center 805 – Control of Uncertainty in Load-Carrying Structures in Mechanical Engineering* at TU Darmstadt [CRC16]. In this CRC there are 13 subprojects dealing with different topics of controlling uncertainty. One subproject considers trusses under uncertainty and this is where the idea and description of active components in trusses comes from.

In this chapter we give a detailed description of this type of problems. Moreover, three different types of models are formulated and extended using constraints modeling discrete cross-sectional bar areas, vibrations, and positioning of active components. The basic idea of all models presented in this chapter are taken from the literature, we give a brief overview below. To our best knowledge the extensions of the models to active components, discrete bar areas, and stress constraints at the same time is only examined in the basic nonconvex model. Especially our kind of active components has not been considered before. Some ideas presented in this chapter result from many discussions with Lars Schewe [Sch12].

5.1 Introduction

A truss is a structure consisting of nodes and bars. The position of the nodes and the potential bars is defined in a so-called ground structure. The problem is to find the optimal topology of a truss under a given external load that minimizes the potential energy in the truss caused by the external load. A very famous example of a truss is the Eiffel Tower, but also bridges, electrical towers, and aircraft landing gears can be modeled as trusses.

Modeling trusses in general leads to nonconvex nonlinear problems. We present three different ways to deal with this nonconvexity. One way is to replace the nonconvex constraint by a big-m formulation with binary variables, so the nonconvexity still exists but is changed to something we can handle: binary variables. This leads to a mixed-integer linear problem. Another way is a complete reformulation as convex program. Doing this we obtain a semidefinite problem. A third way we are looking at, reformulates the model using dual variables and a version of the Farkas-Lemma. This yields a quadratic model. Therefore a mixed-integer linear, a semidefinite, and a quadratic model are presented. We discuss the advantages and disadvantages of these models and show that a semidefinite formulation is well suited for the extensions we are interested in.

Our aim is not just to optimize the topology of a truss, but also to position some special bars within this truss which help to reduce the compliance. For an explanation of the physics behind the truss model, which can be derived from linear elasticity the reader is referred to [Ach93].

There are various studies about designing trusses, most of them use the ground structure approach and only consider continuous cross-sectional areas. The first article known to the author that ever appeared about material optimization is from the beginning of the 20th century [Mic04]. Michell describes frame structures and lower bounds on material needed for carrying a certain external load. Since then Truss Topology Design (TTD) has been an evolving field connecting mechanical engineering and mathematics. A lot of articles have been published in the recent years, a brief overview is given below. This overview does not claim to be a comprehensive listing, it should only give an idea of how wide the field of TTD is.

A detailed survey on trusses with continuous cross-sectional areas is given by Achtziger [Ach93]. He explains the physics behind the problem and gives an extensive mathematical description. The concepts of stable and statically determined trusses are introduced and used to prove for example uniqueness of solutions. Ben-Tal and Nemirovski [BTN01] analyze different ways to model a truss with multiple loads. They consider continuous bar areas and present an SDP and a quadratic model (QP). We will introduce this model in Section 5.5. A MIP formulation is introduced in [Sto07] and [SS03]. This formulation will be presented in detail in Section 5.4. Furthermore, in the book *Topology Optimization* [BS04] a more general way of topology optimization is described. Using this, a ground structure is not needed. They also present ground structure approaches and models for different types of variables and additional constraints on the stability of a truss. We will also take a look at some of these constraints in Section 5.6.3. Moreover, in [BS04] they discuss the so-called SIMP model, where SIMP stands for solid isotropic material with penalization. Using this model they obtain very good approximations for 0-1-problems.

Achtziger and Kočvara [AK06] study truss problems with constraints on the eigenvalues of the constraint matrix. Eigenvalues help to control the minimum eigenfrequency of a truss structure, which can be used for controlling the vibrations. They reformulate the truss as SDP with both minimum compliance and minimum volume as the objective function.

Different formulations of trusses with displacement or stress variables are analyzed by Bendsøe, Ben-Tal, and Zowe [BBTZ94]. An extension to nonconvex global buckling constraints is introduced by Ben-Tal, Jarre, Kočvara, Nemirovski, and Zowe [BTJK$^+$00]. It is also possible to model other types of stress constraints that have a local influence on the stiffness of a truss [Ach99] and to distinguish between tension and compression [Ach96]. These constraints are taken into account in Section 5.6.3. Stress constraints are also considered by Takewaki and Kanno [KT05], [KT04].

A robust programming formulation of Truss Topology Design using semidefinite programming is given in [BTN97]. In this case robust means that ellipsoidal uncertainty sets and only continuous bar areas are considered. In the context of robust Truss Topology Design the question arises how the ellipsoidal uncertainty sets should be constructed. If uncertainty is not only considered in the node with the main load, then a set of nodes where the uncertainty appears must be defined. Certainly, this set will also affect the topology. Therefore Yonekura and Kanno look at design dependent uncertainty sets for robust trusses [YK10]. For getting robust not only in the loads Kanno and Guo add stress constraints to this ideas [KG10].

Kočvara and Outrata model the truss as mathematical program with complementarity constraints and solve this problem using a bilevel-approach [KO06]. The advantage of this approach is that they could guarantee a unique design and unique displacements of the optimal truss. A special primal-dual predictor-corrector interior-point-method for solving Truss Topology Design problems is presented by Jarre, Kočvara, and Zowe [JKZ95]. Furthermore, Achtziger, and Stolpe present relaxations of the truss problem in detail and compute different examples [AS08], [AS09]. In [AS07] they present benchmark examples for trusses with discrete cross-sectional areas. Other ideas for solving trusses, for example for a model in displacement-variables only can be found in [BTB93], [Rin86], [KZN00], and [BGH01].

In the following sections we will present different models for trusses under multiple loads and extend them to discrete bar areas and vibration constraints. Additionally, we position some special active bars. To the authors knowledge no model for this kind of problem extension exists so far. A similar idea but with focus on robust models and ellipsoidal uncertainty sets is pursued by [Hab10]. Numerical results for our models will be presented in Chapter 6.

5.2 A nonconvex, nonlinear model

For modeling a truss certain nodes in a given design-space are allowed as connection points. Additionally, there is a number of possible connection elements between the nodes. In this thesis we are only considering a special kind of these connection elements, which are called bars. They are only able to absorb axial loads. Other connecting elements, for example beams, which can also take transverse loads into account, will not be considered within this thesis.

In the following we will use the value *dim* for determining the degrees of freedom of the system. This value stands for the dimension in which the truss should be build. Since we want to build real-world trusses, we always consider two- or three-dimensional systems only.

Definition 5.2.1. The *ground structure* of a truss consists of potential bars and nodes. The set of nodes is denoted by V, one node $v \in V$ is a vector in $\mathbb{R}^{\text{dim}}$. The free nodes are denoted by $V_f \subset V$ and $d = \text{dim} \cdot |V_f|$ is the number of *degrees of freedom* of the system. The set of bars is denoted by E and $n := |E|$. One bar $e \in E$ consists of two nodes (v, w). □

The position of the fixed nodes $V \setminus V_f$, where the truss will be positioned at the ground or a wall, is also given. For being able to describe this situation in a model, we need to define an order of the nodes. This order has no influence on the model, therefore we can define an arbitrary numbering. For example we could use the lexicographic order of $\mathbb{R}^{\text{dim}}$ on the free nodes first. The fixed nodes will always be the last ones, again in lexicographic order.

Let $s \in \mathbb{N}$. Ordering the nodes is important for being able to define the external global loads $f_i \in \mathbb{R}^d$ for $i = 1, \ldots, s$ which are applied to some free nodes. This global loads vector f_i consists of all local loads being applied to the different nodes. The entries of this load-vector depend on the order of the nodes. Depending on the dimension the first *dim* entries of f_i belong to the first node, the second *dim* entries to the second node and so on.

Of these *dim* entries the first one always corresponds to the x-component of the load, the second to the y-component and if dim=3 the third one stands for the z-component. If the entries corresponding to a node are all equal to zero, then there is no local load applied to this node. Loads cannot be applied to fixed nodes.

Each of the loads is called a scenario and consequently there are s different scenarios. The set of all scenarios is called $S = \{f_1, \ldots, f_s\}$. For $s = 1$ we call it the single-load case if $s > 1$ we have multiple loads. Note that for example it is also possible that two local loads apply to two different nodes in the same scenario. This is illustrated in a small example using Figure 5.1.

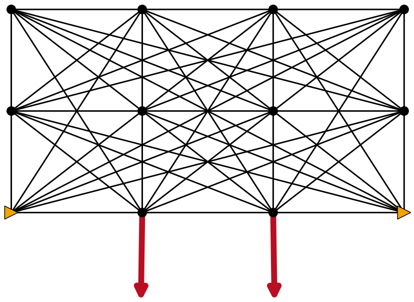


Figure 5.1. – The ground structure of a 4 × 3 truss with two local loads (red arrows) and two fixed nodes (orange triangles).

Example 5.2.2. *The given two-dimensional ground structure of Figure 5.1 has ten free nodes and two fixed nodes. There are* 49 *potential bars and* 20 *degrees of freedom, two for every free node. Two local loads are shown in the figure and there are two cases how these loads can be represented. Note that the order of the entries in the load vectors f_i depend on the numbering of the free nodes, recall that we use the lexicographic order of $\mathbb{R}^{\text{dim}}$ on the free nodes. We consider two possibilities:*

(i) Both local loads apply in the same scenario, so $s = 1$ and the load-vector has this form: $f = (0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, -1, 0, \dots, 0)^T$.

(ii) Every local load applies in an own scenario, so $s = 2$ and the two load-vectors look like this: $f_1 = (\underbrace{0, \dots, 0}_{10}, 0, -1, 0, \dots, 0)^T$ and $f_2 = (0, 0, 0, 0, 0, -1, 0, \dots, 0)^T$.

The optimized trusses can look very different depending on whether the loads are applied in one or two scenarios. Figure 5.2 shows the results of the optimized truss for case (i) with one scenario (left), and case (ii) with two scenarios (right).

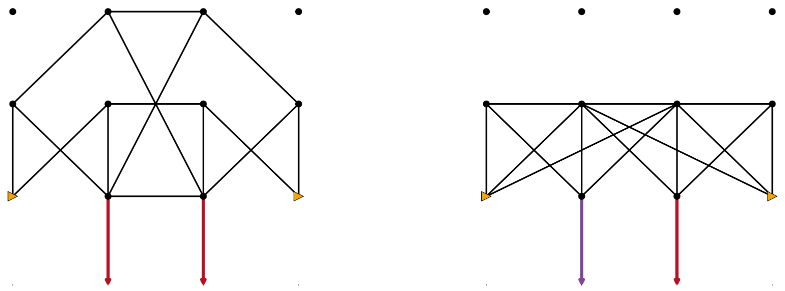


Figure 5.2. – Optimized trusses for one (left) and two scenarios (right).

In the two pictures of figure 5.2 it can be seen that the topology of the truss is different if the loads are applied at the same time in comparison to the case where the loads act in different scenarios. If both loads are applied at the same time, the truss needs to be more rigid to carry them. The difference can also be seen in the objective value of the examples. As objective we consider the compliance, which is a measure for the deformation, and we want to minimize it.

For the left example with both loads applied in one scenario the compliance is 1.91*. For the right example the loads are applied in two scenarios. The objective function is to minimize the maximum of the two compliances and this minimum has the value* 0.77*. So we observe that for the objective function it makes a difference if we choose one or two scenarios and in this example the complicance for the two scenarios is smaller than if we consider both loads in one scenario.*

Our aim is to produce a very stiff truss by selecting only some of the possible bars and adjusting their thickness. The design variables considered are the cross-sectional areas x

of the bars. As measure for the stability of the truss, the potential energy stored in the truss in every scenario, called compliance, is used. This potential energy can be expressed by the product of the loads and the displacement of the nodes u for every scenario:

$$c_{f_i}(u) \equiv \frac{1}{2} f_i^T u_i \quad \text{for } i \in S.$$

The compliance should be minimized for the worst case scenario with respect to a bounded volume of the truss $V_{\max}$.

It is also possible to bound the compliance by some value and minimize the volume of the truss instead. For now we only want to minimize the compliance with restricting the volume. Note that minimizing the volume is equivalent to this problem. We will comment on this fact later in Section 5.3.1 because from an algorithmic point of view it makes a difference which model we choose.

Now we are able to formulate the well-known multiple loads model of Truss Topology Design with design variables x_e for the bar areas for every bar $e = (v, w) \in E$ and variables $u \in \mathbb{R}^{d \times s}$ for the displacements of the nodes in every scenario:

$$\begin{array}{lll} \min & \max\limits_{i \in S} \frac{1}{2} f_i^T u_i & \\ \text{s.t.} & A(x) u_i = f_i & i = 1, \dots, s \\ & \sum\limits_{e=1}^{n} l_e x_e \leq V_{\max} & \\ & u_i \in \mathbb{R}^d & i = 1, \dots, s \\ & x \in \mathbb{R}^n & \\ & x_e \geq 0 & e = 1, \dots, n, \end{array} \tag{5.1}$$

where l_e is the length of the e-th bar, $A(x) = \sum_{e=1}^{n} A_e x_e$, $A_e = \kappa b_e b_e^T$ and

$$b_e = (b_e(k))_{k \in V_f} = \begin{cases} (w - v) \left\| w - v \right\|^{-\frac{3}{2}}, & k = v \\ (v - w) \left\| w - v \right\|^{-\frac{3}{2}}, & k = w \\ 0, & \text{otherwise.} \end{cases}$$

We call $A(x)$ the stiffness matrix of the truss, the A_e are the stiffness matrices for each bar e. The kind of material that is used and its characteristics are included in the parameter κ, also called elasticity modulus. Here k, v, and w are the coordinates of the nodes. At this point we want to remark that the matrices A_e all have rank one.

Note that the objective of the optimization problem (5.1) is to minimize the compliance for the worst case of loads f_i. This objective can also be expressed in the following way:

$$c_{f_i}(x) \equiv \frac{1}{2} f_i^T u_i = \sup_{\tilde{u}_i \in \mathbb{R}^d} [f_i^T \tilde{u}_i - \frac{1}{2} \tilde{u_i}^T A(x) \tilde{u_i}] \quad \text{for } i \in S.$$

This equation is due to the variational description of the compliance. A more detailed explanation is given in [BTN01].

The most important constraints in this model are those ensuring that the truss is able to carry the loads f_i:

$$A(x)u_i = f_i \quad \text{for } i = 1, \dots, s. \tag{5.2}$$

This equilibrium constraint makes this problem nonconvex and nonlinear because x as well as u are variables. So obtaining the global optimum cannot be expected. In the following sections we will show different ways to reformulate this truss model as a convex program or a program with nonconvexity that we are able to deal with. First we need some more definitions and properties of an optimized truss.

Definition 5.2.3. The optimized truss structure under the load f is denoted by $T(\bar{x}, f)$. For a multiple loads truss the notation is $T(\bar{x}, S)$, where S is the set of all scenarios. The vector $\bar{x}$ is the optimal solution of problem (5.1). □

Lemma 5.2.4. *Optimized trusses are convex in terms of the applied loads in the following sense:*

(a) An optimized truss $T(\bar{x}, f)$ that is able to carry load f can also carry load $\hat{f} = -f$.

(b) An optimized truss $T(\bar{x}, S)$ under multiple loads that can carry the loads $f_i \in S$, is also able to carry every convex combination of these loads.

Proof. A truss $T(\bar{x}, f)$ being able to carry a load f means that the there must be an equilibrium between inner and outer forces. This implies that the following system of equations has a solution:

$$A(\bar{x})u = f,$$

where $\bar{x}$ simultaneously is a specific choice of bars and the optimal solution of (5.1). For multiple loads we have $A(\bar{x})u_i = f_i$ for $i = 1, \dots, s$. Note that $A(\bar{x})$ is not necessarily nonsingular, it is positive semidefinite and an eigenvalue equal to zero can exist. An example is given below.

(a) Let $A(\bar{x})\hat{u} = \hat{f} = -f$, then $\hat{u} = -u$ is a solution for this system. So there is a symmetry with respect to the loads.

(b) Now let $\lambda \in [0, 1]$ and $\hat{f} = \lambda f_i + (1 - \lambda) f_j, i \neq j$ and $A(\bar{x})\hat{u} = \hat{f} = \lambda f_i + (1 - \lambda) f_j$ be the system of equations we want to solve. Let $\hat{u} = \lambda u_i + (1 - \lambda) u_j$, then:

$$\begin{array}{rrcl} & A(\bar{x})\hat{u} & = & \hat{f} \\ \Longleftrightarrow & A(\bar{x})(\lambda u_i + (1-\lambda)u_j) & = & \hat{f} \\ \Longleftrightarrow & \lambda A(\bar{x})u_i + (1-\lambda)A(\bar{x})u_j & = & \hat{f} \\ \Longleftrightarrow & \lambda f_i + (1-\lambda) f_j & = & \hat{f}. \end{array}$$

So $\hat{u}$ as a convex combination of u_i and u_j fulfills this equation. □

For a detailed analysis of the solvability of the truss equilibrium the reader is referred to the first chapter of the thesis of Wolfgang Achtziger [Ach93].

Remark 5.2.5. *Lemma 5.2.4 allows an analogy to robust optimization. For a robust formulation of* (5.1) *the loads are given using a matrix Q that defines an ellipsoid with $f \in Q$. In this case the truss is optimized for an ellipsoidal uncertainty set and can carry all loads in the ellipsoid. Analogously and using Lemma 5.2.4 a truss under multiple loads is optimized for a* polyhedral *uncertainty set.*

Example 5.2.6. *In general and using the presented models it cannot be expected that the matrix $A(\bar{x})$ for an optimized truss $T(\bar{x}, f)$ is positive definite. In this example we consider a slightly smaller ground structure than before with only* 4 *free nodes,* 2 *fixed nodes, and* 12 *possible bars. There is only one load applied and we solved it for discrete cross-sectional bar areas, as presented below in Section 5.3.*

Figure 5.3 shows an optimal truss for which the stiffness matrix $A(\bar{x})$ is not positive definite.

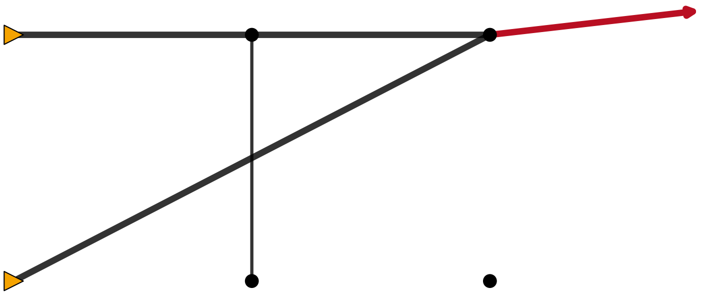


Figure 5.3. – The corresponding stiffness matrix of this truss is not invertible.

There is one vertical bar that does not seem to make sense and this is the bar that makes the matrix singular. This bar could be omitted as it does not have any influence on the objective function. It appears because the volume bound for the truss is not tight and in general we cannot act from the assumption that bounds on volume are good because for bigger and new examples good bounds are unknown. There is some volume left, which is not needed for carrying the load, adding an additional bar does not affect the compliance, so it does not increase the costs and therefore the optimization algorithm positions this bar anywhere.

Adding some costs for bars to the truss model would avoid these redundant bars. For example introducing self-weights or production costs would be a natural form of costs. Both can be added to all the models presented below, but we are not considering them. In other contexts self-weights are necessary, for example [Sch11] considers them in the context of pipes that should be routed through a building, which can also be modeled as a truss with different connection elements. The elements used in this case are called beams.

5.3 An SDP model

For obtaining a global optimal solution we somehow need to deal with the nonconvexity in (5.1). One way is presented in this section. The problem is transformed into an equivalent semidefinite program (SDP), as shown in [BTN97]:

Lemma 5.3.1. *Let A be a positive semidefinite $n \times n$ matrix and let*

$$c = \max_{u \in \mathbb{R}^d} [f^T u - \frac{1}{2} u^T A(x) u].$$

Then the inequality $c \leq \tau$ is equivalent to

$$\mathcal{A} = \begin{pmatrix} 2\tau & f^T \\ f & A(x) \end{pmatrix} \succeq 0.$$

Proof. By re-scaling c it suffices to show that $\tilde{c} = \max_{u \in \mathbb{R}^d} [2f^T u - u^T A(x) u] \leq \tau$ is equivalent to $\tilde{\mathcal{A}} = \begin{pmatrix} \tau & f^T \\ f & A(x) \end{pmatrix} \succeq 0$.

The following equivalences hold:

$$\begin{array}{lrcll}
 & \tau & \geq & \tilde{c} & \\
\Longleftrightarrow & \tau - 2f^T u + u^T A(x) u & \geq & 0 & \text{for all } u \in \mathbb{R}^n \\
 & & & & (\text{multiply with } \lambda^2 \in \mathbb{R}) \\
\Longleftrightarrow & \lambda^2 \tau - \lambda^2 (2f^T u) + \lambda^2 (u^T A(x) u) & \geq & 0 & \text{for all } u \in \mathbb{R}^n \text{ and } \lambda \in \mathbb{R} \\
\Longleftrightarrow & \tau\lambda^2 - 2(\lambda f^T)(\lambda u) + (\lambda u)^T A(x)(\lambda u) & \geq & 0 & \text{for all } u \in \mathbb{R}^n \text{ and } \lambda \in \mathbb{R} \\
 & & & & (\text{set } (-\lambda u) = y) \\
\Longleftrightarrow & \tau\lambda^2 + 2(\lambda f^T) y + y^T A(x) y & \geq & 0 & \text{for all } y \in \mathbb{R}^n \text{ and } \lambda \in \mathbb{R} \\
\Longleftrightarrow & \begin{pmatrix} \lambda \\ y \end{pmatrix}^T \begin{pmatrix} \tau & f^T \\ f & A(x) \end{pmatrix} \begin{pmatrix} \lambda \\ y \end{pmatrix} & \geq & 0 & \text{for all } y \in \mathbb{R}^n \text{ and } \lambda \in \mathbb{R}.
\end{array}$$

This is exactly the definition of positive semidefiniteness. So $\tilde{A} \succeq 0$ if and only if $\tau \geq \tilde{c}$. □

The given proof can be extended to the multiple loads case. Every equation of type (5.2) can be reformulated using τ_i for $i = 1, \ldots, s$. This results in s additional inequalities $\tau_i \leq \hat{\tau}$. Now we are able to reformulate (5.1) as a semidefinite program:

$$
\begin{aligned}
\min \quad & \hat{\tau} \\
\text{s.t.} \quad & \begin{pmatrix} 2\tau_i & f_i^T \\ f_i & A(x) \end{pmatrix} \succeq 0 && i = 1, \ldots, s \\
& \sum_{e=1}^{n} l_e x_e \leq V_{\max} \\
& \tau_i \leq \hat{\tau} && i = 1, \ldots, s \\
& \hat{\tau} \in \mathbb{R} \\
& x \in \mathbb{R}^n \\
& \tau \in \mathbb{R}^s \\
& x_e \geq 0 && e = 1, \ldots, n.
\end{aligned}
\tag{5.3}
$$

Using this program the displacement variables u are omitted, only the bar area variables x and the compliance τ are left. On the other hand, we obtain some new variables τ_i, one for each scenario $i \in S$. Existing interior-point-solvers can solve this SDP model in suitable time. This is well known and discussed extensively in the literature as mentioned in the introduction of this chapter.

Remark 5.3.2. *Due to the fact that the matrix*

$$\begin{pmatrix} 2\tau_i & f_i^T \\ f_i & A(x) \end{pmatrix}$$

has to be positive semidefinite, τ_i must be nonnegative.

Remark 5.3.3. *The problem* (5.3) *is indeed in the transformed dual standard form* (2.1) *as defined in Section 2.3. For the single-load case we present the necessary transformations in the following. Recall the dual standard form:*

$$
\begin{aligned}
\min \quad & \tilde{b}^T y \\
\text{s.t.} \quad & \sum_{i=1}^{m} \tilde{A}_i y_i - \tilde{A}_0 \succeq 0 \\
& y \in \mathbb{R}^m.
\end{aligned}
$$

Here $y = (\hat{\tau}, x_1, \ldots, x_n)^T$, $\tilde{b} = (1, 0, \ldots, 0)^T$ and

$$
\begin{aligned}
\tilde{A}_0 &= \begin{pmatrix} 0 & f^T \\ f & 0 \end{pmatrix}, \\
\tilde{A}_1 &= \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix}, \\
\tilde{A}_i &= \begin{pmatrix} 0 & 0 \\ 0 & A_i \end{pmatrix}, \quad i = 2, \ldots, n+1.
\end{aligned}
$$

Additionally to this SDP constraint, we have the following linear constraints:

$$\begin{aligned} \sum_{e=1}^{n} l_e x_e \leq V_{\max} & \\ \tau_i \leq \hat{\tau} & \qquad i = 1, \dots, s \\ x_e \geq 0 & \qquad e = 1, \dots, n. \end{aligned}$$

The dimension of the variables y is $s + n + 1$.

Discrete cross-sectional bar areas

The assumption that the cross-sectional areas of the bars x_e can have every positive real value is unrealistic due to production restrictions. Therefore from now on we suppose to have m different discrete bar areas β^a chosen from a set B. This is modeled using binary variables for the x_e and the SDP changes to a mixed-integer semidefinite program (MISDP) due to the following additional restriction:

$$x_e^a \in \{0, 1\} \qquad \text{for } e = 1, \dots, n, \quad a = 1, \dots, m.$$

Then $x_e^a = 1$ means that bar e has area a. An additional inequality is needed so that every bar can only have one area or the area zero:

$$\sum_{a=1}^{m} x_e^a \leq 1 \quad \text{ for } e = 1, \dots, n.$$

Observation 5.3.4. *The proof of Lemma 5.3.1 shows that adding additional linear constraints to our truss model does not affect the reformulation. Also changing the variables from continuous to integer or binary can be done and the reformulation still works, i.e., an analogous statement of Lemma 5.3.1 holds.*

This binary model looks as follows:

$$\begin{aligned} \min \quad & \hat{\tau} \\ \text{s.t.} \quad & \begin{pmatrix} 2\tau_i & f_i^T \\ f_i & \sum_e A_e \sum_a \beta^a x_e^a \end{pmatrix} \succeq 0 & i = 1, \dots, s \\ & \sum_{e=1}^{n} l_e \sum_{a=1}^{m} \beta^a x_e^a \leq V_{\max} & \\ & \tau_i \leq \hat{\tau} & i = 1, \dots, s \\ & \sum_{a=1}^{m} x_e^a \leq 1 & e = 1, \dots, n \\ & \hat{\tau} \in \mathbb{R} & \\ & \tau \in \mathbb{R}^s & \\ & x^a \in \{0, 1\}^n & a = 1, \dots, m. \end{aligned}$$

Note that there is a second way to model discrete bar areas. The first one is described above: for every cross-sectional area and every bar there is a binary variable and additional constraints that allow at most one area per bar. The other way is to use integer variables for the discrete cross-sectional areas. Using this approach there is only one variable for every bar and no additional constraint.

Hence this idea reduces the number of variables and the number of constraints and transforms the mixed-binary SDP to a general mixed-integer SDP:

$$\begin{aligned}
\min \quad & \hat{\tau} \\
\text{s.t.} \quad & \begin{pmatrix} 2\tau_i & f_i^T \\ f_i & \sum_e A_e x_e \end{pmatrix} \succeq 0 & i = 1, \dots, s \\
& \sum_{e=1}^{n} l_e x_e \leq V_{\max} \\
& \tau_i \leq \hat{\tau} & i = 1, \dots, s \\
& \hat{\tau} \in \mathbb{R} \\
& \tau \in \mathbb{R}^s \\
& x \in \mathbb{N}_0^n.
\end{aligned}$$

Both models produce the same results, only the computational time needed to solve them is different. As shown in Chapter 6 the second variant can be solved faster. Also from the SDP solver point of view the second alternative should be preferred. In the first model, there will in general be many coefficient matrices which are linear dependent, which means that $A_e^{a_1} = \alpha A_e^{a_2}$, where $\alpha = \frac{\beta^{a_2}}{\beta^{a_1}}$. This is due to the fact that there are different bar areas for the same bar and this leads to the same coefficient matrices except for a linear factor for the different bar areas. This possibly causes trouble in the SDP solver because the solver is not aware of the fact that only one of the corresponding variables $x_e^{a_1}$ and $x_e^{a_2}$ may take the value one.

The impact of discrete bar areas on the topology is illustrated in the following example:

Example 5.3.5. *Again we consider the example of Figure 5.1. We assume that there is only one scenario. The optimal topology for continuous bar areas is shown in Figure 5.4.*

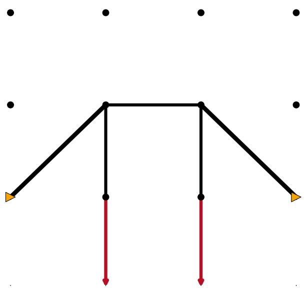


Figure 5.4. – Optimized trusses with continuous bar areas.

Now we switch to discrete bar areas. Figure 5.5 illustrates the change in topology. In the left picture we only decide if a bar exists or not, so there are only two bar areas, namely zero and one. For the right truss, three bar areas are allowed, zero, one, and three. The pictures show that the topology can change completely and it will change again if we choose different areas.

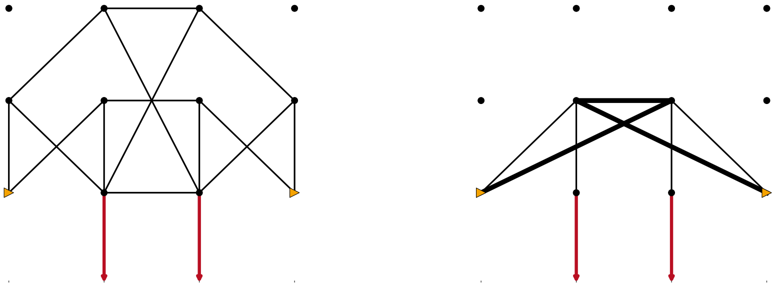


Figure 5.5. – Optimized trusses for one (left) and two different bar areas (right).

5.3.1. Minimizing the volume

There is another way of modeling a truss topology optimization problem, which we will present here for continuous bar areas. It can easily be extended to discrete bar areas.

Instead of minimizing the compliance we bound the compliance by some value and minimize the volume of the truss. The minimum volume notation of the SDP truss model is equivalent to the one with the compliance in the objective value. However, from a solver point of view there are big differences that make the minimum volume formulation easier to solve. Therefore is it worth looking at it:

$$\begin{array}{lll} \min & \sum_e l_e x_e & \\ \text{s.t.} & \begin{pmatrix} 2\tau_i & f_i^T \\ f_i & A(x) \end{pmatrix} \succeq 0, & i = 1, \dots, s \\ & \tau_i \le \tau_{\max} & i = 1, \dots, s \\ & \tau \in \mathbb{R}^s & \\ & x \in \mathbb{R}^n & \\ & x_e \ge 0 & e = 1, \dots, n. \end{array} \tag{5.4}$$

This is a truss model with continuous bar areas and a bound for the maximum compliance of $\tau_{\max}$.

Remark 5.3.6. *Every optimal solution of* (5.3) *with objective value $\bar{\tau}$ leads to the same optimal solution of* (5.4) *with $\tau_{\max} = \bar{\tau}$ and vice versa. For discrete bar areas the bounds on volume and compliance are not necessarily tight because not every volume bound can be* *attained with only a finite set of areas to choose from. The same is true with respect to the compliance.*

With formulation (5.4) finding feasible solutions becomes easy:

Observation 5.3.7. *While problem* (5.3) *normally has a feasible solution – possibly with a very big compliance – but finding one is very hard for solvers, the problem* (5.4) *allows a trivial way to find solutions or detect infeasibility, but sometimes it is infeasible – because the compliance bound cannot be reached.*

A trivial heuristic for finding feasible solutions for (5.4) *makes all bars as thick as possible. In the case of binary bar areas, all bars are used and $x_e = 1$ for $e = 1, \dots, n$. Then there are two possibilities for this solution:*

- *it is feasible and $\tau_i \leq \tau_{\max}$ for $i = 1, \dots, s$,*

- *it is infeasible: then there is no feasible solution and the whole problem is infeasible.*

In conclusion the branch-and-bound framework can always start with an upper bound. Another important point about formulation (5.4) is the fact that all design variables are used in the objective function, whereas in formulation (5.3) they are not. For many standard linear techniques this is an important fact, for example some heuristics use the objective coefficients for decisions. Moreover, branching and node-selection can use information about the impact of a variable to the objective value. This impact is not clear to those modules in the minimum compliance formulation because it is just implicitly given through the semidefinite constraint and cannot be used directly.

Finally, from an engineering point of view it is more natural to bound the compliance of a truss and minimize the volume. Solving problems of type (5.3) with a small volume bound sometimes leads to trusses with a very high compliance, which means that there are very big displacements in the nodes of the truss and this implies that the truss cannot be build this way.

5.4 A MIP model

Solving large SDPs with many binary variables is quite expensive, even solving one large SDP is expensive. This is due to the fact that many standard heuristics, presolve, and cutting plane techniques do not work properly. So it is obvious to try a reformulation of the nonconvex truss as a mixed-integer linear program and solve it using standard solvers like CPLEX [CPL12] and GUROBI [GUR11]. The nonconvexity is not avoided in this case, it is just rewritten into new binary variables and a big-m formulation. Therefore we will now show how to get a linear MIP out of (5.1) as presented in [Sto07]. For simplicity we only look at a single load problem, but note that the problem can also be formulated for multiple loads. The idea of the reformulation only works for binary variables x_e, so the case considered in this formulation is the existence of the bars.

Additionally, bounds on the displacement of each node are needed. These bounds seem to be a hard restriction for the model, but from a application point of view they are reasonable. Bounds on u avoid big displacements and therefore unrealistic truss configurations. In the following section we will see that for some special constraints we explicitly need such bounds.

We start with the following model:

$$\begin{aligned} \min \quad & \frac{1}{2} f^T u \\ \text{s.t.} \quad & A(x)u = f, \\ & \sum_{e=1}^{n} l_e x_e \leq V_{\max} \\ & \qquad u \in [u^{\min}, u^{\max}]^d \\ & \qquad x \in \{0,1\}^n. \end{aligned}$$

Now the model is transformed into something that can be handled. The problematic constraint which we want to reformulate is $A(x)u = f$.

Recall that

$$A(x)u = \left(\sum_{e=1}^{n} A_e x_e \right) u = \left(\sum_{e=1}^{n} x_e \kappa b_e b_e^T \right) u.$$

Additionally introduce a new variable q_e, which indicates the member force, i.e., the force being present in bar e:

$$q_e = \kappa x_e b_e^T u. \tag{5.5}$$

So now the equilibrium looks like $\sum_{e=1}^{n} b_e^T q_e = f$. This equation is convex and linear, but we obtained another nonlinear equation (5.5).

Therefore, as second step, we need to linearize the condition (5.5). As x is binary and u is bounded, we may use standard linearization strategies. We can rewrite q as

$$q = \begin{cases} 0, & x = 0 \\ \kappa b^T u, & x = 1. \end{cases} \tag{5.6}$$

The bounds on q are given by

$$c^{\min} = \min_{u^{\min} \leq u \leq u^{\max}} \kappa b^T u \qquad \text{and} \qquad c^{\max} = \max_{u^{\min} \leq u \leq u^{\max}} \kappa b^T u.$$

Using these bounds to express the first part of (5.6) two linear inequalities are needed:

$$x c^{\min} \leq q \leq x c^{\max}.$$

For the second part of (5.6) another two inequalities are introduced:

$$(1-x) c^{\min} \leq \kappa b^T u - q \leq (1-x) c^{\max}.$$

By now we can state a MIP model of the truss.

$$
\begin{array}{lrll}
\min & & \frac{1}{2}f^T u & \\
\text{s.t.} & x_e c_e^{\max} & \geq q_e & e = 1, \ldots, n \\
& x_e c_e^{\min} & \leq q_e & e = 1, \ldots, n \\
& (1 - x_e) c_e^{\min} & \leq \kappa b_e^T u - q_e & e = 1, \ldots, n \\
& (1 - x_e) c_e^{\max} & \geq \kappa b_e^T u - q_e & e = 1, \ldots, n \\
& \sum_{e=1}^{n} b_e^T q_e & = f & \\
& \sum_{e=1}^{n} l_e x_e & \leq V_{\max} & \\
& u & \in [u^{\min}, u^{\max}]^d & \\
& x_e & \in \{0, 1\}^n. &
\end{array}
\tag{5.7}
$$

As shown in [Sto07] solving this model is easier with a modified objective function, as the fact that $x = x^2$ for $x \in \{0, 1\}$ can be exploited:

$$
\begin{aligned}
f^T u &= u^T A(x) u \\
&= u^T \left(\sum_{e=1}^{n} x_e \kappa A_e \right) u \\
&= u^T \left(\sum_{e=1}^{n} x_e^2 \kappa b_e b_e^T \right) u \\
&= u^T \left(\sum_{e=1}^{n} x_e \kappa b_e x_e b_e^T \right) u \\
&= \sum_{e=1}^{n} u^T \kappa x_e b_e x_e b_e^T u \\
&= \sum_{e=1}^{n} (\kappa x_e b_e^T u)^T (\kappa x_e b_e^T u) \kappa^{-1} \\
&= \frac{q^T q}{\kappa} = \sum_{e=1}^{n} \frac{q_e^2}{\kappa}.
\end{aligned}
$$

This new and equivalent objective function is much easier to solve because there are no variables x and u in it. In contrast this model is not easy to solve as for large bounds on u the $c_{\min}$- and $c_{\max}$-values are getting too big, while for small bounds there will be no feasible solution. We will show this and the advantages of the SDP model in the following chapter.

Remark 5.4.1. *This formulation as linear MIP only works for binary variables due to the reformulation step in* (5.6)*. Therefore the case of continuous bar areas cannot be considered and modeling discrete bar areas is only possible using the binary formulation. Extending this model for multiple loads is possible, then variables q_e^i for every bar in every scenario are needed.*

Remark 5.4.2. *In this linear MIP there are explicit variables on the member forces in each bar: q_e. These variables q_e also have the same value as the stress in the bar, as the stress is q_e per area and we assume that the area is zero or one. This is a very important fact when looking at the stability of a truss. The forces appearing in each bar should be bounded by a constant or by the critical load of a bar. Details can be found in Section 5.6.3.*

Remark 5.4.3. *Formulating the MIP presented above as a problem of minimizing the volume can be done easily as in the case of an SDP. Furthermore, the computational behavior is similar to the case of minimizing the volume in the SDP case. The solving time decreases and finding feasible solutions is easier. Note that the problem type changes when minimizing the volume because the quadratic function of the objective will then be a constraint. The resulting problem is a mixed-integer quadratically constraint problem. Computational results for this model will be presented in Chapter 6.*

5.5 A quadratic model

For the sake of completeness we present a third class of models for Truss Topology Design in this section. From a hierarchical point of view there is one class missing between a linear program and a semidefinite program: a second-order-cone program (SOCP). There is a way to formulate a truss as a quadratic program and transform it into an SOCP, as for example described in [BTN01].

Moreover, the way to handle the nonconvexity is again different from the MIP and the SDP reformulation. For this reformulation we will use duality and a version of the Farkas lemma. This is the most complicated reformulation and the transformed model in the end is not easy to handle, since for example the meaning of the variables in reality is unclear.

In the sequel we will follow [BTN01] for deriving the model. All of the theorems and proofs can be found there. It can be shown that finding the optimal topology of a truss is equivalent to the following problem as we already mentioned above. We want to minimize the potential energy stored in the truss, and this energy can be expressed using the following quadratic form:

$$C_{x,f}(u) = \frac{1}{2}u^T A(x)u - f^T u.$$

Now we present some facts for this quadratic form.

Lemma 5.5.1. *Take a quadratic form*

$$C_{x,f}(u) = \frac{1}{2}u^T A(x)u - f^T u$$

on $\mathbb{R}^m$ with positive semidefinite matrix A. Then the following properties hold:

(i) $C_{x,f}(u)$ is bounded below if and only if it attains its minimum.

(ii) $C_{x,f}(u)$ attains its minimum if and only if the equation

$$Au = f$$

is solvable. Then the set of minimizers of $C_{x,f}(u)$ is exactly the set of solutions of the equation.

(iii) If the minimum exists, its value is equal to $-\frac{1}{2}f^T u$, where u is a solution to $Au = f$.

Proof. We will now proof each of the three statements:

(i) Proving the if part is trivial: If the form attains its minimum it is bounded below by this minimum.

To prove the only if part we have to distinguish two cases:

(a) f is orthogonal to the null space of A, then the system $Au = f$ is solvable due to the fact that a linear system $Dx = b$ is solvable if and only if b is orthogonal to the null space of D^T. We may apply this, as $A^T = A$. The gradient of $C_{x,f}(u)$ is equal to zero at every solution u and as $A(x)$ is positive semidefinite, $C_{x,f}(u)$ is convex and so such a solution u is a minimizer. So if the form is bounded below it attains its minimum.

(b) f has a nonzero projection f' onto the null space of A, then the form is unbounded below because we can take a $\hat{u} = tf'$ and for $t \to \infty$ the form tends to $-\infty$.

(ii) The minimizers of $C_{x,f}(u)$ are the points where the gradient is equal to zero, as $C_{x,f}(u)$ is smooth and convex. The gradient of $C_{x,f}(u)$ is $Au - f$, which is equal to zero if and only if $Au = f$.

(iii) Now let $\bar{u}$ be a solution of $Au = f$, then this is also a minimizer of $C_{x,f}(u)$. Moreover,

$$\bar{u}^T A\bar{u} = \bar{u}^T f.$$

Using this we can rewrite

$$C_{x,f}(u) = \frac{1}{2}f^T\bar{u} - f^T\bar{u} = -\frac{1}{2}f^T\bar{u}.$$ □

Applying this lemma we can state the following theorem.

Theorem 5.5.2. *The compliance is finite and we have*

$$-\mathsf{Compl}_f(x) = \min_u C_{x,f}(u).$$

if and only if the quadratic form

$$C_{x,f}(u) = \frac{1}{2}u^T A(x)u - f^T u \tag{5.8}$$

is bounded below.

Proof. Using Lemma 5.5.1 and the fact that $A(x)$ is positive semidefinite for $x \geq 0$ we know that the quadratic form (5.8) is bounded below if and only if it attains its minimum. If and only if it attains its minimum the equation $Au = f$ is solvable and the minimum value of Lemma 5.5.1 is $-\frac{1}{2}f^T u$ for a solution of $Au = f$. Furthermore, $\frac{1}{2}f^T u$ is exactly the compliance per definition.

Since we only used equivalences the other direction is also true, which finishes the proof. □

Using the results from above we come back to the Truss Topology Design problem. We are interested in the following epigraph:

$$\mathcal{C} = \left\{(x, f; \tau) : x \geq 0, \tau \geq \mathsf{Compl}_f(x)\right\}.$$

For simplicity we will start with a slightly smaller set

$$\mathcal{C}^\circ = \left\{(x, f; \tau) : x \geq 0, \tau > \mathsf{Compl}_f(x)\right\}.$$

Now we can use Theorem 5.5.2 and rewrite this set using a quadratic form. So the set $\mathcal{C}$ consists of all triples (x, f, τ) for nonnegative x such that the quadratic form

$$Q(u) = \frac{1}{2}u^T A(x)u - f^T u + \tau.$$

is nonnegative for all $u \in \mathbb{R}^d$. The set $\mathcal{C}^\circ$ contains all triples (x, f, τ) such that the quadratic form is strictly positive, i.e.,

$$(x, f, \tau) \in \mathcal{C}^\circ \text{ if and only if } Q(u) > 0.$$

This means that the convex quadratic inequality $Q(u) \leq 0$ has no solutions.

We are now looking for the conic quadratic representation of this form. Therefore we first need the definition of the cone we want to look at, which is exactly the Lorentz cone.

Definition 5.5.3 (Lorentz cone or Second-order cone)**.** The following cone is called the *Lorentz Cone* or *Second-Order Cone*:

$$L^k = \{x = (x_1, \ldots, x_k) \in \mathbb{R}^k \mid x_k \geq \sqrt{x_1^2 + \ldots + x_{k-1}^2}\}, \quad k \geq 2.$$

□

Using this cone we can state the definition of conic quadratic representable.

Definition 5.5.4. A set $X \in \mathbb{R}^n$ is *conic quadratic representable* if we can find a system S of finitely many inequalities of the form

$$A_j \begin{pmatrix} x \\ u \end{pmatrix} - b_j \geq_{L^{m_j}} 0$$

with variables $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^d$ and m_j being the dimension of the Lorentz cone such that X is the projection of the solution set of S onto the x-space, i.e.,

$$x \in X \Longleftrightarrow \exists u : A_j \begin{pmatrix} x \\ u \end{pmatrix} - b_j \geq_{L^{m_j}} 0, j = 1, \dots, N. \qquad \square$$

Before explicitly reformulating the truss problem define

$$B(x) := \sqrt{2} \begin{pmatrix} \sqrt{x_1} b_1^T \\ \dots \\ \sqrt{x_n} b_n^T \end{pmatrix}$$

and write the stiffness matrix A in the following way

$$A(x) = \sum_{e=1}^{n} x_e b_e b_e^T = \frac{1}{2} B^T(x) B(x).$$

Exploiting this we can reformulate $Q(u)$:

$$\begin{aligned} Q(u) &= \frac{1}{4} u^T B^T(x) B(x) u - f^T u + \tau \\ &= \frac{1}{4} \left(\|B(x)u\|_2^2 + (1 - f^T u + \tau)^2 - (1 + f^T u - \tau)^2 \right). \end{aligned}$$

As stated above $Q(u) \leq 0$ has no solution for triples in $\mathcal{C}^\circ$. Using the reformulation of $Q(u)$ this is equivalent to the fact that the inequality

$$\left\| \begin{pmatrix} B(x)u \\ 1 - f^T u + \tau \end{pmatrix} \right\|_2 \leq 1 + f^T u - \tau \tag{5.9}$$

has no solutions. To enforce this statement we need the following definition.

Definition 5.5.5. The inequality $Ax - b \geq_K 0$ is *almost solvable* if and only if for all $\varepsilon > 0$ there exists b' such that $\|b' - b\|_2 < \varepsilon$ and the perturbed system $Ax - b' \geq_K 0$ is solvable. □

Using this we can state the following:

A triple (x, f, τ) belongs to $\mathcal{C}^\circ$ if and only if (5.9) is not even almost solvable.

We are not going to prove this fact, the idea of the proof is to perturb the system and show that the minimum of the form depends continuously on the perturbation. The remaining part is proved by contradiction (see [BTN01]).

For rewriting (5.9) we use the following lemma, which is of the same form as Lemma 2.4.8 used for proving strong duality for semidefinite programs:

Lemma 5.5.6 ([BTN01, Lemma 3.4.2])**.** *A triple (x, f, τ) with $x \geq 0$ belongs to the set $\mathcal{C}^\circ$, i.e., $\mathsf{Compl}_f(x) < \tau$ if and only if there exists a vector λ satisfying the relations*

$$A^T\lambda = 0, \quad b^T\lambda > 0, \quad \lambda \geq_{\boldsymbol{L}^k} 0. \tag{5.10}$$

With Definition 5.5.3 we can rewrite (5.9) in conic quadratic form

$$Au - b = \begin{pmatrix} B(x)u \\ -f^T u \\ f^T u \end{pmatrix} - \begin{pmatrix} 0 \\ -1-\tau \\ -1+\tau \end{pmatrix} \geq_{\mathbf{L}^k} 0.$$

Observe that

$$\begin{aligned} A^T &= \left(\sqrt{2x_1}B_1; \ldots; \sqrt{2x_n}B_n; -f; f\right) \\ b &= \left(0; \ldots; 0; -1-\tau; -1+\tau\right). \end{aligned}$$

Then partition $\lambda^T = (w_1; \ldots; w_n; p; q)$ accordingly and rewrite (5.10):

$$\begin{aligned} \sum_{e=1}^n \sqrt{2x_e}B_e w_e &= (p-q)f, \\ p(-1-\tau) + q(-1+\tau) &> 0, \\ \sqrt{\left(\sum_{e=1}^n w_e^T w_e\right) + p^2} &\leq q. \end{aligned} \tag{5.11}$$

Note that in (5.11) every solution satisfies $p \neq q$. So we can finally define new variables $s_e = -(q-p)^{-1}\sqrt{2x_e}w_e$ and (5.11) becomes

$$\begin{aligned} &\sum_{e=1}^n B_e s_e = f \\ &\sum_{e=1}^n \frac{s_e^T s_e}{2x_e} \leq \frac{q+p}{q-p} < \tau. \end{aligned}$$

This leads to the following lemma:

Lemma 5.5.7. *A triple (x, f, τ) belongs to the set $\mathcal{C}^\circ$ if and only if there exist vectors $s_e, e = 1, \dots, n$, satisfying the relations:*

$$\begin{aligned} \sum_{e=1}^{n} B_e s_e &= f, \\ \sum_{e=1}^{n} \frac{s_e^T s_e}{2x_e} &< \tau. \end{aligned} \tag{5.12}$$

Note that all x_e are allowed to be zero, so we need to define $\frac{0}{0} := 0$ and for $a > 0$ we define $\frac{a}{0} := \infty$.

Proof. From Lemma 5.5.6 we know that if $\mathsf{Compl}_f(x) < \tau$ we can find a solution of (5.10) which can be converted to a solution of (5.12).

For proving the if part we take a solution s of (5.12), then we can find $q > \frac{1}{2}$ with

$$\sum_{e=1}^{n} \frac{s_e^T s_e}{2x_e} < 2q - 1 < \tau.$$

Now set $p = q - 1$ and $w_e = -(2x_e)^{-\frac{1}{2}} s_e$, this ensures that w_e is zero whenever x_e is. Using this we get a solution of (5.11), which is equivalent to (5.10). So we can again use Lemma 5.5.6 to obtain $\mathsf{Compl}_f(x) < \tau$. □

What remains to prove is that we can get similar results for the set $\mathcal{C}$ and the inequality $\mathsf{Compl}_f(x) \leq \tau$. Following [BTN01] we state the following proposition.

Proposition 5.5.8. *A triple (x, f, τ) belongs to the set $\mathcal{C}$ if and only if there exist vectors $s_e, e = 1, \dots, n$, satisfying the relations:*

$$\begin{aligned} \sum_{e=1}^{n} B_e s_e &= f, \\ \sum_{e=1}^{n} \frac{s_e^T s_e}{2x_e} &\leq \tau, \\ x &\geq 0. \end{aligned} \tag{5.13}$$

Proof. For proving both directions we use Lemma 5.5.7. If we have a triple (x, f, τ) satisfying (5.13), then for $\tau' > \tau$ we get $\mathsf{Compl}_f(x) < \tau'$ (using Lemma 5.5.7) and therefore $\mathsf{Compl}_f(x) \leq \tau$. For the reverse direction we assume $\mathsf{Compl}_f(x) \leq \tau$. Then we also have $\mathsf{Compl}_f(x) < \tau + 1$. From Lemma 5.5.7 we know that the following optimization problem has a feasible solution:

$$\begin{aligned} \min \quad & \sum_{e=1}^{n} \frac{s_e^T s_e}{2x_e} \\ \text{s.t.} \quad & \sum_{e=1}^{n} B_e s_e = f. \end{aligned}$$

Note that if the variables x_e, representing the existence of bar e, are zero the corresponding variables s_e are also zero. This is why we can just ignore these terms. As this problem is now a convex quadratic optimization problem, with nonnegative objective over an affine plane it is solvable.

Assume that the optimal solution of this problem is $\bar{s}$ with objective value $\bar{\tau}$. Using Lemma 5.5.7 again for $\tau > \mathsf{Compl}_f(x)$ we have

$$\sum_{e=1}^{n} \frac{\bar{s}_e^T \bar{s}_e}{2x_e} = \bar{\tau} < \tau.$$

So if $\tau \geq \mathsf{Compl}_f(x)$ it must be

$$\sum_{e=1}^{n} \frac{\bar{s}_e^T \bar{s}_e}{2x_e} \leq \tau$$

as $\bar{s}$ was the solution of (5.13). □

We proved that our problem can be represented as conic quadratic optimization problem, as it is a sum of fractional-quadratic functions. Minimizing the compliance now corresponds to minimizing $\sum_e \frac{s_e^2}{x_e}$. For being able to handle this objective we add new variables σ_e satisfying

$$s_e^2 \leq 2x_e\sigma_e.$$

This leads to the following quadratic model, where $V_{\max}$ still is the maximum volume allowed for the truss and l_e is the length of bar e:

$$\begin{array}{lll} \min & \tau & \\ \text{s.t.} & \displaystyle\sum_{e=1}^{n} s_e B_e = f & \\ & s_e^2 \leq 2x_e\sigma_e & e = 1, \ldots, n \\ & \displaystyle\sum_{e=1}^{n} \sigma_e \leq \tau & \\ & \displaystyle\sum_{e=1}^{n} l_e x_e \leq V_{\max} & \\ & s_e, \sigma_e, \tau \in \mathbb{R} & e = 1, \ldots, n \\ & x_e, \sigma_e \geq 0 & e = 1, \ldots, n. \end{array}$$

From a quadratic problem to a second order cone problem

It is possible to reformulate the two inequalities

$$s_e^2 \leq 2x_e\sigma_e \text{ and } \sum_{e=1}^{n} \sigma_e \leq \tau$$

as a second order cone using

$$\left\| \frac{s_e}{\sqrt{2x_e}} \right\|^2 \leq \tau$$

and

$$\sigma_e x_e = \frac{1}{4}((\underbrace{\sigma_e + x_e}_{\gamma_e})^2 - (\underbrace{\sigma_e - x_i}_{\delta_e})^2)$$

This leads to the following SOCP constraint:

$$s_e^2 + \frac{1}{4}\delta_e^2 \leq \frac{1}{4}\gamma_e^2.$$

Remark 5.5.9. *The quadratic formulation is some kind of dual reformulation of the problem, therefore extending this model is not as easy as for the two other types of reformulations.*

One problem caused by the dualization is that the physical meaning of the different variables of the model is not always clear. Unchanged is the interpretation of the variables x_e, they represent the different bars and their cross-sectional areas. Additionally, $B_e s_e$ are the reaction forces in bar e caused by the external load f. Due to this interpretation extending the quadratic model is not easy.

Another problem is that not all extensions are possible because Lemma 5.5.6 might not hold in this setting. So for examples general bounds on the s-variables are not allowed.

Remark 5.5.10. *Some of the extensions presented above are also possible in the quadratic model. Multiple loads are no problem, this was already shown in [BTN01]. Additional linear constraints on additional variables or on the x variables are also possible. One such constraint is for example the constraint for the volume bound. As shown by [Sch11] the reformulation is still equivalent if the variables x_e are binary.*

5.6 Stability constraints

By now we have seen three different models, different in the class they belong to and different in the way to handle the nonconvexity of the original formulation. These models are already extended to discrete cross-sectional areas and multiple loads.

Considering uncertain loads is one way to influence the solving process towards a more stable truss. Positioning of active bars in a truss that can produce forces and therefore control uncertainty on their own, is another way. In this section we extend the models to these active bars. Furthermore, we consider different techniques for controlling the stability. We present constraints for stress bounds, local buckling, and vibrations.

5.6.1. Reacting on uncertain loads

Discrete bar areas are one extension for which binary variables are needed. Another one is the positioning of special bars. We want to integrate active components, so called actuators, into the truss.

These special bars are developed within the Collaborative Research Center 805 at TU Darmstadt and they help controlling the uncertainty in the loads. There are different types of actuators, in this section only a specific kind is considered. These active bars can produce forces in the nodes by lengthening and shortening themselves. As the truss model used within this thesis considers exactly axial loads, the actuators can be easily included. The stability of a truss is changed using artificial displacements of the free nodes and working against the displacements caused by the external load. For example the active bars can be realized by putting piezo-ceramics on or in the bars.

Using actuators the uncertainty in the load scenarios of the truss can be controlled in the following way. Usage of the actuators can be chosen independently for every scenario, so they can react on the loads of the different scenarios in a different way. The question is, in which bar should the actuators be positioned such that their impact for all scenarios is maximized.

The modeling is done using the force they can raise to support the system. Hence the equilibrium equation (5.2) changes to

$$A(x)u_i + f_i^{\text{act}} = f_i \qquad \text{for } i = 1, \ldots, s. \tag{5.14}$$

The new forces f_i^{act} are given by:

$$f_{v,i}^{\text{act}} = \sum_{w \in \delta(v)} z_{(v,w),i} \frac{v - w}{\|v - w\|} \qquad \text{for } v \in V_f \text{ and } i \in S,$$

where $z_{(v,w),i}$ is the usage of the actuator in bar (v, w) for scenario i. This usage is bounded by some maximum value the actuators are able to produce. It can be positive as well as negative or zero.
Thereby the three models need to be adapted.

1. In the SDP we define for all $i = 1, \ldots, s$ a new load vector $\hat{f}_i$ with

   $$A(x)u_i + f_i^{\text{act}} = f_i \quad \Longleftrightarrow \quad A(x)u_i = f_i - f_i^{\text{act}} =: \hat{f}_i.$$

   Using this $\hat{f}$ in the proof of Lemma 5.3.1 does not change anything and as stated in Observation 5.3.4 additional constraints can also be added. The extended SDP with additional binary variables y_e arranging the positioning of the actuators and some new constraints looks as follows:

$$
\begin{aligned}
\min \quad & \hat{\tau} \\
\text{s.t.} \quad & \begin{pmatrix} 2\tau_i & f_i^T - f_i^{\text{act}^T} \\ f_i - f_i^{\text{act}} & A(x) \end{pmatrix} \succeq 0, && i = 1, \ldots, s \\
& \sum_{e=1}^{n} l_e \sum_{a=1}^{m} \beta^a x_e^a \leq V_{\max} \\
& \sum_{a=1}^{m} x_e^a \leq 1 && e = 1, \ldots, n \\
& \tau_i \leq \hat{\tau} && i = 1, \ldots, s \\
& \sum_{w \in \delta(v)} z_{(v,w),i} (v - w) \left\| v - w \right\|^{-1} = f_{v,i}^{\text{act}} && v \in V_f, \quad i = 1, \ldots, s \\
& |z_{ei}| \leq y_e f_{\max} && e = 1, \ldots, n, \quad i = 1, \ldots, s \\
& \sum_{e=1}^{n} y_e \leq y_{\max} \\
& y_e \leq \sum_{a=1}^{m} x_e^a && e = 1, \ldots, n \\
& z_i \in \mathbb{R}^n && i = 1, \ldots, s \\
& \hat{\tau} \in \mathbb{R} \\
& x^a \in \{0,1\}^n && a = 1, \ldots, m \\
& y \in \{0,1\}^n.
\end{aligned}
$$

The added constraints model the forces which the actuators are able to produce. Moreover they ensure that bars can only be active if their area is positive. Additionally, there is an upper bound on the number of actuators that can be positioned. Otherwise all bars would be active because in this model costs for actuators are not considered.

2. Also the MIP can be extended to actuators, here the multiple loads case with discrete cross-sectional bar areas is presented.

   The reformulation into a MIP is based on (5.5), in this equation the forces do not appear. Therefore reformulating the new equilibrium equation (5.14) can be done exactly in the same way as before. Even the reformulation of the objective function stays the same because if we use $f - f^{\text{act}} =: \hat{f}$, then

$$(f - f^{\text{act}})^T u = u^T A(x) u = \sum_{e=1}^{n} \frac{q_e^2}{\kappa}.$$

In the resulting model there are additional binary variables y_e for positioning the actuators in the bars, z_e for the usage of the actuators, and the same constraints as stated above:

$$
\begin{array}{lrll}
\min \tau & & & \\
\text{s.t.} & \displaystyle\sum_{e=1}^{n}\sum_{a=1}^{m} \beta^a b_e^T q_{ei}^a & = f_i^T - f_i^{\text{act}T} & i = 1, \dots, s \\
& \dfrac{q_i^T q_i}{\kappa} & \leq \tau & i = 1, \dots, s \\
& x_e^a c_e^{\max} & \geq q_{ei}^a & e = 1, \dots, n, i = 1, \dots, s, a = 1, \dots, m \\
& x_e^a c_e^{\min} & \leq q_{ei}^a & e = 1, \dots, n, i = 1, \dots, s, a = 1, \dots, m \\
& (1 - x_e^a) c_e^{\min} & \leq \beta^a b_e^T u_i - q_{ei}^a & e = 1, \dots, n, i = 1, \dots, s, a = 1, \dots, m \\
& (1 - x_e^a) c_e^{\max} & \geq \beta^a b_e^T u_i - q_{ei}^a & e = 1, \dots, n, i = 1, \dots, s, a = 1, \dots, m \\
& \displaystyle\sum_{e=1}^{n} l_e \sum_{a=1}^{m} \beta^a x_e^a & \leq V_{\max} & \\
& \displaystyle\sum_{a=1}^{m} x_e^a & \leq 1 & e = 1, \dots, n \\
& \displaystyle\sum_{w \in \delta(v)} z_{(v,w),i} \frac{v - w}{\|v - w\|} & = f_{v,i}^{\text{act}} & v \in V_f, i = 1, \dots, s \\
& |z_{ei}| & \leq y_e f_{\max} & e = 1, \dots, n, i = 1, \dots, s \\
& \displaystyle\sum_{e=1}^{n} y_e & \leq y_{\max} & \\
& y_e & \leq \displaystyle\sum_{a=1}^{m} x_e^a & e = 1, \dots, n \\
& u_i & \in [u^{\min}, u^{\max}]^d & i = 1, \dots, s \\
& z_i & \in \mathbb{R}^n & i = 1, \dots, s \\
& x^a & \in \{0, 1\}^n & a = 1, \dots, m \\
& y & \in \{0, 1\}^n. &
\end{array}
$$

3. Extending the quadratic model to active components is also possible. First of all adding actuators changes the objective function in the original problem (5.1). We can again use the transformation $\hat{f} := f - f^{\text{act}}$ and the whole reformulation stays the same. Adding further variables y_e and z_e for positioning the actuators can also be done without affecting the reformulation. This leads to the following model with multiple loads and bar areas equal to zero or one:

$$
\begin{array}{lrll}
\min & \tau & & \\
\text{s.t.} & \displaystyle\sum_{e=1}^{n} s_{ei} B_e = f_i - f_i^{\text{act}} & & i = 1, \ldots, s \\
& s_{ei}^2 \leq 2 x_e \sigma_{ei} & & e = 1, \ldots, n, \quad i = 1, \ldots, s \\
& \displaystyle\sum_{e=1}^{n} \sigma_{ei} \leq \tau & & i = 1, \ldots, s \\
& \displaystyle\sum_{e=1}^{n} l_e x_e \leq V_{\max} & & \\
& s_e, \sigma_{ei}, \tau \in \mathbb{R} & & \\
& \sigma_e \geq 0 & & e = 1, \ldots, n \\
& x \in \{0,1\}^n & & \\
& \displaystyle\sum_{e=1}^{n} l_e x_e \leq V_{\max} & & \\
& \displaystyle\sum_{w \in \delta(v)} z_{(v,w),i} \frac{v-w}{\|v-w\|} = f_{v,i}^{\text{act}} & & v \in V_f, \quad i = 1, \ldots, s \\
& |z_{ei}| \leq y_e f_{\max} & & e = 1, \ldots, n, \quad i = 1, \ldots, s \\
& \displaystyle\sum_{e=1}^{n} y_e \leq y_{\max} & & \\
& y_e \leq x_e & & e = 1, \ldots, n \\
& u_i \in [u^{\min}, u^{\max}]^d & & i = 1, \ldots, s \\
& z_i \in \mathbb{R}^n & & i = 1, \ldots, s \\
& y \in \{0,1\}^n. & &
\end{array}
$$

Definition 5.6.1. If we allow actuators to be positioned in a truss, the truss is called *active.* A truss without actuators is called *passive.* □

As the size of these three new models suggest, they are not easier to solve than those without active components. We have $2ns$ additional variables and $n(s+1)+1$ additional constraints. Due to further dependencies the model becomes more complicated, but as we will see in the following the trouble is worth it. The following example shows the influence of actuators to the topology. The topology changes and the objective value may be improved.

Example 5.6.2. *Look at the ground structure of Figure 5.1, now we apply an additional load in a node in the middle and allow two different bar areas. There are three actuators that can be positioned.*

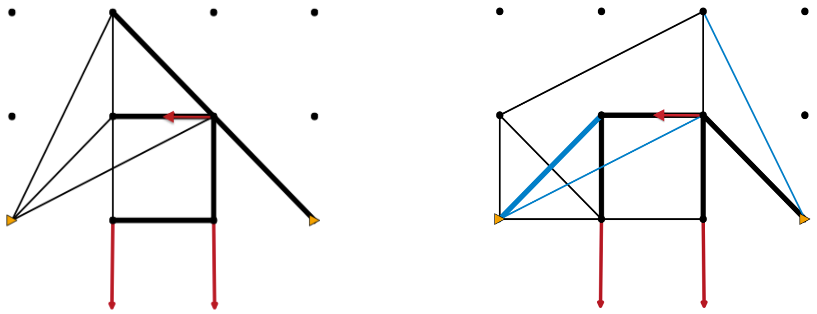


Figure 5.6. – Optimized trusses without (left) and with actuators (right).

The left picture of Figure 5.6 shows the problem without actuators. In the right picture three actuators are positioned (blue bars) and the topology changed. The objective value improved from 0.982 *in the passive case to* 0.908 *for the active truss. Moreover, we can conclude that optimizing the topology first and then positioning the actuators is not as good as doing both at once. The objective value for this procedure in this example is* 0.944 *and therefore worse than the objective for the integrated model.*

For uncertain loads actuators are able to reduce the compliance and therefore they provide a good instrument for controlling uncertainty. From a computational point of view, actuators make the problems more difficult to solve because they add variable upper bounds to the models, which is known to make solving hard. Additionally, the special structure causes numerical trouble in MIP and SDP solvers. Computational results suggest that for example some SDP solvers need special parameter tuning for being able to deal with these constraints induced from active components. More examples and results for active trusses will be presented in Chapter 6.

5.6.2. Controlling vibrations

Another possible extension for stabilizing the truss are vibration constraints. The idea is to control the eigenfrequencies of the structure, thereby controlling the vibrations. This is important for various applications and it can be done using the eigenvalues of a special constraint matrix presented in the following. So it is easy to implement the new constraints in a semidefinite framework. Moreover, piezo-ceramics are used for damping vibrations.

We are going to introduce this topic as presented in [Sti06]. Before we are able to define the problem we have to introduce the element mass matrices M_e and the mass matrix $M(x)$. The element-mass matrices M_e can be obtained in a similar way as A_e using the finite-element-method. So for every bar the mass is calculated and transformed into global coordinates. This is done by the following transformation with α being the angle of e relative to the global coordinate system:

$$M_e = \rho l_e \begin{pmatrix} \cos\alpha & 0 \\ \sin\alpha & 0 \\ 0 & \cos\alpha \\ 0 & \sin\alpha \end{pmatrix} \begin{pmatrix} \frac{1}{3} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{3} \end{pmatrix} \begin{pmatrix} \cos\alpha & \sin\alpha & 0 & 0 \\ 0 & 0 & \cos\alpha & \sin\alpha \end{pmatrix},$$

where l_e is the length of bar e and ρ is the density of the material used. A detailed description of this procedure can be found in [Ste12].

Definition 5.6.3. We call

$$M(x) = M_0 + \sum_{e=1}^{n} x_e M_e$$

the *mass matrix* of a truss. M_0 is the constant mass in the structure and M_e are the element mass matrices for every bar e. □

For controlling vibrations of a structure the lowest eigenfrequency must be considered and a generalized eigenvalue problem of the form

$$A(x) - \lambda M(x) \succeq 0$$

must be solved.

Remark 5.6.4. *The element stiffness matrices A_e can be produced in the same way as the element mass matrices:*

$$A_e = \kappa \begin{pmatrix} \cos\alpha & 0 \\ \sin\alpha & 0 \\ 0 & \cos\alpha \\ 0 & \sin\alpha \end{pmatrix} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} \cos\alpha & \sin\alpha & 0 & 0 \\ 0 & 0 & \cos\alpha & \sin\alpha \end{pmatrix}.$$

Observation 5.6.5. *The mass matrix $M(x)$ has the same dimension as the stiffness matrix $A(x)$ and the same nonzero pattern.*

The problem of controlling the vibrations in a structure can be formulated as a maximization problem over the smallest eigenfrequency. This eigenfrequency can be represented using the minimum eigenvalue $\lambda = \lambda_{\min}$. As shown in Section 2.3.2 this problem can be formulated as SDP:

$$\begin{array}{ll} \max & \lambda \\ \text{s.t.} & A(x) - \lambda M(x) \succeq 0 \\ & x_e \geq 0 \qquad e = 1, \ldots, n. \end{array}$$

However, in the context of the models presented above, where an external load applies, both the compliance and the vibrations should be considered. As we want to consider both at the same time, we first need to choose a lower bound $\bar{\lambda}$ for the minimum eigenfrequency. Using this lower bound we can formulate the problem as linear SDP:

$$\begin{aligned}
\min \quad & \tau \\
\text{s.t.} \quad & \begin{pmatrix} 2\tau & f^T \\ f_i & A(x) \end{pmatrix} \succeq 0 \\
& A(x) - \bar{\lambda} M(x) \succeq 0 \\
& \sum_e x_e \leq V_{\max} \\
& \tau \in \mathbb{R} \\
& x_e \geq 0 \qquad e = 1, \dots, n.
\end{aligned}$$

So including vibration constraints into (5.3) can easily be done. Furthermore, extending this model to discrete bar areas and actuators or formulating it as a volume minimization program is done analogously. We are not going to present these models here, they are just a combination of the models already presented.

Remark 5.6.6. *Controlling vibrations using the MIP or the SOCP model is not possible. As shown above the problem of controlling vibrations yields an SDP in a natural way. A semidefinite constraint must be added and would change the type of the model into an SDP, so all advantages of the other models would be lost. Controlling vibrations can only be done in the context of a semidefinite problem.*

Computational results for vibrations constraints are presented in Section 6.7.3

5.6.3. Considering the stress

The compliance is a measure for the global stability of the whole truss, the local displacement is not taken into account. So for example it is possible that there is a large amount of internal forces in one bar and almost nothing in all other bars. From an application point of view this might not be realistic. Rather bounding the forces for every bar should be in the focus and there are various ways to do this. We present two different kinds of constraints in the context of stress: explicit bounds and local buckling. First of all we define stress in our context of Truss Topology Design.

Definition 5.6.7. We call the force q_e acting on bar e related to its cross-sectional area a_e the *stress* of bar e. This stress can be expressed as the following quotient:

$$\text{stress}_e := \frac{q_e}{a_e} = \frac{\kappa a_e b_e^T u}{l_e a_e} = \frac{\kappa b_e^T u}{l_e},$$

where l_e is the length of bar e and x, b, u are defined as in (5.1). □

This definition might look strange, but now we can use the variables q_e already presented in the MIP in Section 5.4 for modeling stress constraints. This definition is also in accordance with [BS04].

Stress-bounds

A very easy way for considering the stress of specific bars is to give explicit bounds $\overline{\sigma}$ and $\underline{\sigma}$ on the stress. This bounds can be expressed using the stress or the force variables. The constraints we need to add to the model are:

$$\begin{array}{rcccc} & \underline{\sigma} & \leq & \text{stress}_e & \leq & \overline{\sigma} \\ \Longleftrightarrow & \underline{\sigma} a_e & \leq & q_e & \leq & \overline{\sigma} a_e. \end{array}$$

The first line of inequalities is in terms of stress, whereas the second one is in terms of forces. These bounds are just linear inequalities and can easily be added to the MIP (5.7) because it already has variables q_e. For including bounds like this to the SDP or the SOCP, the variables q_e or stress_e need to be added to the respective model. This problem occurs because both models completely reformulate the truss problem due to the nonconvexity induced through the displacement variables:

- In the MIP the nonconvexity is handled using a big-m formulation and therefore the needed variables still exist. The model is stated below together with the constraints for avoiding local buckling.

- In the SDP there are no variables left except bar areas and the compliance, hence the stress for every bar cannot be expressed directly. This is why we need to introduce new variables q_e for the forces and u for the displacements of the nodes. Introducing them would destroy all the general advantages of the SDP: the small number of variables.

- In the SOCP there are variables s_e that are some kind of dual variables due to Lemma 5.5.6. Using this, bounding the s-variables is not allowed and will cause a duality gap. Again we have to introduce new variables for this model as above for the SDP.

Local buckling

The situation gets even more complicated if we consider buckling constraints which are also constraints on the variables q_e. So the difficulty for the SDP and the SOCP stays the same as stated above: a bunch of new variables must be introduced and all advantages of these models will be lost. Therefore we will now focus on the MIP and introduce some additional definitions. While stress-bounds are fixed numbers, local buckling constraints depend on the cross-sectional area of the bar.

Definition 5.6.8. For the volume v_e and the cross-sectional area a_e the *critical buckling stress* of a bar e is defined as

$$\begin{aligned}\sigma_e^{\text{crit}} = \frac{F_{\text{crit}}}{a_e} &= \frac{\kappa\pi^2}{l_e^2}\frac{I}{a_e} \\ &= \frac{\kappa\pi^2}{l_e^2}\frac{\pi r_e^4}{4a_e} \\ &= \frac{\kappa\pi}{4l_e^2}\frac{\pi^2 r_e^4}{a_e} \\ &= \frac{\kappa\pi}{4l_e^2}\frac{(\pi r_e^2)^2}{a_e} \\ &= \frac{\kappa\pi}{4l_e^2}a_e \\ &= \frac{\kappa\pi v_e}{4l_e^3},\end{aligned}$$

where F_{crit} is the the critical buckling load, I is the geometrical moment of inertia and r_e is the radius of the bar e. □

Again there are two inequalities for stating that a bar should not exceed its critical buckling load, one in terms of forces

$$q_e \le \sigma_e^{\text{crit}} a_e = \frac{\kappa\pi}{4l_e^2}a_e^2,$$

and one in terms of stress

$$\text{stress}_e \le \sigma_e^{\text{crit}} = \frac{\kappa\pi}{4l_e^2}a_e.$$

The first inequality in terms of forces is nonconvex, as it is exactly the wrong inequality for a quadratic function. Whereas the second one is linear and therefore convex. As mentioned in Remark 5.4.2 in the MIP the variables q_e have the same value as stress_e. Therefore we can use q_e for these constraints in terms of stress and get convex inequalities which can easily be added to the MIP:

$$q_e \le \frac{\kappa\pi}{4l_e^2}a_e.$$

For the sake of completeness, we present the corresponding MIP. This model includes discrete cross-sectional areas under multiple loads including actuators, stress, and buckling constraints. We will not present computational results for this model because already the model without stress and buckling constraints, and even without actuators caused numerical troubles in CPLEX. Due to the big-m constraints there are no chances of success for this huge model. Therefore we restrict ourselves to the basic MIP and the SDP with all extensions.

$$
\begin{array}{lrll}
\min & \tau & & \\
\text{s.t.} & \displaystyle\sum_{e=1}^{n}\sum_{a=1}^{m}\beta^a b_e^T q_{ei}^a & = f_i^T - f_i^{\text{act}T} & i = 1,\dots,s \\
& \displaystyle\frac{q_i^T q_i}{\kappa} & \leq \tau & i = 1,\dots,s \\
& x_e^a c_e^{\max} & \geq q_{ei}^a & e = 1,\dots,n, i = 1,\dots,s, a = 1,\dots,m \\
& x_e^a c_e^{\min} & \leq q_{ei}^a & e = 1,\dots,n, i = 1,\dots,s, a = 1,\dots,m \\
& (1 - x_e^a)c_e^{\min} & \leq \beta^a b_e^T u_i - q_i^a & e = 1,\dots,n, i = 1,\dots,s, a = 1,\dots,m \\
& (1 - x_e^a)c_e^{\max} & \geq \beta^a b_e^T u_i - q_i^a & e = 1,\dots,n, i = 1,\dots,s, a = 1,\dots,m \\
& \displaystyle\sum_{e=1}^{n} l_e \sum_{a=1}^{m}\beta^a x_e^a & \leq V_{\max} & \\
& \displaystyle\sum_{a=1}^{m} x_e^a & \leq 1 & e = 1,\dots,n \\
& \displaystyle\sum_{w\in\delta(v)} z_{(v,w),i}\frac{v-w}{\|v-w\|} & = f_{v,i}^{\text{act}} & v \in V_f, i = 1,\dots,s \\
& |z_{ei}| & \leq y_e f_{\max} & e = 1,\dots,n, i = 1,\dots,s \\
& \displaystyle\sum_{e=1}^{n} y_e & \leq y_{\max} & \\
& q_{ei} & \leq \displaystyle\frac{\kappa\pi}{4l_e^2}x_e & e = 1,\dots,n, i = 1,\dots,s \\
& q_{ei} & \in [\underline{\sigma},\overline{\sigma}]^s & e = 1,\dots,n \\
& u_i & \in [u^{\min}, u^{\max}]^d & i = 1,\dots,s \\
& x^a & \in \{0,1\}^n & a = 1,\dots,m \\
& y & \in \{0,1\}^n & \\
& \tau & \in \mathbb{R}. &
\end{array}
$$

5.7 Further ideas

Is graph theory useful?

When looking at trusses and having a discrete optimization background there is one idea coming into mind:

> Is this truss not just a graph with some extra properties?

If it were so easy, there would be many techniques from for example network flows that could be used for solving. The answer to the question is not easy. In some sense concerning for example connectivity a truss can be seen as a network and constraints for ensuring the connectivity can be used. Certainly graph theory is not enough to describe a truss completely. The physics in that problem can for example not be put into some flow constraint. We describe the problem in the following.

Definition 5.7.1. Let $G(V, E)$ be the ground structure of a truss with nodes V and potential bars E. Then we could interpret this ground structure as graph with edges E and vertices V. □

Using this definition we can think of a Truss Topology Design problem as a network flow problem in the following way. In this graph there are some sinks, the fixed nodes of the truss ground structure and some sources, the free nodes of the truss, where the loads are applied. The external loads are the flows that need to be transported by the network, designed by the graph. There are positive weights or costs on every edge. The aim is to route all the flow through the network by only using a specific number of edges and minimizing the costs.

This model does not suffice to describe the whole truss problem. One reason for this is that it is important how the bars of the truss lie in space relative to each other because the flow can only go through edges having the right direction.

What we know by now is that these ideas from network flows are not enough to describe the complete problem, but they give good hints for feasible solutions. So our idea is to use them as heuristics. As one intention of this thesis is solving truss problems, we take a closer look on the special structure of trusses especially in their SDP formulation and use it for speeding up the solving process. The ideas presented in the following are not implemented for the general framework for solving MISDPs introduced in Chapter 4 because they are not helpful for general SDPs.

Analyzing the solving process of TTD problems in the minimum compliance formulation shows that finding feasible solutions is very hard. In most of the cases it took a very long time, no standard heuristic is able to find anything. Therefore we decided to look for feasible solutions on our own and hand them over as starting solution.

A very simple idea is to look for the shortest path from a node where the load is applied to a fixed node. We do this for all nodes with loads and all fixed nodes and every time a shortest path was found, we decrease the weights of the edges of this path and search for other shortest paths on the modified graph. This method does not guarantee a feasible solution, but it is a very quick algorithm. If the solution we obtain is not feasible, we fix the solution by adding more edges. Adding more edges can be done using some facts we know about the problem. So for example it is well-known that triangle structures within trusses are very good. This is why we try to add some more edges to our network which produce triangles. The algorithm is presented in the following:

Algorithm 5.7.2. *Shortest-path-heuristic for Truss Topology Design.*
Input: *A graph $G(V, E)$ with a set of sinks $W \subset V$, a set of sources $U \subset V$, where $U \cap W = \emptyset$ and some flow f_u for $u \in U$.*

1. *Define edge-weights $c_e = |U| + |W|$ and set $P = \emptyset$.*
2. *Take $u \in U$ and $w \in W$.*
3. *Compute a shortest path T from u to w and save the edges of T in the set P.*
4. *Reduce the weights on the edges in the path T: $c_t := c_t - 1$ for $t \in T$.*
5. *Set $U := U \setminus \{u\}$ and $W := W \setminus \{w\}$:*
   - *if $U \neq \emptyset$ and $W \neq \emptyset$ take $u \in U$ and $w \in W$ and start again from step 3,*
   - *if one of the sets is empty put the current element back to that set, start again from step 3,*
   - *if both sets are empty go to step 6*
6. *Take all edges saved in P and check the feasiblity. If the solution is feasible* ***stop****, otherwise repair the solution by adding some more edges to get more triangles.*

Additionally, from a graph point of view it is possible to detect infeasible variable fixings within a truss. Here we can use for example connectivity and we can cluster nodes of our truss into components. We then require that at least one of the bars connecting the different components must have a positive bar area. So for example there needs to be a connection between the nodes where the loads are applied and the fixed nodes of the truss.

Detecting such structures quickly and adding appropriate cuts within the branch-and-bound algorithm will be part of the future work.

CHAPTER 6

Computational results in Truss Topology Design

In Chapter 5, we presented various models and extensions to the classical problem of Truss Topology Design. Now we use the techniques introduced in Chapter 3 to solve these models. Also we compare the MISDP with the MIP formulation of the truss. Additionally, we show the performance of the different solving strategies introduced in Chapter 3.

We used SCIP 3.0 and CPLEX 12.4.0.1 and a time limit of two hours. All computational results presented within this thesis are done using a Sun Fire X4600 M2 with 8x8384-Opteron 2.7 Ghz processors. This machine has 32 cores and 320 GB of RAM. As the cores on a single processor share their memory, the solving processes and especially the time required for solving one instance can vary from one run to another. Therefore we made five runs of very instance and show the arithmetic means of these runs within this chapter. In the last section of this chapter we present some statistics on the different solving runs. At that point we show that the solving time can differ by about 30 percent. We say an instance is solved if at least in one of the five runs the instances was solved to optimality. All times presented in this chapter are in seconds.

We start with a short description of our test set in Section 6.1. In Section 6.2 we validate the SDP model using instances from the literature. The introduced features of our software are tested in Section 6.3. In the following chapter we introduce another class of problems that can be formulated as MISDP and can be solved using our software. Solving these types of problems behaved completely different from solving truss problems. Therefore we comment on the solving behavior of trusses in Section 6.4.

As one topic of this thesis are trusses, we also consider the different models we presented before. We introduced continuous and discrete bar areas, we stated a model for minimizing the volume and for minimizing the compliance and we showed that discrete bar areas can be modeled using integer or binary variables. These formulations produce different results, we comment on that fact in Section 6.5. Trusses can be modeled as MISDPs and as linear MIPS, which of the two models can be solved faster we discuss in Section 6.6.

In Section 6.7 we show the solutions of trusses with different stability constraints. As already mentioned in Section 6.8 we discuss the variation of the different solving runs.

6.1 The test set

Within this chapter we use a test set consisting of 30 different instances. We created 22 of these instances, eight are already stated in the literature (see [Sto07] and [AS07]). From the literature we choose instances that were solved using discrete cross-sectional areas and which have the same kind of bars as our instances have. For example, in most of the cases we do not allow overlapping bars, i.e., bars that go over a node of the truss, but do not connect to that node. The 22 instances we created are eleven cantilevers and eleven bridge structures. For all these instances we created the different models, presented in the previous chapter and solved them with a time limit of two hours. Note that we always use a elasticity modulus $\kappa = 1$.

name	fixed nodes	free nodes	potential bars	compliance bound	volume bound	# different bar areas	# actuators
bridge-1	2	10	49	1.45	20.00	1	3
bridge-2	2	10	49	1.10	20.00	2	3
bridge-3	2	10	49	0.81	30.00	2	3
bridge-4	2	10	49	1.44	20.00	1	4
bridge-5	2	10	49	1.10	20.00	2	4
bridge-6	2	10	49	0.80	30.00	2	4
bridge-7	2	14	86	2.00	40.00	3	4
bridge-8	2	18	131	5.00	45.00	3	4
bridge-9	2	10	49	5.00	30.00	3	4
bridge-10	3	12	72	5.00	30.00	3	4
bridge-big	2	22	188	–	70.00	2	7
canti-1	3	9	47	0.26	17.00	1	3
canti-1-m	3	9	47	0.30	17.00	1	3
canti-2	3	9	47	0.22	17.00	2	3
canti-2-m	3	9	47	0.30	17.00	2	3
canti-3	3	9	47	0.15	25.00	2	3
canti-3-m	3	9	47	0.20	25.00	2	3
canti-4	3	9	47	0.25	17.00	1	4
canti-4-m	3	9	47	0.30	17.00	1	4
canti-5	3	9	47	0.21	17.00	2	4
canti-5-m	3	9	47	0.30	17.00	2	4
canti-6	3	9	47	0.14	25.00	2	4
canti-6-m	3	9	47	0.20	25.00	2	4
canti-7	4	12	83	3.00	35.00	3	4
canti-8	3	12	36	11.00	40.00	3	5
canti-9	2	14	86	3.00	30.00	3	4
canti-10	4	11	73	5.00	35.00	3	4

canti-big	4	16	128	3.00	40.00	3	4
lit-as-1	3	6	26	1.00	4.40	1	5
lit-as-2	3	42	630	26.00	40.00	1	5
lit-as-2-big	3	42	632	12.00	100.00	5	5
lit-as-5	2	23	299	5.00	7.00	1	5
lit-as-5-big	2	23	300	5.00	20.00	5	5
lit-as-6	3	17	129	27.00	40.00	1	5
lit-as-6-big	3	17	129	15.00	100.00	5	5
lit-s-1	2	13	104	6.00	7.00	1	5

Table 6.1. – Instance statistics for the Truss Topology Design test set.

Table 6.1 gives an overview about the different characteristics of the instances. We present the number of the free and the fixed nodes and the number of the potential bars. The table also shows the different bar areas allowed for the trusses and the maximum number of actuators we want to position when looking at actuators.

For all these instances it is possible to construct the different models we presented in the previous chapter and all combinations of them. First of all we can take a look at the truss problem with continuous bar areas (*cont*). Then the standard discrete model can be solved as minimum compliance and minimum volume (mV) problem. Additionally, the discrete bar areas can be modeled using integer variables (*int*) instead of binary ones for the minimum compliance and the minimum volume (mV-*int*) formulation. Furthermore, we can position actuators (*act*). This results in different models for each instance.

When looking at stability constraints we can also add more scenarios to the instances. For keeping the computation times small we only looked at two scenarios for all these kinds of problems. Altogether this yields 20 variations.

Moreover, we added two more variations for three scenarios. First we take a look at discrete bar areas and then we extend this model with actuators and the minimum volume formulation and if possible we model the different bar areas as integer variables.

So for each of the 22 instances we have up to 22 variations. Some of the problems only allow one bar area, for these instances modeling the bar areas using integer variables does not change anything, therefore these variations are not created. Additionally, we use some of the variations for the literature instances.

		variables			constraints		SDP
name	kind	bin.	int.	cont.	SDP	linear	blocksize
general	discrete bars	nm	0	1	1	$2n+2$	$d+1$
	minimize volume	nm	0	1	1	$2n+2$	$d+1$
	integer bar areas	0	n	1	1	$2n+1$	$d+1$
	with actuators	$nm+n$	0	$n+1$	1	$7n+3$	$d+1$
	2 scenarios	nm	0	3	2	$2n+4$	$d+1$
	3 scenarios	nm	0	4	3	$2n+5$	$d+1$

Table 6.2. – General model statistics for the MISDP formulation of the truss problems.

As the number of different problems is larger than 500 and the number of variables and constraints can be computed easily using Table 6.1 and Table 6.3 we do not show the statistics in detail. In Table 6.2 we present the general description of a truss with n bars, m different bar areas, and d as degree of freedom. We show how the different extensions effect the number of variables and constraints. Furthermore, we give some examples for some specific instances in Table 6.3.

		variables			constraints		SDP
name	kind	bin.	int.	cont.	SDP	linear	blocksize
bridge-1	discrete bars	49	0	1	1	5	21
	minimize volume	49	0	1	1	4	21
	with actuators	98	0	50	1	153	21
	2 scenarios	49	0	3	2	10	21
	3 scenarios	49	0	4	3	13	21
bridge-2	discrete bars	98	0	1	1	54	21
	minimize volume	98	0	1	1	53	21
	integer bar areas	0	49	1	1	5	21
	with actuators	147	0	50	1	202	21
	2 scenarios	98	0	3	2	59	21
	3 scenarios	98	0	4	3	62	21
bridge-8	discrete bars	393	0	1	1	138	37
	minimize volume	393	0	1	1	137	37
	integer bar areas	0	131	1	1	7	37
	with actuators	524	0	132	1	532	37
	2 scenarios	393	0	3	2	142	37
	3 scenarios	393	0	4	3	146	37
canti-5	discrete bars	94	0	1	1	50	19
	minimize volume	94	0	1	1	49	19
	integer bar areas	3	44	1	1	3	19
	with actuators	141	0	48	1	192	19
	2 scenarios	94	0	3	2	55	19
	3 scenarios	94	0	4	3	59	19
lit-as-5	discrete bars	299	0	1	1	2	47
	minimize volume	299	0	1	1	1	47
	with actuators	598	0	300	1	901	47
lit-s-1	discrete bars	104	0	1	1	2	27
	minimize volume	104	0	1	1	1	27
	with actuators	208	0	105	1	315	27

Table 6.3. – Exemplary model statistics for the MISDP formulation of the Truss Topology Design test set.

6.2 Validation

Most of the ideas presented in the previous chapter are new and there is no literature available, in particular there are no instances available to compare with. For the standard Truss Topology Design problem with discrete bar areas and without any other extensions there is a very small number of examples available for the minimum compliance formulation. We took one example from [Sto07] and four of the instances of [AS07] to validate our model and the solution approaches. From the instances presented in these two papers we could only use four for the validation because the others do not fit to our definitions of multiple loads or used a different set of potential bars. For these four instances we compared the solutions of the model with discrete cross-sectional areas and continuous areas if available. The results are shown in Table 6.4.

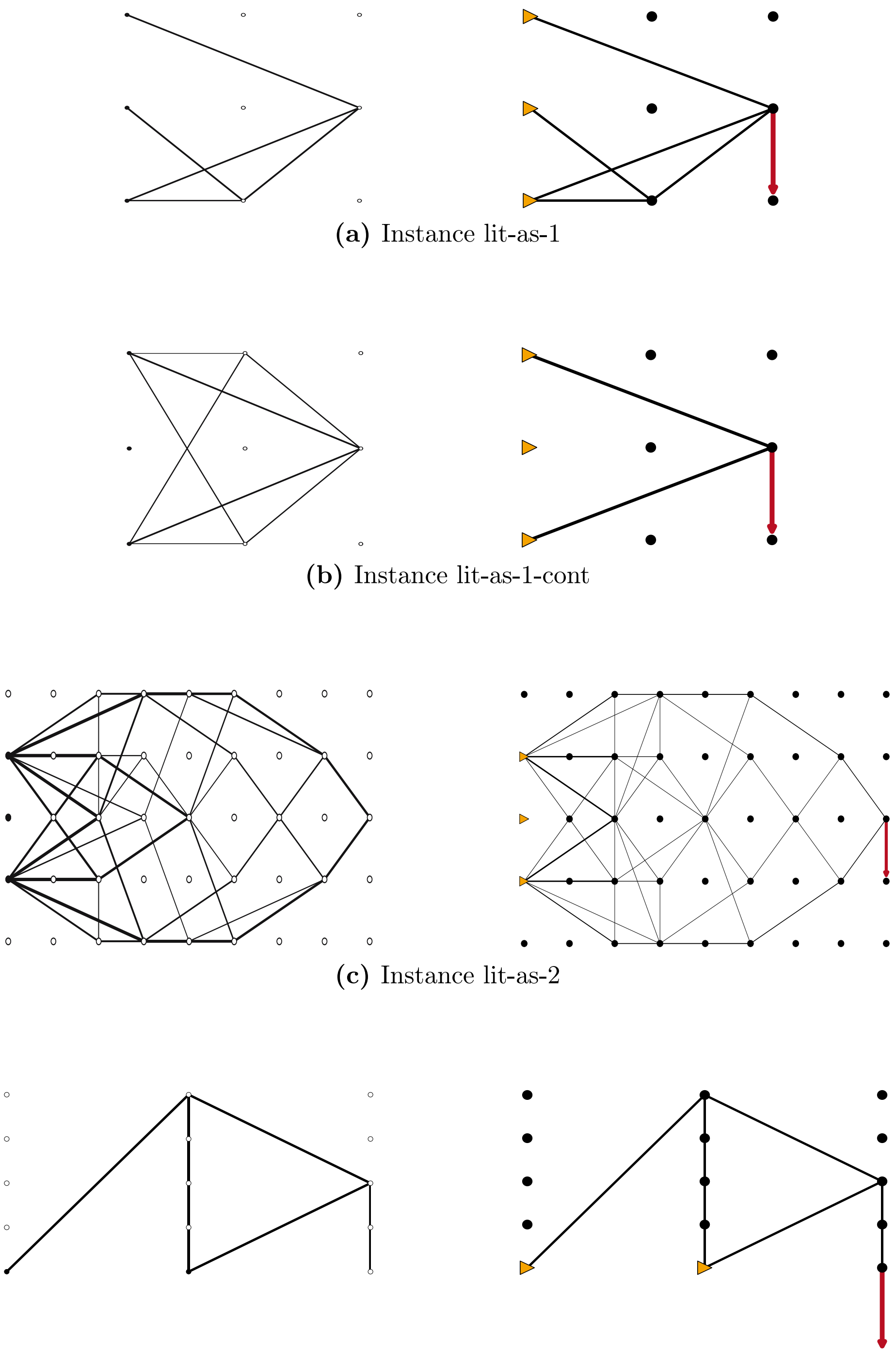
The instances are named in the following way: *lit* stands for literature, *as* for the instances from [AS07] and *s* for the instance from [Sto07]. *cont* means that this is the variant with continuous cross-sectional bar areas and the numbers are numbers of the instances in the appropriate papers.

	literature		MISDP		
name	solution	time	solution	time	status
lit-as-1	0.93	12	0.93	0.23	ok
lit-as-1-cont	0.84	–	0.71	0.03	?
lit-as-2	24.68	89124	–	> 540h	–
lit-as-2-cont	21.99	–	21.93	6.62	ok
lit-as-5	3.00	12184	3.00	86500.00	ok
lit-as-5-cont	2.65	–	2.65	1.03	ok
lit-s-1	3.48	73860	3.47	4991	ok

Table 6.4. – Validation of our MISDP model using some instances from the literature.

We are able to reproduce most of the results. In two cases our solution is smaller than the solution presented in the paper. This might be due to the fact that we used a completely different solving strategy, so these small discrepancies might be due to numerical issues. For one example we were not able to solve the instance to optimality within more than three weeks of computing.

Unfortunately, the instances do not have a unique optimal solution so comparing the solutions does not necessarily show that our algorithm is incorrect if the solutions differ. However, we want to show the solutions of our solver and the solutions from the papers in Figure 6.1. Especially for the instance *lit-as-1-cont* we show the solution which differs from the solution from the literature. The other solutions are identical or very similar with the same objective value.



(a) Instance lit-as-1

(b) Instance lit-as-1-cont

(c) Instance lit-as-2

(d) Instance lit-s-1

Figure 6.1. – Comparing solutions from literature (left) with our solutions (right).

6.3 MISDP Features

We introduced many ideas how to speed up the process of solving MISDPs within a branch-and-bound framework. Some of these ideas we evaluate now: the approximation procedure, the presolving ideas, and the heuristic. Moreover, we present a parameter test of smaller test sets and show results for the best parameter settings. In this section we only show summaries of the computations. The detailed results can be found in the Appendix A.

For all tests in the whole chapter the presolving ideas are used, the impact of these ideas are shown in Section 6.3.3. Furthermore, the heuristic is also used in most of the tests. If it is not used we comment on that.

6.3.1. Approximation

In Section 3.4 we introduced an approximation procedure that uses cuts generated from eigenvectors to approximate the semidefinite constraint. Now we want to use this idea within a branch-and-bound algorithm to compute the solution of a MISDP without solving an SDP relaxation. The results are summarized in Table 6.5. From the 526 instances the MISDP branch-and-bound could solve 398 which is more than the approximation procedure could solve. In return the approximation procedure is faster for 198 instances.

	solving as MISDP			solving with approximation		
time limit	# instances solved	smaller gap, if not solved	fastest solver	# instances solved	smaller gap, if not solved	fastest solver
2 hours	398	17	190	335	62	198

Table 6.5. – Comparison of the two different solving strategies for 526 truss instances.

Figure 6.2 shows the performance profiles for the two different strategies. These profiles also indicate that the approximation procedure is the faster approach but the MISDP is able to solve more instances. Hence it is not clear which strategy is the best. Note that the time in this figure is presented in logarithmic scale.

We want to mention that these results are completely different from those we present in Chapter 7 for the Maximum Cut Problem. For the Maximum Cut Problem, which is a combinatorial optimization problem, the approximation procedure is not able to solve even one of the instances within the time limit, so it is not comparable to the MISDP branch-and-bound. We comment on that fact later in Section 7.2.4.

In Table 6.6 we present some of the instances we solved with both methods. We compare them in terms of solving time, required nodes, and gap if not solved. Additionally, in the last column we show which strategy was the best for this instance. Here *'A'* stands for approximation and *'S'* for the MISDP branch-and-bound algorithm. The detailed results

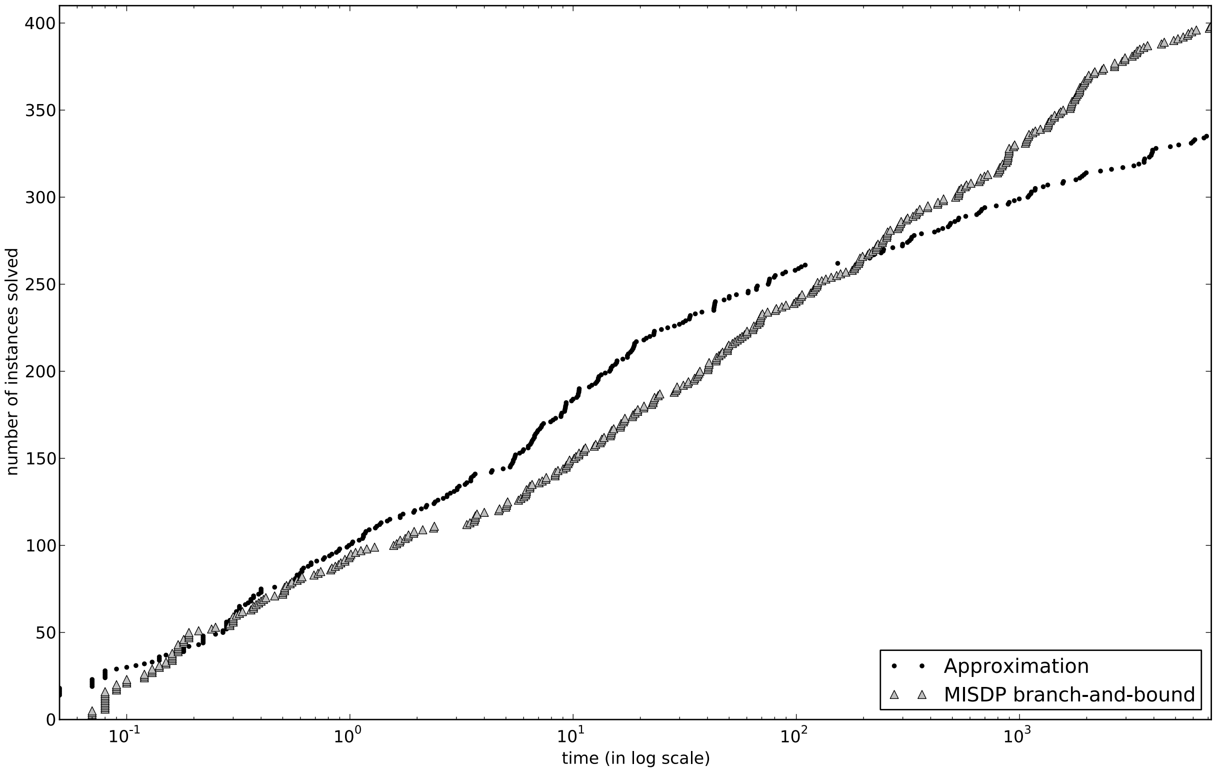


Figure 6.2. – Performance profile for the truss problems comparing pure MISDP branch-and-bound and pure approximation.

for all 526 instances in the same format as Table 6.5 can be found in Table A.1.

	solving as MISDP			solving with approximation			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
bridge-3	40.62	362.00	0.00	12.97	10633.00	0.00	A
bridge-3-act	190.01	712.00	0.00	416.06	399442.00	0.00	S
bridge-3-act-int	951.00	10108.00	0.00	1277.44	1057944.00	0.00	S
bridge-3-3scen	2670.00	10531.00	0.00	947.57	893938.00	0.00	A
bridge-3-2scen	1861.59	10533.00	0.00	1124.12	1100874.00	0.00	A
bridge-3-int	68.91	1149.00	0.00	5.49	3129.00	0.00	A
bridge-3-mV-int	4.64	89.00	0.00	3.58	7019.00	0.00	A
bridge-3-act-mV-int	263.65	1570.00	0.00	7200.40	6497359.00	2.47	S
bridge-3-act-mV	53.05	247.00	0.00	105.32	123699.00	0.00	S
bridge-3-cont	0.10	1.00	0.00	0.08	1.00	0.00	A
bridge-3-cont-act	0.86	3.00	0.00	2.72	731.00	0.00	S
bridge-3-mV	86.10	1014.00	0.00	23.04	48128.00	0.00	A
bridge-3-act-2scen	5658.01	11733.00	0.00	7200.07	2002544.40	4.21	S
bridge-3-act-2scen-int	7200.14	33115.80	–	5167.04	2918134.00	0.00	A
bridge-3-act-2scen-mV	1859.06	5119.00	0.00	7200.14	3956849.60	3.88	S

	solving as MISDP			solving with approximation			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
bridge-3-2scen-mV	2174.15	16846.00	0.00	670.96	844187.00	0.00	A
bridge-3-2scen-int	295.00	2537.00	0.00	42.88	20271.00	0.00	A
bridge-3-2scen-mV-int	40.35	390.00	0.00	1.32	487.00	0.00	A
bridge-3-cont-2scen	0.17	1.00	0.00	0.30	1.00	0.00	S
bridge-3-cont-act-2scen	5.09	11.00	0.00	8.84	508.00	0.00	S
canti-9	254.01	481.00	0.00	18.56	8049.00	0.00	A
canti-9-act	1703.01	1189.00	0.00	678.32	307750.00	0.00	A
canti-9-3scen	7200.38	6472.40	57.77	7200.07	1398209.60	49.19	A
canti-9-2scen	7200.44	8963.40	553.08	7200.09	4157208.60	5.53	A
canti-9-int	1685.27	12650.00	0.00	153.29	65192.00	0.00	A
canti-9-mV-int	99.65	1909.00	0.00	33.51	35299.00	0.00	A
canti-9-act-mV-int	7200.03	40585.40	128.56	7200.41	5157661.40	126.15	A
canti-9-act-mV	7200.81	9526.60	–	7200.19	3402817.00	36.87	A
canti-9-cont	0.18	1.00	0.00	0.18	1.00	0.00	S
canti-9-cont-act	0.60	1.00	0.00	1.18	19.00	0.00	S
canti-9-mV	3367.77	19632.00	0.00	6103.45	5819634.00	0.00	S
canti-9-act-2scen	7201.73	2481.20	–	7200.02	794006.60	11.76	A
canti-9-act-2scen-int	7200.51	9385.80	–	7200.02	566198.40	–	–
canti-9-act-2scen-mV	7200.99	4291.60	–	7200.15	2601883.20	156.12	A
canti-9-2scen-mV	7200.16	17481.80	51.44	7200.45	7326240.40	83.72	S
canti-9-2scen-int	7200.13	27555.00	–	7200.03	719194.40	–	–
canti-9-2scen-mV-int	54.41	355.00	0.00	0.90	775.00	0.00	A
lit-as-6-big	7202.64	1171.00	–	7200.18	2513962.00	–	–
lit-as-6-big-act	7202.62	1189.60	–	7200.17	2598494.00	–	–
lit-as-6-big-act-int	7200.28	15270.40	0.09	7200.07	1197412.60	1050.40	S
lit-as-6-big-int	7200.21	15042.40	0.10	7200.08	1384280.80	1049.77	S
lit-as-6-big-mV-int	571.37	1710.00	0.00	7200.37	4999979.20	4494.28	S
lit-as-6-big-act-mV-int	578.36	1710.00	0.00	7200.39	5017352.20	4494.11	S

Table 6.6. – Comparing the approximation scheme to pure MISDP branch-and-bound for some exemplary instances.

6.3.2. Parameter tests

Using eigenvector cuts to approximate the SDP constraints can be combined with the MISDP branch-and-bound algorithm. This means that in the branching nodes we can solve an SDP relaxation and an LP relaxation of the eigenvector cuts. The frequencies in which one of the two methods can be chosen through a simple parameter. For example if we use a frequency of five for the approximation this means that an LP is solved at the root node and at every node with a depth of five of multiples of five. However, it is not clear which parameter setting is the best. In the previous section we have seen that even the decision between pure approximation and pure MISDP branch-and-bound is not clear.

	3	4	5	6	7	8	9	10	11	12	
instance	time/gap	time/gap	time/gap	time/gap	time/gap	time/gap	time/gap	time/gap	time/gap	time/gap	best
bridge-1-act	25.21	23.04	24.49	25.03	27.27	29.05	22.84	23.66	23.54	28.00	9
bridge-2-act	153.40	304.48	301.71	293.16	512.52	489.41	469.07	568.22	539.50	591.46	3
bridge-4-2scen-mV	156.54	163.19	165.47	164.23	171.62	175.26	171.02	175.11	173.41	178.91	3
bridge-4-mV	0.93	1.09	1.01	1.08	1.39	1.49	1.49	1.47	1.37	1.38	3
bridge-5	89.87	60.08	77.72	97.47	70.60	79.27	88.43	102.06	100.98	113.09	4
bridge-5-2scen	–	–	13.32%	13.10%	935.06%	931.41%	44.71%	44.67%	931.35%	932.11%	6
bridge-5-mV	52.45	44.06	47.61	48.15	50.64	58.76	48.35	56.32	67.99	64.87	4
bridge-8-int	2983.75	2525.23	2442.13	2695.60	2585.78	2453.08	2036.90	2133.01	2210.71	2339.91	9
bridge-8-act-mV-int	–	–	–	–	103.16%	–	–	–	–	–	7
bridge-9-mV-int	33.30	39.68	26.54	61.77	40.27	33.53	24.92	28.45	47.44	27.72	9
canti-10-int	8.36	7.05	8.03	9.21	7.43	8.39	9.18	9.87	10.67	11.45	4
canti-2-int	22.66	14.43	22.12	18.34	23.77	19.61	21.13	21.50	22.05	22.63	4
canti-3-act-mV-int	47.71	98.52	109.50	124.40	100.19	92.34	85.86	115.13	93.76	97.01	3
canti-3-mV	39.41	42.02	39.55	42.73	43.95	42.44	46.14	41.55	45.53	45.31	3
canti-4	40.29	43.37	40.71	41.76	37.16	38.56	35.28	41.16	42.84	42.46	9
canti-4-act-mV	10.54	5.28	10.92	6.04	19.98	20.14	18.21	78.49	5.71	5.87	4
canti-5-act-int	2346.70	6016.12	28.68%	26.01%	262.68%	5968.57	29.90%	29.19%	314.59%	40.87%	3
canti-7-act	1543.15	1402.43	1481.27	1348.42	1432.17	5696.44	1374.52	18.46%	1400.95	1033.45	12
canti-big	2198.58	2085.97	1656.75	2002.85	2194.93	1869.33	2009.07	1502.83	2437.25	3008.24	10
canti-big-act-mV-int	237.12	1061.34	917.39	612.42	725.59	733.19	655.22	662.37	685.69	693.21	3
geom mean run times	2.22	3.41	1.81	1.53	1.97	3.13	2.25	2.97	2.93	2.41	← 6 ↑ 3
bridge-10-int	64.09	66.94	54.80	65.67	65.79	63.66	55.09	59.95	68.03	68.36	5
bridge-10-act-int	21.67%	32736.64%	174.04%	40543.04%	108.08%	36367.09%	1402.99%	155.46%	226.87%	77.79%	3
bridge-2-int	36.63	27.71	32.85	32.78	33.05	29.55	44.42	32.58	44.27	46.98	4
bridge-3-act	667.07	1062.78	456.83	487.44	483.19	410.46	433.42	421.14	370.40	894.12	11
bridge-3-act-mV-int	2772.40	5311.14	416.83	2691.87	305.90	709.87	115.41	1103.94	556.09	249.25	9
bridge-5-act-mV-int	3596.31	1859.67	1323.48	1003.08	896.26	1391.93	870.80	1181.86	1298.20	1693.88	9
bridge-6-act-mV	115.94	108.47	163.74	180.97	138.32	133.37	179.11	137.96	126.87	170.91	4
bridge-6-mV	35.25	34.81	30.61	33.19	37.16	33.23	36.94	41.02	38.09	38.08	5
bridge-7-act	–	–	–	–	–	–	–	–	–	–	–
bridge-9	107.22	84.08	90.96	94.58	80.78	77.16	79.08	95.19	95.66	98.84	8
canti-10	158.84	289.84	260.23	180.57	275.50	189.96	270.93	188.42	283.04	244.90	3
canti-10-2scen	64.12%	–	216.59%	966.12%	–	99.10%	643.41%	853.48%	966.10%	53.28%	12
canti-10-int	8.33	7.09	8.09	9.23	7.42	8.33	9.15	9.89	10.74	11.42	4
canti-10-mV-int	42.95	15.32	15.63	52.30	28.59	47.00	52.84	51.18	49.86	49.31	4
canti-2-act-mV-int	33.04	36.12	25.73	24.35	25.06	24.36	41.76	38.63	23.95	24.77	11
canti-2-mV	13.37	13.09	18.82	17.48	19.35	17.41	17.77	28.98	18.43	18.68	4
canti-4	40.32	43.46	40.83	41.79	37.02	38.64	35.36	41.17	43.16	42.52	9
canti-4-2scen-mV	0.08	0.08	0.08	0.08	0.09	0.08	0.08	0.08	0.08	0.08	3
canti-7-act	1543.40	1413.97	1497.21	1358.43	1398.29	5625.84	1360.91	18.48%	1374.05	1040.49	12
canti-9-mV	2879.95	2101.56	2306.54	2234.52	2283.10	2815.27	2334.77	2627.17	2507.48	2797.05	4
geom mean run times	2.14	10.31	1.92	3.41	3.17	3.35	2.64	2.39	3.13	2.05	← 5 ↑ 4

	3	4	5	6	7	8	9	10	11	12	
instance	time/gap	time/gap	time/gap	time/gap	time/gap	time/gap	time/gap	time/gap	time/gap	time/gap	best
bridge-10-act-int	21.65%	32735.23%	174.08%	40551.88%	108.04%	36356.83%	1402.81%	155.59%	226.95%	77.75%	3
bridge-1-act	25.20	23.00	24.45	24.73	27.18	28.56	22.70	23.60	23.68	28.08	9
bridge-1-mV	32.67	32.39	31.35	33.37	35.65	32.72	34.05	35.73	37.69	35.66	5
bridge-2	89.93	60.27	77.23	97.28	70.05	78.85	87.90	102.48	101.73	112.31	4
bridge-2-act-mV-int	1979.28	2562.13	914.36	1566.65	1094.10	1723.20	380.34	228.66	132.38	173.74	11
bridge-8-act	–	–	–	–	–	–	–	–	–	–	–
bridge-8-mV	2528.87	2126.87	2132.62	1925.86	1944.03	1926.28	2031.05	1927.05	2108.32	1910.59	12
bridge-9-act	569.10	562.19	557.45	387.49	525.79	425.00	699.28	526.99	461.82	413.62	6
bridge-9-act-mV-int	306.14%	261.54%	3100.00	2522.57	6253.71	2170.86	1922.62	1172.23	1345.47	949.08	12
canti-10-2scen-mV	642.65%	–	524.25%	503.48%	347.42%	358.14%	232.85%	411.03%	–	459.23%	9
canti-10-int	8.37	7.05	8.08	9.19	7.39	8.35	9.10	9.86	10.73	11.43	4
canti-1	40.29	43.25	40.94	41.69	36.94	38.54	35.28	41.12	43.05	42.34	9
canti-2	32.50	47.81	39.16	38.11	39.15	39.01	39.32	37.17	38.99	39.76	3
canti-2-2scen	452.42	564.02	446.94	375.75	446.35	448.47	493.25	430.08	464.04	451.70	6
canti-2-int	22.74	14.42	22.25	18.35	23.74	19.60	21.05	21.48	22.09	22.59	4
canti-3-int	10.35	10.92	12.85	9.82	15.08	14.32	14.94	11.90	14.29	14.64	6
canti-7-mV	2882.05	2474.41	2736.40	3841.07	2686.98	3595.45	2669.94	2516.96	5162.47	2766.61	4
canti-9-act-mV	114.69%	13.08%	17.29%	1858.58%	144.05%	1830.52%	1803.68%	156.12%	111.00%	1787.65%	4
canti-9-act-mV-int	188.68%	7.68%	623.80%	–	–	–	1267.56%	–	–	–	4
canti-9-mV-int	106.46	105.55	105.11	126.65	104.99	133.24	142.00	133.12	150.74	175.90	7
geom mean run times	3.47	4.39	2.39	5.27	4.00	6.21	4.84	2.93	6.45	4.86	← 5 ↑ 4

Table 6.7. – Testing different frequencies for the approximation procedure for three test sets.

This is why we decided to take a closer look at different parameter settings. For this reason we constructed three different randomly chosen smaller test sets. Each test set contains 20 instances, every kind of model (such as integer variables, multiple scenarios, actuator positioning, and so on) is represented in each test set. The instances are chosen randomly.

For these test sets we tested ten different parameter settings concerning the frequency in which the approximation procedure is called. As our instances do not have so many variables, the maximal depth of the branch-and-bound tree is for most of the instances smaller than 100. This is why we choose twelve as the smallest frequency and three as the highest frequency. The rounding heuristic (see Section 6.3.4) is turned off for all the parameter tests to be able to concentrate on the impact of the different parameter settings.

The results of our analysis are presented in Table 6.8. The different columns are labeled with the frequencies for the approximation calls. In the columns we present the solving time or the gap if the instance could not be solved to optimality by this parameter setting. If there is a gap shown we add '%' to the number. If there is a '−' this means that the solver reached the time limit and no feasible solution was found so far. The last column shows which parameter setting was the best for this specific instance.

After each test set, we sum the results up by showing the geometric mean of the normalized solving times or gaps. For each instance we normalize the results by setting the best result to one and expressing all other results for this instance relative to this one. If there was no feasible solution found for one instance we take the biggest value of that row and add one, to indicate that this parameter setting was worse than all others. After applying this procedure to all rows we take the geometric mean over the columns. The different values are presented in Table 6.8.

The parameter setting with the smallest geometric mean is shown in the lower right corner of the table for each test set, marked with a ←. Additionally, we count the number of times each parameter setting was the best over all instances of one test set and present this number also in the lower right corner with a ↑.

Which of the parameter settings is the best is ambiguous: when looking at the geometric means the settings 5 and 6 are the best. For the test set where 6 is the best, 5 is second best, therefore we choose 5 for taking a closer look on. Furthermore, if looking at the single instances the setting 4 was the fastest setting for the most instances for two of the three test sets. For the one test set where 3 solved most of the instances faster than the other, 4 was second best. For this reasons we chose to run the settings 4 and 5 on the whole test set with 528 instances.

The summary of the results are presented in Table 6.8. We compare the two parameter setting to the pure MISDP branch-and-bound without any eigenvector cut and to the setting where the approximation is only used in the root node. Again we can see that there is no optimal parameter setting for the whole test set. The parameter settings where the approximation frequency is 4 or 5 can solve the most instances (presented in column two) but they are not as fast as the MISDP branch-and-bound algorithm (see column four). Additionally, we show in column three of Table 6.8 which of the settings was able to produce the smallest gap if it was not able to solve the instance. For this criterion the setting where the eigenvector cuts are applied in a frequency of 5 produces the best results. In the last column of this table we show for the number of times each setting produced the best result for each instance. This means we summed up the last column of Table A.2, where we present the detailed results, for each parameter setting.

parameter setting	# instances solved	smaller gap, if not solved	fastest solver	# times best
MISDP	347	6	153	142
MISDP + Approx. freq. 0	339	14	34	84
MISDP + Approx. freq. 4	378	21	97	114
MISDP + Approx. freq. 5	380	29	85	124

Table 6.8. – Comparison of the four different parameter settings for 528 truss instances.

Additionally, we sum up our parameter test in Figure 6.3 with the four performance profiles for the different settings. The figure illustrates that it is not easy to say which of the settings is the best. There are almost no differences between the setting with an approximation frequency of 4 or 5. Also the pure approximation and the MISDP do not show large differences. In the end it seems that the setting with an approximation frequency of 4 and 5 can solve more instance in less time.

However, this is not everything we can say about parameters. The results show that the best setting depends on the kind of instance we want to solve. Looking again at Table 6.8 we can conclude that on basis of the kind of mode we should decide which parameter setting to choose. If the volume is minimized, this includes the model-types '*mV*', '*act-mV*', and '*2scen-mV*', or the discrete bar areas are modeled as integer variables, model-types '*int*' and '*act-int*', it turned out that solving the approximation with a frequency of four is almost always the best. For some of these instances also 3 and 5 are the best parameter settings, but these are also high frequencies for the approximation such that 4 is still a good choice. For the model with binary variables in one or two scenarios the approximation frequency should be smaller. The setting 9 is a good choice. For actuator positioning ('*act*') the best frequency is 12.

Very ambiguous are the results for the combined models '*mV-int*' and '*act-mV-int*'. For those two it is not clear if the approximation frequency should be high or low. This might be due to the fact that they combine the different advantages and disadvantages. Summing up we can make the following decisions:

- '*mV*', '*act-mV*', '*2scen-mV*', '*int*', '*act-int*' $\Longrightarrow$ choose setting 4,

- discrete bar areas and '*2scen*' $\Longrightarrow$ choose setting 9,

- '*act*' $\Longrightarrow$ choose setting 12 and

- '*mV-int*' and '*act-mV-int*' $\Longrightarrow$?.

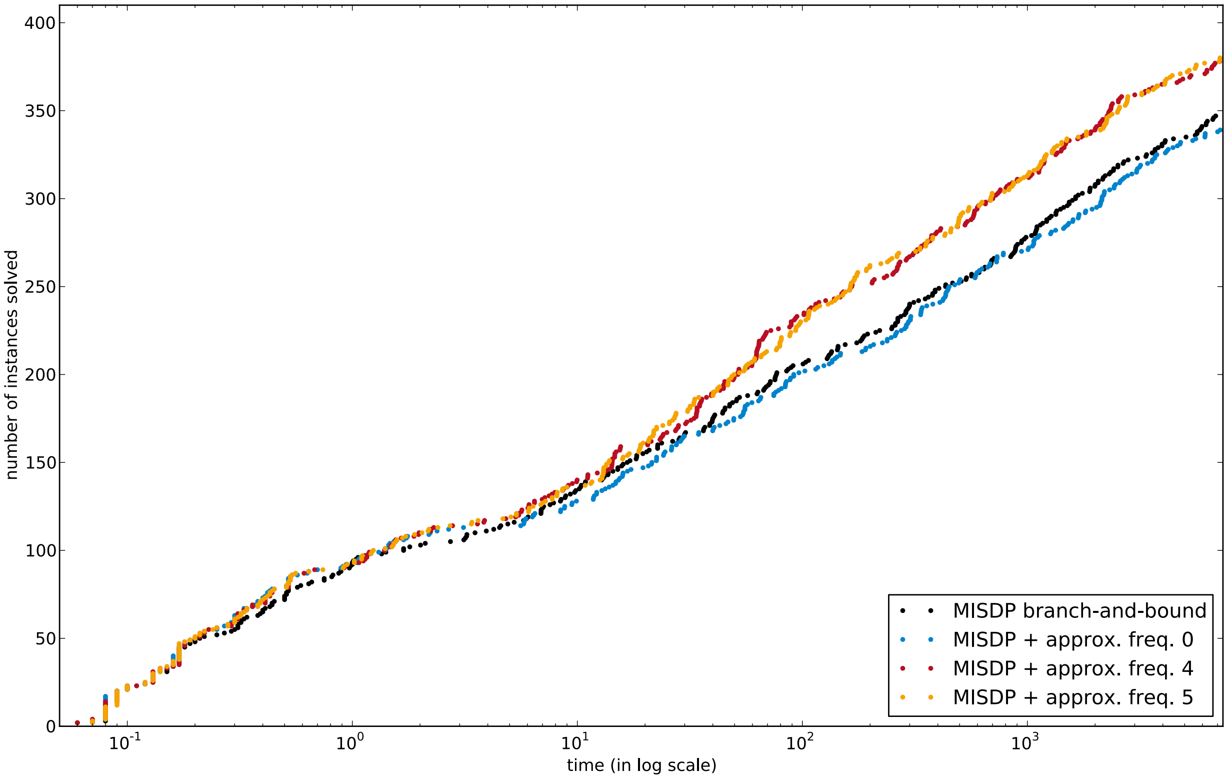


Figure 6.3. – Performance profile for the truss problems using different parameter settings.

6.3.3. Presolving

In Section 3.3.2 we stated different presolving methods. Their respective impact on the solving process is summarized in Table 6.9. For these tests the heuristic (see Section 6.3.4) is not used because we want to look at the impact of presolving only. Detailed results for all instances can be found in Table A.3.

	solving as MISDP			without presolving		
number of instances	# instances solved	smaller gap, if not solved	fastest solver	# instances solved	smaller gap, if not solved	fastest solver
415	291	12	138	290	6	136

Table 6.9. – Summary of the impact of the presolving ideas on truss problems.

Additionally, we sum up the computations in Figure 6.4. Looking at this figure it seems that both solving runs, with and without the SDP presolving, behave similarly. Both algorithms can solve almost the same number of instances. For half of the solved instances using our presolving ideas decreases the solving time. For the other half the solving time is increased if our presolving ideas are used. This can also be seen in Table 6.9. Moreover, this table shows that the gap for those instances that could not be solved to optimality within the time limit is smaller if using presolving.

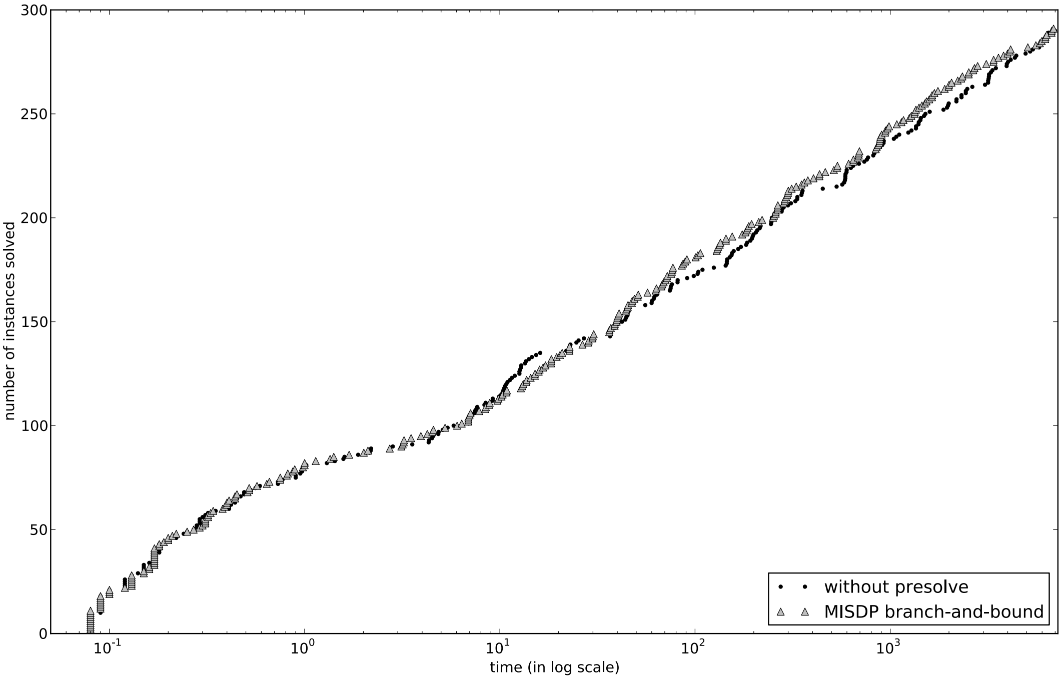


Figure 6.4. – Performance profile for the impact of the presolving ideas on the truss problems modeled.

However, presolving helps to prevent numerical difficulties. Thus we are always going to use it.

6.3.4. Heuristic

In Section 3.3.1 we stated a simple rounding heuristic, whose impact on the solving process is summarized in Table 6.10. Additionally, we sum up the computations in Figure 6.5. This figure shows that the impact of the heuristic is significant: if the heuristic is used, more instances can be solved and the time the solving process needs can be decreased. Moreover, we show the number of nodes needed to solve the instances in Figure 6.6. In this figure we can see that the number of nodes the branch-and-bound algorithm needs to solve the instances can be decreased.

The detailed results for all instances can be found in Table A.4. This shows that a heuristic is needed for solving MISDPs.

	solving as MISDP			without using the heuristic		
number of inst.	# inst. solved	smaller gap, if not solved	fastest solver	# inst. solved	smaller gap, if not solved	fastest solver
426	342	36	290	300	0	30

Table 6.10. – Summary of the impact of the heuristic in the pure MISDP branch-and-bound.

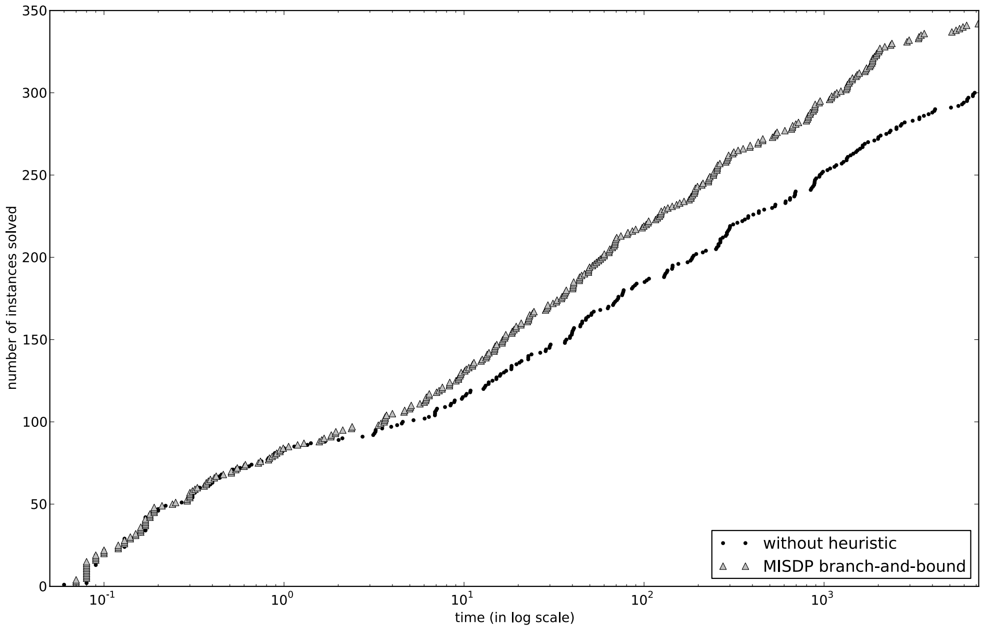


Figure 6.5. – Performance profile for the truss problems solved with the MISDP using the rounding heuristic or not using it.

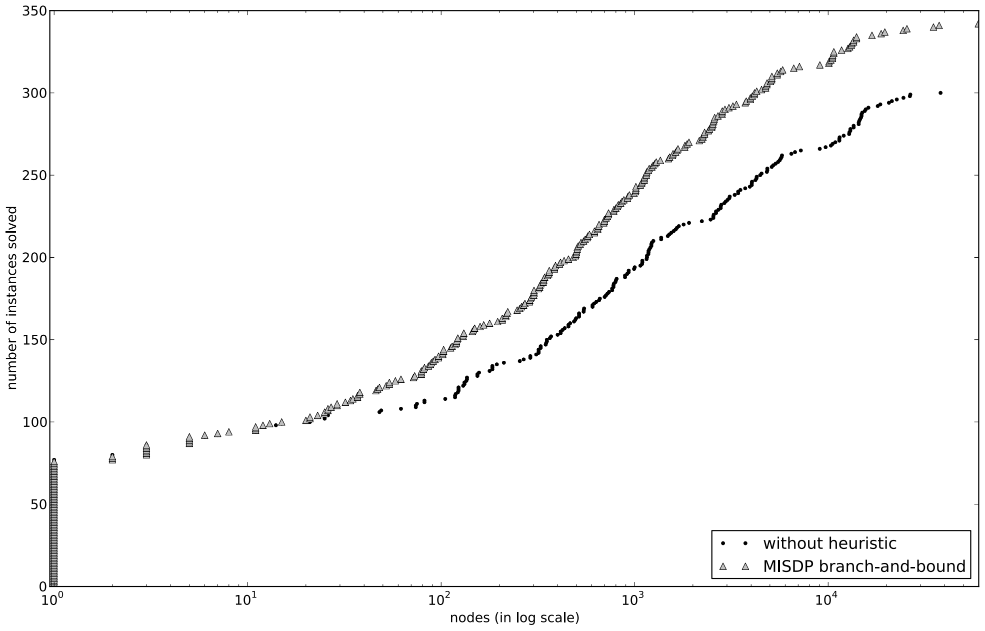


Figure 6.6. – Performance profile in terms of nodes for the truss problems solved with the MISDP using the rounding heuristic or not using it.

6.4 A note on the solving behavior

The first thing we want to remark is that solving one node within the MISDP branch-and-bound tree takes much longer than solving an LP relaxation. The SDP relaxation is stronger than the LP relaxation but solving takes longer. We already mentioned this difference in Chapter 3.

In this section we also want to take a closer look at the solving behavior for the different truss models in the pure MISDP branch-and-bound algorithm. We want to compare the formulation that minimizes the volume to the formulation that minimizes the compliance and we can see large differences and some analogies. So for example the quality of the relaxation solution obtained in the root node of the branch-and-bound tree is always good.

For the 219 instances of our test set that were solved to optimality we analyzed the solving characteristics and summarized them in Table 6.11. The geometric mean of the normalized root gap is about 6 and 21. The maximum root gap over all instances is for the minimum volume formulation 308.2% and for the minimum compliance formulation only 56%. Differences can for example be found in the quality of the first solution found. As already mentioned before, finding a feasible solution in the minimum volume formulation is trivial hence the quality of this solution is bad. Whereas the quality of the first solution for the formulation that minimizes the compliance is clearly better.

The number of nodes that have to be processed to prove optimality for a feasible solution found also differs. For the minimum volume formulation the maximum percentage of nodes that needs to be processed after the optimal solution was found is about 99% for one instance. The solving process stops immediately in about 20% of the instances if the optimal solution is found. This number is even larger for the minimum compliance formulation. For this model the solving process stops in about 30% of the instances if the optimal solution is found. The highest number of nodes that need to be processed subsequently over all instances is 75% of the entire nodes, needed for this instance.

The detailed results for all 219 instances can be found in Tables A.5 and A.6.

instance	geometric mean root sol.	geometric mean first sol.	min. nodes best sol. found	max. nodes best sol. found	min. overall nodes	max. overall nodes	geom. mean nodes after best sol.
min. com.	6.56	4.89	46	59109	46	59718	1.93
min. vol.	21.06	28.53	1	13567	13	37477	23.04

Table 6.11. – Summary of the geometric means for the relative root solution gap, the first solution gap and the nodes needed after the optimal solution is found.

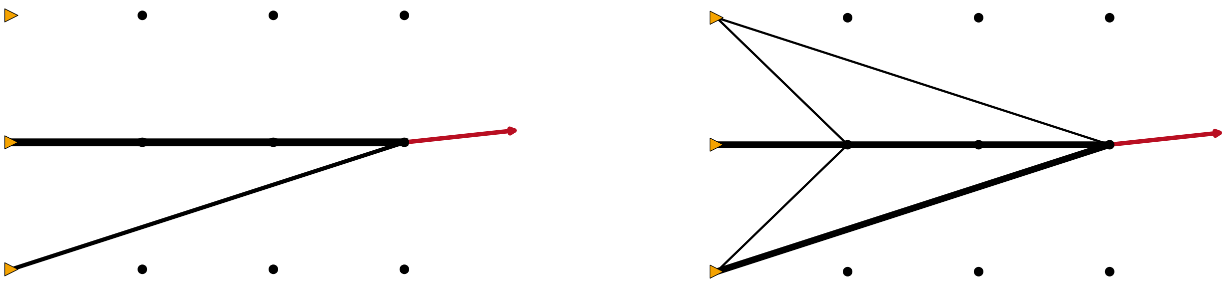


Figure 6.7. – Using discrete bar areas (right picture) can change the topology completely (compliance = 0.19). The left picture shows an optimized truss with continuous bar areas (compliance = 0.16).

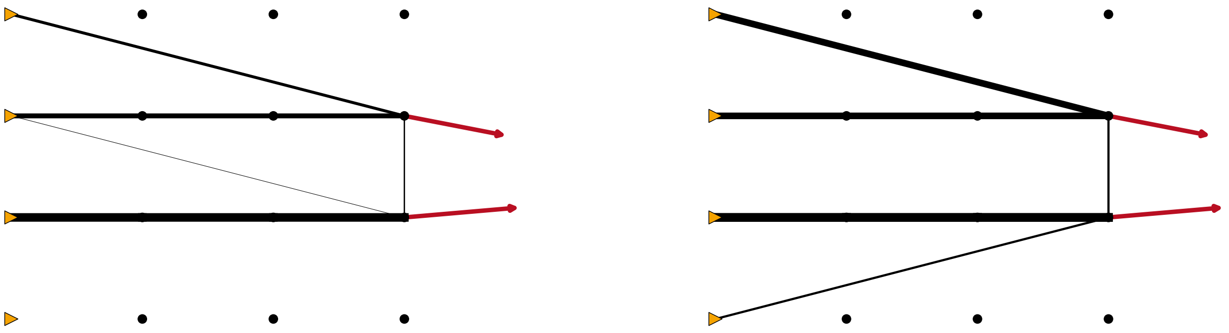


Figure 6.8. – This is a cantilever with twelve free nodes, four fixed nodes, and two loads, again the right picture shows the variant with discrete bar areas (compliance = 0.43). The optimal compliance of the problem with continuous bar areas (left picture) is equal to 0.40.

6.5 Comparing the models – volume minimization and discrete bars

After evaluating the features and the solving strategies of our software, we look at the different formulations for trusses. We start with three figures that illustrate that solving the continuous and the discrete Truss Topology Design problem really makes a difference for the topology. We demonstrate this fact in the Figures 6.7, 6.8, and 6.9.

All figures present cantilevers. The first figure shows a cantilever with nine free nodes and one external load. The second and the third picture show a larger cantilever with twelve free nodes and two loads in one scenario (Figure 6.8) and four loads in two scenarios (Figure 6.9).

These figures present the different topologies obtained if solving with continuous (left) or discrete (right) cross-sectional areas. Additionally, they show that the optimal solution

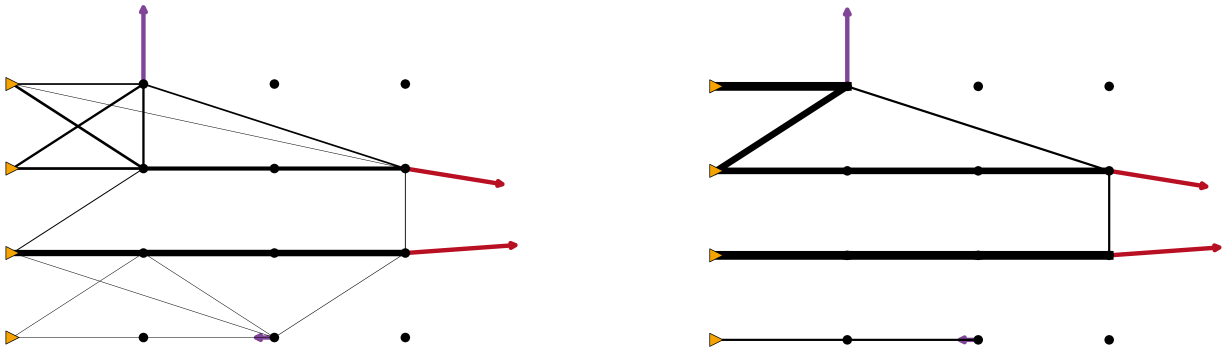


Figure 6.9. – This is the same example as presented in the previous figure with an additional scenario. The optimal compliance of the problem with continuous bar areas is equal to 0.53, whereas the discrete problem has a optimal value of 0.64.

value gets worse, when using discrete bar areas. This is due to the fact that not all bar areas are allowed anymore to compensate the external load, as the areas can only be chosen from a fixed set.

We do not show computational results for this comparison, as solving the truss problem with continuous bar areas yields an SDP and not a MISDP. Therefore any SDP solver can be used.

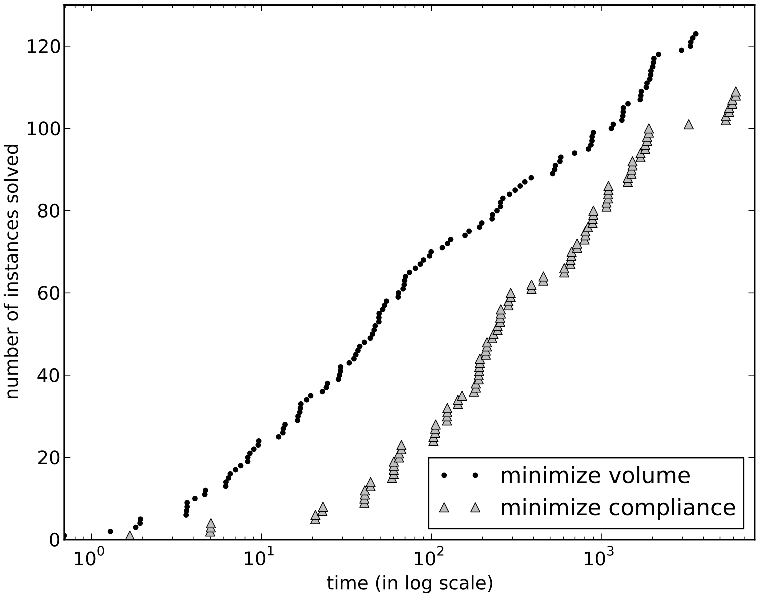


Figure 6.10. – Performance profile for the truss problems modeled with two different objective funtions: minimizing the compliance and minimizing the volume.

Additionally, we take a closer look at the two different objective functions a truss problem can be modeled with: minimizing the compliance and minimizing the volume. In Section 5.3.1 we formulated this model because it seems that the solver could use the structure of this model and therefore these model are easier to solve. The computational results do not substantially support this assumption in general. Figure 6.10 shows that in the end the minimum volume formulation is able to solve more problems and is faster for more instances. Certainly, for the instances with actuator positioning the minimum compliance formulation can solve more instances.

The same bottom line can be drawn from the first part of Table 6.12, where we summarized our solving runs. In this table we look at each kind of instances in a single row and sum up the results in the overall row. The details for all instances we present in Table A.7.

Moreover, we want to compare the two ways to model the discrete bar areas. As presented in Chapter 5 for the MISDP they can be modeled using binary or integer variables. We claimed that the variant using integer variables is better. Looking at the whole test set this is true. As we show in the second part of Table 6.12 using the integer formulation the solver was able to solve more instances in less of time.

Again we look a the different kinds of instance in a separate row and in the end sum up the results. The summary of our computations are presented in a performance profile in Figure 6.11 (the details can be found in Table A.8). However, if we look at the instances from one specific kind in our test set it is not true that the integer formulation is faster. We observed that for actuator positioning the binary formulation is significantly better.

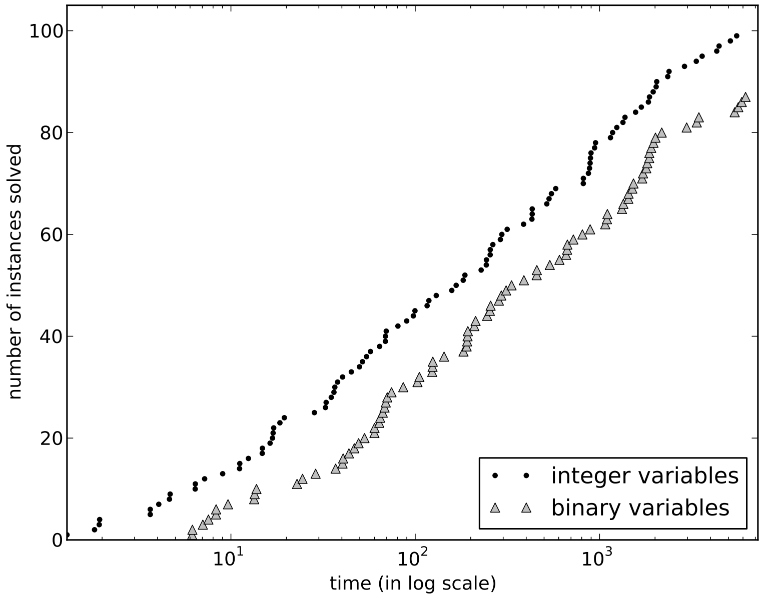


Figure 6.11. – Performance profile for the truss problems modeled with binary and integer variables.

kind of inst.	# inst.	minimizing the compliance # inst. solved	smaller gap, if not solved	fastest solver	minimizing the volume # inst. solved	smaller gap, if not solved	fastest solver
binary vars	29	23	0	10	23	1	15
actuators	29	22	0	8	18	1	15
int. vars	20	16	0	0	19	1	19
int. vars + act.	20	16	0	5	18	0	14
2 scenarios	68	33	6	4	49	4	42
overall	166	109	7	30	127	7	105
		using binary variables			using integer variables		
binary vars	21	17	0	4	18	1	15
actuators	21	16	0	15	10	1	2
min. volume	17	15	0	0	17	0	17
min. volume + act.	19	11	0	8	18	1	10
2 scenarios	55	28	3	17	36	7	25
overall	133	87	3	44	99	10	69

Table 6.12. – Summary of the comparison of optimizing trusses with two different objective functions and discrete cross-sectional areas using binary or integer variables.

6.6 Comparing the models – MIP and MISDP

It is possible to model the truss topology problem as linear MIP. We introduced this model in Section 5.4, the question is now whether this model can be solved more easily and faster as the MISDP we introduced. Therefore we first show the general rules for computing the number of variables and constraints in Table 6.13 and then take a look at the problem characteristics of the MIP for some example instances in Table 6.14.

		variables			linear
name	kind	bin.	int.	cont.	constraints
general	discrete bars	nm	0	$nm + d$	$4nm + 1 + d + n$
	minimize volume	nm	0	$nm + d$	$4nm + d + n$
	with actuators	$nm + n$	0	$nm + n + d$	$4nm + 2 + 2d + 4n$
	2 scenarios	nm	0	$2nm + 2d + 1$	$8nm + 1 + 2d + n$
	3 scenarios	nm	0	$3nm + 3d + 1$	$12nm + 1 + 3d + n$

Table 6.13. – General model statistics for the MIP formulation of the truss problems.

		variables			linear
name	kind	bin.	int.	cont.	constraints
bridge-1	discrete bars	49	0	69	217
	minimize volume	49	0	69	216
	with actuators	98	0	138	385
	2 scenarios	49	0	139	433
	3 scenarios	49	0	208	649
bridge-2	discrete bars	98	0	118	216
	minimize volume	98	0	118	215
	with actuators	147	0	187	630
	2 scenarios	98	0	237	874
	3 scenarios	98	0	355	1286
bridge-8	discrete bars	393	0	429	1740
	minimize volume	393	0	1	1739
	with actuators	524	0	596	2170
	2 scenarios	393	0	859	3348
	3 scenarios	393	0	1288	4956
canti-5	discrete bars	94	0	112	442
	minimize volume	94	0	112	441
	with actuators	141	0	177	602
	2 scenarios	94	0	209	836
	3 scenarios	94	0	313	1230
lit-as-5	discrete bars	299	0	345	1243
	minimize volume	299	0	345	1242
	with actuators	598	0	690	2187
lit-s-1	discrete bars	104	0	130	443
	minimize volume	104	0	130	169
	with actuators	208	0	260	782

Table 6.14. – Exemplary model statistics for the MIP formulation of the Truss Topology Design test set.

We tested the MIP formulation on 104 single load instances of our test set. The results are presented in Tables A.9 and A.10. Additionally, we provide an overview in Table 6.15 and a summary in Table 6.16. Table A.9 only shows those instances for which at least one of the models could be solved to optimality and compares MISDP and MIP in terms of time, nodes, and gap. The last column of this table additionally states which of the models could be solved faster: '*M*' stands for the MIP and '*S*' for the MISDP.

We can observe a very interesting fact: all minimum volume formulations can be solved faster using the MIP, whereas the minimum compliance models are always solved faster with the MISDP. If we now divide our test set into one set with the minimum compliance models and one with those that minimize the volume of a truss, we see exactly the same. For minimizing the volume the MIP is a very good model and for minimizing the compliance we should choose the MISDP.

In the exemplary instances Table 6.15 we can also see this behavior: the minimum volume formulations can be solved faster using the MIP and for the minimum compliance formulations the MISDP is the model of choice. The MIP is not able to solve one of the minimum compliance instances presented here.

The results are summed up in Table 6.16. This table shows that the MISDP solves more instances, but the MIP can be solved faster for more instances. Here we only show results for instances where at least one of the models could be solved to optimality. If we take a closer look at those instances both models could not solve within the time limit, we can conclude that the MIP is able to find feasible solutions very quickly but has still a large gap. In contrast, the MISDP is not able to find a feasible solution in most of the cases.

	solving as MISDP			solving as MIP			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
bridge-2	105.32	733.00	0.00	7200.01	3995826.20	53.07	S
bridge-2-act	284.27	1124.00	0.00	7200.01	819100.00	56.54	S
bridge-3	40.62	362.00	0.00	7200.01	1861801.20	44.47	S
bridge-3-act	190.01	712.00	0.00	7200.02	1256853.20	46.20	S
canti-8	64.53	302.00	0.00	7200.06	539359.80	0.14	S
canti-8-act	143.33	293.00	0.00	7200.05	574799.60	0.13	S
canti-9	254.01	481.00	0.00	7200.01	2980492.50	73.06	S
canti-9-act	1703.01	1189.00	0.00	7200.01	982240.40	75.35	S
canti-10	192.92	409.00	0.00	7200.01	2933699.75	57.86	S
canti-10-act	888.75	1093.00	0.00	7200.01	828881.80	56.41	S
canti-big	1096.81	1273.00	0.00	7200.01	2251844.25	85.04	S
canti-big-act	5409.97	2225.00	0.00	7200.00	1120412.60	83.44	S
bridge-2-act-mV	7.55	29.00	0.00	2.83	1948.00	0.00	M
bridge-2-mV	69.18	1015.00	0.00	2.84	3402.00	0.00	M
bridge-3-act-mV	53.05	247.00	0.00	2.83	1948.00	0.00	M
bridge-3-mV	86.10	1014.00	0.00	2.83	3402.00	0.00	M
canti-8-act-mV	22.83	91.00	0.00	0.58	918.00	0.00	M
canti-8-mV	13.76	122.00	0.00	0.62	918.00	0.00	M
canti-9-act-mV	7200.81	9526.60	–	2137.02	559311.00	0.01	M
canti-9-mV	3367.77	19632.00	0.00	657.35	354823.00	0.01	M
canti-10-act-mV	7200.09	38623.20	–	86.59	34390.00	0.00	M
canti-10-mV	2968.92	35012.00	0.00	18.87	14699.00	0.00	M
canti-big-act-mV	7201.02	3169.60	–	3841.72	542136.00	0.01	M
canti-big-mV	7200.09	28768.40	3.82	666.29	167430.00	0.01	M

Table 6.15. – Comparing the MIP and the MISDP model for some exemplary instances within the time limit of two hours.

	solving as MISDP			solving as MIP		
kind of model	# instances solved	smaller gap, if not solved	fastest solver	# instances solved	smaller gap, if not solved	fastest solver
min. compliance	46	0	42	12	0	4
min. volume	58	1	7	56	0	53
overall	104	1	49	68	17	57

Table 6.16. – Comparison of the two different models for 129 truss instances.

Figure 6.12 sums up all the results in a performance profile. This profile shows the behavior stated above. Note that the time is plotted in logarithmic scale. We are going to use the logarithmic scale for all our performance profiles.

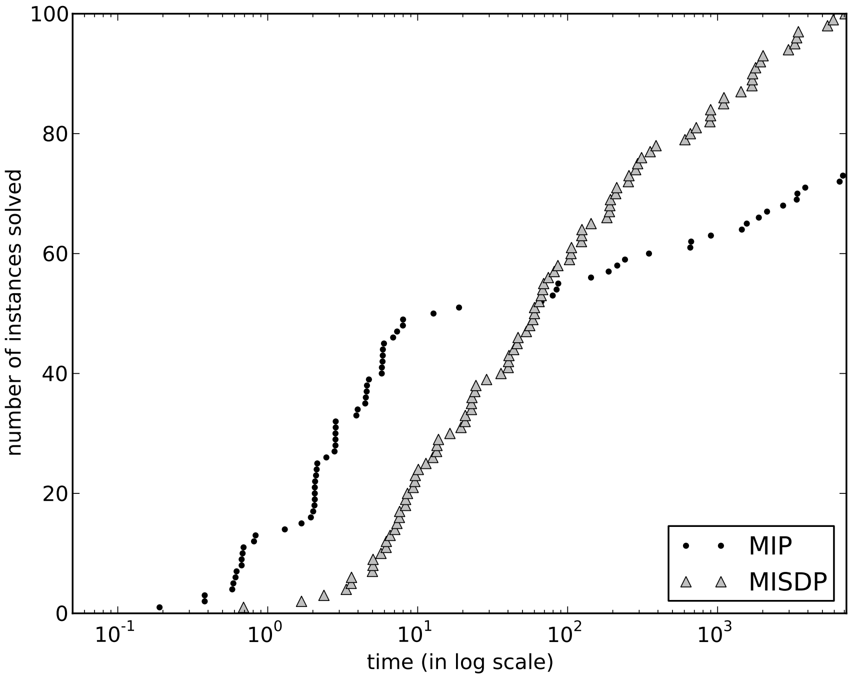


Figure 6.12. – Performance profile for the truss problems modeled as MIP and MISDP for a time limit of two hours.

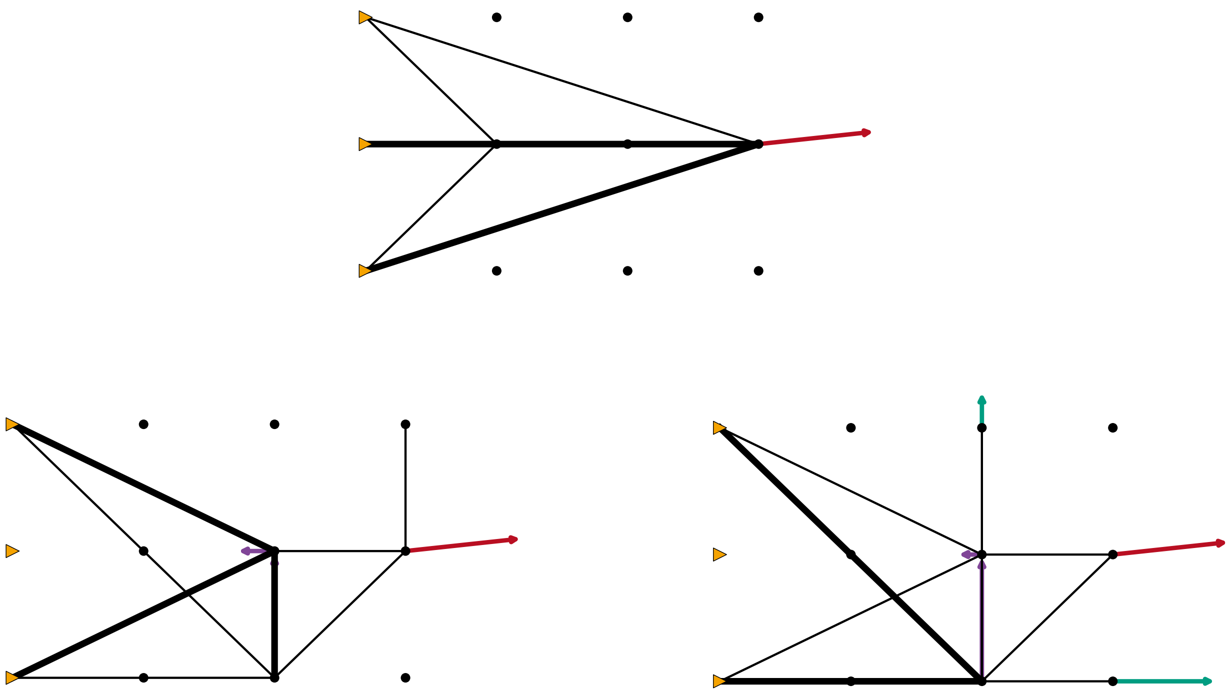


Figure 6.13. – These three trusses show that the topology can change significantly if we apply different loads in different scenarios. The figure on top shows a truss with one scenario (compliance = 0.19). The figures on the bottom have two (left, compliance = 0.86) and three (right, compliance = 0.95) scenarios.

6.7 Stability constraints

Controlling uncertainty in load-carrying structures like trusses is the main goal of the Collaborative Research Center 805. We will control the uncertainty using the stability constraints we presented in Section 6.7. Now we show the numerical results for multiple loads, actuator positioning, and vibration constraints.

6.7.1. Multiple loads

Considering multiple loads is one way to control uncertainty. Certainly the optimal topology of a truss will change if we apply these loads to nodes that were not present in the optimal topology. We show an example in Figure 6.13, where we consider a truss with one, two, and three scenarios. If the additional loads are applied in addition to an already existing load the bar areas of the used bars will change. This can be seen in Figure 6.14.

Modeling uncertain load scenarios can be done using the multiple load formulation we

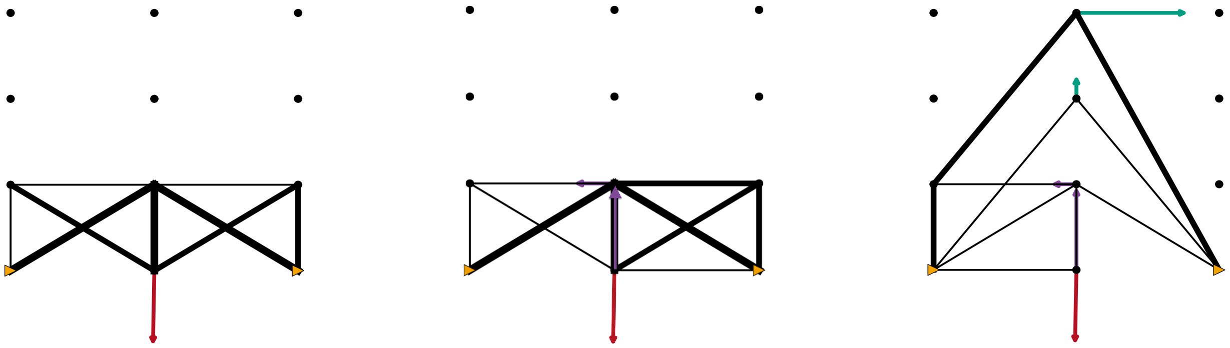


Figure 6.14. – Three bridge trusses with ten free nodes for one (left, compliance = 0.17), two (middle, compliance = 0.21), and three (right, compliance = 1.41) scenarios.

introduced for the MIP and the MISDP. Solving the MIP can cause serious difficulties concerning the numerics. If we let CPLEX solve the two and three scenario instances it aborts the solving process due to numerical difficulties for many of the instances.

We present some examples in Table 6.17. These instances are those of Figures 6.13 and 6.14. In the figures we only show the instances without actuator positioning. The results for the whole test set of 100 instances can be found in Table A.11.

	solving as MISDP			solving as MIP		
instance	time	nodes	gap [%]	time	nodes	gap [%]
bridge-9-3scen	3747.72	9217.00	0.00	7200.01	56488.20	100.00
bridge-9-2scen	245.43	652.00	0.00	7200.01	93841.50	77.55
bridge-9-act-2scen	807.06	939.00	0.00	–	–	–
bridge-9-act-2scen-mV	63.80	147.00	0.00	–	–	–
bridge-9-2scen-mV	36.98	255.00	0.00	–	–	–
canti-3-3scen	1062.21	4736.00	0.00	7200.01	110184.40	100.00
canti-3-2scen	455.97	2831.00	0.00	–	–	–
canti-3-act-2scen	1505.96	4274.00	0.00	7200.01	120770.17	100.00
canti-3-m-act-2scen-mV	49.27	121.00	0.00	–	–	–
canti-3-m-2scen-mV	29.16	216.00	0.00	–	–	–

Table 6.17. – Comparing the MIP and the MISDP model for the truss instances with more than one scenario.

In Table 6.18 we sum up the results subdivided into the instances with two and with three scenarios. This table shows that the solving takes much longer than for single scenario instances because not so many instances could be solved to optimality. This is surely due to the fact that there is one semidefinite constraint per scenario. The MISDP is able to solve more than half of the instances, whereas the MIP aborted solving for more than half of the instances. The MIP was not able to solve even a single instance but for a quarter of the instances is was able to find a feasible solution.We do not present a performance profile because the performance of the MIP was too bad.

		MISDP	solving as MIP		
number of instances	number of scenarios	# instances solved	# instances solved	feasible sol. found	solving aborted
80	2	56	0	13	54
20	3	11	0	9	2

Table 6.18. – Summing up the results of the solving runs for truss instances with two or three scenarios.

6.7.2. Actuators

This section covers positioning actuators within a truss and optimizing the topology at the same time. We discussed the different models in which the actuators we consider can be modeled. Now we compare again the MIP and the MISDP formulation. The results are shown in Table A.12.

At this point we only want to sum up the results in Table 6.19 and Figure 6.15. Again, we can conclude that the MISDP is the model of choice. Taking a look at the detailed results shows the same behavior as presented in the previous section: the MIP is faster for the minimum volume formulation and the MISDP can solve the minimum compliance formulations in shorter time.

Using actuators within a truss has a direct influence to the compliance of the truss. In Table 6.20 we compare the active and the passive models for the instances that could be solved to optimality. For all instances the objective value of the optimal solution for the active model was better or equal the one of the passive model. In the cases where the values are equal the instances were modeled as minimum volume formulation and actuators do not necessarily influence the volume if looking at discrete bar areas. In terms of the compliance adding actuators is always a good idea.

	solving as MISDP			solving as MIP		
# instances	# instances solved	smaller gap, if not solved	fastest solver	# instances solved	smaller gap, if not solved	fastest solver
78	55	0	42	23	0	18

Table 6.19. – Summing up the results of the solving runs for truss instances where we want to position actuators.

How the actuators can influence the topology of a truss we show in three figures (Figures 6.17, 6.16, and 6.18). Bars with actuators on them are marked as light blue bars. In all three figures we present the model with discrete bar areas, the one where actuators are positioned and the continuous model with actuators. All these instances minimize the compliance. In Figure 6.17, we show a bridge where the topology does not change if considering actuators. This is completely different in Figure 6.16, for this cantilever the

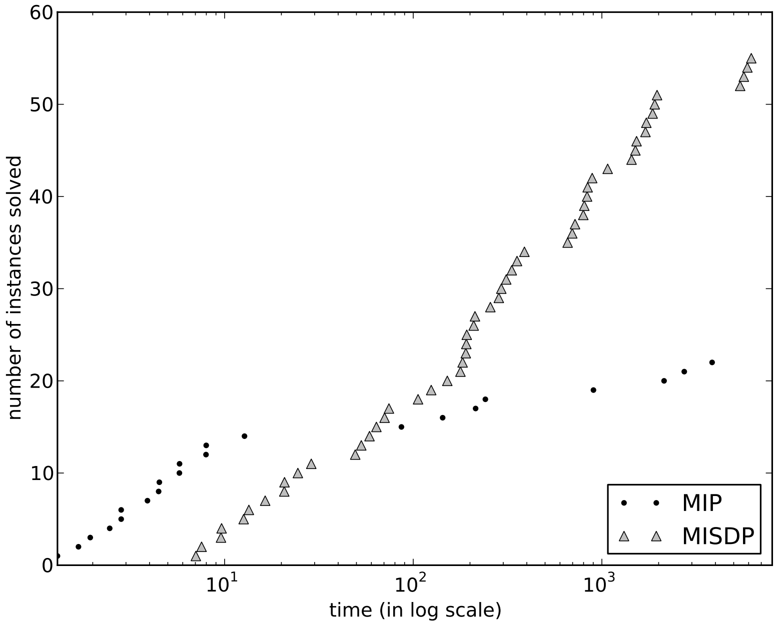


Figure 6.15. – Performance profile for the truss problems modeled as MIP and MISDP for a time limit of two hours and actuator positioning.

topology changes if want to position actuators. The third figure (Figure 6.18) presents a slightly larger cantilever.

	passive			active			
instance	time	nodes	opt. value	time	nodes	opt. value	best
bridge-1	5.04	79	1.55	20.72	130	1.44	A
bridge-1-mV	35.94	847	23.43	355.10	2576	21.96	A
bridge-1-2scen	232.17	2299	1.67	797.30	2158	1.55	A
bridge-1-2scen-mV	288.33	3773	24.19	696.68	2436	21.96	A
bridge-2	105.32	733	1.17	284.27	1124	1.10	A
bridge-2-mV	69.18	1015	22.90	7.55	29	18.49	A
bridge-2-2scen-mV	1349.46	13544	25.37	1437.06	3728	23.14	A
bridge-3	40.62	362	0.85	190.01	712	0.80	A
bridge-3-mV	86.10	1014	33.14	53.05	247	29.14	A
bridge-3-2scen	1861.59	10533	0.92	5658.01	11733	0.85	A
bridge-3-2scen-mV	2174.15	16846	34.85	1859.06	5119	31.20	A
bridge-4	4.99	79	1.55	20.78	114	1.42	A
bridge-4-mV	24.05	555	23.43	12.62	81	19.96	A
bridge-4-2scen	228.23	2299	1.67	835.05	2563	1.52	A
bridge-4-2scen-mV	198.07	2592	24.19	840.77	3072	21.78	A
bridge-5	102.84	733	1.17	293.04	1179	1.08	A

	passive			active			
instance	time	nodes	opt. value	time	nodes	opt. value	best
bridge-5-mV	68.17	1015	22.90	7.04	27	18.49	A
bridge-5-2scen-mV	1346.56	13544	25.37	1961.70	5686	23.31	A
bridge-6	40.33	362	0.85	209.19	800	0.78	A
bridge-6-mV	43.68	501	33.14	124.55	584	29.14	A
bridge-6-2scen	1814.74	10533	0.92	6195.92	13265	0.84	A
bridge-6-2scen-mV	1323.15	10086	34.85	1721.03	4825	30.79	A
bridge-7	1432.54	2504	0.64	5908.73	4164	0.58	A
bridge-7-mV	13.37	38	12.49	311.16	260	12.49	-
bridge-9	66.65	323	0.17	388.64	1070	0.14	A
bridge-9-mV	46.71	498	4.83	74.36	290	4.83	-
bridge-9-2scen	245.43	652	0.21	807.06	939	0.18	A
bridge-9-2scen-mV	36.98	255	4.83	63.80	147	4.83	-
bridge-10	1102.61	6642	0.24	720.74	861	0.16	A
canti-1	23.01	620	0.34	106.06	723	0.26	A
canti-1-2scen	43.88	505	1.42	177.99	655	1.29	A
canti-1-mV	3.62	103	13.97	9.57	73	10.63	A
canti-2	60.07	698	0.28	182.73	1629	0.21	A
canti-2-2scen	669.06	4754	1.42	1906.94	5111	1.29	A
canti-2-mV	6.16	102	12.16	28.81	166	10.16	A
canti-2-2scen-mV	49.13	385	55.72	333.06	923	48.91	A
canti-3	124.15	1121	0.19	256.42	993	0.15	A
canti-3-2scen	455.97	2831	0.86	1505.96	4274	0.78	A
canti-3-mV	8.30	120	15.65	9.64	52	12.16	A
canti-3-2scen-mV	29.16	216	91.91	49.27	121	74.52	A
canti-4	22.82	620	0.34	58.65	360	0.24	A
canti-4-2scen	43.86	505	1.42	151.59	523	1.26	A
canti-4-mV	3.60	103	13.97	16.41	131	10.63	A
canti-5	60.15	698	0.28	191.21	1655	0.20	A
canti-5-2scen	666.57	4754	1.42	1527.60	4151	1.26	A
canti-5-mV	6.18	102	12.16	24.46	196	10.16	A
canti-5-2scen-mV	49.14	385	55.72	192.23	528	46.78	A
canti-6	123.61	1121	0.19	212.51	781	0.14	A
canti-6-2scen	456.76	2831	0.86	1073.52	2517	0.76	A
canti-6-mV	8.31	120	15.65	13.45	72	12.16	A
canti-6-2scen-mV	29.23	216	91.91	70.49	178	70.45	A
canti-7	605.74	1517	0.43	658.47	495	0.34	A
canti-9	254.01	481	0.59	1703.01	1189	0.50	A
canti-10	192.92	409	0.18	888.75	1093	0.15	A
canti-big	1096.81	1273	0.37	5409.97	2225	0.31	A

Table 6.20. – Comparing the solution values and solving times of active and passive trusses.

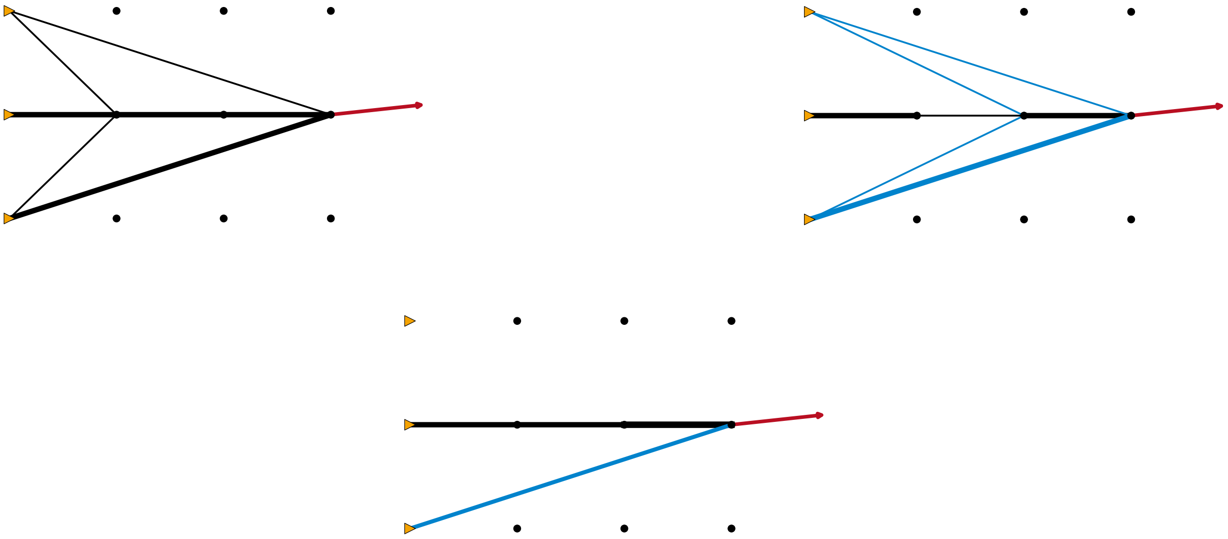


Figure 6.16. – Actuators can also be positioned in the Truss Topology Design problem with continuous bar areas (bottom, compliance = 0.11). In the two upper pictures we show the model with discrete bar areas (left, compliance = 0.19) and actuators (right, compliance = 0.14).

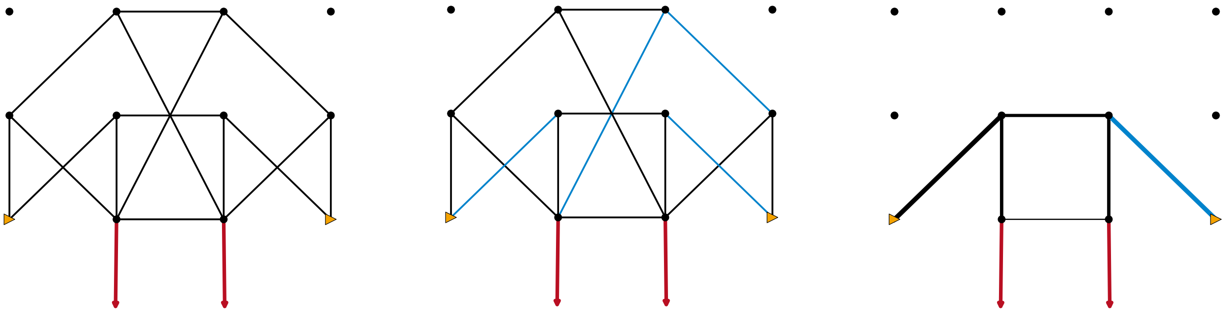


Figure 6.17. – Positioning actuators in trusses (marked as blue bars) does not always change the topology. The left figure shows the truss with discrete bar areas (compliance = 1.55, actuators are positioned in the middle for discrete bar areas (compliance = 1.45) and for continuous bar areas we show the results in the right figure (compliance = 0.97).

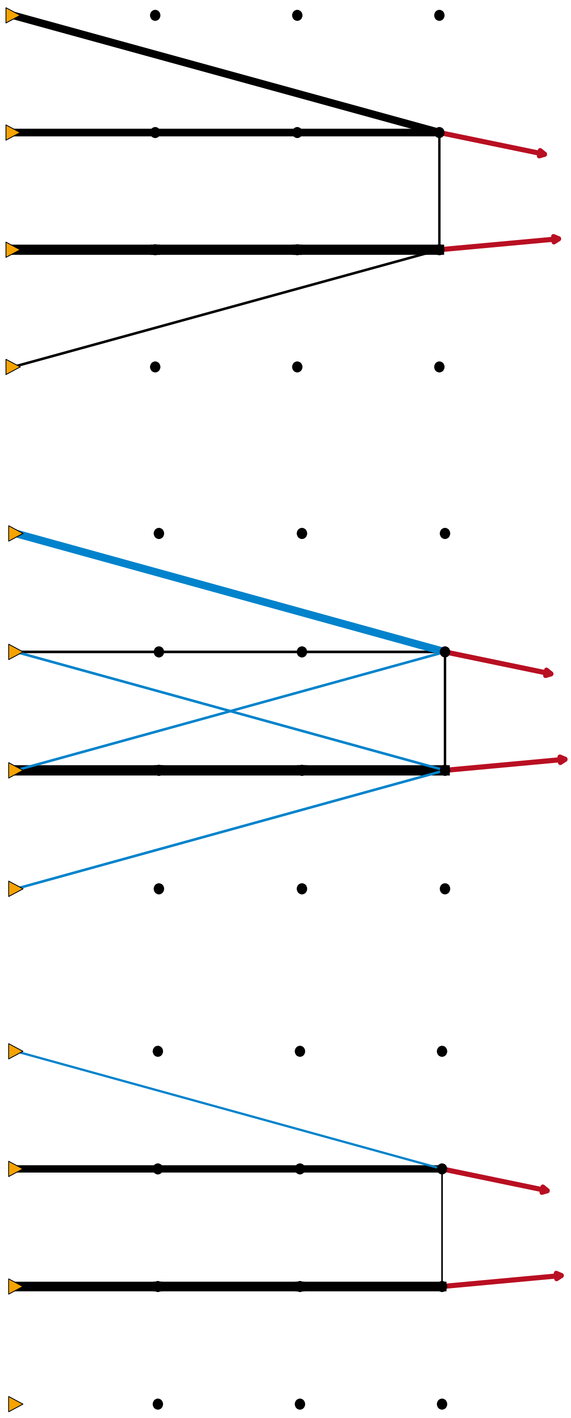


Figure 6.18. – Positioning actuator in a truss can also change the value of the optimal solution. Actuators reduce the value of the optimal solution. In this figure the left picture at the top has a optimal value of 0.43 and uses discrete bar areas, the right picture in the top row uses actuators and can reduce the optimal compliance to 0.34. At the bottom we show the topology for the continuous bar areas problem with actuator positioning. Here the compliance is again reduced to 0.32.

6.7.3. Vibrations

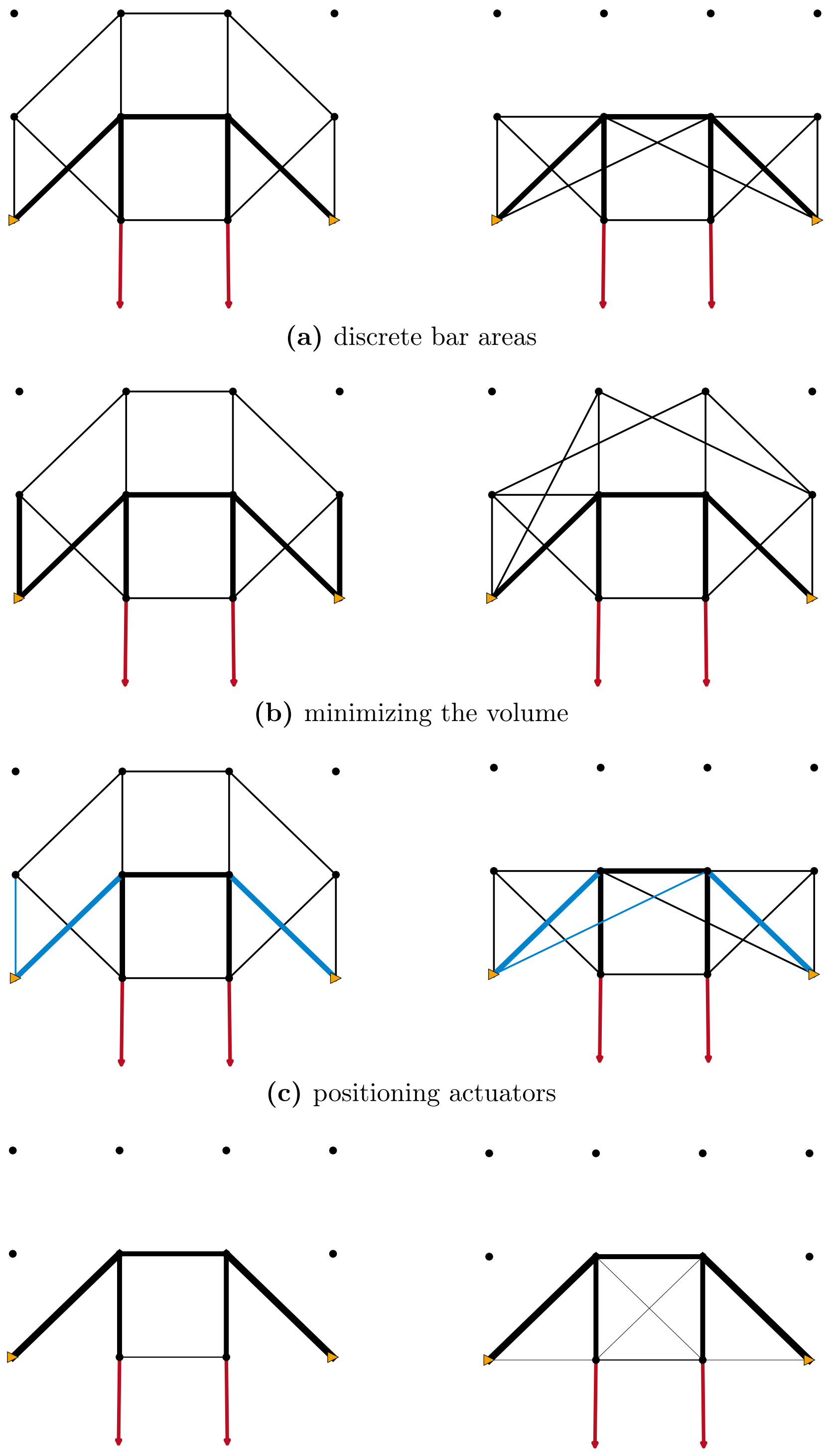
Finally, we also want to consider vibration constraints. Vibrations can only be considered in the MISDP. This is why we can only show results for the MISDP. For these instances we choose a normalized density of 1.0, just like we did with the elasticity modulus κ and a minimum eigenvalue $\bar{\lambda} = 0.01$. We used 104 instances for our tests. The rounding heuristic (see Section 6.3.4) was turned off for these tests because we want to show the pure impact of adding vibrations constraints.

We compare the results and the different solution values for some example instances in Table 6.21. The results for the whole test set can be found in Table A.13. The tables show that it does not always make a difference in the optimal solution value if considering vibrations, but sometimes the value changes.

	using vibration constraints				without considering vibrations			
			gap	optimal			gap	optimal
instance	time	nodes	[%]	value	time	nodes	[%]	value
bridge-3	580.39	3216.00	0.00	0.88	44.21	415.00	0.00	0.85
bridge-3-mV	2111.13	16959.00	0.00	34.44	90.72	1071.00	0.00	33.14
bridge-3-cont	0.16	1.00	0.00	0.69	0.10	1.00	0.00	0.68
bridge-3-act	1069.90	3330.00	0.00	0.81	184.18	724.00	0.00	0.80
bridge-3-2scen	3522.83	14156.00	0.00	0.92	2519.36	14388.00	0.00	0.92
bridge-big	7201.04	3460.40	–	–	7200.83	4854.20	–	–
bridge-big-cont	0.78	1.00	0.00	2.87	0.51	1.00	0.00	2.87
bridge-big-act	7205.44	905.40	–	–	7203.62	875.20	–	–
bridge-big-2scen	7201.46	2558.00	–	–	7201.16	3221.20	–	–
canti-1-mV	5.92	86.00	0.00	19.45	4.57	123.00	0.00	13.97
canti-1-cont	0.13	1.00	0.00	0.23	0.08	1.00	0.00	0.23
canti-1-act	219.74	1028.00	0.00	0.26	174.47	1153.00	0.00	0.26
canti-3-mV	6.36	51.00	0.00	21.65	9.72	136.83	0.00	16.65
canti-3-cont	0.13	1.00	0.00	0.16	0.08	1.00	0.00	0.16
canti-3-act	483.64	1495.00	0.00	0.15	265.58	993.00	0.00	0.15

Table 6.21. – Comparing solving times and solution values for trusses with and without vibration constraints for some exemplary instances.

To show that vibration constraints have an impact on the optimal topology we present the solutions of the instances in Table 6.21 in the Figures 6.20 and 6.19. Figure 6.20 shows one of the large examples for continuous cross-sectional ares. In Figure 6.19 we show a small bridge example with ten free nodes. For this example we show all the kinds of models we used for comparing vibration constraints (right) to standard trusses. We start with the standard truss model with discrete bar areas modeled as binary variables. In the second row we show the formulation that minimizes the volume as objective function. Then we position actuators and finally we also present the results for continuous bar areas. What can be seen at this example is the topology always changes if vibrations are considered.



(a) discrete bar areas

(b) minimizing the volume

(c) positioning actuators

(d) continuous bar areas

Figure 6.19. – A small overview of how vibrations can change different variations of one instance, models with vibrations are shown in the right column. Also the optimal solution value can change. For example, the first row shows the bridge with discrete bar areas, the compliance changes from 0.85 withour vibration constraint to 0.88 with vibrations.

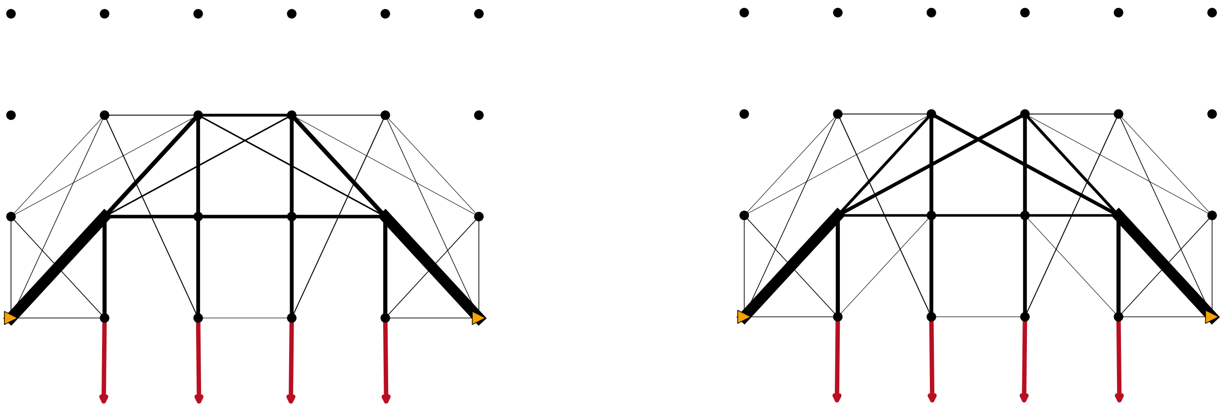


Figure 6.20. – Considering vibrations for continuous bar areas (right) can change the topology but in this case does not change the objective value of the optimal solution.

6.8 Statistics

As we use multi-core processors that share for example memory with other processes on the same system, we cannot guarantee that we can reproduce the solving time we present within this thesis. We detected huge differences in the solving time between two different solving runs for one instance. Therefore we did all the computations five times and present arithmetic means in all tables. Within this section we want to show how much the solving time or the number of nodes solved within the time limit can differ. We present six different tables, three for the MIP solving runs using CPLEX and three for the MISDP solving runs with SCIP.

We sum up the results presented in the six tables in Table 6.22. For this summary we ignored all instances where the solving time was smaller than 0.1 seconds because for these small solving times we cannot guarantee that the time measurement works correctly. So these instances could falsify our results. For example the instance *lit-as-1-cont* was one time solved in 0.01 seconds, another solving time for the same instance was 0.03. The difference is only 0.02 seconds, but these are 66.67% of the maximum solving time. As this time is so short we cannot be sure that the time was measured correctly.

Table 6.22 shows that the maximum differences for the MISDP are smaller and also the arithmetic means of these differences are smaller than for the MIP. This might be due to the fact that the MISDP needs less nodes than the MIP. Solving many nodes within a branch-and-bound algorithm is often accompanied with a lot of memory accesses because the nodes are located at many different places within the branch-and-bound tree and this is exactly one of the things that affect the solving time. Solving MISDPs we do not need so many nodes and therefore we do not jump that lot in the tree. Thus the variation of solving times or number of nodes needed, if the time limit is reached, is smaller than for the MIP.

	time		nodes		gap	
model	max. diff. [%]	geometric mean of diff.	max. diff. [%]	geometric mean of diff.	max. diff. [%]	geometric mean of diff.
MIP	47.36	9.02	37.81	8.98	91.93	3.70
MISDP	30.00	2.98	17.24	2.43	22.73	0.29

Table 6.22. – Variations in the different solving runs with time limit two hours, comparing MIP and MISDP.

We show the difference in the solving times for MIP and MISDP for the instances solved to optimality in Tables A.14 and A.17. Within these tables we present the average solving time for the five runs, the minimum and the maximum solving time. Additionally, we present the difference between maximum and minimum solving time in fifth column, called range. The last column shows the percentage of the range with respect to the maximum solving time.

In Tables A.15 and A.18 we consider the instances the MIP and the MISDP were not able to solve to optimality. Therefore we show the differences in the number of nodes.

Finally, Tables A.16 and A.19 show the differences concerning the gap for all instances that were not solved to optimality.

Finally, we like to mention that we also observed problems with the time limit. While CPLEX terminates within $[7199, 7201]$ if the time limit is set to 7200, SCIP sometimes needs more time to terminate. This can happen if the problems are large and solving one relaxations takes a lot of time. Then the time limit might be reached during the solving process of one SDP relaxation. Within this solving the time limit cannot be checked because the solving is done by the solver DSDP, so SCIP needs to wait until DSDP has finished and for our test set for the large instances this could be up to 100 seconds.

Additionally, we want to remark that using a node limit instead of a time limit is not a good idea for comparing the MIP and the MISDP formulation. The MISDP naturally needs only a fraction of the nodes the MIP needs so it would be very unfair to set a small node limit because the MIP would not be able to do much, whereas a large node limit yield a very long solving time for the MISDP, probably until it is finished.

CHAPTER 7

Results for the Maximum Cut Problem

To show the variety of SDPs and their applications this section will give another example of problems with semidefinite formulations. This example stands for a whole class of problems, namely combinatorial optimization problems. Various of these problems have SDP formulations which are normally used to compute good lower bounds as a starting point for the branch-and-bound algorithm. We take a closer look at the Maximum Cut Problem. The ideas and models presented in Section 7.1 are well-known from the literature – except for the transformation of the SDP.

For example in [Hel00] Maximum Cut Problems and their SDP formulation are extensively discussed. There also exist special solvers like the BiqMac solver [RRW10] for efficiently solving Maximum Cut Problems. The intention of this chapter is to show that our software is able to solve general mixed-integer semidefinite programs (MISDP), therefore we do not want to give a detailed survey on Maximum Cut Problems. We want to show that in the solving process problems like this behave completely different as for example Truss Topology Design problems. Therefore, the focus of this chapter lies again on computational results which are presented in Section 7.2.

Again we compare the solving times of a MIP to an MISDP formulation for the Maximum Cut Problem. Additionally, we discuss the differences in the solving process compared to Truss Topology Design. The test set is randomly generated. For validating that our code and the models are correct we compared the results to the solutions of the BiqMac solver.

7.1 The Maximum Cut Problem

One well-known example where SDP relaxations are used in combinatorial optimization is the Maximum Cut Problem. This is completely different from the problem of Truss Topology Design presented in Chapter 5. The semidefinite formulation does not arise from the physical background of the problem, here it is entirely artificial and there is nearly no structure in the coefficient matrices.

Before looking at the semidefinite formulation of the Maximum Cut Problem, we give a brief introduction into the topic.

Definition 7.1.1. Given an edge-weighted, undirected graph $G = (V, E)$ with vertices $V = \{1, \ldots, n\}$, edges $ij \in E$, where i and j are the endpoints of an edge ij. The egde-weights are c_{ij}, with $c_{ii} = 0$ and $c_{ij} = 0$ for $ij \notin E$. A *cut* $\delta(S)$, with $S \subseteq V$, is defined as the set of edges ij with $i \in S$ and $j \in V \setminus S$. □

Now we want to find a set S, such that the sum of all weights of the edges in the cut is maximum.

Definition 7.1.2. The *maximum cut* for an edge-weighted, undirected graph $G = (V, E)$ and weights c_{ij} is defined as

$$\max_{S \subset V} \sum_{ij \in \delta(S)} c_{ij}. \tag{7.1}$$

□

This problem can now be written using variables $x_i \in \{-1, 1\}$ for every vertex i, where $x_i = 1$ indicates that vertex i belongs to the set S and $x_i = -1$ that vertex i is in the set $V \setminus S$. This leads to

$$\begin{aligned} \max \quad & \sum_{i<j} c_{ij} \frac{1 - x_i x_j}{2} \\ \text{s.t.} \quad & x_i \in \{-1, 1\}, \quad i = 1, \ldots, n. \end{aligned}$$

The product of two variables $x_i x_j$ is positive if the vertices i and j belong to the same set, i.e., $i, j \in S$ or $i, j \in V \setminus S$. It is negative if the corresponding edge ij is in the cut, so the two edges belong to different sets.

Remark 7.1.3. *Every vector x with entries -1 and 1 is a feasible solution of the Maximum Cut Problem. For example take the vector of all ones. This means that all vertices are in the same set and no edge is in the cut. This solution is feasible. Therefore, finding feasible solutions for the max-cut problem is trivial.*

The Maximum Cut Problem can be formulated as a semidefinite program [Hel00]. To this end, we need to transform the objective function. Let e be the vector of all ones, then

$$\sum_{i<j} c_{ij} \frac{1 - x_i x_j}{2} = \frac{1}{4} \sum_{i=1}^{n} \left(\sum_{j=1}^{n} c_{ij} x_i x_i - \sum_{j=1}^{n} c_{ij} x_i x_j \right) = \frac{1}{4} x^T (\mathrm{Diag}(Ce) - C) x),$$

where $\mathrm{Diag}(Ce)$ maps the vector Ce onto a diagonal matrix in the following way:

$$\operatorname{Diag}(Ce) = \begin{pmatrix} \sum_i c_{1i} & 0 & 0 & \dots & 0 \\ 0 & \sum_i c_{2i} & 0 & \dots & 0 \\ \vdots & 0 & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & 0 \\ 0 & 0 & \dots & 0 & \sum_i c_{ni} \end{pmatrix}.$$

For $L(G) = \frac{1}{4}\operatorname{Diag}(Ce) - C$ we can reformulate the problem into:

$$\max_{x \in \{-1,1\}^n} x^T L(G) x.$$

A semidefinite formulation

The semidefinite relaxation is obtained using the facts that

(i) $x^T L(G) x = \langle L(G)x, x\rangle = \langle L(G), xx^T\rangle$ and

(ii) xx^T is positive semidefinite for all $x \in \{-1,1\}^n$.

The diagonal entries of xx^T are always equal to one and its rank is also equal to one. Characterizing the matrix $X = xx^T$ using these constraints results in the following MISDP:

$$\begin{aligned} \max \quad & \langle L(G), X\rangle \\ \text{s.t.} \quad & \operatorname{diag}(X) = e \\ & X \succeq 0 \\ & x_{ij} \in \{-1,1\} \qquad i,j = 1,\dots,n. \end{aligned} \tag{7.2}$$

Now the problem has variables x_{ij} for every edge ij, indicating if the edge is in the cut ($x_{ij} = -1$) or not, then $x_{ij} = 1$.

The transformed SDP

Note that for modeling this problem in the standard SDP form, a transformation is needed because it is not possible to model $x_{ij} \in \{-1,1\}$ and especially $x_{ij} \neq 0$ in the SDPA format [FFK+08]. Therefore, we take the complement and transform the problem into a binary program. So we introduce new variables $y_{ij} \in \{0,1\}$ with

$$\begin{aligned} y_{ij} = 0 \quad &\Leftrightarrow \quad x_{ij} = -1 \\ y_{ij} = 1 \quad &\Leftrightarrow \quad x_{ij} = 1. \end{aligned}$$

Then $y_{ij} = 0$ if ij is in the cut and it is equal to one if ij is not in the cut. The transformation used is

$$y_{ij} = \frac{x_{ij}+1}{2} \quad \text{or} \quad x_{ij} = 2y_{ij} - 1.$$

Not only the variables need to be transformed, also the constants need the transformation. For $x_{ii} = 1$ nothing changes because $y_{ii} = \frac{x_{ii}+1}{2} = 1$. Hence

$$X = 2Y - Y_0,$$

where Y_0 is the matrix with all diagonal entries equal to one and all other entries equal to minus one. The objective changes to $\langle L(G), 2Y - Y_0\rangle = \langle L(G), 2Y\rangle - \langle L(G), Y_0\rangle$. Now define a constant $s := \langle L(G), Y_0\rangle$. Using this constant shift-factor the transformed problem of (7.2) has the following form:

$$\begin{array}{lrl} \max & \langle L(G), 2Y\rangle - s & \\ \text{s.t.} & 2Y - Y_0 \succeq 0 & \\ & y_{ii} = 1 & i = 1, \dots, n \\ & y_{ij} \in \{0, 1\} & i, j = 1, \dots, n. \end{array}$$

Remark 7.1.4. *The formulation stated above is the complement of* (7.2). *It would have been more natural to transform the problem using variables $z_{ij} \in \{0,1\}$ with $z_{ij} = 1$ for ij being in the cut and then to maximize $\sum_{i,j=1}^{n} c_{ij} z_{ij}$. However, this transformation ($z_{ij} = \frac{1-x_{ij}}{2}$) leads to zero entries on the diagonal of the variable matrix:*

$$z_{ii} = \frac{1 - x_{ii}}{2} = 0.$$

With all diagonal entries equal to zero the matrix is no longer positive semidefinite anymore because, for example, the 2×2 principal minors have a negative determinant:

$$\det \begin{pmatrix} z_{11} & z_{12} \\ z_{12} & z_{22} \end{pmatrix} = \det \begin{pmatrix} 0 & z_{12} \\ z_{12} & 0 \end{pmatrix} = 0 - z_{12}^2 \leq 0.$$

Thereby this minor is negative definite and by Proposition 2.1.4 the matrix obtained is not positive semidefinite if $z_{12} \neq 0$.

This is why we choose the complement formulation and perform a variable flip. After solving we also need to reverse the transformation for being able to compare the result to other formulations.

A formulation as linear integer program

For getting an idea of the strengths and weaknesses of an SDP formulation, we again compare the model stated above with a linear IP formulation. To this end we present one possible formulation as linear binary problem. We start with (7.1) and introduce binary variables z_e for each edge e. If an edge ij is in the cut, then $z_{ij} = 1$ and $z_{ij} = 0$ otherwise. So the objective function is the sum of all variables multiplied with the edge-weights:

$$\sum_{ij \in E} c_{ij} z_{ij}.$$

As constraints we add some triangle inequalities that state if an edge is in the cut and this edge is part of a triangle, then there is exactly one other edge in this triangle in the cut:

$$\left.\begin{aligned} z_{ij} + z_{ik} + z_{jk} &\leq 2, \\ z_{ij} + z_{ik} - z_{jk} &\geq 0, \\ z_{jk} + z_{ij} - z_{ik} &\geq 0, \\ z_{ik} + z_{jk} - z_{ij} &\geq 0, \end{aligned}\right\} \text{ for } i, j, k \in V \text{ with } i < j < k.$$

Since these inequalities are needed for all combinations for i, j and k this is a very large formulation. For the sake of completeness note that $z_{ii} = 0$ and $z_{ij} = z_{ji}$.

7.2 Computational Results

In this section we show that our software is able to solve Maximum Cut Problems and that the behavior of the Maximum Cut Problems within the MISDP solving process is completely different from the behavior of the truss problems in the previous chapter. Some of the most remarkable differences of the max-cut instances are:

- finding feasible solutions is trivial (see Remark 7.1.3),
- the quality of the first solution is very bad (see Table A.24),
- the approximation procedure is completely useless (see Table 7.5) and
- using MISDPs for solving max-cut problems cannot compete with specific max-cut software (see Table 7.2).

Before presenting the results we first show the validation of our model. Then we again compare a mixed-integer linear and a mixed-integer semidefinite formulation. Additionally, we again compare the approximation procedure with the pure branch-and-bound algorithm. Then we give a detailed description of the differences just mentioned above. Finally, we make some remarks on the variations of the different runs as we already did in Chapter 6. Again we solved all the instances five times and the solution time and the number of nodes presented here are arithmetic means. The solution time is always presented in seconds and the detailed results can be found in Appendix A.2.

The test set

As most of the literature instances are too big for our solver we generated our own dense random graph instances. These instances consist of $30, 35, 40, 45$ or 50 vertices and a random number of edges. The edge weights c_{ij} are chosen from the set $C = \{1, 2, \ldots, 130\}$.

For each size we generated 20 instances, their names depend on the number of edges and a number of zero from 20 for numeration. So for example *mc30-0* is the first max-cut instance (*mc* stands for max-cut) with 30 vertices and *mc45-6* is the seventh instance with 45 vertices.

A detailed description of the instance characteristics can be found in Table A.20. Now we only want to give an overview of these characteristics in Table 7.1. In this table we show the number of vertices of the five different types of instances, the minimum and the maximum number of edges of the 20 instances of one type. Additionally, we present the minimum and the maximum degree over all vertices of all 20 instances of one type. To give an idea of the graphs we present the optimal solution of instance *mc30-6* in Figure 7.1.

instance	vertices	min. number of edges	max. number of edges	min. degree of vertex	max. degree of vertex
mc30-*	30	266	314	24	29
mc35-*	35	335	475	27	34
mc40-*	40	491	608	35	39
mc45-*	45	587	750	39	44
mc50-*	50	733	898	43	49

Table 7.1. – Overview of the number of vertices and the number of edges in the five different types of instances.

7.2.1. Model and algorithm validation

To ensure that the models presented in Section 7.1 and our implementation are correct we solved some of our generated instances using a special max-cut solver. There exist many special solvers for max-cut problems, the BiqMac solver ([RRW10]) is one of them. These solvers are very fast because they use problem specific information about max-cut problems.

We used the BiqMac solver to solve some of our instances and then compared the objective function values with our MIP and our MISDP solution. Note that the MISDP solution must be converted first as described in the previous section. We used the BiqMac solver through the web-interface. It runs on a Intel(R) Xeon(R) CPU 5160 with 3.00GHz. This machine is slightly faster than the one we used for our tests, but this fact does not explain the short solving times. Furthermore note, that we only tested each of the instances once with the BiqMac solver, as this solver was only available through the web-interface.

As Table 7.2 shows in all instances we tested the objective function values are identical or we could not solve the instance with one day. This is represented by the status *'ok'*. Some instance are so large that the MIP was not able to solve them within a day, therefore a validation was not possible. In all other cases our models produce the same solution as the BiqMac solver, so there is a good chance that our model and our algorithms are correct.

instance		solving with BiqMac			MIP	MISDP	conv. sol.	
name	edges	time	nodes	solution	solution	solution	MISDP	status
mc30-0	314	3.42	7	12516	12516	7761	12516	ok
mc30-1	284	0.28	1	12603	12603	6518	12603	ok
mc30-2	297	0.24	1	11928	11928	6865	11928	ok
mc30-3	276	0.32	1	11723	11723	6366	11723	ok
mc30-4	297	0.54	1	12180	12180	7156	12180	ok
mc30-5	295	0.94	1	13095	13095	7178	13095	ok
mc30-6	292	0.76	1	12404	12404	7104	12404	ok
mc30-7	304	2.62	5	13075	13075	7957	13075	ok
mc30-8	287	0.16	1	11630	11630	6246	11630	ok
mc30-9	290	0.32	1	12176	12176	6797	12176	ok
mc40-0	575	3.02	3	23594	23594	15249	23594	ok
mc40-1	503	3.17	5	20755	20755	12782	20755	ok
mc40-2	491	4.11	5	19650	19650	12073	19650	ok
mc40-3	537	7.74	9	21133	21133	13526	21133	ok
mc40-4	540	0.98	1	21597	21597	12750	21597	ok
mc40-5	513	3.92	5	20483	20483	12524	20483	ok
mc40-6	508	2.64	3	19189	19189	11254	19189	ok
mc40-7	520	9.51	13	20617	20617	13096	20617	ok
mc40-8	541	0.96	1	21181	21181	13281	21181	ok
mc40-9	483	1.32	1	20550	20550	12074	20550	ok
mc50-0	733	15.67	13	29711	time > 24h	18726	29711	ok
mc50-1	871	19.87	13	33706	time > 24h	22084	33706	ok
mc50-2	873	10.63	7	34552	time > 24h	22206	34552	ok
mc50-3	765	1.25	1	31358	time > 24h	19382	31358	ok
mc50-4	863	15.91	13	34184	time > 24h	23460	34184	ok
mc50-5	833	14.41	11	33536	time > 24h	22295	33536	ok
mc50-6	888	1.26	1	34870	time > 24h	22835	34870	ok
mc50-7	816	6.40	5	32158	time > 24h	20943	32158	ok
mc50-8	870	13.70	11	34435	time > 24h	22943	34435	ok
mc50-9	881	9.44	7	34306	time > 24h	22913	34306	ok

Table 7.2. – Instance statistics and results for solving some of the max-cut instances with the BiqMac solver, the MIP and the MISDP model.

Additionally, this table shows the solving time and the number of branch-and-bound nodes the BiqMac solver needed. Both values are very small. The solving times and number of nodes our general MISDP solver needs are really huge compared to those values. This will be explained in the following sections. We show the solving times of our MISDP solver for example in Tables A.21, A.22 and A.23. By now we can conclude that for the Maximum Cut Problem the solver presented in this thesis cannot compete with these highly problem specific solvers.

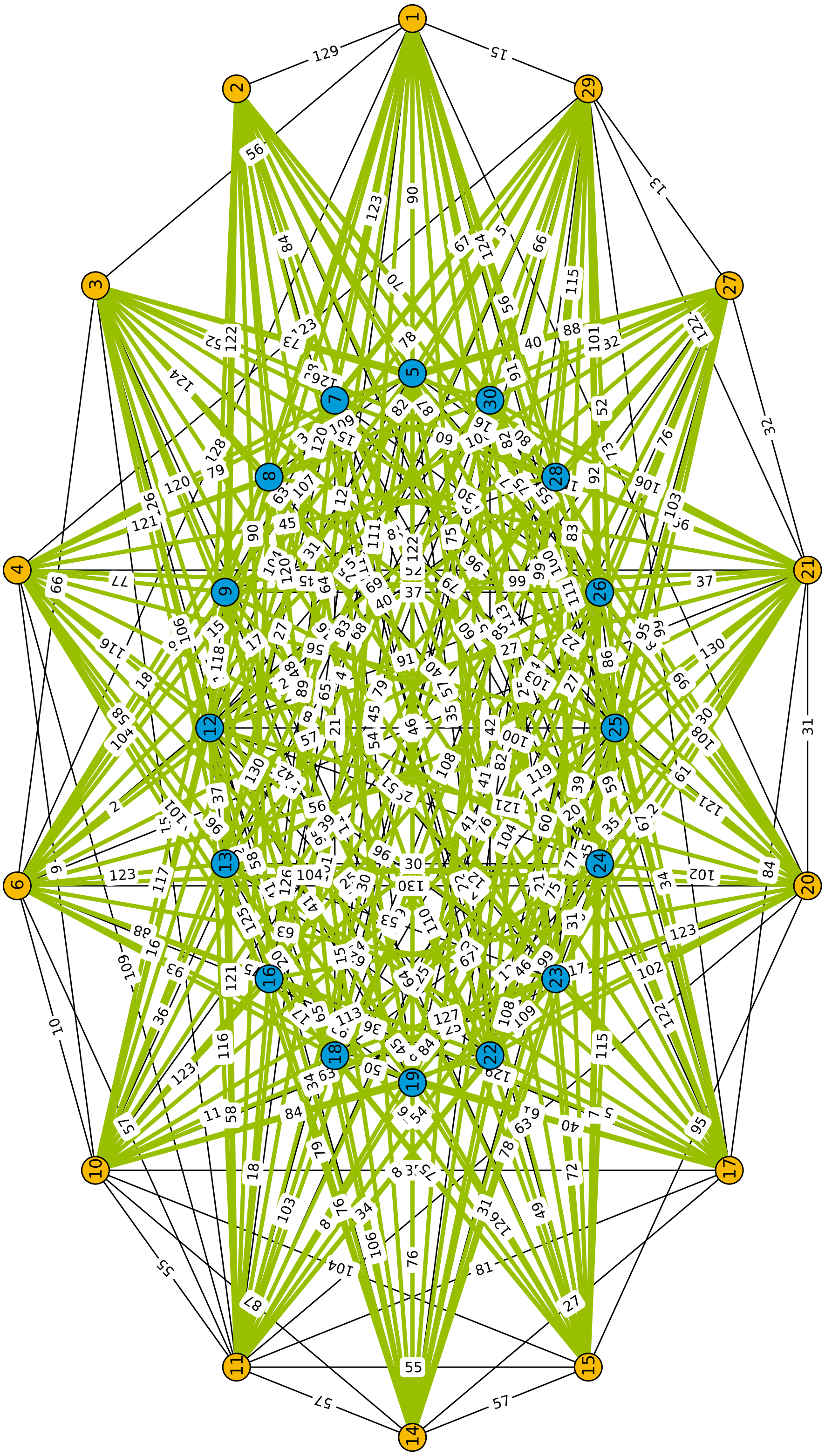


Figure 7.1. – Visualization of the optimal solution of instance mc30-6.

7.2.2. Comparing MIP and MISDP

For the whole test set of one hundred instances we tested the MIP formulation and the pure MISDP formulation. The results of these runs are presented in detail in Tables A.21, A.22, and A.23.

First we give an idea of the size of the instances in Table 7.3. The size only depends on the number of vertices in the graph, so we have five different types of instances. The number of variables is the same for the MIP and the MISDP, only the number of constraints is different. For the MISDP we always have one semidefinite constraint. In the MIP formulation the number grows with the number of vertices.

instance	variables	cons in MISDP	cons in MIP
mc30-*	435	1	16240
mc35-*	595	1	26180
mc40-*	780	1	39520
mc45-*	990	1	56760
mc50-*	1225	1	78400

Table 7.3. – Number of variables and number of constraints in the five different types of instances.

For the small instances, those with 30 and 35 vertices, the MIP formulation, solved using CPLEX 12.4.0.1 [CPL12], is always faster. This can be seen in Table A.21. This table shows all instances where at least one of the models was able to solve the problem instance to optimality within the time limit of two hours. For the middle-sized instances with 40 and 45 vertices there are some cases where the MISDP is faster.

Good solutions for the larger instances (40, 45, and 50 vertices) are also obtained faster by the MIP. We again use our standard time limit of two hours. For the instances where this limit was reached by both models, the gap of the MIP formulation is always smaller than the gap of the MISDP formulation (see Table A.22). However, if we choose a time limit of five hours, one third of the instances can be solved faster using the MISDP approach. These results are presented in Table A.23.

If we increase the time limit the number of instances where the MISDP is able to solve to optimality will also increase, whereas the MIP still has a small gap for most of the cases. The results are summarized in Table 7.4.

	solving as MISDP			solving as MIP		
time limit	# instances solved	smaller gap, if not solved	fastest solver	# instances solved	smaller gap, if not solved	fastest solver
2 hours	60	0	17	54	42	41
5 hours	73	1	29	67	28	42

Table 7.4. – Comparison of the two models for the Maximum Cut Problem for different time limits.

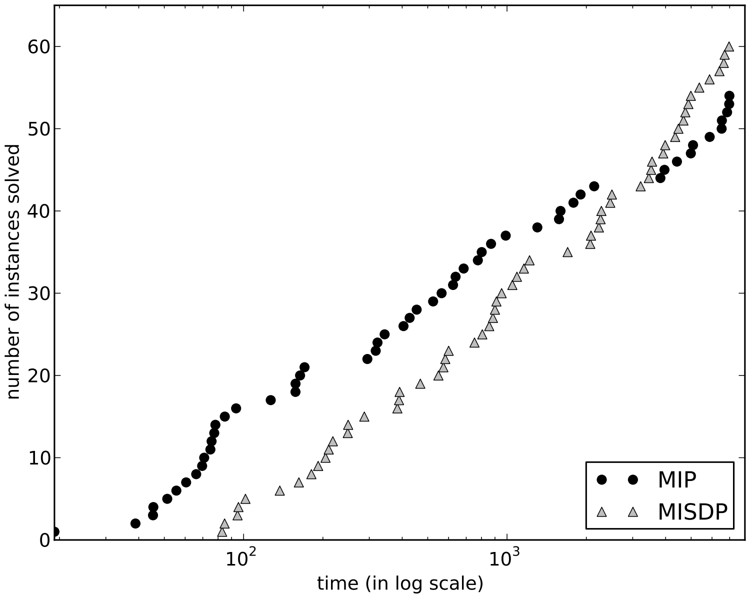


Figure 7.2. – Performance profile for the Maximum Cut Problems modeled as MIP and MISDP for a time limit of two hours.

In this table the first column shows the different time limits. Then, in three columns respectively we present the number of instances in MIP and MISDP formulation that could be solved. Additionally, we show for how many instances the corresponding method produced the smallest gap within the time limit and in each third column we state the number of solving runs in which this method was faster than the other.

This table shows that the MISDP formulation is able to solve more instances in the given time limit. If we want to obtain a good feasible solution in a short time, the MIP is the model of choice. Because it is able to reduce the gap very quickly, it has a feasible solution that is at most a small percentage away from the optimum, whereas the gap in the MISDP is very large for a long time We sum up these results in three performance profiles in Figures 7.2 and 7.3. Note that the time is plotted in logarithmic scale.

The two performance profiles in Figures 7.2 and 7.3 show that the performance of the MIP is better for the short time limit. For the larger time limit the MISDP is able to solve more problems. To demonstrate the big differences between the MIP and the MISDP we additionally show the number of branch-and-bound nodes needed to solve each instance for a time limit of five hours in the Figures 7.4, note that the number of nodes is presented in logarithmic scale. This figure directly shows that the MISDP is able so solve more instances using less nodes than the MIP.

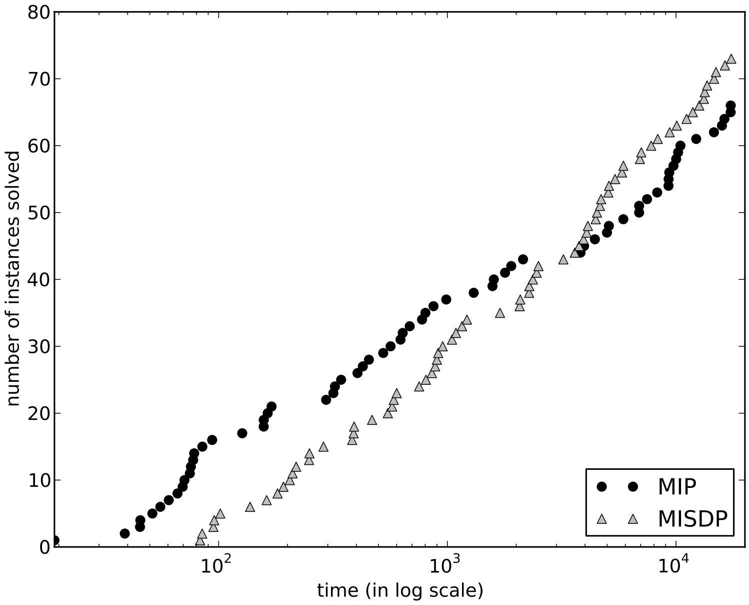


Figure 7.3. – Performance profile for the Maximum Cut Problems modeled as MIP and MISDP for a time limit of five hours.

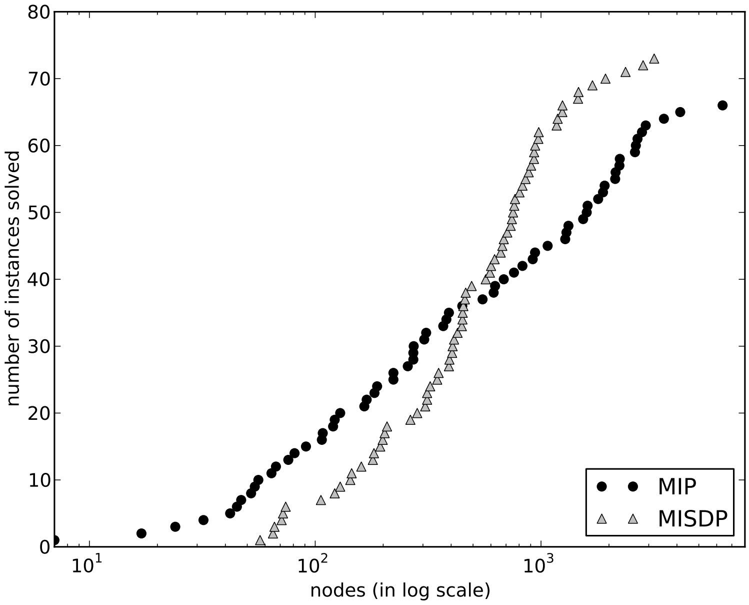


Figure 7.4. – Performance profile for the Maximum Cut Problems modeled as MIP and MISDP for a time limit of five hours in terms of nodes.

7.2.3. Comparing the different solving strategies

As already shown in Chapter 6 the pure MISDP branch-and-bound approach is not always the best choice. In the truss examples the approximation procedure or a combination of MISDP and approximation produces good results. This is completely different in the case of the Maximum Cut Problem.

In Table 7.5 we show that even for the small instances the approximation procedure was not able to solve even one of them within the time limit of two hours. In this table we present results for 20 instances with 30 and 35 vertices. We do not show the results for all the other instances, as they look exactly the same. The second and the third column show the solving time and number of branch-and-bound nodes the MISDP branch-and-bound algorithm needed. It was able to solve all the instances. Recall, that this is the solving procedure where we solve an SDP relaxation in every node of the branch-and-bound tree and do not add any linear cuts. The next three columns present the results of the pure approximation procedure introduced in Section 3.4. For all twenty instances the time limit was reached and the gap is still huge. Additionally, it is possible to use the SDP relaxation only in the root and solve the other nodes using the approximation procedure. The corresponding results are shown in the last three columns and they are similar to the results, where only the approximation procedure is used.

	only MISDP		approximation			approx. + root SDP		
instance	time	nodes	time	nodes	gap [%]	time	nodes	gap [%]
mc30-0	911.55	709.00	7200.00	1647.20	200.82	7200.01	953.33	172.96
mc30-1	94.86	66.00	7200.00	1758.80	225.46	7200.00	1268.67	201.74
mc30-2	95.61	72.00	7200.00	1508.40	205.03	7200.01	1058.67	178.81
mc30-3	191.90	143.00	7200.00	1221.20	219.28	7200.01	1066.83	196.88
mc30-4	248.53	194.00	7200.01	1909.00	208.21	7200.00	1206.33	182.66
mc30-5	137.20	106.00	7200.01	1962.80	218.09	7200.01	1367.83	195.78
mc30-6	249.41	208.00	7200.01	1297.40	208.62	7200.00	955.83	186.99
mc30-7	582.86	464.00	7200.01	1414.60	201.86	7200.00	948.83	178.25
mc30-8	101.63	71.00	7200.01	2180.60	222.84	7200.00	1314.50	196.20
mc30-9	204.62	160.00	7200.00	1793.00	213.95	7200.01	1222.33	188.91
mc35-0	1090.37	404.00	7200.01	769.40	202.39	7200.15	414.60	173.71
mc35-1	855.90	347.00	7200.00	711.40	215.66	7200.01	342.40	180.31
mc35-2	899.95	313.00	7200.01	830.60	229.55	7200.01	352.00	191.98
mc35-3	2070.39	854.00	7200.00	857.20	218.08	7200.02	336.00	181.41
mc35-4	3215.60	1245.00	7200.00	1294.00	188.74	7200.02	518.40	165.80
mc35-5	1047.74	391.00	7200.01	913.60	201.34	7200.01	361.40	175.08
mc35-6	2083.98	767.00	7200.01	700.20	252.27	7200.01	352.20	209.80
mc35-7	389.11	129.00	7200.01	741.80	220.08	7200.00	448.20	186.58
mc35-8	1217.64	493.00	7200.00	739.80	225.49	7200.01	365.80	186.04
mc35-9	752.56	264.00	7200.00	907.40	206.85	7200.00	412.40	179.75

Table 7.5. – Comparison of the pure branch-and-bound algorithm for MISDPs and the approximation procedure.

One reason is for sure that the semidefinite constraint and the constraint matrices within do not have a rich structure. As shown in Section 7.1 there is exactly one entry in the coefficient matrices A_i.

	only MISDP		approx. in root + MISDP			
instance	time	nodes	time	nodes	gap [%]	best
mc30-0	911.55	709	4396.89	744.00	0.00	1
mc30-1	94.86	66	413.43	67.00	0.00	1
mc30-2	95.61	72	402.60	93.00	0.00	1
mc30-3	191.90	143	1173.53	143.00	0.00	1
mc30-4	248.53	194	1586.56	221.00	0.00	1
mc30-5	137.20	106	669.31	106.00	0.00	1
mc30-6	249.41	208	938.39	221.00	0.00	1
mc30-7	582.86	464	1966.69	424.00	0.00	1
mc30-8	101.63	71	384.17	78.00	0.00	1
mc30-9	204.62	160	828.50	163.00	0.00	1
mc30-10	162.12	122	584.62	108.00	0.00	1
mc30-11	180.88	145	735.44	149.00	0.00	1
mc30-12	549.00	453	2733.65	451.00	0.00	1
mc30-13	82.90	57	295.74	59.00	0.00	1
mc30-14	383.51	307	1920.99	307.00	0.00	1
mc30-15	391.07	323	1525.34	292.00	0.00	1
mc30-16	218.20	180	1255.82	194.00	0.00	1
mc30-17	84.71	65	324.42	66.00	0.00	1
mc30-18	287.26	199	1198.72	202.00	0.00	1
mc30-19	600.39	446	2662.85	494.00	0.00	1
mc35-0	1090.37	404	5457.27	477.00	0.00	1
mc35-1	855.90	347	7208.51	628.00	2.55	1
mc35-2	899.95	313	4117.61	313.00	0.00	1
mc35-3	2070.39	854	7203.46	673.80	169.59	1
mc35-4	3215.60	1245	7203.24	752.60	155.68	1
mc35-5	1047.74	391	4656.17	388.00	0.00	1
mc35-6	2083.98	767	7206.92	559.20	190.86	1
mc35-7	389.11	129	1808.40	145.00	0.00	1
mc35-8	1217.64	493	5714.31	493.00	0.00	1
mc35-9	752.56	264	2718.62	265.00	0.00	1
mc35-10	884.18	313	4050.84	313.00	0.00	1
mc35-11	2501.49	977	7205.23	726.20	160.76	1
mc35-12	953.90	352	3568.86	378.00	0.00	1
mc35-13	2281.37	931	7206.10	617.20	168.98	1
mc35-14	468.62	182	2330.78	214.00	0.00	1
mc35-15	1158.87	393	6617.28	555.00	0.00	1
mc35-16	573.46	203	2216.35	182.00	0.00	1
mc35-17	210.48	74	1018.63	77.00	0.00	1
mc35-18	805.69	283	3304.41	280.00	0.00	1

	only MISDP		approx. in root + MISDP			
instance	time	nodes	time	nodes	gap [%]	best
mc35-19	1698.72	663	7208.62	547.40	167.01	1
mc40-0	4675.02	904	7212.70	401.00	156.10	1
mc40-1	4880.23	942	7211.65	363.00	164.56	1
mc40-4	2232.79	406	7209.01	358.80	169.87	1
mc40-5	4348.02	828	7213.26	346.00	165.38	1
mc40-6	3916.87	734	7205.74	334.00	172.27	1
mc40-8	3523.96	743	7213.07	393.20	160.91	1
mc40-9	2265.56	450	7213.38	356.20	170.96	1
mc40-10	3984.38	761	7214.24	319.20	151.52	1
mc40-11	2465.58	449	7212.22	303.40	158.98	1
mc40-12	4986.98	882	7208.18	367.60	165.19	1
mc40-14	3454.60	674	7205.71	338.20	162.60	1
mc40-17	3552.70	684	7211.30	305.00	156.96	1
mc40-19	5369.73	973	7210.67	348.00	161.44	1
mc45-1	4473.02	427	7224.38	132.20	163.27	1
mc45-9	4751.70	412	7224.05	188.40	156.67	1
mc45-17	6702.10	594	7222.09	166.40	168.21	1
mc45-19	6403.76	623	7217.93	161.00	159.88	1

Table 7.6. – Comparing MISDP branch-and-bound to additionally using approximation in the root node. Results for the instances which MISDP branch-and-bound was able to solve within a time limit of two hours. For these instances pure MISDP branch-and-bound is always the fastest.

Furthermore, it is possible to use the approximation procedure in the root node only. The results for this comparison are presented in Table 7.6 for the instances that could be solved to optimality within the time limit of two hours using the pure MISDP branch-and-bound algorithm. The second and third columns show again the solving time and the number of nodes the MISDP approach needs. The next three columns present the results for using the approximation only in the root node. The last column indicates which of the three procedures was faster, where '*1*' stands for the pure MISDP approach. This one is always the fastest.

The bigger instances that could not be solved within the time limit are shown in Table 7.7. Again we compare the two different approaches discussed above. Which of the two procedures yields the best gap is shown in the last column and again the pure MISDP approach is always the best choice. Its gap after two hours of solving is always smaller than the gap the other procedure produced.

	only MISDP			approx. in root + MISDP			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
mc40-2	7202.90	1426.00	163.42	7213.75	408.00	166.48	1
mc40-3	7202.71	1549.00	158.13	6041.28	305.40	161.27	1
mc40-7	7202.93	1350.00	159.50	7205.30	373.40	162.43	1
mc40-13	6649.80	1145.80	32.04	7210.55	282.20	163.18	1
mc40-15	7202.06	1440.60	159.14	7208.62	350.40	161.42	1
mc40-16	7202.41	1346.20	157.74	7206.06	375.20	160.23	1
mc40-18	7203.31	1459.60	157.86	7210.92	344.60	160.91	1
mc45-0	7205.84	680.80	153.85	7222.73	169.80	155.50	1
mc45-2	7204.59	734.80	157.10	7226.19	180.20	159.25	1
mc45-3	7204.64	825.40	158.76	7226.37	192.60	161.43	1
mc45-4	7206.62	696.40	153.59	7209.26	192.00	155.64	1
mc45-5	7208.24	672.80	156.48	7219.76	168.40	159.40	1
mc45-6	7205.73	757.40	162.42	7218.34	177.00	165.51	1
mc45-7	7206.40	687.80	165.50	7216.96	196.20	168.75	1
mc45-8	7205.37	735.80	157.33	7218.91	153.40	159.74	1
mc45-10	7207.61	756.40	160.85	7224.89	156.60	163.61	1
mc45-11	5872.94	560.20	31.92	7225.40	152.40	161.75	1
mc45-12	7205.25	745.80	170.82	7222.06	157.00	173.87	1
mc45-13	7205.37	749.60	160.75	7229.24	141.40	164.54	1
mc45-14	7204.86	758.00	159.04	7220.58	154.40	161.09	1
mc45-15	7206.99	843.00	158.47	7232.92	154.60	161.12	1
mc45-16	7206.42	754.20	161.16	7229.52	165.40	164.12	1
mc45-18	6965.07	703.20	66.11	7217.18	128.20	167.38	1
mc50-0	7208.86	418.20	162.81	7268.32	82.20	165.30	1
mc50-1	7211.28	394.00	156.86	7257.61	95.20	159.11	1
mc50-2	7206.04	428.00	157.59	7232.57	95.20	159.64	1
mc50-3	7207.13	390.00	162.61	7262.78	90.00	165.42	1
mc50-4	7204.86	472.60	149.06	7225.20	92.00	150.83	1
mc50-5	7206.66	468.20	154.63	7253.87	93.20	156.68	1
mc50-6	7217.39	415.20	152.80	7246.73	101.80	154.72	1
mc50-7	7208.33	496.20	155.59	7238.94	108.80	157.83	1
mc50-8	7206.46	478.40	153.03	7228.70	102.40	154.55	1
mc50-9	7207.15	470.00	152.71	7245.12	115.80	154.25	1
mc50-10	7208.08	461.80	154.77	7242.10	82.60	156.99	1
mc50-11	7211.03	463.00	154.26	7288.17	69.60	156.93	1
mc50-12	7209.88	424.00	158.76	7231.35	62.40	161.75	1
mc50-13	7205.75	429.80	159.98	7267.99	75.20	163.16	1
mc50-14	7207.56	442.80	151.73	7299.63	74.00	154.15	1
mc50-15	7214.48	444.40	152.60	7263.70	70.80	155.25	1
mc50-16	7212.39	415.00	161.38	7251.09	65.00	163.73	1
mc50-17	7213.16	356.60	161.50	7285.60	75.40	165.40	1
mc50-18	7208.45	459.40	160.39	7256.38	71.00	163.36	1

	only MISDP			approx. in root + MISDP			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
mc50-19	7208.77	438.60	149.16	7257.30	97.20	151.25	1

Table 7.7. – Comparing MISDP branch-and-bound to additionally using approximation in the root node. Results for the instances wich MISDP branch-and-bound could not solve.

7.2.4. Differences between the Maximum Cut Problem and Truss Topology Design

As we have already seen in the previous section the behavior of max-cut instances is different from the behavior we had seen for truss problems. For truss problems it is not clear which of the presented solution procedures is the best. This is completely different for max-cut problems: the pure MISDP branch-and-bound is always the fastest method.

However, these are not the only differences we can observe. In Table 7.8 we present aggregated information about the quality of the root relaxation and the first solution. The detailed results for each instance can be found in Table A.24. For problems from Truss Topology Design we already stated such a table (see Table 6.11). These two tables demonstrate big differences in the solving runs.

For the Maximum Cut Problem we looked at 71 instances, all instances that could be solved by the MISDP within five hours. Again the lower bound given by the relaxation solution of the root node of the branch-and-bound tree is of good quality. The maximum gap over all 71 instances is 6.9%. For one instance the gap between the root solution and the optimal solution is only 1.8%. This is the smallest root gap obtained for all instances and this is similar to the results we presented for trusses. Completely different from trusses – at least for the minimum compliance formulation – is the quality of the first solution found within the solving process.

This first feasible solution is always found in the root node by the SCIP-heuristic called trivial and is of very poor quality. Its gap to the optimal solution is between 150% and 193.4%. For trusses this gap was smaller in most of the cases for the minimum compliance formulation. For the minimum volume model this gap was also very large. For many truss examples the first solution found is already the optimal solution. The detailed results we already presented in Chapter 6.

As finding feasible solutions is so easy for the Maximum Cut Problem our heuristic does not help, therefore we do not present computational results for it. Additionally, we do not show the effects of presolving in this context because we have already seen that there is insufficient structure in the coefficient matrices. Therefore, all our presolving ideas would result in trivial inequalities.

Another interesting fact that can be seen in Table A.24 is that the branch-and-bound algorithm terminates in more than 56% of the cases immediately if the optimal solution is found. In 40 of the 71 instances the solving process stops immediately. However, there are instances where 26.15% of the nodes are processed after the optimal solution is found to prove optimality of the solution.

instance	geometric mean root sol.	geometric mean first sol.	min. nodes best sol. found	max. nodes best sol. found	min. overall nodes	max. overall nodes	geom. mean nodes after best sol.
mc30-*	2.164	1.097	48	709	57	709	1.633
mc35-*	1.547	1.090	74	1245	74	1245	1.548
mc40-*	1.321	1.171	406	2368	406	2368	1.735
mc45-*	1.242	1.042	402	1466	412	1466	1.386
mc50-*	1.305	1.036	460	803	460	803	1.386
overall	2.105	1.113	74	2368	57	2368	1.580

Table 7.8. – Summary of the geometric means for the relative root solution gap, the first solution gap an the nodes needed after the optimal solution is found.

7.2.5. A note on some statistics

As already done in Section 6.8 we want to comment on the statistics of the solving runs. Again we did all the computations five times and presented the arithmetic means of the running time and the number of nodes in all the previous tables. In this section we want to show how much the running times and the number of nodes (needed until the time limit is reached) can vary.

As we already presented a similar detailed examination in the previous chapter for the problems from Truss Topology Design, we now only want to give an overview. We summarized the differences in Table 7.9. Additionally, we show the summary for the solving runs with a time limit of five hours in Table 7.10.

Comparing these two tables we can conclude that the absolute difference for the solving time, the number of nodes needed, and the gap, if the time limit is reached, decreases if the time limit increases. Especially the MISDP code could reduce the differences in the solving runs. This might be due to the fact that for solving the MISDP there are not so many nodes needed and therefore most of the time is spend solving the SDP. Each time a new node is processed and solved, the data of this specific node is needed from the memory and as the different cores share the same memory this access to the memory makes the different solving runs sometimes faster and sometimes slower because it is not clear what the other solving runs being processed by the same processor are doing.

Therefore the more nodes are needed the more the solving time will vary for each node. As MISDPs do not need as much nodes as MIPs their variation is smaller and it gets even smaller if the time limit is increased.

The details for all test runs and the results for each instance can be found in Tables A.25 and A.26 for the MIP and in Tables A.27 and A.28 for the MISDP.

	time		nodes		gap	
model	max. diff. [%]	geometric mean of diff.	max. diff. [%]	geometric mean of diff.	max. diff. [%]	geometric mean of diff.
MIP	52.02	21.98	87.76	27.32	46.89	5.70
MISDP	41.89	17.76	47.86	22.75	0.54	0.25

Table 7.9. – Variations in the different solving runs with time limit two hours, comparing MIP and MISDP.

	time		nodes		gap	
model	max. diff. [%]	geometric mean of diff.	max. diff. [%]	geometric mean of diff.	max. diff. [%]	geometric mean of diff.
MIP	60.00	19.70	49.12	21.82	40.23	4.56
MISDP	31.12	16.38	29.61	16.80	0.28	0.17

Table 7.10. – Variations in the different solving runs with time limit five hours, comparing MIP and MISDP.

CHAPTER 8

Conclusion

Many applications and combinatorial optimization problems can be modeled using mixed-integer semidefinite programs (MISDP). Within this thesis we presented two of them. We extensively discussed the problem of Truss Topology Design and also analyzed the Maximum Cut Problem. We extended the standard SDP for Truss Topology Design to contain vibrations, actuator positioning, multiple loads, and discrete cross-sectional areas and obtained a MISDP. Additionally, we stated a mixed-integer linear model (MIP) for optimizing the topology of trusses and compared it to the MISDP.

Solving general MISDPs was not possible using any before available solver. This is why we implemented a software package that is able to solve MISDP using SCIP as a the branch-and-bound framework and the SDP solver DSDP for solving the relaxations. This software is able to solve general MISDPs and already has some features to speed up the solving process. Furthermore, it is able to solve MISDPs without using an SDP solver.

Using the application of Truss Topology Design and the Maximum Cut Problem we extensively analyzed the solving process of our software. We compared different strategies for solving MISDPs, evaluated the features of our software and discussed different parameter settings. Additionally, we compared the MISDPs for the two kinds of problems with a MIP formulation and we concluded that small problems can be solved using a MIP but for larger problems the MISDP is faster.

Our software is able to solve general MISDPs. Hence it does not take the structure of a specific kind of problem into account. For example, it cannot compete with special max-cut solvers. However, adding problem specific cuts to our solver can easily be done, as we implemented it as a plugin for SCIP. Our software is constructed modularly and therefore extensions can be added easily.

Thus integrating another SDP solver to our solver interface is not complicated. This is one of the things we want to do in the future. We introduced three different algorithms for solving SDPs: interior-point-methods, the augmented Lagrangian method, and the spectral bundle method. In our case we only implemented an interface to an interior-point-method due to algorithmic problems for the other two methods. It would be very interesting if the other two methods could be included in a modified version that is able to deal with unboundedness and infeasibility.

Moreover, we have seen that solving one SDP relaxation in a node of the branch-and-bound tree takes a lot of time, but in return we do not have to solve many nodes. This is why it would be a good idea to parallelize the node solving process. There are parallel SDP solvers available that should be tested. For example, the parallel version of SDPA would be a good choice. Another advantage of this code is that it is able to exploit the problem structure to speed up the solving process.

Detecting some kinds of problem structure is also one of our future tasks. We want to use this structure to split large SDP blocks into smaller ones and to be able to handle our data more efficiently. However, most of the solving time is spent solving the SDP relaxation – more than 96% for all instances we tested. This is why speeding up the solving of the nodes or detecting infeasibility of the nodes before solving started should be in the main focus.

To decrease the number of required nodes and because finding a feasible solution for the truss problems seems to be hard for the solver, we want to implement more heuristics. The first idea is to generalize the heuristics for MIPs to general MISDPs.

Additionally to the further development of the software, the application of Truss Topology Design could be extended, too. We considered vibration constraints and actuator positioning apart from each other because the actuators we use to not affect vibrations in a truss. Indeed, there are actuators that can absorb vibrations. This is why considering these special actuators and modeling them for the MISDP is another future task.

For problems from Truss Topology Design, our solver works already very well because there only linear and semidefinite constraints appear. However, the solver can be extended to deal with applications with more difficult nonlinear constraints. Due to its modular structure and the modular structure of SCIP, only the new constraint has to be added.

APPENDIX A

Tables

In this appendix we present all the tables that were too long to fit in the Chapters 6 and 7. To present the different tables we use the same structure of sections as in the different chapters.

All computational results presented in this thesis are done using a Sun Fire X4600 M2 with 8x8384-Opteron 2.7 Ghz processors. This machine has 32 cores and 320 GB of RAM. We used SCIP 3.0 with CPLEX 12.4.0.1 and DSDP 5.8 for the MISDP solving runs and CPLEX 12.4.0.1 for the MIPs.

Remark A.0.1. *Moreover, we want to mention that all results presented here are arithmetic means of five different solving runs for each instance and each solving strategy. This is why it is possible that an instance with a solving time smaller than* 7200 *seconds has a positive gap, anyhow. This is due to the fact that this instance was possibly solved one or more times and hit the time limit in the other runs.*

A.1 Results for Truss Topology Design

The detailed results for the different models for Truss Topology Design are presented in this chapter. For all computational results for truss problems the solving time is always presented in seconds and the gap is in percent. Additionally, for the truss instances we always have a time limit of two hours.

The test set has 528 instances. We do not use all the instances in all tables because sometimes we want to compare special features or components and therefore only one part of the instances is interesting.

A.1.1. MISDP Features

In this section we present the different Tables for testing the features our software has.

Approximation

First we want to compare the different solving strategies for MISDP. We contrast pure branch-and-bound for MISDP with the approximation procedure that does not use an SDP solver. In Table A.1 we compare both strategies in solving time, nodes needed, and gap. In the last column we show which of the two strategies was better for this instance: '*A*' stands for the approximation procedure, which approximates the SDP constraints using eigenvalue cuts, and '*S*' for the branch-and-bound where SDPs are solved in all branching nodes and no additional cuts are added. We compared 526 instances.

	solving as MISDP			solving with approximation			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
bridge-1	5.04	79.00	0.00	0.81	283.00	0.00	A
bridge-1-act	20.72	130.00	0.00	8.35	6576.00	0.00	A
bridge-1-3scen	345.54	2300.00	0.00	24.87	11266.00	0.00	A
bridge-1-2scen	232.17	2299.00	0.00	17.51	8234.00	0.00	A
bridge-1-act-mV	355.10	2576.00	0.00	16.70	16890.00	0.00	A
bridge-1-cont	0.08	1.00	0.00	0.07	1.00	0.00	A
bridge-1-cont-act	0.82	3.00	0.00	2.41	529.00	0.00	S
bridge-1-mV	35.94	847.00	0.00	3.34	2778.00	0.00	A
bridge-1-act-2scen	797.30	2158.00	0.00	223.27	126809.00	0.00	A
bridge-1-act-2scen-mV	696.68	2436.00	0.00	5877.60	3639122.00	0.00	S
bridge-1-2scen-mV	288.33	3773.00	0.00	11.82	8416.00	0.00	A
bridge-1-cont-2scen	0.18	1.00	0.00	0.32	1.00	0.00	S
bridge-1-cont-act-2scen	10.63	26.00	0.00	6.55	451.00	0.00	A
bridge-2	105.32	733.00	0.00	1.15	471.00	0.00	A
bridge-2-act	284.27	1124.00	0.00	9.31	6435.00	0.00	A
bridge-2-act-int	7200.07	112487.40	4.53	298.75	262732.00	0.00	A
bridge-2-3scen	7200.14	28564.80	45.04	6887.03	2790184.80	0.12	A
bridge-2-2scen	7200.08	43287.20	42.14	7200.01	2873775.60	1.35	A
bridge-2-act-3scen-mV-int	4906.39	9005.00	0.00	7200.13	2771086.00	4.96	S
bridge-2-int	36.62	645.00	0.00	2.19	1362.00	0.00	A
bridge-2-mV-int	17.06	302.00	0.00	4.34	4891.00	0.00	A
bridge-2-act-mV-int	157.85	918.00	0.00	7200.13	5663605.00	1.27	S
bridge-2-act-mV	7.55	29.00	0.00	1.14	370.00	0.00	A
bridge-2-cont	0.08	1.00	0.00	0.07	1.00	0.00	A
bridge-2-cont-act	0.83	3.00	0.00	2.38	529.00	0.00	S
bridge-2-mV	69.18	1015.00	0.00	9.84	18596.00	0.00	A
bridge-2-act-2scen	7200.16	13911.00	–	7200.03	1931721.80	9.17	A
bridge-2-act-2scen-int	7200.12	35691.00	105673.87	898.99	340808.00	0.00	A
bridge-2-act-2scen-mV	1437.06	3728.00	0.00	7200.04	4118716.20	1.44	S
bridge-2-2scen-mV	1349.46	13544.00	0.00	321.41	369042.00	0.00	A
bridge-2-2scen-int	431.15	3964.00	0.00	42.70	15345.00	0.00	A
bridge-2-2scen-mV-int	229.01	2254.00	0.00	33.17	29990.00	0.00	A
bridge-2-cont-2scen	0.17	1.00	0.00	0.32	1.00	0.00	S

	solving as MISDP			solving with approximation			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
bridge-2-cont-act-2scen	10.29	26.00	0.00	6.50	451.00	0.00	A
bridge-2-act-2scen-mV-int	3348.08	9062.00	0.00	7200.17	3343257.80	4.56	S
bridge-3	40.62	362.00	0.00	12.97	10633.00	0.00	A
bridge-3-act	190.01	712.00	0.00	416.06	399442.00	0.00	S
bridge-3-act-int	951.00	10108.00	0.00	1277.44	1057944.00	0.00	S
bridge-3-3scen	2670.00	10531.00	0.00	947.57	893938.00	0.00	A
bridge-3-2scen	1861.59	10533.00	0.00	1124.12	1100874.00	0.00	A
bridge-3-act-3scen-mV-int	3195.60	5819.00	0.00	7200.19	3193843.60	3.70	S
bridge-3-int	68.91	1149.00	0.00	5.49	3129.00	0.00	A
bridge-3-mV-int	4.64	89.00	0.00	3.58	7019.00	0.00	A
bridge-3-act-mV-int	263.65	1570.00	0.00	7200.40	6497359.00	2.47	S
bridge-3-act-mV	53.05	247.00	0.00	105.32	123699.00	0.00	S
bridge-3-cont	0.10	1.00	0.00	0.08	1.00	0.00	A
bridge-3-cont-act	0.86	3.00	0.00	2.72	731.00	0.00	S
bridge-3-mV	86.10	1014.00	0.00	23.04	48128.00	0.00	A
bridge-3-act-2scen	5658.01	11733.00	0.00	7200.07	2002544.40	4.21	S
bridge-3-act-2scen-int	7200.14	33115.80	–	5167.04	2918134.00	0.00	A
bridge-3-act-2scen-mV	1859.06	5119.00	0.00	7200.14	3956849.60	3.88	S
bridge-3-2scen-mV	2174.15	16846.00	0.00	670.96	844187.00	0.00	A
bridge-3-2scen-int	295.00	2537.00	0.00	42.88	20271.00	0.00	A
bridge-3-2scen-mV-int	40.35	390.00	0.00	1.32	487.00	0.00	A
bridge-3-cont-2scen	0.17	1.00	0.00	0.30	1.00	0.00	S
bridge-3-cont-act-2scen	5.09	11.00	0.00	8.84	508.00	0.00	S
bridge-3-act-2scen-mV-int	2027.43	5821.00	0.00	7200.30	4370823.80	4.03	S
bridge-4	4.99	79.00	0.00	0.83	283.00	0.00	A
bridge-4-act	20.78	114.00	0.00	22.08	12399.00	0.00	S
bridge-4-3scen	343.13	2300.00	0.00	23.13	11266.00	0.00	A
bridge-4-2scen	228.23	2299.00	0.00	17.51	8234.00	0.00	A
bridge-4-act-mV	12.62	81.00	0.00	60.90	67089.00	0.00	S
bridge-4-cont	0.08	1.00	0.00	0.07	1.00	0.00	A
bridge-4-cont-act	0.74	3.00	0.00	5.38	1196.00	0.00	S
bridge-4-mV	24.05	555.00	0.00	3.02	2786.00	0.00	A
bridge-4-act-2scen	835.05	2563.00	0.00	1338.13	823944.00	0.00	S
bridge-4-act-2scen-mV	840.77	3072.00	0.00	7200.06	3941220.40	2.22	S
bridge-4-2scen-mV	198.07	2592.00	0.00	9.25	6199.00	0.00	A
bridge-4-cont-2scen	0.17	1.00	0.00	0.31	1.00	0.00	S
bridge-4-cont-act-2scen	9.49	21.00	0.00	9.23	1097.00	0.00	A
bridge-5	102.84	733.00	0.00	1.15	471.00	0.00	A
bridge-5-act	293.04	1179.00	0.00	15.67	11245.00	0.00	A
bridge-5-act-int	7200.05	106126.60	30.48	1172.88	1132202.00	0.00	A
bridge-5-3scen	7200.12	28478.80	45.07	6704.88	2854464.00	0.00	A
bridge-5-2scen	7200.07	43462.60	42.12	7200.01	2912701.60	1.25	A
bridge-5-act-3scen-mV-int	5708.07	10530.00	0.00	7200.14	2411765.40	7.98	S
bridge-5-int	36.22	645.00	0.00	2.09	1362.00	0.00	A
bridge-5-mV-int	16.81	302.00	0.00	4.30	4891.00	0.00	A
bridge-5-act-mV-int	2042.50	13175.00	0.00	7200.22	5539530.40	2.07	S
bridge-5-act-mV	7.04	27.00	0.00	2.21	889.00	0.00	A
bridge-5-cont	0.08	1.00	0.00	0.07	1.00	0.00	A
bridge-5-cont-act	0.72	3.00	0.00	5.33	1196.00	0.00	S

	solving as MISDP			solving with approximation			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
bridge-5-mV	68.17	1015.00	0.00	9.30	18596.00	0.00	A
bridge-5-act-2scen	7200.34	13934.60	–	7200.03	1724581.60	8.23	A
bridge-5-act-2scen-int	7200.16	33020.80	99341.99	7200.01	2582693.00	0.69	A
bridge-5-act-2scen-mV	1961.70	5686.00	0.00	7200.15	3147303.20	7.01	S
bridge-5-2scen-mV	1346.56	13544.00	0.00	326.93	369042.00	0.00	A
bridge-5-2scen-int	429.55	3964.00	0.00	43.12	15345.00	0.00	A
bridge-5-2scen-mV-int	227.79	2254.00	0.00	31.92	29990.00	0.00	A
bridge-5-cont-2scen	0.17	1.00	0.00	0.32	1.00	0.00	S
bridge-5-cont-act-2scen	9.31	21.00	0.00	9.16	1097.00	0.00	A
bridge-5-act-2scen-mV-int	3601.90	10558.00	0.00	7200.20	3160493.40	8.51	S
bridge-6	40.33	362.00	0.00	12.94	10633.00	0.00	A
bridge-6-act	209.19	800.00	0.00	487.01	471567.00	0.00	S
bridge-6-act-int	7200.02	104381.60	19.52	3425.60	3018956.00	0.00	A
bridge-6-3scen	2667.76	10531.00	0.00	997.26	893938.00	0.00	A
bridge-6-2scen	1814.74	10533.00	0.00	1131.03	1100874.00	0.00	A
bridge-6-act-3scen-mV-int	2964.04	5453.00	0.00	7200.24	3509216.40	5.98	S
bridge-6-int	68.87	1149.00	0.00	5.40	3129.00	0.00	A
bridge-6-mV-int	16.31	337.00	0.00	14.80	31074.00	0.00	A
bridge-6-act-mV-int	1146.08	7105.00	0.00	7200.37	6118576.80	1.99	S
bridge-6-act-mV	124.55	584.00	0.00	491.40	539004.00	0.00	S
bridge-6-cont	0.10	1.00	0.00	0.08	1.00	0.00	A
bridge-6-cont-act	0.61	2.00	0.00	5.46	1535.00	0.00	S
bridge-6-mV	43.68	501.00	0.00	14.57	27526.00	0.00	A
bridge-6-act-2scen	6195.92	13265.00	0.00	7200.07	2327342.00	4.93	S
bridge-6-act-2scen-int	7200.12	30386.40	–	7200.02	3966772.00	2.66	A
bridge-6-act-2scen-mV	1721.03	4825.00	0.00	7200.13	3716080.00	3.09	S
bridge-6-2scen-mV	1323.15	10086.00	0.00	657.85	769393.00	0.00	A
bridge-6-2scen-int	289.64	2537.00	0.00	42.72	20271.00	0.00	A
bridge-6-2scen-mV-int	97.66	1009.00	0.00	12.13	8280.00	0.00	A
bridge-6-cont-2scen	0.17	1.00	0.00	0.31	1.00	0.00	S
bridge-6-cont-act-2scen	3.45	7.00	0.00	6.89	1079.00	0.00	S
bridge-6-act-2scen-mV-int	1953.60	5455.00	0.00	7200.24	4021949.00	4.28	S
bridge-7	1432.54	2504.00	0.00	1562.40	1914877.00	0.00	S
bridge-7-act	5908.73	4164.00	0.00	7200.10	2907375.00	2.43	S
bridge-7-act-int	7200.02	33381.80	118.24	7200.12	2340282.60	8.28	A
bridge-7-3scen	7200.36	6987.20	152.85	7200.05	826940.60	116.40	A
bridge-7-2scen	7200.43	8118.20	–	7200.07	4758100.20	1.29	A
bridge-7-act-3scen-mV-int	2664.79	1852.00	0.00	7200.05	873136.60	–	S
bridge-7-int	12.44	118.00	0.00	1.95	744.00	0.00	A
bridge-7-mV-int	1.29	13.00	0.00	0.29	49.00	0.00	A
bridge-7-act-mV-int	255.21	780.00	0.00	7200.22	3359438.20	8.78	S
bridge-7-act-mV	311.16	260.00	0.00	3946.53	1854329.00	0.00	S
bridge-7-cont	0.16	1.00	0.00	0.13	1.00	0.00	A
bridge-7-cont-act	8.33	15.00	0.00	1090.48	101795.00	0.00	S
bridge-7-mV	13.37	38.00	0.00	6.71	3400.00	0.00	A
bridge-7-act-2scen	7201.90	2465.60	–	7200.07	1258568.00	15.42	A
bridge-7-act-2scen-int	7200.22	17251.00	205.52	7200.05	1138793.80	23.23	A
bridge-7-act-2scen-mV	7201.31	3583.40	–	7200.08	1520042.00	26.07	A
bridge-7-2scen-mV	536.99	1358.00	0.00	786.63	765082.00	0.00	S

	solving as MISDP			solving with approximation			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
bridge-7-2scen-int	315.31	1156.00	0.00	244.90	46117.00	0.00	A
bridge-7-2scen-mV-int	69.62	414.00	0.00	18.08	5113.00	0.00	A
bridge-7-cont-2scen	0.33	1.00	0.00	0.25	1.00	0.00	A
bridge-7-cont-act-2scen	31.18	25.00	0.00	239.83	19431.00	0.00	S
bridge-7-act-2scen-mV-int	1176.80	1246.00	0.00	7200.06	1100729.60	–	S
bridge-8	7201.01	4361.40	–	7200.06	865553.20	25.76	A
bridge-8-act	7201.87	1313.80	–	7200.02	371213.20	93.76	A
bridge-8-act-int	7200.48	8372.00	–	7200.02	502432.00	35.50	A
bridge-8-3scen	7200.90	2718.80	–	7200.02	427687.40	–	–
bridge-8-2scen	7201.54	2999.20	–	7200.03	550563.80	291.43	A
bridge-8-act-3scen-mV-int	7201.73	1594.40	–	7200.02	449302.00	–	–
bridge-8-int	1568.68	4834.00	0.00	314.62	48135.00	0.00	A
bridge-8-mV-int	64.08	324.00	0.00	79.58	60771.00	0.00	S
bridge-8-act-mV-int	1840.51	1815.00	0.00	7200.07	1008953.60	–	S
bridge-8-act-mV	7201.33	2186.00	–	7200.06	1028191.60	2440.45	A
bridge-8-cont	0.30	1.00	0.00	4.86	1.00	0.00	S
bridge-8-cont-act	15.20	11.00	0.00	7200.00	5392.20	2.59	S
bridge-8-mV	2010.62	2837.00	0.00	7200.17	2506964.20	454.74	S
bridge-8-act-2scen	7205.50	720.60	–	7200.02	301710.20	–	–
bridge-8-act-2scen-int	7200.95	4410.60	–	7200.03	521275.40	–	–
bridge-8-act-2scen-mV	7203.99	1125.20	–	7200.05	1023853.40	33.66	A
bridge-8-2scen-mV	7200.36	7087.20	4.90	7200.16	2678961.00	528.88	S
bridge-8-2scen-int	2889.06	5038.00	0.00	452.93	71607.00	0.00	A
bridge-8-2scen-mV-int	115.97	328.00	0.00	66.94	37036.00	0.00	A
bridge-8-cont-2scen	0.51	1.00	0.00	6.72	1.00	0.00	S
bridge-8-cont-act-2scen	34.76	11.00	0.00	7200.01	4502.40	2.18	S
bridge-8-act-2scen-mV-int	7201.32	3035.00	0.94	7200.06	1078867.40	–	S
bridge-9	66.65	323.00	0.00	2583.09	5544790.00	0.00	S
bridge-9-act	388.64	1070.00	0.00	7200.32	5414218.00	18.91	S
bridge-9-act-int	5125.62	59718.00	0.00	7200.20	7528110.00	2.22	S
bridge-9-3scen	3747.72	9217.00	0.00	7200.35	5458421.40	19.95	S
bridge-9-2scen	245.43	652.00	0.00	2306.45	3651533.00	0.00	S
bridge-9-act-3scen-mV-int	2148.21	3867.00	0.00	7200.37	3942925.40	–	S
bridge-9-int	57.17	1088.00	0.00	5.22	8067.00	0.00	A
bridge-9-mV-int	16.94	545.00	0.00	20.75	38652.00	0.00	S
bridge-9-act-mV-int	255.01	1867.00	0.00	7200.82	8373192.00	–	S
bridge-9-act-mV	74.36	290.00	0.00	6.96	6283.00	0.00	A
bridge-9-cont	0.10	1.00	0.00	0.08	1.00	0.00	A
bridge-9-cont-act	13.23	82.00	0.00	86.88	10611.00	0.00	S
bridge-9-mV	46.71	498.00	0.00	0.46	719.00	0.00	A
bridge-9-act-2scen	807.06	939.00	0.00	7200.18	4352504.20	12.47	S
bridge-9-act-2scen-int	548.16	3220.00	0.00	7200.19	5666307.20	4.66	S
bridge-9-act-2scen-mV	63.80	147.00	0.00	7.25	4895.00	0.00	A
bridge-9-2scen-mV	36.98	255.00	0.00	0.61	743.00	0.00	A
bridge-9-2scen-int	182.06	1579.00	0.00	5.26	6893.00	0.00	A
bridge-9-2scen-mV-int	18.44	319.00	0.00	6.63	8109.00	0.00	A
bridge-9-cont-2scen	0.18	1.00	0.00	0.10	1.00	0.00	A
bridge-9-cont-act-2scen	22.48	79.00	0.00	479.13	21532.00	0.00	S
bridge-9-act-2scen-mV-int	517.27	1495.00	0.00	7205.78	5296278.80	68.43	S

	solving as MISDP			solving with approximation			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
bridge-10	1102.61	6642.00	0.00	299.44	388985.00	0.00	A
bridge-10-act	720.74	861.00	0.00	7200.10	3506759.60	9.73	S
bridge-10-act-int	7200.10	27943.00	49.87	7200.25	5995477.00	5.67	A
bridge-10-3scen	7200.29	11057.80	4668.82	7200.22	3564426.60	51.09	A
bridge-10-2scen	7189.18	18789.00	0.02	7200.11	5090068.20	3.10	S
bridge-10-act-3scen-mV-int	7200.58	6379.00	–	7200.20	3029778.00	2043.35	A
bridge-10-int	49.82	455.00	0.00	5.94	4074.00	0.00	A
bridge-10-mV-int	130.01	2929.00	0.00	21.28	36716.00	0.00	A
bridge-10-act-mV-int	531.97	1819.00	0.00	7200.47	5885611.60	–	S
bridge-10-act-mV	7200.18	45895.80	51.60	513.03	283171.00	0.00	A
bridge-10-cont	0.14	1.00	0.00	0.14	1.00	0.00	S
bridge-10-cont-act	0.46	1.00	0.00	0.60	11.00	0.00	S
bridge-10-mV	7200.03	81793.00	33.60	109.61	117637.00	0.00	A
bridge-10-act-2scen	7200.93	4566.00	203.30	7200.07	1193143.40	25.93	A
bridge-10-act-2scen-int	7200.10	12193.20	–	7200.06	1710789.60	41.98	A
bridge-10-act-2scen-mV	7200.14	25639.40	–	7200.16	2655729.40	2102.97	A
bridge-10-2scen-mV	7200.07	34747.00	–	7200.54	8037534.00	31.85	A
bridge-10-2scen-int	7200.07	48638.80	16.51	1104.52	392360.00	0.00	A
bridge-10-2scen-mV-int	1339.31	10334.00	0.00	574.17	615927.00	0.00	A
bridge-10-cont-2scen	0.19	1.00	0.00	0.71	1.00	0.00	S
bridge-10-cont-act-2scen	6.12	6.00	0.00	10.63	23.00	0.00	S
bridge-10-act-2scen-mV-int	7200.34	12816.60	1892.98	7200.17	2384196.40	508.24	A
bridge-big	1792.50	1228.00	0.00	7200.00	124412.20	–	S
bridge-big-act	7204.64	865.60	–	7200.01	81196.60	–	–
bridge-big-act-int	7202.14	3016.60	–	7200.01	38853.40	–	–
bridge-big-3scen	7201.24	2535.20	–	7200.01	140728.40	–	–
bridge-big-2scen	7201.16	3132.80	–	7200.02	322668.80	–	–
bridge-big-act-3scen-mV-int	7205.19	1259.50	–	0.14	1.00	–	–
bridge-big-int	7200.22	14620.60	–	7200.00	54635.00	–	–
bridge-big-cont	0.54	1.00	0.00	8.13	1.00	0.00	S
bridge-big-cont-act	40.67	12.00	0.00	7200.00	3907.40	10.26	S
bridge-big-act-2scen	7209.36	430.80	–	7200.01	95050.20	–	–
bridge-big-act-2scen-int	7203.39	1114.20	–	7200.01	42637.60	–	–
bridge-big-act-2scen-mV	7203.51	841.80	–	0.21	1.00	–	–
bridge-big-2scen-mV	1.62	1.00	–	0.05	1.00	–	–
bridge-big-2scen-int	7200.50	8395.20	–	7200.00	50183.80	–	–
bridge-big-2scen-mV-int	0.55	1.00	–	0.02	1.00	–	–
bridge-big-cont-2scen	0.89	1.00	0.00	10.38	1.00	0.00	S
bridge-big-cont-act-2scen	7201.57	2741.80	0.01	7200.01	4901.40	5.59	S
bridge-big-act-2scen-mV-int	7202.58	2581.20	–	0.09	1.00	–	–
canti-1	23.01	620.00	0.00	3.03	5029.00	0.00	A
canti-1-act	106.06	723.00	0.00	53.89	77799.00	0.00	A
canti-1-3scen	534.35	4719.00	0.00	74.82	77573.00	0.00	A
canti-1-2scen	43.88	505.00	0.00	10.65	16406.00	0.00	A
canti-1-act-mV	3.34	23.00	0.00	1.02	572.00	0.00	A
canti-1-cont	0.08	1.00	0.00	0.02	1.00	0.00	A
canti-1-cont-act	0.99	5.00	0.00	0.37	29.00	0.00	A
canti-1-mV	2.37	62.00	0.00	0.67	1089.00	0.00	A
canti-1-act-2scen	177.99	655.00	0.00	102.29	78473.00	0.00	A

	solving as MISDP			solving with approximation			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
canti-1-act-2scen-mV	0.29	1.00	–	0.12	1.00	–	–
canti-1-2scen-mV	0.09	1.00	–	0.01	1.00	–	–
canti-1-cont-2scen	0.13	1.00	0.00	0.28	1.00	0.00	S
canti-1-cont-act-2scen	0.36	1.00	0.00	1.36	9.00	0.00	S
canti-1-m-act-mV	9.57	73.00	0.00	6.00	5814.00	0.00	A
canti-1-m-mV	3.62	103.00	0.00	0.57	623.00	0.00	A
canti-1-m-act-2scen-mV	0.30	1.00	–	0.40	1.00	–	–
canti-1-m-2scen-mV	0.08	1.00	–	0.05	1.00	–	–
canti-2	60.07	698.00	0.00	6.43	10183.00	0.00	A
canti-2-act	182.73	1629.00	0.00	15.02	14721.00	0.00	A
canti-2-act-int	185.98	2695.00	0.00	26.53	27499.00	0.00	A
canti-2-3scen	1767.22	10880.00	0.00	7200.13	6135403.00	5.12	S
canti-2-2scen	669.06	4754.00	0.00	3889.43	5115814.00	0.00	S
canti-2-act-3scen-mV-int	220.09	347.00	0.00	7200.66	7472468.40	–	S
canti-2-int	14.79	339.00	0.00	1.68	1103.00	0.00	A
canti-2-mV-int	2.39	58.00	0.00	1.10	2476.00	0.00	A
canti-2-act-mV-int	18.54	130.00	0.00	10.61	14629.00	0.00	A
canti-2-act-mV	9.33	47.00	0.00	3.48	2705.00	0.00	A
canti-2-cont	0.07	1.00	0.00	0.03	1.00	0.00	A
canti-2-cont-act	0.95	5.00	0.00	0.40	29.00	0.00	A
canti-2-mV	19.45	343.00	0.00	5.76	11179.00	0.00	A
canti-2-act-2scen	1906.94	5111.00	0.00	7200.10	3048329.20	11.30	S
canti-2-act-2scen-int	7200.12	41560.60	1315.22	4739.60	2423752.00	0.00	A
canti-2-act-2scen-mV	1388.02	4528.00	0.00	7200.55	7707285.00	–	S
canti-2-2scen-mV	0.15	1.00	–	0.22	1.00	–	–
canti-2-2scen-int	816.04	14020.00	0.00	66.31	32222.00	0.00	A
canti-2-2scen-mV-int	0.08	1.00	–	0.04	1.00	–	–
canti-2-cont-2scen	0.12	1.00	0.00	0.28	1.00	0.00	S
canti-2-cont-act-2scen	0.37	1.00	0.00	1.38	9.00	0.00	S
canti-2-act-2scen-mV-int	14.90	37.00	0.00	7200.76	11095761.40	–	S
canti-2-act-3scen-mV-int	357.34	776.00	0.00	7200.52	5218587.20	6.09	S
canti-2-m-mV-int	1.93	54.00	0.00	0.08	3.00	0.00	A
canti-2-m-act-mV-int	28.37	221.00	0.00	7.10	7316.00	0.00	A
canti-2-m-act-mV	28.81	166.00	0.00	17.68	18887.00	0.00	A
canti-2-m-mV	6.16	102.00	0.00	0.58	580.00	0.00	A
canti-2-m-act-2scen-mV	333.06	923.00	0.00	7200.12	5178997.40	0.72	S
canti-2-m-2scen-mV	49.13	385.00	0.00	1948.59	2965943.00	0.00	S
canti-2-m-act-2scen-mV-int	166.71	573.00	0.00	7200.64	6886427.40	2.53	S
canti-3	124.15	1121.00	0.00	15.73	32381.00	0.00	A
canti-3-act	256.42	993.00	0.00	432.80	518800.00	0.00	S
canti-3-act-int	242.99	1675.00	0.00	214.98	292155.00	0.00	A
canti-3-3scen	1062.21	4736.00	0.00	3616.18	4754646.00	0.00	S
canti-3-2scen	455.97	2831.00	0.00	534.61	972886.00	0.00	S
canti-3-act-3scen-mV-int	0.40	1.00	–	0.57	1.00	–	–
canti-3-int	11.14	207.00	0.00	0.89	1049.00	0.00	A
canti-3-mV-int	1.57	34.00	0.00	7.20	25346.00	0.00	S
canti-3-act-mV-int	23.18	159.00	0.00	75.34	118666.00	0.00	S
canti-3-act-mV	7.27	32.00	0.00	1.30	719.00	0.00	A
canti-3-cont	0.08	1.00	0.00	0.03	1.00	0.00	A

	solving as MISDP			solving with approximation			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
canti-3-cont-act	0.95	5.00	0.00	0.31	29.00	0.00	A
canti-3-mV	40.15	569.00	0.00	329.11	849340.00	0.00	S
canti-3-act-2scen	1505.96	4274.00	0.00	7200.07	4974080.00	1.29	S
canti-3-act-2scen-int	7200.08	45527.40	49.39	3637.51	2672747.00	0.00	A
canti-3-act-2scen-mV	0.41	1.00	–	0.34	1.00	–	–
canti-3-2scen-mV	0.15	1.00	–	0.17	1.00	–	–
canti-3-2scen-int	888.54	10700.00	0.00	50.01	39174.00	0.00	A
canti-3-2scen-mV-int	0.09	1.00	–	0.04	1.00	–	–
canti-3-cont-2scen	0.13	1.00	0.00	0.30	1.00	0.00	S
canti-3-cont-act-2scen	1.06	2.00	0.00	8.90	81.00	0.00	S
canti-3-act-2scen-mV-int	0.25	1.00	–	0.28	1.00	–	–
canti-3-act-3scen-mV-int	103.17	217.00	0.00	7200.64	5962551.00	6.01	S
canti-3-m-mV-int	3.65	97.00	0.00	3.48	6531.00	0.00	A
canti-3-m-act-mV-int	19.49	145.00	0.00	6.79	6688.00	0.00	A
canti-3-m-act-mV	9.64	52.00	0.00	7.37	7616.00	0.00	A
canti-3-m-mV	8.30	120.00	0.00	1.47	1820.00	0.00	A
canti-3-m-act-2scen-mV	49.27	121.00	0.00	1574.21	1557673.00	0.00	S
canti-3-m-2scen-mV	29.16	216.00	0.00	7201.25	13683437.40	3.90	S
canti-3-m-act-2scen-mV-int	45.01	149.00	0.00	7200.46	7410932.60	0.86	S
canti-4	22.82	620.00	0.00	3.09	5029.00	0.00	A
canti-4-act	58.65	360.00	0.00	75.64	103502.00	0.00	S
canti-4-3scen	535.78	4719.00	0.00	75.91	77573.00	0.00	A
canti-4-2scen	43.86	505.00	0.00	10.52	16406.00	0.00	A
canti-4-act-mV	11.35	86.00	0.00	1.93	1213.00	0.00	A
canti-4-cont	0.07	1.00	0.00	0.03	1.00	0.00	A
canti-4-cont-act	0.19	1.00	0.00	0.61	15.00	0.00	S
canti-4-mV	10.10	267.00	0.00	60.77	154493.00	0.00	S
canti-4-act-2scen	151.59	523.00	0.00	192.21	114800.00	0.00	S
canti-4-act-2scen-mV	0.30	1.00	–	0.14	1.00	–	–
canti-4-2scen-mV	0.08	1.00	–	0.01	1.00	–	–
canti-4-cont-2scen	0.12	1.00	0.00	0.28	1.00	0.00	S
canti-4-cont-act-2scen	0.37	1.00	0.00	0.97	17.00	0.00	S
canti-4-m-act-mV	16.41	131.00	0.00	19.20	21992.00	0.00	S
canti-4-m-mV	3.60	103.00	0.00	0.56	623.00	0.00	A
canti-4-m-act-2scen-mV	0.29	1.00	–	0.39	1.00	–	–
canti-4-m-2scen-mV	0.08	1.00	–	0.05	1.00	–	–
canti-5	60.15	698.00	0.00	6.29	10183.00	0.00	A
canti-5-act	191.21	1655.00	0.00	18.69	16690.00	0.00	A
canti-5-act-int	1865.97	25536.00	0.00	43.36	52521.00	0.00	A
canti-5-3scen	1772.46	10880.00	0.00	7200.13	5594000.00	5.51	S
canti-5-2scen	666.57	4754.00	0.00	3911.01	5115814.00	0.00	S
canti-5-act-3scen-mV-int	7200.36	13104.40	6.37	7200.65	7483601.40	–	S
canti-5-int	14.82	339.00	0.00	1.68	1103.00	0.00	A
canti-5-mV-int	2.12	48.00	0.00	0.52	1264.00	0.00	A
canti-5-act-mV-int	121.63	880.00	0.00	80.50	94876.00	0.00	A
canti-5-act-mV	7.58	37.00	0.00	2.93	1863.00	0.00	A
canti-5-cont	0.08	1.00	0.00	0.03	1.00	0.00	A
canti-5-cont-act	0.19	1.00	0.00	0.62	15.00	0.00	S
canti-5-mV	5.69	88.00	0.00	1.73	2110.00	0.00	A

	solving as MISDP			solving with approximation			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
canti-5-act-2scen	1527.60	4151.00	0.00	7200.08	3621700.60	5.98	S
canti-5-act-2scen-int	7200.05	44512.40	2434.12	7200.02	3752314.40	5.21	A
canti-5-act-2scen-mV	135.63	286.00	0.00	7200.40	7796504.40	–	S
canti-5-2scen-mV	0.16	1.00	–	0.22	1.00	–	–
canti-5-2scen-int	816.87	14020.00	0.00	66.29	32222.00	0.00	A
canti-5-2scen-mV-int	0.09	1.00	–	0.05	1.00	–	–
canti-5-cont-2scen	0.13	1.00	0.00	0.27	1.00	0.00	S
canti-5-cont-act-2scen	0.39	1.00	0.00	0.99	17.00	0.00	S
canti-5-act-2scen-mV-int	7200.10	21183.20	6.41	7200.86	10757623.20	–	S
canti-5-act-3scen-mV-int	1082.54	2286.00	0.00	7200.55	5519582.40	7.84	S
canti-5-m-mV-int	1.94	54.00	0.00	0.07	3.00	0.00	A
canti-5-m-act-mV-int	35.03	287.00	0.00	5.52	4400.00	0.00	A
canti-5-m-act-mV	24.46	196.00	0.00	14.74	14946.00	0.00	A
canti-5-m-mV	6.18	102.00	0.00	0.58	580.00	0.00	A
canti-5-m-act-2scen-mV	192.23	528.00	0.00	1789.28	1327825.00	0.00	S
canti-5-m-2scen-mV	49.14	385.00	0.00	1907.79	2965943.00	0.00	S
canti-5-m-act-2scen-mV-int	243.41	815.00	0.00	7200.63	6655755.40	5.24	S
canti-6	123.61	1121.00	0.00	15.47	32381.00	0.00	A
canti-6-act	212.51	781.00	0.00	337.30	377452.00	0.00	S
canti-6-act-int	2343.84	24420.00	0.00	270.54	371783.00	0.00	A
canti-6-3scen	1070.28	4736.00	0.00	3616.66	4754646.00	0.00	S
canti-6-2scen	456.76	2831.00	0.00	530.89	972886.00	0.00	S
canti-6-act-3scen-mV-int	0.58	1.00	–	0.57	1.00	–	–
canti-6-int	11.17	207.00	0.00	0.87	1049.00	0.00	A
canti-6-mV-int	1.67	38.00	0.00	213.14	567652.00	0.00	S
canti-6-act-mV-int	16.28	112.00	0.00	7200.54	8390352.80	2.30	S
canti-6-act-mV	55.89	301.00	0.00	246.11	290204.00	0.00	S
canti-6-cont	0.07	1.00	0.00	0.03	1.00	0.00	A
canti-6-cont-act	0.19	1.00	0.00	0.67	31.00	0.00	S
canti-6-mV	81.38	1293.00	0.00	1990.35	4189785.00	0.00	S
canti-6-act-2scen	1073.52	2517.00	0.00	7200.03	4550435.40	0.60	S
canti-6-act-2scen-int	7200.09	42176.20	83.96	1862.74	1433562.00	0.00	A
canti-6-act-2scen-mV	0.42	1.00	–	0.36	1.00	–	–
canti-6-2scen-mV	0.16	1.00	–	0.22	1.00	–	–
canti-6-2scen-int	892.59	10700.00	0.00	50.08	39174.00	0.00	A
canti-6-2scen-mV-int	0.09	1.00	–	0.04	1.00	–	–
canti-6-cont-2scen	0.12	1.00	0.00	0.28	1.00	0.00	S
canti-6-cont-act-2scen	0.38	1.00	0.00	10.00	143.00	0.00	S
canti-6-act-2scen-mV-int	0.30	1.00	–	0.27	1.00	–	–
canti-6-act-3scen-mV-int	119.44	238.00	0.00	7200.63	5680687.40	3.95	S
canti-6-m-mV-int	3.65	97.00	0.00	3.50	6531.00	0.00	A
canti-6-m-act-mV-int	4.68	35.00	0.00	2.48	2128.00	0.00	A
canti-6-m-act-mV	13.45	72.00	0.00	13.93	13636.00	0.00	S
canti-6-m-mV	8.31	120.00	0.00	1.51	1820.00	0.00	A
canti-6-m-act-2scen-mV	70.49	178.00	0.00	7200.21	5722510.60	1.22	S
canti-6-m-2scen-mV	29.23	216.00	0.00	7201.16	13833636.20	3.89	S
canti-6-m-act-2scen-mV-int	89.84	271.00	0.00	7200.66	7903603.40	4.03	S
canti-7	605.74	1517.00	0.00	37.76	43012.00	0.00	A
canti-7-act	658.47	495.00	0.00	22.98	14766.00	0.00	A

	solving as MISDP			solving with approximation			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
canti-7-act-int	1373.81	5119.00	0.00	89.53	72554.00	0.00	A
canti-7-3scen	7200.57	7887.00	–	7200.13	2212334.80	17.34	A
canti-7-2scen	7200.21	11006.60	3.58	7200.10	4755489.40	2.24	A
canti-7-act-3scen-mV-int	7200.90	5090.20	4272.15	7200.12	1998101.00	–	S
canti-7-int	32.57	219.00	0.00	3.28	1496.00	0.00	A
canti-7-mV-int	32.86	500.00	0.00	13.06	9802.00	0.00	A
canti-7-act-mV-int	882.18	2403.00	0.00	7200.35	4654032.00	44.28	S
canti-7-act-mV	7200.46	10477.60	–	7200.21	3588723.20	36.95	A
canti-7-cont	0.16	1.00	0.00	0.11	1.00	0.00	A
canti-7-cont-act	1.83	3.00	0.00	0.36	3.00	0.00	A
canti-7-mV	3457.21	37477.00	0.00	4087.47	3281867.00	0.00	S
canti-7-act-2scen	7201.61	3137.80	–	7200.07	1440281.40	16.91	A
canti-7-act-2scen-int	7200.35	13857.20	–	7200.03	2444042.40	3.33	A
canti-7-act-2scen-mV	7200.53	5361.40	–	7200.13	2256051.20	424.68	A
canti-7-2scen-mV	7200.12	23404.40	24.81	7200.61	8560794.40	69.16	S
canti-7-2scen-int	1240.38	5460.00	0.00	98.77	24159.00	0.00	A
canti-7-2scen-mV-int	9.02	93.00	0.00	0.18	149.00	0.00	A
canti-7-cont-2scen	0.24	1.00	0.00	0.40	1.00	0.00	S
canti-7-cont-act-2scen	6.01	5.00	0.00	3.64	53.00	0.00	A
canti-7-act-2scen-mV-int	7200.26	8583.60	1921.40	7200.18	2851879.20	–	S
canti-8	64.53	302.00	0.00	7200.70	8727956.60	79.53	S
canti-8-act	143.33	293.00	0.00	7200.32	3668021.40	51.46	S
canti-8-act-mV	22.83	91.00	0.00	35.27	39504.00	0.00	S
canti-8-cont	0.07	1.00	0.00	2.73	1.00	0.00	S
canti-8-cont-act	0.91	3.00	0.00	363.65	240.00	0.00	S
canti-8-mV	13.76	122.00	0.00	18.86	37530.00	0.00	S
canti-9	254.01	481.00	0.00	18.56	8049.00	0.00	A
canti-9-act	1703.01	1189.00	0.00	678.32	307750.00	0.00	A
canti-9-act-int	7200.10	23209.80	363497.06	7200.08	2291051.00	55.04	A
canti-9-3scen	7200.38	6472.40	57.77	7200.07	1398209.60	49.19	A
canti-9-2scen	7200.44	8963.40	553.08	7200.09	4157208.60	5.53	A
canti-9-act-3scen-mV-int	7200.82	4503.60	–	7200.03	680780.40	225.16	A
canti-9-int	1685.27	12650.00	0.00	153.29	65192.00	0.00	A
canti-9-mV-int	99.65	1909.00	0.00	33.51	35299.00	0.00	A
canti-9-act-mV-int	7200.03	40585.40	128.56	7200.41	5157661.40	126.15	A
canti-9-act-mV	7200.81	9526.60	–	7200.19	3402817.00	36.87	A
canti-9-cont	0.18	1.00	0.00	0.18	1.00	0.00	S
canti-9-cont-act	0.60	1.00	0.00	1.18	19.00	0.00	S
canti-9-mV	3367.77	19632.00	0.00	6103.45	5819634.00	0.00	S
canti-9-act-2scen	7201.73	2481.20	–	7200.02	794006.60	11.76	A
canti-9-act-2scen-int	7200.51	9385.80	–	7200.02	566198.40	–	–
canti-9-act-2scen-mV	7200.99	4291.60	–	7200.15	2601883.20	156.12	A
canti-9-2scen-mV	7200.16	17481.80	51.44	7200.45	7326240.40	83.72	S
canti-9-2scen-int	7200.13	27555.00	–	7200.03	719194.40	–	–
canti-9-2scen-mV-int	54.41	355.00	0.00	0.90	775.00	0.00	A
canti-9-cont-2scen	0.32	1.00	0.00	0.35	1.00	0.00	S
canti-9-cont-act-2scen	1.19	1.00	0.00	1.22	5.00	0.00	S
canti-9-act-2scen-mV-int	7200.33	8032.80	–	7200.08	1255552.00	2189.83	A
canti-10	192.92	409.00	0.00	7.93	11646.00	0.00	A

	solving as MISDP			solving with approximation			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
canti-10-act	888.75	1093.00	0.00	698.91	743374.00	0.00	A
canti-10-act-int	939.82	4054.00	0.00	1176.29	1201179.00	0.00	S
canti-10-3scen	7200.39	10192.80	–	7200.15	2807625.00	9.24	A
canti-10-2scen	7200.21	12704.80	421.00	7200.16	7029535.00	1.61	A
canti-10-act-3scen-mV-int	7200.45	8074.80	–	7200.09	1487428.60	271.44	A
canti-10-int	6.41	46.00	0.00	2.62	4310.00	0.00	A
canti-10-mV-int	51.73	823.00	0.00	0.76	1005.00	0.00	A
canti-10-act-mV-int	869.97	3356.00	0.00	642.86	652460.00	0.00	A
canti-10-act-mV	7200.09	38623.20	–	888.93	485276.00	0.00	A
canti-10-cont	0.14	1.00	0.00	0.08	1.00	0.00	A
canti-10-cont-act	3.71	8.00	0.00	18.35	2120.00	0.00	S
canti-10-mV	2968.92	35012.00	0.00	43.01	58444.00	0.00	A
canti-10-act-2scen	7200.78	4654.60	475.07	7200.05	2191698.00	2.51	A
canti-10-act-2scen-int	7200.11	25469.00	144.82	3942.95	2133927.00	0.00	A
canti-10-act-2scen-mV	7200.60	6881.40	–	7200.15	2542168.40	520.10	A
canti-10-2scen-mV	7200.17	25858.20	–	7200.28	6305739.60	49.31	A
canti-10-2scen-int	2382.31	12917.00	0.00	28.43	29566.00	0.00	A
canti-10-2scen-mV-int	1.82	20.00	0.00	0.18	220.00	0.00	A
canti-10-cont-2scen	0.21	1.00	0.00	0.37	1.00	0.00	S
canti-10-cont-act-2scen	18.94	25.00	0.00	47.47	1757.00	0.00	S
canti-10-act-2scen-mV-int	7200.31	13706.60	2593.77	7200.22	3296624.80	135.80	A
canti-big	1096.81	1273.00	0.00	31.08	32056.00	0.00	A
canti-big-act	5409.97	2225.00	0.00	7200.03	2887491.80	0.40	S
canti-big-act-int	7200.36	11657.80	79.97	7200.12	4170821.60	5.21	A
canti-big-3scen	7201.23	3037.60	–	7200.05	968583.80	17.72	A
canti-big-2scen	7201.09	3618.80	–	7200.11	2815116.20	5.05	A
canti-big-act-3scen-mV-int	7201.69	1781.60	–	7200.03	644583.00	214.30	A
canti-big-int	118.57	432.00	0.00	2.82	1376.00	0.00	A
canti-big-mV-int	80.54	693.00	0.00	18.81	10931.00	0.00	A
canti-big-act-mV-int	387.04	511.00	0.00	3253.43	971328.00	0.00	S
canti-big-act-mV	7201.02	3169.60	–	7200.09	2569215.00	62.16	A
canti-big-cont	0.31	1.00	0.00	0.22	1.00	0.00	A
canti-big-cont-act	4.00	2.00	0.00	1.17	5.00	0.00	A
canti-big-mV	7200.09	28768.40	3.82	7200.66	8230343.40	18.27	S
canti-big-act-2scen	7204.02	783.60	–	7200.04	791930.80	15.29	A
canti-big-act-2scen-int	7200.93	3276.80	–	7200.05	918741.40	–	–
canti-big-act-2scen-mV	7201.39	1520.00	–	7200.07	1360736.20	3526.21	A
canti-big-2scen-mV	7200.50	7355.60	–	7200.38	5873697.00	119.34	A
canti-big-2scen-int	7200.15	15132.20	–	193.53	40560.00	0.00	A
canti-big-2scen-mV-int	6.40	29.00	0.00	0.05	21.00	0.00	A
canti-big-cont-2scen	0.51	1.00	0.00	0.77	1.00	0.00	S
canti-big-cont-act-2scen	14.68	5.00	0.00	6.30	31.00	0.00	A
canti-big-act-2scen-mV-int	7201.44	3427.60	–	7200.04	670048.20	2077.90	A
lit-as-1	1.68	85.00	0.00	0.22	273.00	0.00	A
lit-as-1-act	5.03	79.00	0.00	0.65	300.00	0.00	A
lit-as-1-act-mV	8.55	233.00	0.00	10.55	12737.00	0.00	S
lit-as-1-cont	0.02	1.00	0.00	0.03	1.00	0.00	S
lit-as-1-cont-act	1.77	37.00	0.00	0.19	11.00	0.00	A
lit-as-1-mV	0.69	51.00	0.00	0.15	150.00	0.00	A

	solving as MISDP			solving with approximation			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
lit-as-2	7203.22	1205.60	–	7200.00	11694.60	–	–
lit-as-2-act	7246.01	46.20	–	7200.01	2010.00	–	–
lit-as-2-act-mV	7238.32	95.60	–	7200.03	586099.00	–	–
lit-as-2-cont	5.83	1.00	0.00	7200.04	1.00	–	S
lit-as-2-cont-act	7220.84	104.80	0.04	7200.02	28.00	–	S
lit-as-2-mV	7202.52	1821.40	2285.68	7200.07	1297978.60	10832.01	S
lit-as-2-big	7385.51	12.60	–	7200.01	14778.00	–	–
lit-as-2-big-act	8169.05	5.00	–	7200.01	3490.00	–	–
lit-as-2-big-act-int	7226.34	82.60	–	7200.01	7619.20	–	–
lit-as-2-big-int	7203.79	1085.40	–	7200.00	30317.20	–	–
lit-as-2-big-mV-int	7202.33	1756.60	2.04	7200.07	1246502.80	70431.57	S
lit-as-2-big-act-mV-int	7249.71	93.80	–	7200.01	178442.00	–	–
lit-as-2-big-cont	6.18	1.00	0.00	7200.06	1.00	–	S
lit-as-2-big-mV	7329.79	27.20	–	7200.04	657085.40	–	–
lit-as-5	7200.69	7242.00	–	7200.08	1267787.00	–	–
lit-as-5-act	7207.23	498.20	–	7200.02	302661.80	–	–
lit-as-5-act-mV	7085.71	1456.40	0.09	7200.02	355201.00	219.92	S
lit-as-5-cont	1.01	1.00	0.00	29.94	1.00	0.00	S
lit-as-5-cont-act	10.60	1.00	0.00	7200.00	740.80	99.11	S
lit-as-5-mV	1931.00	3636.00	0.00	7200.33	5205523.00	420.59	S
lit-as-5-big	7240.45	121.60	–	7200.02	500411.00	–	–
lit-as-5-big-act	7224.05	109.20	–	7200.02	333144.20	–	–
lit-as-5-big-act-int	4323.24	4463.00	0.00	7200.02	269348.60	–	S
lit-as-5-big-int	4450.68	4463.00	0.00	7200.02	385181.40	–	S
lit-as-5-big-mV-int	883.51	1762.00	0.00	7200.30	4641160.20	419.93	S
lit-as-5-big-act-mV-int	898.65	1762.00	0.00	6000.24	2575696.33	12861.75	S
lit-as-5-big-act-mV	7215.69	248.60	–	7200.12	2079297.80	4590.25	A
lit-as-5-big-cont	1.00	1.00	0.00	33.31	1.00	0.00	S
lit-as-5-big-cont-act	1.12	1.00	0.00	2903.74	158.00	1.26	S
lit-as-5-big-mV	7216.83	242.40	–	7200.13	2418290.80	1769.70	A
lit-as-6	898.80	1983.00	0.00	7200.52	6942063.40	–	S
lit-as-6-act	898.11	1983.00	0.00	7200.48	5604146.60	–	S
lit-as-6-act-mV	1711.35	5558.00	0.00	7200.54	7277743.60	2900.60	S
lit-as-6-cont	0.50	1.00	0.00	8.82	1.00	0.00	S
lit-as-6-cont-act	0.52	1.00	0.00	9.19	1.00	0.00	S
lit-as-6-mV	1695.02	5558.00	0.00	7200.61	7382207.00	2894.46	S
lit-as-6-big	7202.64	1171.00	–	7200.18	2513962.00	–	–
lit-as-6-big-act	7202.62	1189.60	–	7200.17	2598494.00	–	–
lit-as-6-big-act-int	7200.28	15270.40	0.09	7200.07	1197412.60	1050.40	S
lit-as-6-big-int	7200.21	15042.40	0.10	7200.08	1384280.80	1049.77	S
lit-as-6-big-mV-int	571.37	1710.00	0.00	7200.37	4999979.20	4494.28	S
lit-as-6-big-act-mV-int	578.36	1710.00	0.00	7200.39	5017352.20	4494.11	S
lit-as-6-big-act-mV	7202.36	1936.40	–	7200.23	3776372.40	–	–
lit-as-6-big-cont	0.50	1.00	0.00	12.78	1.00	0.00	S
lit-as-6-big-cont-act	0.51	1.00	0.00	12.59	1.00	0.00	S
lit-as-6-big-mV	7201.96	1967.20	–	7200.24	3791624.20	–	–
lit-s-1	3272.54	21610.00	0.00	7200.32	8186697.60	5.25	S
lit-s-1-act	7200.27	11110.00	2063.86	7200.06	2991362.60	6.71	A
lit-s-1-act-mV	46.05	106.00	0.00	7200.05	2805057.00	7.71	S

	solving as MISDP			solving with approximation			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
lit-s-1-cont	0.16	1.00	0.00	1.03	1.00	0.00	S
lit-s-1-cont-act	7200.13	31.00	–	3818.55	12455.00	0.00	A
lit-s-1-mV	6.55	83.00	0.00	13.37	23857.00	0.00	S

Table A.1. – Comparing the approximation scheme with pure MISDP branch-and-bound.

Parameter tests

Additionally, we present the detailed results for the parameter tests in Table A.2. We compared four different settings, each is presented in three columns. The first setting was the pure MISDP branch-and-bound algorithm, where we use SCIP with its standard settings and solve an SDP relaxation in every branching node. The second setting is to solve additionally one approximation-LP in the root node. This approximation is build using the eigenvalue cuts presented in Section 3.4. For the third setting we choose to solve the approximation at each node with a depth of four and multiples of four, we call this a frequency. Finally, we take a setting where the approximation frequency is five.

Again the last column shows which of the four settings solved the instance in the shortest time or has the smallest gap if none of the parameter settings was able to solve. We compared 528 instances.

Note that we had to shorten the names for this table. So the bridge-instances are now only called '*b*' and the cantilever-instances are called '*c*'. Moreover, if there are two or three scenarios of loads in the instance we now write '*2s*' or '*3s*'.

instance	only MISDP time	nodes	gap	MISDP + approx. freq. 0 time	nodes	gap	MISDP + approx. freq. 4 time	nodes	gap	MISDP + approx. freq. 5 time	nodes	gap	best
b-1	7.00	132.00	0.00	8.44	119.00	0.00	6.09	68.00	0.00	6.09	68.00	0.00	3
b-1-act	20.33	130.00	0.00	26.26	110.00	0.00	24.12	96.00	0.00	25.70	106.00	0.00	1
b-1-3s	388.78	2781.00	0.00	433.72	2638.00	0.00	375.62	2374.00	0.00	493.41	3111.00	0.00	3
b-1-2s	265.19	2786.00	0.00	260.07	2544.00	0.00	264.11	2395.00	0.00	263.82	2386.00	0.00	2
b-1-act-mV	2061.81	16147.00	0.00	2146.00	14563.00	0.00	9.00	48.00	0.00	8.11	43.00	0.00	4
b-1-cont	0.08	1.00	0.00	0.07	1.00	0.00	0.09	1.00	0.00	0.09	1.00	0.00	2
b-1-cont-act	0.82	3.00	0.00	1.18	3.00	0.00	1.18	3.00	0.00	1.22	3.00	0.00	1
b-1-mV	38.84	932.00	0.00	39.95	868.00	0.00	32.98	792.00	0.00	31.71	784.00	0.00	4
b-1-act-2s	867.47	2469.00	0.00	1066.16	2615.00	0.00	1014.88	2561.00	0.00	998.40	2488.00	0.00	1
b-1-act-2s-mV	3971.74	15466.00	0.00	4640.39	14949.00	0.00	896.55	2893.00	0.00	866.86	2662.00	0.00	4
b-1-2s-mV	296.97	3939.00	0.00	296.41	3945.00	0.00	237.32	3535.00	0.00	238.77	3576.00	0.00	3
b-1-cont-2s	0.18	1.00	0.00	0.17	1.00	0.00	0.18	1.00	0.00	0.17	1.00	0.00	2
b-1-cont-act-2s	10.20	26.00	0.00	14.78	24.00	0.00	34.92	25.00	0.00	19.11	28.00	0.00	1
b-2	134.94	1158.00	0.00	238.69	1990.00	0.00	62.64	487.00	0.00	79.42	581.00	0.00	3
b-2-act	288.50	1177.00	0.00	627.31	2209.00	0.00	323.96	817.00	0.00	318.48	842.00	0.00	1
b-2-act-int	7200.03	82811.60	–	7200.05	59037.80	–	7200.12	46933.80	20.54	3794.51	25077.00	0.00	4
b-2-3s	7200.15	26590.80	929.69	7200.20	22224.40	–	7200.08	23605.40	12.11	7200.22	23684.60	42.69	3
b-2-2s	7200.10	39747.60	907.06	7200.14	32017.80	934.06	7200.13	28564.80	–	7200.10	31499.20	13.44	4
b-2-act-3s-mV-int	7200.42	12353.40	–	7200.22	9639.00	–	7200.30	10731.80	11.38	7200.29	11026.40	3.52	4
b-2-int	40.45	765.00	0.00	54.68	856.00	0.00	28.17	452.00	0.00	33.25	524.00	0.00	3
b-2-mV-int	18.35	327.00	0.00	26.30	310.00	0.00	20.55	327.00	0.00	18.75	332.00	0.00	1
b-2-act-mV-int	7200.08	43084.00	–	7200.10	29170.40	–	2624.31	11666.00	0.00	933.31	4101.00	0.00	4
b-2-act-mV	1902.13	13018.00	0.00	2320.98	12749.00	0.00	12.27	33.00	0.00	12.62	32.00	0.00	3
b-2-cont	0.08	1.00	0.00	0.08	1.00	0.00	0.08	1.00	0.00	0.08	1.00	0.00	2
b-2-cont-act	0.81	3.00	0.00	1.16	3.00	0.00	1.19	3.00	0.00	1.21	3.00	0.00	1
b-2-mV	76.09	1096.00	0.00	84.27	1032.00	0.00	45.15	692.00	0.00	48.24	708.00	0.00	3
b-2-act-2s	7200.28	14171.40	–	7200.37	10036.00	–	7200.54	10448.20	–	7200.30	10396.80	–	–
b-2-act-2s-int	7200.12	34991.40	–	7200.11	30064.40	–	7200.15	20430.40	–	7200.15	22879.40	–	–
b-2-act-2s-mV	6311.59	24501.00	0.00	7200.34	16639.00	326.79	1033.42	2757.00	0.00	1831.76	4329.00	0.00	3
b-2-2s-mV	1338.64	13544.00	0.00	1682.69	13857.00	0.00	1125.80	10896.00	0.00	1192.49	11019.00	0.00	3
b-2-2s-int	433.71	4031.00	0.00	437.20	3647.00	0.00	651.16	4921.00	0.00	551.08	4321.00	0.00	1
b-2-2s-mV-int	258.56	2551.00	0.00	302.36	2383.00	0.00	205.19	2118.00	0.00	199.69	1953.00	0.00	4
b-2-cont-2s	0.17	1.00	0.00	0.17	1.00	0.00	0.18	1.00	0.00	0.17	1.00	0.00	2
b-2-cont-act-2s	10.34	26.00	0.00	14.86	24.00	0.00	34.95	25.00	0.00	18.99	28.00	0.00	1
b-2-act-2s-mV-int	7200.20	20041.00	–	7200.30	14837.40	–	7200.19	16019.60	5.45	7200.25	16010.60	3.84	4
b-3	44.21	415.00	0.00	74.69	477.00	0.00	64.12	408.00	0.00	88.17	512.00	0.00	1
b-3-act	184.18	724.00	0.00	336.42	731.00	0.00	1141.77	3090.00	0.00	494.54	1065.00	0.00	1
b-3-act-int	7200.06	71202.00	–	7200.05	51060.20	–	7200.06	42846.80	46.37	7200.07	50720.00	26.00	4
b-3-3s	3686.99	14378.00	0.00	4771.75	15431.00	0.00	6267.05	20398.00	0.00	5610.05	18408.00	0.00	1
b-3-2s	2526.70	14388.00	0.00	2591.84	11437.00	0.00	3341.04	14600.00	0.00	4076.25	17262.00	0.00	1
b-3-act-3s-mV-int	7200.42	12467.80	–	7200.38	9640.00	18.04	7200.42	11367.80	2.50	7200.31	10779.80	5.39	3
b-3-int	71.79	1220.00	0.00	91.54	1215.00	0.00	89.17	1169.00	0.00	80.69	1144.00	0.00	1
b-3-mV-int	6.38	123.00	0.00	0.30	1.00	0.00	0.31	1.00	0.00	0.30	1.00	0.00	2
b-3-act-mV-int	7200.09	43322.20	–	7200.09	30785.40	–	5335.76	25199.00	0.00	429.19	1977.00	0.00	4
b-3-act-mV	7200.07	40837.00	13.74	7200.08	29509.60	255.09	162.13	701.00	0.00	106.89	356.00	0.00	4

	only MISDP			MISDP + approx. freq. 0			MISDP + approx. freq. 4			MISDP + approx. freq. 5			
instance	time	nodes	gap	time	nodes	gap	time	nodes	gap	time	nodes	gap	best
b-3-cont	0.10	1.00	0.00	0.10	1.00	0.00	0.11	1.00	0.00	0.10	1.00	0.00	2
b-3-cont-act	0.87	3.00	0.00	1.53	4.00	0.00	1.56	4.00	0.00	1.58	4.00	0.00	1
b-3-mV	90.90	1071.00	0.00	147.45	1358.00	0.00	63.38	829.00	0.00	66.03	860.00	0.00	3
b-3-act-2s	6223.10	12837.00	0.00	7200.22	11522.40	58.78	7200.17	11603.80	–	7200.20	11481.80	60.20	1
b-3-act-2s-int	7200.13	33003.20	–	7200.14	21255.40	–	7200.22	18211.80	–	7200.15	18856.20	–	–
b-3-act-2s-mV	7200.17	20622.80	–	7200.25	15254.00	253.85	2099.07	5287.00	0.00	2112.64	5190.00	0.00	3
b-3-2s-mV	2345.37	18040.00	0.00	3576.35	23075.00	0.00	1785.09	15531.00	0.00	1830.41	15684.00	0.00	3
b-3-2s-int	299.80	2634.00	0.00	289.10	2382.00	0.00	338.45	2591.00	0.00	356.36	2851.00	0.00	2
b-3-2s-mV-int	47.57	464.00	0.00	1.69	11.00	0.00	1.97	13.00	0.00	1.73	11.00	0.00	2
b-3-cont-2s	0.17	1.00	0.00	0.17	1.00	0.00	0.17	1.00	0.00	0.17	1.00	0.00	2
b-3-cont-act-2s	5.24	11.00	0.00	15.25	25.00	0.00	44.65	24.00	0.00	27.47	26.00	0.00	1
b-3-act-2s-mV-int	7200.16	20045.80	–	7200.17	15185.40	–	6582.55	16338.00	0.00	7200.15	16615.40	0.89	3
b-4	7.08	132.00	0.00	8.42	119.00	0.00	6.10	68.00	0.00	6.07	68.00	0.00	4
b-4-act	22.73	136.00	0.00	27.69	109.00	0.00	23.55	84.00	0.00	26.98	101.00	0.00	1
b-4-3s	389.21	2781.00	0.00	430.42	2638.00	0.00	372.92	2374.00	0.00	496.39	3111.00	0.00	3
b-4-2s	265.70	2786.00	0.00	258.12	2544.00	0.00	264.48	2395.00	0.00	263.37	2386.00	0.00	2
b-4-act-mV	1994.79	15646.00	0.00	2114.32	14319.00	0.00	15.45	86.00	0.00	15.99	92.00	0.00	3
b-4-cont	0.08	1.00	0.00	0.08	1.00	0.00	0.08	1.00	0.00	0.08	1.00	0.00	2
b-4-cont-act	0.75	3.00	0.00	1.45	5.00	0.00	1.49	5.00	0.00	1.53	5.00	0.00	1
b-4-mV	26.51	618.00	0.00	1.42	27.00	0.00	1.14	19.00	0.00	1.02	17.00	0.00	4
b-4-act-2s	852.55	2563.00	0.00	941.93	2417.00	0.00	1048.84	2704.00	0.00	1154.67	2917.00	0.00	1
b-4-act-2s-mV	3813.22	14780.00	0.00	4437.83	14343.00	0.00	1124.21	3555.00	0.00	1150.25	3657.00	0.00	3
b-4-2s-mV	250.93	3289.00	0.00	207.24	2640.00	0.00	165.78	2451.00	0.00	164.93	2492.00	0.00	4
b-4-cont-2s	0.17	1.00	0.00	0.17	1.00	0.00	0.17	1.00	0.00	0.17	1.00	0.00	2
b-4-cont-act-2s	9.84	21.00	0.00	22.51	48.00	0.00	33.85	45.00	0.00	21.90	39.00	0.00	1
b-5	133.85	1158.00	0.00	238.06	1990.00	0.00	62.57	487.00	0.00	79.41	581.00	0.00	3
b-5-act	292.69	1179.00	0.00	612.95	2045.00	0.00	390.73	1000.00	0.00	378.90	908.00	0.00	1
b-5-act-int	7200.06	80355.20	–	7200.04	55316.00	–	7200.09	43960.60	8.71	7200.05	51513.20	37.75	4
b-5-3s	7200.14	26584.40	929.74	7200.17	22502.20	–	7200.21	23273.60	12.23	7200.10	23708.20	42.68	3
b-5-2s	7200.13	39869.00	906.85	7200.14	32279.60	933.77	7200.17	28509.40	–	7200.09	31274.20	13.51	4
b-5-act-3s-mV-int	7200.35	12337.20	–	7200.28	9558.80	–	7200.20	11156.60	19.27	7200.26	11007.00	6.34	4
b-5-int	40.83	765.00	0.00	54.26	856.00	0.00	28.06	452.00	0.00	33.13	524.00	0.00	3
b-5-mV-int	18.37	327.00	0.00	26.18	310.00	0.00	20.57	327.00	0.00	18.83	332.00	0.00	1
b-5-act-mV-int	7200.18	43057.80	–	7200.11	28237.80	–	1920.55	8132.00	0.00	1357.20	5644.00	0.00	4
b-5-act-mV	2309.51	14623.00	0.00	2912.66	14611.00	0.00	14.32	37.00	0.00	13.11	33.00	0.00	4
b-5-cont	0.08	1.00	0.00	0.08	1.00	0.00	0.08	1.00	0.00	0.08	1.00	0.00	2
b-5-cont-act	0.75	3.00	0.00	1.45	5.00	0.00	1.48	5.00	0.00	1.52	5.00	0.00	1
b-5-mV	75.54	1096.00	0.00	84.91	1032.00	0.00	45.19	692.00	0.00	48.63	708.00	0.00	3
b-5-act-2s	7200.23	14070.80	–	7200.22	9909.80	–	7200.31	10258.40	–	7200.30	9815.20	19.09	4
b-5-act-2s-int	7200.08	31867.00	–	7200.13	31945.20	–	7200.14	21305.60	> 5000	7200.24	22513.60	> 5000	4
b-5-act-2s-mV	7200.18	25459.00	1.04	7200.26	16138.00	330.24	1201.29	2967.00	0.00	1420.37	3822.00	0.00	3
b-5-2s-mV	1339.94	13544.00	0.00	1690.68	13857.00	0.00	1130.24	10896.00	0.00	1184.47	11019.00	0.00	3
b-5-2s-int	432.99	4031.00	0.00	433.90	3647.00	0.00	652.48	4921.00	0.00	549.02	4321.00	0.00	1
b-5-2s-mV-int	259.59	2551.00	0.00	303.41	2383.00	0.00	204.20	2118.00	0.00	199.24	1953.00	0.00	4
b-5-cont-2s	0.18	1.00	0.00	0.18	1.00	0.00	0.17	1.00	0.00	0.17	1.00	0.00	3

	only MISDP			MISDP + approx. freq. 0			MISDP + approx. freq. 4			MISDP + approx. freq. 5			
instance	time	nodes	gap	time	nodes	gap	time	nodes	gap	time	nodes	gap	best
b-5-cont-act-2s	9.72	21.00	0.00	22.51	48.00	0.00	34.05	45.00	0.00	21.83	39.00	0.00	1
b-5-act-2s-mV-int	7200.21	19246.60	–	7200.14	15099.40	–	7200.12	16999.40	4.28	7200.33	16206.40	3.29	4
b-6	44.44	415.00	0.00	74.65	477.00	0.00	64.22	408.00	0.00	87.90	512.00	0.00	1
b-6-act	212.02	800.00	0.00	443.20	1119.00	0.00				544.41	1391.00	0.00	1
b-6-act-int	7200.06	66028.80	–	7200.09	51188.80	–	7200.13	40204.40	123.38	7200.12	41720.20	57.54	4
b-6-3s	3697.17	14378.00	0.00	4798.04	15431.00	0.00	6283.19	20398.00	0.00	5629.80	18408.00	0.00	1
b-6-2s	2523.85	14388.00	0.00	2569.38	11437.00	0.00	3304.52	14600.00	0.00	4096.55	17262.00	0.00	1
b-6-act-3s-mV-int	7200.22	12416.00	–	7200.34	9149.80	–	7200.32	10388.60	18.00	7200.38	10039.20	35.08	4
b-6-int	72.09	1220.00	0.00	91.54	1215.00	0.00	89.21	1169.00	0.00	80.66	1144.00	0.00	1
b-6-mV-int	19.55	400.00	0.00	21.35	339.00	0.00	14.18	330.00	0.00	13.07	290.00	0.00	4
b-6-act-mV-int	7200.12	43208.80	–	7200.12	29698.60	–	2032.06	9006.00	0.00	3221.17	14377.00	0.00	3
b-6-act-mV	7200.06	40219.40	–	7200.10	29244.80	245.67	114.74	491.00	0.00	174.78	628.00	0.00	3
b-6-cont	0.10	1.00	0.00	0.10	1.00	0.00	0.10	1.00	0.00	0.10	1.00	0.00	2
b-6-cont-act	0.64	2.00	0.00	0.89	2.00	0.00	0.92	2.00	0.00	0.91	2.00	0.00	1
b-6-mV	47.47	548.00	0.00	59.69	511.00	0.00	35.75	460.00	0.00	31.05	418.00	0.00	4
b-6-act-2s	6727.72	14479.00	0.00	7192.90	11219.20	0.12	7200.34	10846.80	130.60	7200.24	10261.60	–	1
b-6-act-2s-int	7200.04	30007.60	–	7200.13	24485.00	–	7200.20	19172.20	–	7200.17	19664.20	–	–
b-6-act-2s-mV	7200.16	20231.00	–	7200.30	14261.60	270.58	1996.56	4143.00	0.00	3701.52	8641.00	0.00	3
b-6-2s-mV	1449.97	10890.00	0.00	2120.05	13171.00	0.00	1182.64	9689.00	0.00	1070.91	9017.00	0.00	4
b-6-2s-int	299.47	2634.00	0.00	289.75	2382.00	0.00	336.30	2591.00	0.00	353.35	2851.00	0.00	2
b-6-2s-mV-int	106.47	1092.00	0.00	124.65	1027.00	0.00	78.30	933.00	0.00	94.05	1094.00	0.00	3
b-6-cont-2s	0.17	1.00	0.00	0.16	1.00	0.00	0.17	1.00	0.00	0.17	1.00	0.00	2
b-6-cont-act-2s	3.51	7.00	0.00	6.41	9.00	0.00	6.45	9.00	0.00	6.36	9.00	0.00	1
b-6-act-2s-mV-int	7200.14	19716.80	–	7200.18	15738.20	–	7200.16	16699.00	10.56	7200.21	15873.40	35.51	4
b-7	1683.93	3520.00	0.00	2647.72	3522.00	0.00	2299.26	2587.00	0.00	2774.76	3233.00	0.00	1
b-7-act	5841.99	4455.00	0.00	7200.67	2777.40	–	7201.34	2729.00	–	7200.88	2677.20	–	1
b-7-act-int	7200.24	30518.00	–	7200.25	18411.00	–	7200.18	22723.20	32.26	7200.19	25456.80	28.25	4
b-7-3s	7200.38	6760.40	152.98	7200.86	4199.20	–	7200.55	4410.00	–	7200.86	4502.60	–	1
b-7-2s	7200.22	8013.20	–	7200.78	5682.00	–	7200.33	5823.80	–	7200.52	5729.00	–	–
b-7-act-3s-mV-int	7200.89	4625.40	–	7201.20	3040.00	–	7201.16	3432.60	195.21	7201.41	3317.20	–	3
b-7-int	12.82	135.00	0.00	14.39	116.00	0.00	14.23	102.00	0.00	13.49	86.00	0.00	1
b-7-mV-int	1.41	14.00	0.00	1.40	9.00	0.00	1.39	9.00	0.00	1.40	9.00	0.00	3
b-7-act-mV-int	843.02	4210.00	0.00	781.09	1565.00	0.00	3964.90	6778.00	0.00	2797.65	4609.00	0.00	2
b-7-act-mV	312.72	267.00	0.00	495.79	307.00	0.00	739.00	536.00	0.00	497.28	299.00	0.00	1
b-7-cont	0.17	1.00	0.00	0.16	1.00	0.00	0.16	1.00	0.00	0.16	1.00	0.00	2
b-7-cont-act	8.41	15.00	0.00	7200.15	57.00	–	268.46	66.00	0.00	7200.13	307.00	7.67	1
b-7-mV	20.86	49.00	0.00	29.88	52.00	0.00	15.58	28.00	0.00	17.02	33.00	0.00	3
b-7-act-2s	7201.14	2335.00	–	7202.90	1647.00	–	7202.18	1643.60	–	7201.82	1713.60	–	–
b-7-act-2s-int	7200.22	16729.80	–	7200.25	10704.00	–	7200.47	6705.00	131.83	7200.43	6552.60	4870.70	4
b-7-act-2s-mV	7200.50	3734.00	–	7201.50	2879.00	1235.34	7200.77	3680.80	5.31	7200.81	3644.00	5.30	4
b-7-2s-mV	642.07	1484.00	0.00	729.62	1439.00	0.00	541.03	1229.00	0.00	687.56	1454.00	0.00	3
b-7-2s-int	328.49	1228.00	0.00	597.94	1555.00	0.00	725.57	1790.00	0.00	855.69	2003.00	0.00	1
b-7-2s-mV-int	86.54	486.00	0.00	114.66	527.00	0.00	89.21	452.00	0.00	106.83	469.00	0.00	1
b-7-cont-2s	0.33	1.00	0.00	0.32	1.00	0.00	0.33	1.00	0.00	0.33	1.00	0.00	2
b-7-cont-act-2s	30.27	25.00	0.00	27.55	15.00	0.00	30.26	17.00	0.00	30.90	17.00	0.00	2

	only MISDP			MISDP + approx. freq. 0			MISDP + approx. freq. 4			MISDP + approx. freq. 5			
instance	time	nodes	gap	time	nodes	gap	time	nodes	gap	time	nodes	gap	best
b-7-act-2s-mV-int	7200.14	17797.80	–	7200.27	6927.80	–	7200.65	5819.80	–	7200.46	5782.00	2493.49	4
b-8	7200.47	4664.60	–	7201.86	2652.60	–	7201.23	2693.80	–	7201.12	2710.00	–	–
b-8-act	7202.03	1411.20	–	7204.83	865.00	–	7204.49	845.40	–	7203.49	812.40	–	–
b-8-act-int	7200.54	8143.80	–	7200.72	4653.40	> 5000	7200.94	4669.80	> 5000	7200.78	4764.80	> 5000	3
b-8-3s	7201.08	2578.40	–	7201.48	1679.20	–	7202.12	1817.40	–	7200.81	1783.40	–	–
b-8-2s	7200.94	3061.40	–	7201.64	2237.80	–	7200.90	2214.60	–	7201.84	2253.00	–	–
b-8-act-3s-mV-int	7202.41	1564.80	–	7201.31	1100.00	–	7201.90	1232.00	–	7202.11	1240.80	–	–
b-8-int	1636.89	5354.00	0.00	2462.30	5609.00	0.00	2580.47	5129.00	0.00	2527.04	5041.00	0.00	1
b-8-mV-int	70.72	365.00	0.00	117.05	412.00	0.00	92.06	379.00	0.00	76.56	344.00	0.00	1
b-8-act-mV-int	7200.15	8002.60	–	7200.24	5491.80	–	7200.42	4604.40	–	7200.94	4535.40	–	–
b-8-act-mV	7200.96	2263.60	–	7202.44	2045.80	1489.58	7202.53	1577.00	1562.95	7201.18	1597.60	1544.31	2
b-8-cont	0.29	1.00	0.00	0.30	1.00	0.00	0.31	1.00	0.00	0.31	1.00	0.00	1
b-8-cont-act	14.39	11.00	0.00	7200.23	21.00	–	7201.21	14.00	–	7200.21	18.00	–	1
b-8-mV	2217.31	3085.00	0.00	2149.64	2698.00	0.00	2290.53	2875.00	0.00	2240.03	2840.00	0.00	2
b-8-act-2s	7208.01	691.60	–	7206.03	493.00	–	7207.92	443.20	–	7206.55	465.80	–	–
b-8-act-2s-int	7201.21	4006.40	–	7201.20	2990.40	–	7201.82	2452.60	–	7201.93	2610.40	–	–
b-8-act-2s-mV	7202.75	1155.40	–	7202.60	904.00	1772.39	7203.69	862.40	1609.22	7205.51	794.00	1550.56	4
b-8-2s-mV	7200.60	6018.20	–	7200.66	4294.40	–	7200.69	6559.00	9.93	7200.58	5994.60	15.36	4
b-8-2s-int	3085.78	5509.00	0.00	3678.87	5494.00	0.00	3606.70	4965.00	0.00	3625.58	5075.00	0.00	1
b-8-2s-mV-int	128.84	370.00	0.00	184.23	416.00	0.00	166.54	422.00	0.00	152.51	369.00	0.00	1
b-8-cont-2s	0.52	1.00	0.00	0.51	1.00	0.00	0.53	1.00	0.00	0.51	1.00	0.00	2
b-8-cont-act-2s	36.33	11.00	0.00	7203.59	1.00	–	7203.24	1.00	–	7202.47	1.00	–	1
b-8-act-2s-mV-int	7201.41	3053.40	–	7202.25	2530.40	–	7202.20	2237.00	–	7201.95	2186.00	–	–
b-9	76.64	435.00	0.00	96.54	449.00	0.00	87.77	366.00	0.00	93.36	390.00	0.00	1
b-9-act	363.05	1001.00	0.00	408.33	1017.00	0.00	606.82	1076.00	0.00	610.86	1130.00	0.00	1
b-9-act-int	7200.05	65900.20	–	7200.11	50491.00	–	7200.04	44835.40	115.83	7200.04	56861.40	110.84	4
b-9-3s	3963.39	9879.00	0.00	6163.10	12080.00	0.00	7200.27	14948.20	0.17	7200.18	14476.80	16.22	1
b-9-2s	378.53	1249.00	0.00	498.56	1261.00	0.00	375.66	732.00	0.00	379.93	752.00	0.00	3
b-9-act-3s-mV-int	1534.41	2680.00	0.00	2141.09	2742.00	0.00	6832.43	10641.00	0.00	5561.60	8489.00	0.00	1
b-9-int	69.09	1206.00	0.00	88.16	1130.00	0.00	93.25	1336.00	0.00	69.36	1034.00	0.00	1
b-9-mV-int	16.66	545.00	0.00	28.71	687.00	0.00	40.51	795.00	0.00	27.11	640.00	0.00	1
b-9-act-mV-int	404.55	2907.00	0.00	734.09	3129.00	0.00	7200.07	37679.00	261.97	3222.89	14790.00	0.00	1
b-9-act-mV	284.25	915.00	0.00	338.59	888.00	0.00	152.39	520.00	0.00	98.29	352.00	0.00	4
b-9-cont	0.10	1.00	0.00	0.11	1.00	0.00	0.10	1.00	0.00	0.10	1.00	0.00	1
b-9-cont-act	13.02	82.00	0.00	15.85	78.00	0.00	63.92	89.00	0.00	42.79	81.00	0.00	1
b-9-mV	45.36	498.00	0.00	22.46	243.00	0.00	34.05	422.00	0.00	21.05	209.00	0.00	4
b-9-act-2s	880.19	1180.00	0.00	1113.30	1336.00	0.00	2007.16	2047.00	0.00	1022.52	970.00	0.00	1
b-9-act-2s-int	7200.06	58181.60	–	7200.06	27307.80	–	7200.18	22065.60	117.59	7200.13	22116.00	123.54	4
b-9-act-2s-mV	249.35	425.00	0.00	262.55	415.00	0.00	72.46	185.00	0.00	176.40	302.00	0.00	3
b-9-2s-mV	36.87	255.00	0.00	29.43	172.00	0.00	39.51	191.00	0.00	93.56	485.00	0.00	2
b-9-2s-int	465.21	5745.00	0.00	733.93	10311.00	0.00	255.57	1887.00	0.00	223.40	1844.00	0.00	4
b-9-2s-mV-int	18.25	319.00	0.00	46.72	516.00	0.00	63.03	642.00	0.00	59.46	618.00	0.00	1
b-9-cont-2s	0.17	1.00	0.00	0.17	1.00	0.00	0.17	1.00	0.00	0.16	1.00	0.00	4
b-9-cont-act-2s	22.73	79.00	0.00	55.73	39.00	0.00	69.26	23.00	0.00	55.75	25.00	0.00	1
b-9-act-2s-mV-int	508.01	1511.00	0.00	7200.15	46694.60	121.43	1410.71	4390.00	0.00	7176.31	20236.80	71.60	1

	only MISDP			MISDP + approx. freq. 0			MISDP + approx. freq. 4			MISDP + approx. freq. 5			
instance	time	nodes	gap	time	nodes	gap	time	nodes	gap	time	nodes	gap	best
b-10	1081.06	6742.00	0.00	1267.94	6674.00	0.00	758.59	1790.00	0.00	790.70	2208.00	0.00	3
b-10-act	968.78	1791.00	0.00	1131.14	1787.00	0.00	1153.05	1068.00	0.00	1012.81	844.00	0.00	1
b-10-act-int	7200.19	22660.60	–	7200.29	16483.00	–	7200.25	15174.00	> 5000	7200.27	15370.20	174.64	4
b-10-3s	7200.29	11306.60	4666.71	7200.33	7488.60	4509.00	7200.66	7441.80	3223.54	7200.40	7819.80	–	3
b-10-2s	7200.18	18738.00	0.18	7200.19	12389.60	76.31	7200.20	11515.80	56.25	7200.23	13112.80	1.75	1
b-10-act-3s-mV-int	7200.88	6570.20	–	7201.13	4708.00	–	7200.76	5711.80	–	7200.45	5691.60	–	–
b-10-int	48.95	455.00	0.00	76.22	508.00	0.00	67.97	453.00	0.00	55.72	430.00	0.00	1
b-10-mV-int	131.47	2958.00	0.00	142.51	2520.00	0.00	23.91	370.00	0.00	23.98	396.00	0.00	3
b-10-act-mV-int	1169.86	5204.00	0.00	1818.00	4828.00	0.00	7200.10	24281.80	96.39	7200.17	22269.20	99.83	1
b-10-act-mV	7200.07	45856.40	–	7200.07	43383.40	–	572.01	864.00	0.00	360.05	950.00	0.00	4
b-10-cont	0.13	1.00	0.00	0.13	1.00	0.00	0.14	1.00	0.00	0.13	1.00	0.00	2
b-10-cont-act	0.44	1.00	0.00	0.42	1.00	0.00	0.43	1.00	0.00	0.45	1.00	0.00	2
b-10-mV	7200.06	61355.40	214.52	7200.05	57556.60	157.44	312.56	1097.00	0.00	170.71	676.00	0.00	4
b-10-act-2s	7201.06	4227.00	–	7201.31	2694.20	–	7201.04	3018.20	–	7201.12	2877.80	–	–
b-10-act-2s-int	7200.32	12062.80	–	7200.46	8633.20	–	7200.54	8616.00	–	7200.58	8252.40	–	–
b-10-act-2s-mV	7200.14	26314.00	–	7200.05	22916.00	2293.63	7200.42	5800.80	1896.01	7200.54	5304.40	1775.79	4
b-10-2s-mV	7200.11	34298.40	–	7200.10	32286.80	–	7200.20	16435.20	152.98	7200.16	17705.60	62.18	4
b-10-2s-int	7200.06	46746.00	4017.17	7200.03	41892.00	186.70	7200.14	38628.80	15.68	7200.09	35918.60	19.08	3
b-10-2s-mV-int	1350.61	10334.00	0.00	1590.19	10069.00	0.00	1358.47	9478.00	0.00	1320.29	9281.00	0.00	4
b-10-cont-2s	0.20	1.00	0.00	0.19	1.00	0.00	0.19	1.00	0.00	0.19	1.00	0.00	2
b-10-cont-act-2s	6.04	6.00	0.00	8.79	7.00	0.00	8.61	7.00	0.00	8.98	7.00	0.00	1
b-10-act-2s-mV-int	7200.17	14274.40	–	7200.56	8503.00	–	7200.46	9477.20	–	7200.42	9195.20	–	–
b-big	7200.83	4854.20	–	7201.49	2816.00	–	3777.65	1364.00	0.00	4407.03	1460.00	0.00	3
b-big-act	7203.62	875.20	–	7204.82	556.20	–	7204.33	531.60	–	7207.37	508.00	–	–
b-big-act-int	7201.35	3387.20	–	7202.04	2130.80	–	7201.76	1432.60	–	7202.75	1674.00	–	–
b-big-3s	7201.17	2497.40	–	7201.86	1742.20	–	7202.69	1609.60	–	7201.09	1714.20	–	–
b-big-2s	7201.16	3221.20	–	7202.08	2078.40	–	7201.28	2065.00	–	7202.27	2108.80	–	–
b-big-act-3s-mV-int	7204.51	1341.20	–	11.80	1.00	–	9.12	1.00	–	10.84	1.00	–	3
b-big-int	7200.29	14332.60	–	7200.32	9123.60	–	7200.49	8035.60	–	7200.44	8335.20	–	–
b-big-cont	0.51	1.00	0.00	0.53	1.00	0.00	0.54	1.00	0.00	0.53	1.00	0.00	1
b-big-cont-act	36.45	12.00	0.00	3185.92	441.00	0.00	7203.23	11.00	–	7201.30	12.00	–	1
b-big-act-2s	7208.83	445.80	–	7216.68	278.80	–	7209.60	316.40	–	7207.41	249.20	–	–
b-big-act-2s-int	7202.74	1330.20	–	7202.00	1429.20	–	7206.58	1049.40	–	7206.21	813.80	–	–
b-big-act-2s-mV	7202.71	1026.00	–	11.77	1.00	–	9.50	1.00	–	12.88	1.00	–	3
b-big-2s-mV	1.69	1.00	–	1.76	1.00	–	1.53	1.00	–	1.56	1.00	–	3
b-big-2s-int	7200.36	8132.20	–	7200.77	5807.80	–	7200.82	5952.80	–	7200.24	5928.80	–	–
b-big-2s-mV-int	0.57	1.00	–	0.57	1.00	–	0.52	1.00	–	0.54	1.00	–	3
b-big-cont-2s	0.89	1.00	0.00	0.91	1.00	0.00	0.95	1.00	0.00	0.95	1.00	0.00	1
b-big-cont-act-2s	7199.61	2817.20	0.00	502.98	42.00	0.00	1674.03	32.00	0.00	5131.33	70.00	0.00	2
b-big-act-2s-mV-int	7201.87	2542.60	–	5.91	1.00	–	4.81	1.00	–	6.11	1.00	–	3
c-1	29.96	892.00	0.00	49.59	807.00	0.00	44.60	654.00	0.00	41.73	654.00	0.00	1
c-1-act	174.47	1153.00	0.00	200.34	927.00	0.00	118.45	580.00	0.00	144.67	640.00	0.00	3
c-1-3s	565.24	4984.00	0.00	585.55	4589.00	0.00	528.11	4043.00	0.00	506.96	3965.00	0.00	4
c-1-2s	50.89	606.00	0.00	55.55	494.00	0.00	59.98	524.00	0.00	61.38	513.00	0.00	1
c-1-act-mV	7200.06	76117.00	6.28	7200.04	64596.60	3.07	3.60	25.00	0.00	3.62	21.00	0.00	3

	only MISDP			MISDP + approx. freq. 0			MISDP + approx. freq. 4			MISDP + approx. freq. 5			
instance	time	nodes	gap	time	nodes	gap	time	nodes	gap	time	nodes	gap	best
c-1-cont	0.08	1.00	0.00	0.08	1.00	0.00	0.08	1.00	0.00	0.07	1.00	0.00	4
c-1-cont-act	0.98	5.00	0.00	2.19	7.00	0.00	2.24	7.00	0.00	2.19	7.00	0.00	1
c-1-mV	3.13	82.00	0.00	0.25	1.00	0.00	0.23	1.00	0.00	0.24	1.00	0.00	3
c-1-act-2s	181.29	655.00	0.00	337.18	958.00	0.00	288.90	756.00	0.00	354.59	960.00	0.00	1
c-1-act-2s-mV	0.30	1.00	–	0.29	1.00	–	0.29	1.00	–	0.28	1.00	–	4
c-1-2s-mV	0.09	1.00	–	0.09	1.00	–	0.09	1.00	–	0.09	1.00	–	2
c-1-cont-2s	0.13	1.00	0.00	0.12	1.00	0.00	0.13	1.00	0.00	0.12	1.00	0.00	2
c-1-cont-act-2s	0.40	1.00	0.00	0.38	1.00	0.00	0.41	1.00	0.00	0.39	1.00	0.00	2
c-1-act-mV	683.28	5774.00	0.00	782.35	5750.00	0.00	8.68	50.00	0.00	8.58	57.00	0.00	4
c-1-m-mV	4.57	123.00	0.00	5.82	114.00	0.00	5.50	129.00	0.00	5.36	127.00	0.00	1
c-1-m-act-2s-mV	0.34	1.00	–	0.33	1.00	–	0.31	1.00	–	0.34	1.00	–	3
c-1-m-2s-mV	0.09	1.00	–	0.08	1.00	–	0.08	1.00	–	0.08	1.00	–	2
c-2	63.44	777.00	0.00	40.18	558.00	0.00	49.23	614.00	0.00	40.06	518.00	0.00	4
c-2-act	188.59	1629.00	0.00	340.86	1950.00	0.00	299.46	1279.00	0.00	163.05	785.00	0.00	4
c-2-act-int	7200.03	85305.20	1720.99	7200.10	62977.00	1785.84	489.00	3383.00	0.00	3443.22	25855.00	0.00	3
c-2-3s	1890.65	11398.00	0.00	2183.94	11394.00	0.00	2246.91	11730.00	0.00	2214.48	11804.00	0.00	1
c-2-2s	690.41	4849.00	0.00	423.57	2654.00	0.00	575.91	3554.00	0.00	456.38	2776.00	0.00	2
c-2-act-3s-mV-int	710.79	1090.00	0.00	7200.28	9151.20	0.35	14.87	17.00	0.00	77.27	99.00	0.00	3
c-2-int	15.12	352.00	0.00	24.25	497.00	0.00	14.67	313.00	0.00	22.51	460.00	0.00	3
c-2-mV-int	3.18	75.00	0.00	3.12	59.00	0.00	2.30	62.00	0.00	2.40	66.00	0.00	3
c-2-act-mV-int	695.43	4529.00	0.00	666.52	3500.00	0.00	37.49	209.00	0.00	26.51	143.00	0.00	4
c-2-act-mV	1044.02	7226.00	0.00	729.56	3121.00	0.00	101.74	385.00	0.00	80.14	340.00	0.00	4
c-2-cont	0.08	1.00	0.00	0.08	1.00	0.00	0.07	1.00	0.00	0.08	1.00	0.00	3
c-2-cont-act	1.00	5.00	0.00	2.22	7.00	0.00	2.23	7.00	0.00	2.17	7.00	0.00	1
c-2-mV	28.54	455.00	0.00	21.44	320.00	0.00	13.35	244.00	0.00	19.11	325.00	0.00	3
c-2-act-2s	2003.68	5111.00	0.00	3698.37	7469.00	0.00	4751.75	9255.00	0.00	4782.55	9559.00	0.00	1
c-2-act-2s-int	7200.11	42722.80	–	7200.23	34150.40	–	7200.08	26470.60	–	7200.13	26168.20	–	–
c-2-act-2s-mV	4082.25	12036.00	0.00	1939.01	4190.00	0.00	230.99	390.00	0.00	173.97	288.00	0.00	4
c-2-2s-mV	0.16	1.00	–	0.16	1.00	–	0.17	1.00	–	0.17	1.00	–	2
c-2-2s-int	886.69	14875.00	0.00	1013.81	12163.00	0.00	701.99	8057.00	0.00	697.69	8289.00	0.00	4
c-2-2s-mV-int	0.09	1.00	–	0.08	1.00	–	0.08	1.00	–	0.09	1.00	–	2
c-2-cont-2s	0.12	1.00	0.00	0.13	1.00	0.00	0.13	1.00	0.00	0.14	1.00	0.00	1
c-2-cont-act-2s	0.38	1.00	0.00	0.38	1.00	0.00	0.41	1.00	0.00	0.39	1.00	0.00	2
c-2-act-2s-mV-int	3363.03	9051.00	0.00	3597.89	8733.00	0.00	27.29	57.00	0.00	21.83	46.00	0.00	4
c-2-act-3s-mV-int	7200.13	13794.20	–	7200.29	11609.00	–	2105.17	3932.00	0.00	2570.05	4835.00	0.00	3
c-2-m-mV-int	3.23	74.00	0.00	0.13	1.00	0.00	0.13	1.00	0.00	0.14	1.00	0.00	2
c-2-m-act-mV-int	7200.10	59788.60	3.67	2200.06	14120.00	0.00	30.68	184.00	0.00	46.64	249.00	0.00	3
c-2-m-act-mV	612.03	3727.00	0.00	226.82	1335.00	0.00	106.69	536.00	0.00	103.62	488.00	0.00	4
c-2-m-mV	8.88	118.00	0.00	0.37	1.00	0.00	0.36	1.00	0.00	0.38	1.00	0.00	3
c-2-m-act-2s-mV	7200.14	21004.20	–	7200.21	16370.60	142.42	369.72	843.00	0.00	620.24	1177.00	0.00	3
c-2-m-2s-mV	67.95	516.00	0.00	83.88	481.00	0.00	51.72	387.00	0.00	47.14	308.00	0.00	4
c-2-m-act-2s-mV-int	3585.56	11463.00	0.00	4103.65	9878.00	0.00	61.53	165.00	0.00	56.33	135.00	0.00	4
c-3	143.96	1371.00	0.00	135.92	880.00	0.00	149.73	994.00	0.00	163.38	1107.00	0.00	2
c-3-act	265.58	993.00	0.00	302.13	1040.00	0.00	583.03	2011.00	0.00	333.39	1104.00	0.00	1
c-3-act-int	7200.04	61270.60	886.71	7200.04	52498.20	896.38	684.17	3849.00	0.00	170.56	917.00	0.00	4

	only MISDP			MISDP + approx. freq. 0			MISDP + approx. freq. 4			MISDP + approx. freq. 5			
instance	time	nodes	gap	time	nodes	gap	time	nodes	gap	time	nodes	gap	best
c-3-3s	1101.55	4833.00	0.00	1509.89	5386.00	0.00	1488.73	5224.00	0.00	1498.94	5260.00	0.00	1
c-3-2s	536.50	3427.00	0.00	683.51	3650.00	0.00	409.32	2202.00	0.00	486.32	2492.00	0.00	3
c-3-act-3s-mV-int	0.43	1.00	–	0.44	1.00	–	0.45	1.00	–	0.43	1.00	–	1
c-3-int	13.68	319.00	0.00	15.92	318.00	0.00	11.13	149.00	0.00	13.08	169.00	0.00	3
c-3-mV-int	7.84	194.00	0.00	8.76	165.00	0.00	1.88	41.00	0.00	2.04	39.00	0.00	3
c-3-act-mV-int	7200.10	46609.80	–	102.64	669.00	0.00	101.64	573.00	0.00	113.07	649.00	0.00	3
c-3-act-mV	7200.04	53503.00	5.56	7200.14	32504.60	278.85	56.96	204.00	0.00	67.80	235.00	0.00	3
c-3-cont	0.08	1.00	0.00	0.08	1.00	0.00	0.08	1.00	0.00	0.08	1.00	0.00	2
c-3-cont-act	1.00	5.00	0.00	2.68	7.00	0.00	2.79	7.00	0.00	2.72	7.00	0.00	1
c-3-mV	57.12	781.00	0.00	64.97	672.00	0.00	43.01	529.00	0.00	40.27	459.00	0.00	4
c-3-act-2s	1584.97	4274.00	0.00	2788.40	5450.00	0.00	2322.12	3896.00	0.00	2318.51	4110.00	0.00	1
c-3-act-2s-int	7200.13	41198.80	–	7200.14	26827.60	–	7200.14	22729.20	99.94	6131.47	20599.00	0.00	4
c-3-act-2s-mV	0.45	1.00	–	0.43	1.00	–	0.44	1.00	–	0.43	1.00	–	2
c-3-2s-mV	0.15	1.00	–	0.16	1.00	–	0.17	1.00	–	0.15	1.00	–	1
c-3-2s-int	944.60	11425.00	0.00	1068.56	11424.00	0.00	597.51	5845.00	0.00	833.99	7314.00	0.00	3
c-3-2s-mV-int	0.09	1.00	–	0.09	1.00	–	0.09	1.00	–	0.09	1.00	–	2
c-3-cont-2s	0.13	1.00	0.00	0.14	1.00	0.00	0.13	1.00	0.00	0.13	1.00	0.00	1
c-3-cont-act-2s	1.14	2.00	0.00	13.85	3.00	0.00	14.33	3.00	0.00	13.90	3.00	0.00	1
c-3-act-2s-mV-int	0.27	1.00	–	0.27	1.00	–	0.29	1.00	–	0.28	1.00	–	2
c-3-act-3s-mV-int	7200.31	13298.60	–	7200.49	10877.80	1.31	358.64	581.00	0.00	4094.29	7291.00	0.00	3
c-3-m-mV-int	6.90	184.00	0.00	9.45	198.00	0.00	3.84	103.00	0.00	3.40	92.00	0.00	4
c-3-m-act-mV-int	3387.09	22634.00	0.00	0.70	1.00	0.00	0.68	1.00	0.00	0.74	1.00	0.00	3
c-3-m-act-mV	379.51	3024.00	0.00	24.55	95.00	0.00	14.25	80.00	0.00	19.60	77.00	0.00	3
c-3-m-mV	10.82	154.00	0.00	12.19	152.00	0.00	9.98	128.00	0.00	7.73	102.00	0.00	4
c-3-m-act-2s-mV	6868.89	20610.00	0.00	7200.13	16095.20	95.17	138.60	303.00	0.00	97.37	168.00	0.00	4
c-3-m-2s-mV	39.82	289.00	0.00	34.56	207.00	0.00	33.78	244.00	0.00	32.93	231.00	0.00	4
c-3-m-act-2s-mV-int	7200.18	21778.20	0.46	6172.38	15885.00	0.00	7.95	19.00	0.00	11.47	27.00	0.00	3
c-4	29.82	892.00	0.00	49.53	807.00	0.00	44.67	654.00	0.00	41.99	654.00	0.00	1
c-4-act	89.06	516.00	0.00	139.60	676.00	0.00	114.39	494.00	0.00	88.87	359.00	0.00	4
c-4-3s	565.16	4984.00	0.00	592.51	4589.00	0.00	532.62	4043.00	0.00	504.32	3965.00	0.00	4
c-4-2s	51.19	606.00	0.00	55.57	494.00	0.00	59.72	524.00	0.00	61.21	513.00	0.00	1
c-4-act-mV	351.45	2803.00	0.00	379.91	2501.00	0.00	5.64	30.00	0.00	11.62	73.00	0.00	3
c-4-cont	0.08	1.00	0.00	0.08	1.00	0.00	0.08	1.00	0.00	0.08	1.00	0.00	2
c-4-cont-act	0.19	1.00	0.00	0.21	1.00	0.00	0.22	1.00	0.00	0.20	1.00	0.00	1
c-4-mV	13.18	349.00	0.00	16.70	350.00	0.00	14.68	330.00	0.00	13.78	303.00	0.00	1
c-4-act-2s	185.90	634.00	0.00	335.44	864.00	0.00	348.98	846.00	0.00	516.77	1247.00	0.00	1
c-4-act-2s-mV	0.31	1.00	–	0.30	1.00	–	0.31	1.00	–	0.30	1.00	–	2
c-4-2s-mV	0.08	1.00	–	0.08	1.00	–	0.08	1.00	–	0.09	1.00	–	2
c-4-cont-2s	0.13	1.00	0.00	0.14	1.00	0.00	0.12	1.00	0.00	0.13	1.00	0.00	3
c-4-cont-act-2s	0.41	1.00	0.00	0.40	1.00	0.00	0.41	1.00	0.00	0.42	1.00	0.00	2
c-4-m-act-mV	1141.98	10544.00	0.00	1279.10	10473.00	0.00	34.03	231.00	0.00	20.82	134.00	0.00	4
c-4-m-mV	4.51	123.00	0.00	5.80	114.00	0.00	5.49	129.00	0.00	5.36	127.00	0.00	1
c-4-m-act-2s-mV	0.31	1.00	–	0.30	1.00	–	0.30	1.00	–	0.32	1.00	–	2
c-4-m-2s-mV	0.09	1.00	–	0.08	1.00	–	0.09	1.00	–	0.09	1.00	–	2
c-5	62.88	777.00	0.00	40.61	558.00	0.00	49.74	614.00	0.00	40.23	518.00	0.00	4

	only MISDP			MISDP + approx. freq. 0			MISDP + approx. freq. 4			MISDP + approx. freq. 5			
instance	time	nodes	gap	time	nodes	gap	time	nodes	gap	time	nodes	gap	best
c-5-act	194.91	1655.00	0.00	284.60	2392.00	0.00	267.58	1192.00	0.00	134.98	686.00	0.00	4
c-5-act-int	7200.03	95888.20	–	7200.03	63776.40	–	6150.93	41441.00	0.00	7200.08	49894.00	28.86	3
c-5-3s	1896.61	11398.00	0.00	2195.56	11394.00	0.00	2262.07	11730.00	0.00	2189.69	11804.00	0.00	1
c-5-2s	689.91	4849.00	0.00	420.20	2654.00	0.00	579.25	3554.00	0.00	458.78	2776.00	0.00	2
c-5-act-3s-mV-int	7200.31	12602.40	–	7200.19	10274.00	–	7200.42	10196.20	3.27	7200.32	10231.80	5.77	4
c-5-int	15.15	352.00	0.00	24.14	497.00	0.00	14.65	313.00	0.00	22.49	460.00	0.00	3
c-5-mV-int	2.73	62.00	0.00	0.22	1.00	0.00	0.21	1.00	0.00	0.21	1.00	0.00	3
c-5-act-mV-int	7200.03	46931.00	–	7200.14	33830.60	–	62.25	333.00	0.00	79.33	432.00	0.00	3
c-5-act-mV	7200.07	60929.80	1.30	7200.09	34937.80	375.55	52.26	255.00	0.00	52.47	248.00	0.00	3
c-5-cont	0.08	1.00	0.00	0.08	1.00	0.00	0.07	1.00	0.00	0.09	1.00	0.00	3
c-5-cont-act	0.21	1.00	0.00	0.21	1.00	0.00	0.21	1.00	0.00	0.21	1.00	0.00	2
c-5-mV	8.48	122.00	0.00	5.77	78.00	0.00	5.32	78.00	0.00	4.67	66.00	0.00	4
c-5-act-2s	1639.65	4260.00	0.00	3062.53	6374.00	0.00	2389.50	4393.00	0.00	2256.70	4267.00	0.00	1
c-5-act-2s-int	7200.05	44286.80	–	7200.14	27908.60	–	7200.15	23666.40	–	7200.22	23189.40	> 5000	4
c-5-act-2s-mV	6667.27	21351.00	0.00	7200.35	34521.80	21.35	68.69	100.00	0.00	119.42	198.00	0.00	3
c-5-2s-mV	0.17	1.00	–	0.16	1.00	–	0.17	1.00	–	0.17	1.00	–	2
c-5-2s-int	890.97	14875.00	0.00	1006.80	12163.00	0.00	705.94	8057.00	0.00	696.22	8289.00	0.00	4
c-5-2s-mV-int	0.09	1.00	–	0.08	1.00	–	0.09	1.00	–	0.09	1.00	–	2
c-5-cont-2s	0.13	1.00	0.00	0.13	1.00	0.00	0.13	1.00	0.00	0.13	1.00	0.00	2
c-5-cont-act-2s	0.40	1.00	0.00	0.41	1.00	0.00	0.41	1.00	0.00	0.41	1.00	0.00	1
c-5-act-2s-mV-int	7200.20	20407.20	–	7200.22	18405.40	–				869.40	1962.00	0.00	4
c-5-act-3s-mV-int	5924.36	11553.00	0.00	7200.37	10172.80	–	3000.75	4700.00	0.00	2170.82	3558.00	0.00	4
c-5-m-mV-int	3.22	74.00	0.00	0.12	1.00	0.00	0.13	1.00	0.00	0.12	1.00	0.00	2
c-5-m-act-mV-int	1404.58	13033.00	0.00	1808.23	10965.00	0.00	35.28	205.00	0.00	192.28	1015.00	0.00	3
c-5-m-act-mV	904.02	5681.00	0.00	445.34	2189.00	0.00	52.23	311.00	0.00	141.22	633.00	0.00	3
c-5-m-mV	8.82	118.00	0.00	0.36	1.00	0.00	0.36	1.00	0.00	0.38	1.00	0.00	2
c-5-m-act-2s-mV	7200.09	21549.20	–	7200.27	16666.60	151.17	396.30	796.00	0.00	1205.48	2501.00	0.00	3
c-5-m-2s-mV	67.25	516.00	0.00	84.09	481.00	0.00	51.69	387.00	0.00	46.93	308.00	0.00	4
c-5-m-act-2s-mV-int	7200.11	23141.60	2.57	7200.17	17416.20	5.61	271.81	691.00	0.00	168.36	400.00	0.00	4
c-6	144.11	1371.00	0.00	135.34	880.00	0.00	149.73	994.00	0.00	164.11	1107.00	0.00	2
c-6-act	220.64	795.00	0.00	294.06	1196.00	0.00	324.58	1269.00	0.00	333.48	1257.00	0.00	1
c-6-act-int	7200.04	59571.60	–	7200.06	54939.00	–	1903.07	12083.00	0.00	2554.63	14179.00	0.00	3
c-6-3s	1094.68	4833.00	0.00	1528.85	5386.00	0.00	1478.33	5224.00	0.00	1492.94	5260.00	0.00	1
c-6-2s	533.75	3427.00	0.00	677.65	3650.00	0.00	412.45	2202.00	0.00	490.12	2492.00	0.00	3
c-6-act-3s-mV-int	0.59	1.00	–	0.63	1.00	–	0.61	1.00	–	0.56	1.00	–	4
c-6-int	13.72	319.00	0.00	15.97	318.00	0.00	11.14	149.00	0.00	13.04	169.00	0.00	3
c-6-mV-int	2.11	48.00	0.00	2.38	39.00	0.00	1.98	46.00	0.00	1.89	38.00	0.00	4
c-6-act-mV-int	4101.91	26448.00	0.00	5427.78	26302.00	0.00	25.00	134.00	0.00	21.63	112.00	0.00	4
c-6-act-mV	7200.04	46444.60	9.13	5605.60	23983.00	0.00	147.23	555.00	0.00	155.38	559.00	0.00	3
c-6-cont	0.08	1.00	0.00	0.08	1.00	0.00	0.08	1.00	0.00	0.08	1.00	0.00	2
c-6-cont-act	0.20	1.00	0.00	0.20	1.00	0.00	0.20	1.00	0.00	0.20	1.00	0.00	2
c-6-mV	103.50	1547.00	0.00	131.60	1534.00	0.00	109.35	1421.00	0.00	96.14	1326.00	0.00	4
c-6-act-2s	1534.15	4041.00	0.00	2029.24	4100.00	0.00	1736.49	2640.00	0.00	1305.55	1983.00	0.00	4
c-6-act-2s-int	7200.06	37936.80	–	7200.08	26613.00	–	7200.17	22130.80	–	7200.13	21929.60	–	–
c-6-act-2s-mV	0.44	1.00	–	0.42	1.00	–	0.44	1.00	–	0.44	1.00	–	2

	only MISDP			MISDP + approx. freq. 0			MISDP + approx. freq. 4			MISDP + approx. freq. 5			
instance	time	nodes	gap	time	nodes	gap	time	nodes	gap	time	nodes	gap	best
c-6-2s-mV	0.16	1.00	–	0.16	1.00	–	0.17	1.00	–	0.17	1.00	–	2
c-6-2s-int	940.53	11425.00	0.00	1067.64	11424.00	0.00	599.52	5845.00	0.00	826.62	7314.00	0.00	3
c-6-2s-mV-int	0.09	1.00	–	0.09	1.00	–	0.09	1.00	–	0.09	1.00	–	2
c-6-cont-2s	0.13	1.00	0.00	0.13	1.00	0.00	0.13	1.00	0.00	0.13	1.00	0.00	2
c-6-cont-act-2s	0.39	1.00	0.00	0.40	1.00	0.00	0.43	1.00	0.00	0.40	1.00	0.00	1
c-6-act-2s-mV-int	0.31	1.00	–	0.29	1.00	–	0.30	1.00	–	0.30	1.00	–	2
c-6-act-3s-mV-int	7200.29	13095.00	–	7200.18	10572.00	–	208.78	322.00	0.00	2752.55	4386.00	0.00	3
c-6-m-mV-int	6.89	184.00	0.00	9.49	198.00	0.00	3.87	103.00	0.00	3.38	92.00	0.00	4
c-6-m-act-mV-int	5572.80	38106.00	0.00	7200.07	34888.60	–	5.67	28.00	0.00	8.60	42.00	0.00	3
c-6-m-act-mV	610.11	4824.00	0.00	0.94	1.00	0.00	0.93	1.00	0.00	1.02	1.00	0.00	3
c-6-m-mV	10.87	154.00	0.00	12.17	152.00	0.00	9.93	128.00	0.00	7.73	102.00	0.00	4
c-6-m-act-2s-mV	7200.21	21112.40	–	7200.28	14417.60	103.72	64.33	111.00	0.00	106.32	192.00	0.00	3
c-6-m-2s-mV	39.67	289.00	0.00	34.66	207.00	0.00	33.71	244.00	0.00	32.92	231.00	0.00	4
c-6-m-act-2s-mV-int	6241.48	18606.00	0.00	7200.14	17218.40	0.61	39.34	97.00	0.00	106.04	248.00	0.00	3
c-7	647.95	1692.00	0.00	588.88	1349.00	0.00	561.92	1233.00	0.00	432.00	1095.00	0.00	4
c-7-act	884.80	737.00	0.00	1334.81	728.00	0.00	1535.26	702.00	0.00	1682.04	748.00	0.00	1
c-7-act-int	7200.17	20471.40	–	7200.34	15554.00	–	581.04	1009.00	0.00	455.99	783.00	0.00	4
c-7-3s	7200.34	7370.20	–	7200.64	5355.60	–	7200.86	5362.20	–	7200.79	5303.80	–	–
c-7-2s	7200.44	10093.40	–	7200.46	7115.40	–	7200.49	6658.60	–	7200.33	6608.00	–	–
c-7-act-3s-mV-int	7200.85	4718.20	–	7200.93	3543.20	–	7200.92	4157.40	115.11	7201.30	3414.00	–	3
c-7-int	40.00	347.00	0.00	54.88	355.00	0.00	35.59	198.00	0.00	49.75	243.00	0.00	3
c-7-mV-int	38.74	546.00	0.00	63.85	674.00	0.00	59.63	582.00	0.00	34.65	408.00	0.00	4
c-7-act-mV-int	2804.67	9703.00	0.00	3762.08	9537.00	0.00	7200.21	17234.40	43.32	629.20	1509.00	0.00	4
c-7-act-mV	7200.44	9545.20	–	7200.75	7098.60	1900.04	4622.94	8251.00	0.00	7200.36	6199.00	122.97	3
c-7-cont	0.17	1.00	0.00	0.18	1.00	0.00	0.17	1.00	0.00	0.17	1.00	0.00	1
c-7-cont-act	2.01	3.00	0.00	13.31	3.00	0.00	14.27	3.00	0.00	13.41	3.00	0.00	1
c-7-mV	7200.07	32494.40	32.44	7200.25	24331.20	–	2537.76	22814.00	0.00	2786.49	25964.00	0.00	3
c-7-act-2s	7201.39	2790.60	–	7202.84	2099.60	–	7202.61	1918.40	–	7202.23	1997.60	–	–
c-7-act-2s-int	7200.41	13098.50	–	7200.30	8629.40	–	7200.43	6872.00	–	7200.42	7037.00	–	–
c-7-act-2s-mV	7200.71	5090.33	–	7201.40	3823.40	1644.64	7201.19	3435.80	1681.80	7200.90	3375.20	1695.22	2
c-7-2s-mV	7200.19	19787.83	–	7200.19	14987.00	–	7200.29	12748.40	–	7200.39	13201.80	–	–
c-7-2s-int	1277.31	5616.00	0.00	1080.82	3827.00	0.00	903.27	3198.00	0.00	1214.01	3913.00	0.00	3
c-7-2s-mV-int	45.19	310.00	0.00	57.06	336.00	0.00	27.61	160.00	0.00	27.38	151.00	0.00	4
c-7-cont-2s	0.25	1.00	0.00	0.25	1.00	0.00	0.25	1.00	0.00	0.25	1.00	0.00	2
c-7-cont-act-2s	6.90	5.00	0.00	20.70	7.00	0.00	21.28	7.00	0.00	20.71	7.00	0.00	1
c-7-act-2s-mV-int	7200.23	8540.67	–	7200.50	6691.20	–	7200.30	7148.60	–	7200.70	6934.80	–	–
c-8	143.08	711.00	0.00	191.60	706.00	0.00	126.92	389.00	0.00	115.74	361.00	0.00	4
c-8-act	273.87	661.00	0.00	357.09	667.00	0.00	1188.74	3194.00	0.00	250.23	354.00	0.00	4
c-8-act-mV	52.57	157.00	0.00	80.69	210.00	0.00	66.50	192.00	0.00	57.61	155.00	0.00	1
c-8-cont	0.06	1.00	0.00	0.07	1.00	0.00	0.06	1.00	0.00	0.07	1.00	0.00	1
c-8-cont-act	0.89	3.00	0.00	65.40	6.00	0.00	67.17	6.00	0.00	65.44	6.00	0.00	1
c-8-mV	23.70	178.00	0.00	19.58	143.00	0.00	31.98	233.00	0.00	25.65	175.00	0.00	2
c-9	288.71	700.00	0.00	475.16	1114.00	0.00	252.10	243.00	0.00	162.46	175.00	0.00	4
c-9-act	7200.80	4865.00	–	3054.99	1584.00	0.00	2134.96	1032.00	0.00	4307.30	1779.00	0.00	3
c-9-act-int	7200.22	20785.60	–				7200.34	11445.40	–	7200.25	11907.00	–	–

	only MISDP			MISDP + approx. freq. 0			MISDP + approx. freq. 4			MISDP + approx. freq. 5			
instance	time	nodes	gap	time	nodes	gap	time	nodes	gap	time	nodes	gap	best
c-9-3s	7200.59	6265.80	57.87	7200.46	4412.40	–	7200.57	5032.20	–	7200.76	5051.60	–	1
c-9-2s	7200.50	8371.60	554.14	7200.55	6325.40	–	7200.40	7107.20	57.96	7200.62	6324.60	114.48	3
c-9-act-3s-mV-int	7201.31	4316.00	–	7201.17	3097.00	177.21	7201.05	3793.20	221.15	7201.22	3411.80	215.32	2
c-9-int	1755.88	12880.00	0.00	2099.04	12210.00	0.00	3475.09	17454.00	0.00	2781.19	14036.00	0.00	1
c-9-mV-int	100.71	1910.00	0.00	275.17	2671.00	0.00	107.35	1310.00	0.00	107.34	1478.00	0.00	1
c-9-act-mV-int	7200.11	38860.40	–	7200.19	26421.40	–	7200.21	12204.60	11.83	7200.24	12112.00	625.13	4
c-9-act-mV	7200.38	8819.60	–	7200.82	6317.20	1737.92	7200.23	10462.00	14.41	7200.55	6695.80	20.04	3
c-9-cont	0.17	1.00	0.00	0.18	1.00	0.00	0.18	1.00	0.00	0.18	1.00	0.00	1
c-9-cont-act	0.66	1.00	0.00	0.64	1.00	0.00	0.64	1.00	0.00	0.64	1.00	0.00	2
c-9-mV	5070.35	26591.00	0.00	6150.73	26445.00	0.00	2163.84	9473.00	0.00	2379.92	9500.00	0.00	3
c-9-act-2s	7202.07	2171.20	–	7201.97	1636.60	–	7201.72	1636.60	–	7202.14	1591.40	–	–
c-9-act-2s-int	7200.35	9002.00	–	7200.68	6440.20	–	7200.93	5329.20	–	7200.71	5328.80	–	–
c-9-act-2s-mV	7201.12	3984.83	–	7200.61	3108.20	163.65	7201.14	2886.80	172.47	7201.31	2829.80	170.99	2
c-9-2s-mV	7200.32	15817.50	–	7200.34	13005.00	225.05	7200.33	11348.80	228.25	7200.16	12319.60	16.32	4
c-9-2s-int	7200.13	27379.83	–	7200.11	21576.20	–	7200.16	20235.40	–	7200.12	19598.40	> 5000	4
c-9-2s-mV-int	85.10	493.00	0.00	7200.06	57006.20	97.15	812.07	8715.00	0.00	130.09	1133.00	0.00	1
c-9-cont-2s	0.32	1.00	0.00	0.33	1.00	0.00	0.33	1.00	0.00	0.34	1.00	0.00	1
c-9-cont-act-2s	1.35	1.00	0.00	1.31	1.00	0.00	1.40	1.00	0.00	1.24	1.00	0.00	4
c-9-act-2s-mV-int	7200.51	7600.17	–	7200.31	5943.40	–	7200.77	5819.00	–	7200.36	5939.80	1652.34	4
c-10	986.13	2712.00	0.00	449.34	1045.00	0.00	309.16	534.00	0.00	268.31	457.00	0.00	4
c-10-act	2663.99	3100.00	0.00	1053.37	1112.00	0.00	857.25	679.00	0.00	911.95	568.00	0.00	3
c-10-act-int	7200.22	24836.40	–	7200.20	17531.00	–	7200.19	17096.60	8.22	7178.64	18470.20	7.92	4
c-10-3s	7200.24	9592.20	–	7200.53	6987.60	–	7200.39	6984.00	–	7200.41	7212.40	–	–
c-10-2s	7200.27	12644.20	421.05	7200.34	8989.40	965.17	7200.26	8644.60	–	7200.64	9288.00	216.80	4
c-10-act-3s-mV-int	7200.56	7530.40	–	7200.91	5306.80	–	7200.48	6132.80	–	7200.76	6043.80	–	–
c-10-int	15.99	211.00	0.00	17.32	208.00	0.00	7.17	43.00	0.00	8.15	51.00	0.00	3
c-10-mV-int	77.06	1192.00	0.00	53.08	720.00	0.00	15.51	303.00	0.00	15.88	286.00	0.00	3
c-10-act-mV-int	1504.91	6456.00	0.00	7200.16	24809.00	–	7200.12	25300.00	3225.15	7200.15	25018.60	> 5000	1
c-10-act-mV	7200.10	39296.40	–	7200.05	34669.20	3060.46	4908.21	16194.00	0.00	4162.94	8690.00	0.00	4
c-10-cont	0.15	1.00	0.00	0.14	1.00	0.00	0.15	1.00	0.00	0.14	1.00	0.00	2
c-10-cont-act	3.94	8.00	0.00	15.45	25.00	0.00	23.29	26.00	0.00	31.73	24.00	0.00	1
c-10-mV	7200.05	49142.80	–	7200.13	39068.00	–	751.61	7801.00	0.00	1418.98	10013.00	0.00	3
c-10-act-2s	7200.99	3927.80	475.66	7201.74	2902.20	749.42	7201.03	2697.80	–	7201.40	2736.20	751.36	1
c-10-act-2s-int	7200.17	24520.17	–	7200.32	11526.80	–	7200.25	9275.20	156.42	7200.21	9818.80	–	3
c-10-act-2s-mV	7200.60	6445.33	–	7201.17	5690.40	1847.28	7200.73	4542.60	1869.87	7200.52	4672.20	1861.14	2
c-10-2s-mV	7200.13	25495.83	–	7200.24	19478.40	–	7200.31	14567.80	–	7200.13	15375.80	524.89	4
c-10-2s-int	2687.78	15006.00	0.00	3420.22	14687.00	0.00	1191.17	5165.00	0.00	1139.21	4501.00	0.00	4
c-10-2s-mV-int	15.75	119.00	0.00	29.03	205.00	0.00	62.29	448.00	0.00	16.71	134.00	0.00	1
c-10-cont-2s	0.22	1.00	0.00	0.22	1.00	0.00	0.22	1.00	0.00	0.22	1.00	0.00	2
c-10-cont-act-2s	22.77	25.00	0.00	78.41	64.00	0.00	292.17	104.00	0.00	193.15	74.00	0.00	1
c-10-act-2s-mV-int	7200.25	13182.83	–	7200.41	8653.80	127.29	7200.38	10469.60	179.07	7200.54	10021.60	156.57	2
c-big	1253.32	1590.00	0.00	3085.07	2222.00	0.00	2355.99	1176.00	0.00	1828.27	974.00	0.00	1
c-big-act	5972.22	2225.00	0.00	7202.16	1255.40	92.08	7162.38	1084.20	57.39	7126.87	1556.40	3.02	1
c-big-act-int	7200.65	10002.80	–	7200.57	7207.60	–	7200.63	6508.60	18.16	2511.75	2346.00	0.00	4
c-big-3s	7201.23	2781.00	–	7201.92	1966.60	–	7201.87	2004.60	–	7201.89	1990.20	–	–

	only MISDP			MISDP + approx. freq. 0			MISDP + approx. freq. 4			MISDP + approx. freq. 5			
instance	time	nodes	gap	time	nodes	gap	time	nodes	gap	time	nodes	gap	best
c-big-2s	7201.10	3374.80	–	7202.00	2353.40	–	7201.22	2292.40	–	7201.51	2242.20	–	–
c-big-act-3s-mV-int	7202.67	1781.00	–	7203.98	1171.20	145.87	7203.06	1146.20	154.57	7202.43	1385.40	153.10	2
c-big-int	155.05	807.00	0.00	190.72	808.00	0.00	163.52	502.00	0.00	150.75	516.00	0.00	4
c-big-mV-int	130.11	931.00	0.00	147.88	846.00	0.00	94.63	666.00	0.00	139.18	766.00	0.00	3
c-big-act-mV-int	7200.29	22467.20	54.56	2702.25	3646.00	0.00	1189.57	787.00	0.00	965.13	711.00	0.00	4
c-big-act-mV	7201.13	2818.20	–	7201.80	2543.60	3695.23	7202.09	1896.00	4143.83	7201.18	2091.20	1802.30	4
c-big-cont	0.32	1.00	0.00	0.33	1.00	0.00	0.34	1.00	0.00	0.32	1.00	0.00	1
c-big-cont-act	4.25	2.00	0.00	45.77	4.00	0.00	49.86	4.00	0.00	46.79	4.00	0.00	1
c-big-mV	7200.37	11109.00	–	7200.33	8008.20	271.66	7200.13	26845.60	6.93	7200.14	26124.60	4.31	4
c-big-act-2s	7201.90	805.80	–	7203.89	617.40	–	7208.27	554.40	–	7207.03	604.60	–	–
c-big-act-2s-int	7201.36	3223.83	–	7201.60	2389.60	–	7201.36	2191.00	–	7201.80	2078.00	–	–
c-big-act-2s-mV	7204.06	1449.83	–	7202.92	1218.00	3079.11	7202.30	1165.00	3147.58	7202.67	1132.20	3123.59	2
c-big-2s-mV	7200.51	7202.17	–	7200.76	5402.00	347.96	7200.79	5369.40	350.39	7200.56	5157.60	350.06	2
c-big-2s-int	7200.27	14926.17	–	7200.29	11669.60	–	7200.32	11365.60	–	7200.47	10861.20	> 5000	4
c-big-2s-mV-int	28.28	105.00	0.00	42.96	146.00	0.00	6.03	24.00	0.00	8.12	31.00	0.00	3
c-big-cont-2s	0.52	1.00	0.00	0.51	1.00	0.00	0.52	1.00	0.00	0.52	1.00	0.00	2
c-big-cont-act-2s	17.14	5.00	0.00	248.45	6.00	0.00	261.38	6.00	0.00	244.32	6.00	0.00	1
c-big-act-2s-mV-int	7201.10	3351.67	–	7201.32	2632.80	–	7201.55	2681.80	–	7200.95	2612.80	–	–
lit-as-1	1.69	85.00	0.00	1.74	64.00	0.00	1.65	53.00	0.00	1.67	57.00	0.00	3
lit-as-1-act	4.91	79.00	0.00	5.60	62.00	0.00	6.45	70.00	0.00	5.07	57.00	0.00	1
lit-as-1-act-mV	10.42	256.00	0.00	12.85	246.00	0.00	13.70	298.00	0.00	13.01	270.00	0.00	1
lit-as-1-cont	0.02	1.00	0.00	0.02	1.00	0.00	0.02	1.00	0.00	0.02	1.00	0.00	2
lit-as-1-cont-act	1.80	37.00	0.00	1.47	9.00	0.00	1.55	9.00	0.00	1.50	9.00	0.00	2
lit-as-1-mV	0.97	71.00	0.00	1.07	52.00	0.00	1.16	57.00	0.00	0.92	50.00	0.00	4
lit-as-2	7202.72	1175.40	–	7206.37	727.40	2127.83	7205.29	746.00	2162.67	7204.71	723.00	2160.38	2
lit-as-2-act	7279.63	54.20	–	7274.44	41.00	–	7272.40	39.60	–	7282.27	40.00	–	–
lit-as-2-act-mV	7243.74	91.20	–	7234.53	79.20	–	7248.26	59.00	–	7254.24	64.60	–	–
lit-as-2-cont	5.77	1.00	0.00	6.31	1.00	0.00	7.72	1.00	0.00	6.88	1.00	0.00	1
lit-as-2-cont-act	7204.88	43.00	–	7236.84	1.00	–	7237.05	1.00	–	7280.40	1.00	–	–
lit-as-2-mV	7203.17	1796.00	2285.93	7202.97	1483.40	2289.40	7201.88	1488.20	2289.90	7202.76	1553.60	2289.97	1
lit-as-2-big	7476.55	13.60	–	7710.54	10.20	–	7638.37	10.60	–	7607.50	10.00	–	–
lit-as-2-big-act	8155.19	4.80	–	8155.33	3.00	–	7979.66	2.40	–	7805.57	2.20	–	–
lit-as-2-big-act-int	7244.26	84.20	–	7251.86	62.00	3877.13	7251.63	64.40	3877.22	7235.44	62.40	3877.35	2
lit-as-2-big-int	7202.46	1107.80	–	7202.58	903.80	–	7203.34	935.60	–	7203.87	984.60	–	–
lit-as-2-big-mV-int	7202.49	1862.20	> 5000	7202.82	1133.60	> 5000	7203.13	1133.40	5.89	7203.78	1112.60	7.14	3
lit-as-2-big-act-mV-int	7239.88	102.40	–	7258.12	77.80	–	7213.04	84.20	–	7258.81	85.20	–	–
lit-as-2-big-act-mV	7743.18	10.80	–	7652.12	7.00	–							–
lit-as-2-big-cont	5.94	1.00	0.00	6.48	1.00	0.00	7.49	1.00	0.00	7.04	1.00	0.00	1
lit-as-2-big-cont-act	7249.13	87.20	–	7232.38	1.00	–	7227.26	1.00	–	7278.08	1.00	–	–
lit-as-2-big-mV	7311.05	27.80	–	7264.43	19.20	–	7306.50	18.20	–	7313.95	19.80	–	–
lit-as-5	7200.46	6852.80	–	7200.94	3939.00	–	7201.16	4064.60	–	7200.62	3995.20	–	–
lit-as-5-act	7208.55	463.20	–	7208.14	315.80	–	7210.03	346.00	–	7207.91	321.00	–	–
lit-as-5-act-mV	7205.72	927.20	–	7209.80	556.20	464.97	7206.47	589.80	481.97	7205.86	626.00	18.31	4
lit-as-5-cont	1.01	1.00	0.00	1.05	1.00	0.00	1.16	1.00	0.00	1.10	1.00	0.00	1
lit-as-5-cont-act	9.31	1.00	0.00	11.90	1.00	0.00	11.10	1.00	0.00	12.89	1.00	0.00	1

	only MISDP			MISDP + approx. freq. 0			MISDP + approx. freq. 4			MISDP + approx. freq. 5			
instance	time	nodes	gap	time	nodes	gap	time	nodes	gap	time	nodes	gap	best
lit-as-5-mV	2065.63	3693.00	0.00	3153.15	3282.00	0.00	2539.43	3324.00	0.00	2720.86	3418.00	0.00	1
lit-as-5-big	7225.82	107.40	–	7245.38	98.80	> 5000	7231.22	88.20	> 5000	7237.15	92.80	> 5000	2
lit-as-5-big-act	7223.13	113.80	–	7240.68	95.40	> 5000	7256.74	98.60	> 5000	7221.60	98.20	> 5000	2
lit-as-5-big-act-int	5676.15	5402.00	0.00	7200.64	5712.20	> 5000	7200.66	5677.20	0.28	7200.66	5515.20	> 5000	1
lit-as-5-big-int	5688.25	5402.00	0.00	7200.66	5532.00	> 5000	7200.25	5774.00	0.27	7200.50	5321.00	> 5000	1
lit-as-5-big-mV-int	1087.96	1949.00	0.00	1697.08	1808.00	0.00	1421.47	1836.00	0.00	1288.30	1736.00	0.00	1
lit-as-5-big-act-mV-int	1102.83	1949.00	0.00	1596.30	1808.00	0.00	1336.79	1836.00	0.00	1288.62	1736.00	0.00	1
lit-as-5-big-act-mV	7232.06	184.60	–	7213.55	196.20	–	7220.94	188.80	–	7220.42	200.40	–	–
lit-as-5-big-cont	1.04	1.00	0.00	1.05	1.00	0.00	1.12	1.00	0.00	1.10	1.00	0.00	1
lit-as-5-big-cont-act	1.06	1.00	0.00	1.09	1.00	0.00	1.07	1.00	0.00	1.11	1.00	0.00	1
lit-as-5-big-mV	7212.27	257.00	–	7225.61	186.20	–	7212.40	203.20	–	7214.74	212.00	–	–
lit-as-6	2411.97	5229.00	0.00	4228.72	6609.00	0.00	2349.83	3927.00	0.00	2250.07	3711.00	0.00	4
lit-as-6-act	2419.81	5229.00	0.00	4164.43	6609.00	0.00	2364.84	3927.00	0.00	2259.57	3711.00	0.00	4
lit-as-6-act-mV	2175.08	6978.00	0.00	2427.05	7449.00	0.00	1456.18	4150.00	0.00	1126.62	3197.00	0.00	4
lit-as-6-cont	0.51	1.00	0.00	0.52	1.00	0.00	0.53	1.00	0.00	0.52	1.00	0.00	1
lit-as-6-cont-act	0.50	1.00	0.00	0.52	1.00	0.00	0.52	1.00	0.00	0.53	1.00	0.00	1
lit-as-6-mV	2161.52	6978.00	0.00	2432.12	7449.00	0.00	1458.10	4150.00	0.00	1110.42	3197.00	0.00	4
lit-as-6-big	7202.45	1122.60	–	7203.56	771.00	–	7203.62	794.40	–	7205.27	814.00	–	–
lit-as-6-big-act	7202.82	1116.80	–	7204.44	772.60	–	7203.22	801.60	–	7204.53	828.80	–	–
lit-as-6-big-act-int	7200.19	14550.60	–	7200.38	13113.60	–	6633.28	11918.00	0.00	7200.24	13199.60	0.80	3
lit-as-6-big-int	7200.26	14542.00	–	7200.21	12863.60	–	6748.14	11918.00	0.00	7200.28	12959.80	0.83	3
lit-as-6-big-mV-int	1219.35	3696.00	0.00	1471.90	3594.00	0.00	830.18	1881.00	0.00	690.46	1607.00	0.00	4
lit-as-6-big-act-mV-int	1217.89	3696.00	0.00	1447.09	3594.00	0.00	803.06	1881.00	0.00	686.62	1607.00	0.00	4
lit-as-6-big-act-mV	7202.84	1904.40	–	7202.62	1648.80	–	7201.52	2024.20	–	7201.44	1956.00	–	–
lit-as-6-big-cont	0.50	1.00	0.00	0.52	1.00	0.00	0.53	1.00	0.00	0.51	1.00	0.00	1
lit-as-6-big-cont-act	0.50	1.00	0.00	0.52	1.00	0.00	0.52	1.00	0.00	0.53	1.00	0.00	1
lit-as-6-big-mV	7202.58	1907.80	–	7201.92	1647.00	–	7202.00	1974.20	–	7201.59	1974.60	–	–
lit-s-1	4459.78	29779.00	0.00	7006.64	29452.00	0.00	5265.58	29942.00	0.00	5237.65	29566.00	0.00	1
lit-s-1-act	7200.35	10697.00	–	7200.76	5646.80	2182.32	7200.47	7330.20	20.84	7200.35	6981.20	35.32	3
lit-s-1-act-mV	75.62	140.00	0.00	88.70	104.00	0.00	62.58	117.00	0.00	54.90	136.00	0.00	4
lit-s-1-cont	0.15	1.00	0.00	0.16	1.00	0.00	0.17	1.00	0.00	0.16	1.00	0.00	1
lit-s-1-cont-act	7200.12	31.00	–	7202.36	1.00	–	7201.47	1.00	–	7203.36	1.00	–	–
lit-s-1-mV	7.56	88.00	0.00	9.91	84.00	0.00	6.98	89.00	0.00	7.22	83.00	0.00	3

Table A.2. – Comparing different parameter settings for 528 the truss instances.

Presolving

We also solved most instances of our test set, precisely 426 instances, without any presolving idea, we presented in Section 3.3. The results are presented in Table A.3, if using the presolving ideas (‘*S*’) or not (‘*nP*’) is better is presented in the last column.

	solving as MISDP			without presolve			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
bridge-1	7.00	132.00	0.00	6.92	132.00	0.00	nP
bridge-1-act	20.33	130.00	0.00	21.90	154.17	0.00	S
bridge-1-2s	265.19	2786.00	0.00	278.83	2968.00	0.00	S
bridge-1-act-mV	2061.81	16147.00	0.00	1977.46	16157.00	0.00	nP
bridge-1-cont	0.08	1.00	0.00	0.09	1.00	0.00	S
bridge-1-cont-act	0.82	3.00	0.00	0.90	3.83	0.00	S
bridge-1-mV	38.84	932.00	0.00	36.72	941.00	0.00	nP
bridge-1-act-2s	867.47	2469.00	0.00	909.95	2740.00	0.00	S
bridge-1-act-2s-mV	3992.90	15466.00	0.00	4143.17	16349.00	0.00	S
bridge-1-2s-mV	296.97	3939.00	0.00	299.38	4021.00	0.00	S
bridge-1-cont-2s	0.18	1.00	0.00	0.17	1.00	0.00	nP
bridge-1-cont-act-2s	10.20	26.00	0.00	6.92	18.00	0.00	nP
bridge-2	134.94	1158.00	0.00	247.02	2349.67	0.00	S
bridge-2-act	288.50	1177.00	0.00	590.24	2602.00	0.00	S
bridge-2-act-int	7200.03	82811.60	–	7200.03	84858.40	–	–
bridge-2-2s	7200.10	39747.60	907.06	7200.13	39162.40	5217.84	S
bridge-2-int	40.45	765.00	0.00	60.15	1123.00	0.00	S
bridge-2-mV-int	18.35	327.00	0.00	12.85	328.00	0.00	nP
bridge-2-act-mV-int	7200.08	43084.00	–	7200.05	55578.20	–	–
bridge-2-act-mV	1902.13	13018.00	0.00	580.78	5035.00	0.00	nP
bridge-2-cont	0.08	1.00	0.00	0.09	1.00	0.00	S
bridge-2-cont-act	0.81	3.00	0.00	0.90	3.83	0.00	S
bridge-2-mV	76.09	1096.00	0.00	146.08	2677.00	0.00	S
bridge-2-act-2s	7200.28	14171.40	–	7200.17	15282.40	–	–
bridge-2-act-2s-int	7200.12	34997.50	–	7200.06	42299.60	–	–
bridge-2-act-2s-mV	6336.03	24501.00	0.00	7200.10	25286.80	2.68	S
bridge-2-2s-mV	1334.01	13544.00	0.00	3172.33	33150.00	0.00	S
bridge-2-2s-int	434.50	4031.00	0.00	597.40	5983.00	0.00	S
bridge-2-2s-mV-int	259.32	2551.00	0.00	245.00	2515.00	0.00	nP
bridge-2-cont-2s	0.17	1.00	0.00	0.18	1.00	0.00	S
bridge-2-cont-act-2s	10.34	26.00	0.00	6.96	18.00	0.00	nP
bridge-2-act-2s-mV-int	7200.24	19941.00	–	7200.23	20727.40	–	–
bridge-3	44.21	415.00	0.00	44.35	415.00	0.00	S
bridge-3-act	184.18	724.00	0.00	182.72	729.83	0.00	nP
bridge-3-act-int	7200.06	71202.00	–	7200.09	76835.20	–	–
bridge-3-2s	2526.70	14388.00	0.00	2183.28	12383.00	0.00	nP
bridge-3-int	71.79	1220.00	0.00	76.11	1369.00	0.00	S

	solving as MISDP			without presolve			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
bridge-3-mV-int	6.38	123.00	0.00	4.50	123.00	0.00	nP
bridge-3-act-mV-int	7200.09	43322.20	–	7200.11	54485.60	–	–
bridge-3-act-mV	7200.07	40837.00	13.74	7200.07	42051.00	13.98	S
bridge-3-cont	0.10	1.00	0.00	0.10	1.00	0.00	S
bridge-3-cont-act	0.87	3.00	0.00	0.95	3.83	0.00	S
bridge-3-mV	90.90	1071.00	0.00	81.37	1071.00	0.00	nP
bridge-3-act-2s	6223.10	12837.00	0.00	5210.81	11182.00	0.00	nP
bridge-3-act-2s-int	7200.13	33016.00	–	7200.10	37647.60	–	–
bridge-3-act-2s-mV	7200.13	20692.25	–	7200.14	21586.20	–	–
bridge-3-2s-mV	2343.01	18040.00	0.00	2439.63	19240.00	0.00	S
bridge-3-2s-int	300.62	2634.00	0.00	333.88	3015.00	0.00	S
bridge-3-2s-mV-int	47.68	464.00	0.00	45.99	464.00	0.00	nP
bridge-3-cont-2s	0.17	1.00	0.00	0.18	1.00	0.00	S
bridge-3-cont-act-2s	5.24	11.00	0.00	6.99	18.00	0.00	S
bridge-3-act-2s-mV-int	7200.16	19829.00	–	7200.14	21067.80	–	–
bridge-4	7.08	132.00	0.00	7.00	132.00	0.00	nP
bridge-4-act	22.73	136.00	0.00	22.76	136.00	0.00	S
bridge-4-2s	265.70	2786.00	0.00	277.76	2968.00	0.00	S
bridge-4-act-mV	1994.79	15646.00	0.00	1994.19	15675.00	0.00	nP
bridge-4-cont	0.08	1.00	0.00	0.09	1.00	0.00	S
bridge-4-cont-act	0.75	3.00	0.00	0.77	3.00	0.00	S
bridge-4-mV	26.51	618.00	0.00	25.32	618.00	0.00	nP
bridge-4-act-2s	852.55	2563.00	0.00	884.16	2769.00	0.00	S
bridge-4-act-2s-mV	3808.52	14780.00	0.00	3957.01	15663.00	0.00	S
bridge-4-2s-mV	251.05	3289.00	0.00	253.95	3369.00	0.00	S
bridge-4-cont-2s	0.17	1.00	0.00	0.18	1.00	0.00	S
bridge-4-cont-act-2s	9.84	21.00	0.00	7.39	18.00	0.00	nP
bridge-5	133.85	1158.00	0.00	247.62	2349.67	0.00	S
bridge-5-act	292.69	1179.00	0.00	628.63	2575.67	0.00	S
bridge-5-act-int	7200.06	80355.20	–	7200.03	70841.60	–	–
bridge-5-2s	7200.13	39869.00	906.85	7200.05	39176.20	5217.92	S
bridge-5-int	40.83	765.00	0.00	59.92	1123.00	0.00	S
bridge-5-mV-int	18.37	327.00	0.00	12.88	328.00	0.00	nP
bridge-5-act-mV-int	7200.18	43057.80	–	7200.04	56032.80	–	–
bridge-5-act-mV	2309.51	14623.00	0.00	691.58	4739.00	0.00	nP
bridge-5-cont	0.08	1.00	0.00	0.08	1.00	0.00	S
bridge-5-cont-act	0.75	3.00	0.00	0.73	3.00	0.00	nP
bridge-5-mV	75.54	1096.00	0.00	145.34	2677.00	0.00	S
bridge-5-act-2s	7200.23	14070.80	–	7200.23	14825.80	–	–
bridge-5-act-2s-int	7200.09	31797.50	–	7200.07	39490.00	–	–
bridge-5-act-2s-mV	7200.20	25463.00	1.05	7200.08	22908.80	9.05	S
bridge-5-2s-mV	1337.17	13544.00	0.00	3175.99	33150.00	0.00	S
bridge-5-2s-int	433.91	4031.00	0.00	598.38	5983.00	0.00	S

	solving as MISDP			without presolve			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
bridge-5-2s-mV-int	259.10	2551.00	0.00	246.18	2515.00	0.00	nP
bridge-5-cont-2s	0.18	1.00	0.00	0.17	1.00	0.00	nP
bridge-5-cont-act-2s	9.72	21.00	0.00	7.45	18.00	0.00	nP
bridge-5-act-2s-mV-int	7200.26	19076.50	–	7200.19	20205.00	–	–
bridge-6	44.44	415.00	0.00	43.94	415.00	0.00	nP
bridge-6-act	212.02	800.00	0.00	208.39	805.83	0.00	nP
bridge-6-act-int	7200.06	66028.80	–	7200.06	74710.60	–	–
bridge-6-2s	2523.85	14388.00	0.00	2186.68	12383.00	0.00	nP
bridge-6-int	72.09	1220.00	0.00	75.19	1369.00	0.00	S
bridge-6-mV-int	19.55	400.00	0.00	14.09	400.00	0.00	nP
bridge-6-act-mV-int	7200.12	43208.80	–	7200.06	55914.80	–	–
bridge-6-act-mV	7200.06	40219.40	–	7200.09	41307.00	–	–
bridge-6-cont	0.10	1.00	0.00	0.10	1.00	0.00	S
bridge-6-cont-act	0.64	2.00	0.00	0.73	2.83	0.00	S
bridge-6-mV	47.47	548.00	0.00	42.26	548.00	0.00	nP
bridge-6-act-2s	6727.72	14479.00	0.00	6045.23	13292.00	0.00	nP
bridge-6-act-2s-int	7200.03	29885.75	–	7200.10	35843.00	–	–
bridge-6-act-2s-mV	7200.18	20334.75	–	7200.12	21243.00	–	–
bridge-6-2s-mV	1453.11	10890.00	0.00	1514.91	11571.00	0.00	S
bridge-6-2s-int	299.63	2634.00	0.00	334.58	3015.00	0.00	S
bridge-6-2s-mV-int	106.53	1092.00	0.00	103.10	1091.00	0.00	nP
bridge-6-cont-2s	0.17	1.00	0.00	0.18	1.00	0.00	S
bridge-6-cont-act-2s	3.51	7.00	0.00	7.66	18.00	0.00	S
bridge-6-act-2s-mV-int	7200.16	19462.00	–	7200.20	20449.00	–	–
bridge-7	1683.93	3520.00	0.00	1593.38	3517.50	0.00	nP
bridge-7-act	5841.99	4455.00	0.00	5381.06	4535.83	0.00	nP
bridge-7-act-int	7200.24	30518.00	–	7200.11	31711.00	–	–
bridge-7-2s	7200.22	8013.20	–	7200.15	8524.80	–	–
bridge-7-int	12.82	135.00	0.00	45.04	344.00	0.00	S
bridge-7-mV-int	1.41	14.00	0.00	5.10	47.00	0.00	S
bridge-7-act-mV-int	843.02	4210.00	0.00	7200.20	30795.80	1.21	S
bridge-7-act-mV	312.72	267.00	0.00	1874.67	2778.00	0.00	S
bridge-7-cont	0.17	1.00	0.00	0.17	1.00	0.00	S
bridge-7-cont-act	8.41	15.00	0.00	10.18	19.17	0.00	S
bridge-7-mV	20.86	49.00	0.00	153.84	427.00	0.00	S
bridge-7-act-2s	7201.14	2335.00	–	7201.44	2670.80	–	–
bridge-7-act-2s-int	7200.19	16759.00	–	7200.14	16179.20	–	–
bridge-7-act-2s-mV	7200.61	3696.50	–	7200.88	4199.20	–	–
bridge-7-2s-mV	646.36	1484.00	0.00	1238.26	3035.00	0.00	S
bridge-7-2s-int	329.92	1228.00	0.00	924.67	3269.00	0.00	S
bridge-7-2s-mV-int	86.89	486.00	0.00	157.82	891.00	0.00	S
bridge-7-cont-2s	0.33	1.00	0.00	0.35	1.00	0.00	S
bridge-7-cont-act-2s	30.27	25.00	0.00	23.00	21.00	0.00	nP

	solving as MISDP			without presolve			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
bridge-7-act-2s-mV-int	7200.16	17316.00	–	7200.43	16953.20	–	–
bridge-8	7200.47	4664.60	–	7200.66	5017.33	–	–
bridge-8-act	7202.03	1411.20	–	7202.38	1485.50	–	–
bridge-8-act-int	7200.54	8143.80	–	7200.23	10188.60	–	–
bridge-8-2s	7200.94	3061.40	–	7200.97	3233.20	–	–
bridge-8-int	1636.89	5354.00	0.00	7200.16	25369.80	–	S
bridge-8-mV-int	70.72	365.00	0.00	451.20	2675.00	0.00	S
bridge-8-act-mV-int	7200.15	8002.60	–	7200.08	14281.20	–	–
bridge-8-act-mV	7200.96	2263.60	–	7201.28	2112.20	–	–
bridge-8-cont	0.29	1.00	0.00	0.29	1.00	0.00	S
bridge-8-cont-act	14.39	11.00	0.00	7.58	5.17	0.00	nP
bridge-8-mV	2217.31	3085.00	0.00	7200.21	12898.80	2.16	S
bridge-8-act-2s	7208.01	691.60	–	7205.63	749.60	–	–
bridge-8-act-2s-int	7201.26	4034.00	–	7200.82	4761.00	–	–
bridge-8-act-2s-mV	7203.12	1132.75	–	7202.25	1263.60	–	–
bridge-8-2s-mV	7200.68	5928.25	–	7200.34	6318.20	–	–
bridge-8-2s-int	3106.06	5509.00	0.00	7200.34	13546.00	–	S
bridge-8-2s-mV-int	129.16	370.00	0.00	844.83	2549.00	0.00	S
bridge-8-cont-2s	0.52	1.00	0.00	0.53	1.00	0.00	S
bridge-8-cont-act-2s	36.33	11.00	0.00	12.71	4.00	0.00	nP
bridge-8-act-2s-mV-int	7201.45	2974.75	–	7200.49	3718.00	–	–
bridge-9	76.64	435.00	0.00	74.95	435.00	0.00	nP
bridge-9-act	363.05	1001.00	0.00	355.50	1017.67	0.00	nP
bridge-9-act-int	7200.05	65900.20	–	7200.06	69735.60	–	–
bridge-9-2s	378.53	1249.00	0.00	350.93	1109.00	0.00	nP
bridge-9-int	69.09	1206.00	0.00	205.30	7506.00	0.00	S
bridge-9-mV-int	16.66	545.00	0.00	22.74	926.00	0.00	S
bridge-9-act-mV-int	404.55	2907.00	0.00	214.42	1767.00	0.00	nP
bridge-9-act-mV	284.25	915.00	0.00	124.87	408.00	0.00	nP
bridge-9-cont	0.10	1.00	0.00	0.10	1.00	0.00	S
bridge-9-cont-act	13.02	82.00	0.00	10.80	57.00	0.00	nP
bridge-9-mV	45.36	498.00	0.00	74.32	694.00	0.00	S
bridge-9-act-2s	880.19	1180.00	0.00	818.94	1180.00	0.00	nP
bridge-9-act-2s-int	7200.07	58127.25	–	7200.14	38289.20	–	–
bridge-9-act-2s-mV	254.04	425.00	0.00	166.43	300.00	0.00	nP
bridge-9-2s-mV	37.05	255.00	0.00	10.63	121.00	0.00	nP
bridge-9-2s-int	464.83	5745.00	0.00	588.98	10129.00	0.00	S
bridge-9-2s-mV-int	18.31	319.00	0.00	10.34	177.00	0.00	nP
bridge-9-cont-2s	0.17	1.00	0.00	0.18	1.00	0.00	S
bridge-9-cont-act-2s	22.73	79.00	0.00	14.60	35.00	0.00	nP
bridge-9-act-2s-mV-int	514.44	1511.00	0.00	310.02	973.00	0.00	nP
bridge-10	1081.06	6742.00	0.00	1074.79	6759.50	0.00	nP
bridge-10-act	968.78	1791.00	0.00	926.59	1790.17	0.00	nP

	solving as MISDP			without presolve			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
bridge-10-act-int	7200.19	22660.60	–	7200.14	24556.60	–	–
bridge-10-2s	7200.18	18738.00	0.18	6988.02	16885.00	0.00	nP
bridge-10-int	48.95	455.00	0.00	46.10	449.00	0.00	nP
bridge-10-mV-int	131.47	2958.00	0.00	150.56	4182.00	0.00	S
bridge-10-act-mV-int	1169.86	5204.00	0.00	588.88	2841.00	0.00	nP
bridge-10-act-mV	7200.07	45856.40	–	7200.09	46763.80	–	–
bridge-10-cont	0.13	1.00	0.00	0.13	1.00	0.00	S
bridge-10-cont-act	0.44	1.00	0.00	0.45	1.00	0.00	S
bridge-10-mV	7200.06	61355.40	214.52	7200.05	65443.40	464.62	S
bridge-10-act-2s	7201.06	4227.00	–	7200.53	4778.60	–	–
bridge-10-act-2s-int	7200.26	12056.25	–	7200.30	14025.20	–	–
bridge-10-act-2s-mV	7200.14	26216.50	–	7200.07	31199.20	–	–
bridge-10-2s-mV	7200.12	34258.25	–	7200.08	35915.00	–	–
bridge-10-2s-int	7200.07	46690.50	4017.47	7200.07	46018.40	5709.04	S
bridge-10-2s-mV-int	1354.79	10334.00	0.00	2482.44	20420.00	0.00	S
bridge-10-cont-2s	0.20	1.00	0.00	0.20	1.00	0.00	S
bridge-10-cont-act-2s	6.04	6.00	0.00	5.40	6.00	0.00	nP
bridge-10-act-2s-mV-int	7200.19	14099.25	–	7200.28	12287.80	–	–
bridge-big	7200.83	4854.20	–	3060.06	2594.00	0.00	nP
bridge-big-act	7203.62	875.20	–	7205.63	879.00	–	–
bridge-big-act-int	7201.35	3387.20	–	7201.32	3017.20	–	–
bridge-big-2s	7201.16	3221.20	–	7201.06	3349.20	–	–
bridge-big-int	7200.29	14332.60	–	7200.37	15417.40	–	–
bridge-big-cont	0.51	1.00	0.00	0.49	1.00	0.00	nP
bridge-big-cont-act	36.45	12.00	0.00	7200.16	322.00	–	S
bridge-big-act-2s	7208.83	445.80	–	7206.69	496.60	–	–
bridge-big-act-2s-int	7202.74	1330.20	–	7202.02	1621.60	–	–
bridge-big-act-2s-mV	7202.71	1026.00	–	7205.68	881.40	–	–
bridge-big-2s-mV	1.69	1.00	–	1.58	1.00	–	–
bridge-big-2s-int	7200.36	8132.20	–	7200.49	8633.60	–	–
bridge-big-2s-mV-int	0.57	1.00	–	0.56	1.00	–	–
bridge-big-cont-2s	0.89	1.00	0.00	1.02	1.00	0.00	S
bridge-big-cont-act-2s	7199.61	2817.20	0.00	7200.27	322.00	144.40	S
bridge-big-act-2s-mV-int	7201.87	2542.60	–	9.92	1.00	–	–
canti-1	29.96	892.00	0.00	37.07	892.00	0.00	S
canti-1-act	174.47	1153.00	0.00	171.88	1148.83	0.00	nP
canti-1-2s	50.89	606.00	0.00	197.37	2532.00	0.00	S
canti-1-cont	0.08	1.00	0.00	0.08	1.00	0.00	S
canti-1-cont-act	0.98	5.00	0.00	0.97	5.00	0.00	nP
canti-1-mV	3.13	82.00	0.00	2.83	82.00	0.00	nP
canti-1-act-2s	181.29	655.00	0.00	736.28	2699.00	0.00	S
canti-1-act-2s-mV	0.30	1.00	–	0.29	1.00	–	–
canti-1-2s-mV	0.09	1.00	–	0.08	1.00	–	–

	solving as MISDP			without presolve			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
canti-1-cont-2s	0.13	1.00	0.00	0.12	1.00	0.00	nP
canti-1-cont-act-2s	0.40	1.00	0.00	11.53	75.00	0.00	S
canti-1-act-mV	683.28	5774.00	0.00	645.60	5775.00	0.00	nP
canti-1-m-mV	4.57	123.00	0.00	4.32	123.00	0.00	nP
canti-1-m-act-2s-mV	0.34	1.00	–	0.28	1.00	–	–
canti-1-m-2s-mV	0.09	1.00	–	0.08	1.00	–	–
canti-2	63.44	777.00	0.00	61.42	790.33	0.00	nP
canti-2-act	188.59	1629.00	0.00	184.74	1625.67	0.00	nP
canti-2-act-int	7200.03	85305.20	1720.99	7200.04	92124.00	1722.70	S
canti-2-2s	690.41	4849.00	0.00	2319.10	18449.00	0.00	S
canti-2-int	15.12	352.00	0.00	12.60	352.00	0.00	nP
canti-2-mV-int	3.18	75.00	0.00	2.15	75.00	0.00	nP
canti-2-act-mV-int	695.43	4529.00	0.00	531.32	4529.00	0.00	nP
canti-2-cont	0.08	1.00	0.00	0.08	1.00	0.00	S
canti-2-cont-act	1.00	5.00	0.00	0.97	5.00	0.00	nP
canti-2-mV	28.54	455.00	0.00	26.98	455.00	0.00	nP
canti-2-act-2s	2003.68	5111.00	0.00	6135.15	18237.00	0.00	S
canti-2-act-2s-int	7200.12	42579.50	–	7200.14	48304.00	–	–
canti-2-act-2s-mV	4139.22	12036.00	0.00	3948.61	12075.00	0.00	nP
canti-2-2s-mV	0.16	1.00	–	0.14	1.00	–	–
canti-2-2s-int	887.02	14875.00	0.00	1354.56	22993.00	0.00	S
canti-2-2s-mV-int	0.09	1.00	–	0.09	1.00	–	–
canti-2-cont-2s	0.12	1.00	0.00	0.12	1.00	0.00	S
canti-2-cont-act-2s	0.38	1.00	0.00	11.28	75.00	0.00	S
canti-2-act-2s-mV-int	3380.69	9051.00	0.00	3209.68	9070.00	0.00	nP
canti-2-m-mV-int	3.23	74.00	0.00	2.19	74.00	0.00	nP
canti-2-m-act-mV-int	7200.10	59788.60	3.67	6386.83	64218.00	0.00	nP
canti-2-m-act-mV	612.03	3727.00	0.00	585.93	3759.00	0.00	nP
canti-2-m-mV	8.88	118.00	0.00	9.20	129.00	0.00	S
canti-2-m-act-2s-mV	7200.14	21004.20	–	7200.20	22666.00	–	–
canti-2-m-2s-mV	67.95	516.00	0.00	63.72	499.00	0.00	nP
canti-2-m-act-2s-mV-int	3585.56	11463.00	0.00	3340.71	11452.00	0.00	nP
canti-3	143.96	1371.00	0.00	145.60	1371.83	0.00	S
canti-3-act	265.58	993.00	0.00	255.31	989.67	0.00	nP
canti-3-act-int	7200.04	61270.60	886.71	7200.07	70608.60	876.81	nP
canti-3-2s	536.50	3427.00	0.00	1427.82	8997.00	0.00	S
canti-3-int	13.68	319.00	0.00	13.62	319.00	0.00	nP
canti-3-mV-int	7.84	194.00	0.00	5.80	196.00	0.00	nP
canti-3-act-mV-int	7200.10	46609.80	–	7200.11	58306.40	–	–
canti-3-act-mV	7200.04	53503.00	5.56	7200.11	54234.80	6.25	S
canti-3-cont	0.08	1.00	0.00	0.09	1.00	0.00	S
canti-3-cont-act	1.00	5.00	0.00	0.98	5.00	0.00	nP
canti-3-mV	57.12	781.00	0.00	55.60	781.00	0.00	nP

	solving as MISDP			without presolve			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
canti-3-act-2s	1584.97	4274.00	0.00	3476.21	9302.00	0.00	S
canti-3-act-2s-int	7200.13	41022.25	–	7200.14	36922.40	–	–
canti-3-2s-int	946.47	11425.00	0.00	1396.01	17085.00	0.00	S
canti-3-2s-mV-int	0.09	1.00	–	0.08	1.00	–	–
canti-3-cont-2s	0.13	1.00	0.00	0.12	1.00	0.00	nP
canti-3-cont-act-2s	1.14	2.00	0.00	8.46	52.00	0.00	S
canti-3-act-2s-mV-int	0.27	1.00	–	0.27	1.00	–	–
canti-3-m-mV-int	6.90	184.00	0.00	4.86	186.00	0.00	nP
canti-3-m-act-mV-int	3387.09	22634.00	0.00	2634.17	22639.00	0.00	nP
canti-3-m-mV	10.82	154.00	0.00	10.53	154.00	0.00	nP
canti-3-m-act-2s-mV	6868.89	20610.00	0.00	6461.69	20505.00	0.00	nP
canti-3-m-2s-mV	39.82	289.00	0.00	38.88	289.00	0.00	nP
canti-3-m-act-2s-mV-int	7200.18	21778.20	0.46	7200.19	23202.20	0.42	nP
canti-4	29.82	892.00	0.00	37.22	892.00	0.00	S
canti-4-act	89.06	516.00	0.00	91.12	536.83	0.00	S
canti-4-2s	51.19	606.00	0.00	195.65	2532.00	0.00	S
canti-4-act-mV	351.45	2803.00	0.00	352.79	2809.00	0.00	S
canti-4-cont	0.08	1.00	0.00	0.08	1.00	0.00	S
canti-4-cont-act	0.19	1.00	0.00	0.22	1.00	0.00	S
canti-4-mV	13.18	349.00	0.00	11.93	349.00	0.00	nP
canti-4-act-2s	185.90	634.00	0.00	759.15	2782.00	0.00	S
canti-4-cont-2s	0.13	1.00	0.00	0.12	1.00	0.00	nP
canti-4-cont-act-2s	0.41	1.00	0.00	0.42	1.00	0.00	S
canti-4-m-act-mV	1141.98	10544.00	0.00	1111.94	10526.00	0.00	nP
canti-4-m-mV	4.51	123.00	0.00	4.34	123.00	0.00	nP
canti-4-m-act-2s-mV	0.31	1.00	–	0.28	1.00	–	–
canti-4-m-2s-mV	0.09	1.00	–	0.08	1.00	–	–
canti-5	62.88	777.00	0.00	61.94	790.33	0.00	nP
canti-5-act	194.91	1655.00	0.00	192.06	1630.83	0.00	nP
canti-5-act-int	7200.03	95888.20	–	7200.05	100921.60	–	–
canti-5-2s	689.91	4849.00	0.00	2319.00	18449.00	0.00	S
canti-5-int	15.15	352.00	0.00	12.60	352.00	0.00	nP
canti-5-mV-int	2.73	62.00	0.00	1.88	62.00	0.00	nP
canti-5-act-mV-int	7200.03	46931.00	–	7200.04	59216.60	–	–
canti-5-cont	0.08	1.00	0.00	0.09	1.00	0.00	S
canti-5-cont-act	0.21	1.00	0.00	0.22	1.00	0.00	S
canti-5-mV	8.48	122.00	0.00	8.34	122.00	0.00	nP
canti-5-act-2s	1639.65	4260.00	0.00	5791.65	17287.00	0.00	S
canti-5-act-2s-int	7200.04	44463.00	–	7200.06	46312.00	–	–
canti-5-act-2s-mV	6707.97	21351.00	0.00	6246.36	20769.00	0.00	nP
canti-5-2s-mV	0.17	1.00	–	0.15	1.00	–	–
canti-5-2s-int	890.64	14875.00	0.00	1356.51	22993.00	0.00	S
canti-5-2s-mV-int	0.09	1.00	–	0.09	1.00	–	–

	solving as MISDP			without presolve			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
canti-5-cont-2s	0.13	1.00	0.00	0.13	1.00	0.00	S
canti-5-cont-act-2s	0.40	1.00	0.00	0.44	1.00	0.00	S
canti-5-act-2s-mV-int	7200.21	20236.00	–	7200.13	21001.40	–	–
canti-5-m-mV-int	3.22	74.00	0.00	2.18	74.00	0.00	nP
canti-5-m-act-mV-int	1404.58	13033.00	0.00	1043.67	12991.00	0.00	nP
canti-5-m-act-mV	904.02	5681.00	0.00	867.28	5720.00	0.00	nP
canti-5-m-mV	8.82	118.00	0.00	9.19	129.00	0.00	S
canti-5-m-act-2s-mV	7200.09	21549.20	–	7200.13	22694.20	–	–
canti-5-m-2s-mV	67.25	516.00	0.00	63.90	499.00	0.00	nP
canti-5-m-act-2s-mV-int	7200.11	23141.60	2.57	7200.18	24371.60	2.53	nP
canti-6	144.11	1371.00	0.00	143.53	1371.83	0.00	nP
canti-6-act	220.64	795.00	0.00	216.61	804.17	0.00	nP
canti-6-act-int	7200.04	59571.60	–	7200.06	61377.20	–	–
canti-6-2s	533.75	3427.00	0.00	1434.65	8997.00	0.00	S
canti-6-int	13.72	319.00	0.00	13.47	319.00	0.00	nP
canti-6-mV-int	2.11	48.00	0.00	1.60	53.00	0.00	nP
canti-6-act-mV-int	4101.91	26448.00	0.00	3193.40	26465.00	0.00	nP
canti-6-act-mV	7200.04	46444.60	9.13	7200.05	48480.80	8.99	nP
canti-6-cont	0.08	1.00	0.00	0.09	1.00	0.00	S
canti-6-cont-act	0.20	1.00	0.00	0.19	1.00	0.00	nP
canti-6-mV	103.50	1547.00	0.00	98.50	1547.00	0.00	nP
canti-6-act-2s	1534.15	4041.00	0.00	3281.22	8535.00	0.00	S
canti-6-act-2s-int	7200.06	38074.00	–	7200.08	41631.80	–	–
canti-6-act-2s-mV	0.44	1.00	–	0.41	1.00	–	–
canti-6-2s-mV	0.16	1.00	–	0.15	1.00	–	–
canti-6-2s-int	941.43	11425.00	0.00	1400.49	17085.00	0.00	S
canti-6-2s-mV-int	0.09	1.00	–	0.09	1.00	–	–
canti-6-cont-2s	0.13	1.00	0.00	0.12	1.00	0.00	nP
canti-6-cont-act-2s	0.39	1.00	0.00	0.44	1.00	0.00	S
canti-6-act-2s-mV-int	0.31	1.00	–	0.30	1.00	–	–
canti-6-m-mV-int	6.89	184.00	0.00	4.84	186.00	0.00	nP
canti-6-m-act-mV-int	5572.80	38106.00	0.00	4350.62	38116.00	0.00	nP
canti-6-m-mV	10.87	154.00	0.00	10.43	154.00	0.00	nP
canti-6-m-act-2s-mV	7200.21	21112.40	–	7200.20	22130.60	–	–
canti-6-m-2s-mV	39.67	289.00	0.00	38.89	289.00	0.00	nP
canti-6-m-act-2s-mV-int	6241.48	18606.00	0.00	5791.95	18619.00	0.00	nP
canti-7	647.95	1692.00	0.00	567.82	1588.00	0.00	nP
canti-7-act	884.80	737.00	0.00	829.00	737.00	0.00	nP
canti-7-act-int	7200.17	20471.40	–	7200.13	22616.80	–	–
canti-7-2s	7200.48	10025.71	–	7200.47	10889.20	353.73	nP
canti-7-int	40.00	347.00	0.00	40.44	347.00	0.00	S
canti-7-mV-int	38.74	546.00	0.00	45.11	770.00	0.00	S
canti-7-act-mV-int	2804.67	9703.00	0.00	3204.69	12649.00	0.00	S

	solving as MISDP			without presolve			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
canti-7-act-mV	7200.44	9545.20	–	7200.40	11380.40	–	–
canti-7-cont	0.17	1.00	0.00	0.16	1.00	0.00	nP
canti-7-cont-act	2.01	3.00	0.00	0.49	1.00	0.00	nP
canti-7-mV	7200.07	32494.40	32.44	7200.11	35430.00	–	S
canti-7-act-2s	7201.23	2761.00	–	7201.08	3140.60	–	–
canti-7-act-2s-int	7200.37	13058.57	–	7200.54	10813.40	–	–
canti-7-act-2s-mV	7200.79	4998.57	–	7200.57	6111.00	–	–
canti-7-2s-mV	7200.17	19372.14	–	7200.15	22790.20	–	–
canti-7-2s-int	1283.04	5616.00	0.00	1489.67	6572.00	0.00	S
canti-7-2s-mV-int	45.25	310.00	0.00	45.68	329.00	0.00	S
canti-7-cont-2s	0.25	1.00	0.00	0.25	1.00	0.00	S
canti-7-cont-act-2s	6.90	5.00	0.00	4.56	3.00	0.00	nP
canti-7-act-2s-mV-int	7200.26	8418.71	–	7200.52	8735.60	–	–
canti-9	288.71	700.00	0.00	284.46	705.00	0.00	nP
canti-9-act	7200.80	4865.00	–	1954.24	1591.00	0.00	nP
canti-9-act-int	7200.22	20785.60	–	7200.19	21381.20	–	–
canti-9-2s	7200.46	8407.86	554.10	7200.37	8812.60	556.25	S
canti-9-int	1755.88	12880.00	0.00	2438.23	15497.00	0.00	S
canti-9-mV-int	100.71	1910.00	0.00	103.98	1956.00	0.00	S
canti-9-act-mV-int	7200.11	38860.40	–	7200.06	59012.40	–	–
canti-9-act-mV	7200.38	8819.60	–	7200.18	9759.20	–	–
canti-9-cont	0.17	1.00	0.00	0.17	1.00	0.00	S
canti-9-cont-act	0.66	1.00	0.00	0.59	1.00	0.00	nP
canti-9-mV	5070.35	26591.00	0.00	4937.57	28706.00	0.00	nP
canti-9-act-2s	7201.56	2214.14	–	7201.84	2275.40	–	–
canti-9-act-2s-int	7200.24	8919.71	–	7200.25	8657.40	–	–
canti-9-act-2s-mV	7200.83	3864.71	–	7200.35	4233.60	–	–
canti-9-2s-mV	7200.31	15661.86	–	7200.21	17024.60	–	–
canti-9-2s-int	7200.14	27338.71	–	7200.08	27293.40	–	–
canti-9-2s-mV-int	85.62	493.00	0.00	81.47	470.00	0.00	nP
canti-9-cont-2s	0.32	1.00	0.00	0.31	1.00	0.00	nP
canti-9-cont-act-2s	1.35	1.00	0.00	1.30	1.00	0.00	nP
canti-9-act-2s-mV-int	7200.58	7511.14	–	7200.34	7673.60	–	–
canti-10	986.13	2712.00	0.00	326.64	1284.00	0.00	nP
canti-10-act	2663.99	3100.00	0.00	769.91	1279.00	0.00	nP
canti-10-act-int	7200.22	24836.40	–	7200.14	27851.00	–	–
canti-10-2s	7200.29	12623.29	421.05	7200.22	13859.40	918.64	S
canti-10-int	15.99	211.00	0.00	16.10	221.00	0.00	S
canti-10-mV-int	77.06	1192.00	0.00	109.21	2586.00	0.00	S
canti-10-act-mV-int	1504.91	6456.00	0.00	4014.74	27808.00	0.00	S
canti-10-act-mV	7200.10	39296.40	–	7200.04	50308.00	–	–
canti-10-cont	0.15	1.00	0.00	0.15	1.00	0.00	S
canti-10-cont-act	3.94	8.00	0.00	3.56	8.00	0.00	nP

	solving as MISDP			without presolve			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
canti-10-mV	7200.05	49142.80	–	7200.06	63367.80	–	–
canti-10-act-2s	7200.95	3942.43	475.68	7200.55	5004.00	426.62	nP
canti-10-act-2s-int	7200.14	24388.57	–	7200.12	30819.80	–	–
canti-10-act-2s-mV	7200.49	6194.29	–	7200.25	13571.40	–	–
canti-10-2s-mV	7200.11	25193.57	–	7200.16	31988.00	–	–
canti-10-2s-int	2702.72	15006.00	0.00	4423.74	23290.00	0.00	S
canti-10-2s-mV-int	15.82	119.00	0.00	15.35	121.00	0.00	nP
canti-10-cont-2s	0.22	1.00	0.00	0.24	1.00	0.00	S
canti-10-cont-act-2s	22.77	25.00	0.00	24.68	34.00	0.00	S
canti-10-act-2s-mV-int	7200.27	12884.29	–	7200.19	15382.80	–	–
canti-big	1253.32	1590.00	0.00	1284.83	1590.00	0.00	S
canti-big-act	5972.22	2225.00	0.00	5920.95	2133.00	0.00	nP
canti-big-int	155.05	807.00	0.00	154.87	797.00	0.00	nP
canti-big-mV-int	130.11	931.00	0.00	199.37	1676.00	0.00	S
canti-big-act-mV-int	7200.29	22467.20	54.56	7200.25	29387.80	64.95	S
canti-big-cont	0.32	1.00	0.00	0.32	1.00	0.00	S
canti-big-cont-act	4.25	2.00	0.00	1.43	1.00	0.00	nP
canti-big-2s-mV-int	28.39	105.00	0.00	36.87	145.00	0.00	S
canti-big-cont-2s	0.52	1.00	0.00	0.47	1.00	0.00	nP
canti-big-cont-act-2s	17.14	5.00	0.00	10.95	3.00	0.00	nP

Table A.3. – Comparing the solving process with and without the presolving ideas for pure MISDP branch-and-bound.

Heuristic

For 426 instances we tested our rounding heuristic. The results are shown in Table A.4. The table is of the same kind as we have seen before, three columns for MISDP without heuristic and with heuristic and a columns that tells us which method was the best for this instance. The last column shows which solving algorithm was better: using the heuristic (‘*S*’) or turning the heuristic off (‘*nH*’).

	solving as MISDP			without heuristic			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
bridge-1	5.04	79.00	0.00	7.00	132.00	0.00	S
bridge-1-act	20.72	130.00	0.00	20.33	130.00	0.00	nH
bridge-1-2s	232.17	2299.00	0.00	265.19	2786.00	0.00	S
bridge-1-act-mV	355.10	2576.00	0.00	2061.81	16147.00	0.00	S
bridge-1-cont	0.08	1.00	0.00	0.08	1.00	0.00	S
bridge-1-cont-act	0.82	3.00	0.00	0.82	3.00	0.00	S
bridge-1-mV	35.94	847.00	0.00	38.84	932.00	0.00	S
bridge-1-act-2s	797.30	2158.00	0.00	867.47	2469.00	0.00	S

	solving as MISDP			without heuristic			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
bridge-1-act-2s-mV	696.68	2436.00	0.00	3992.90	15466.00	0.00	S
bridge-1-2s-mV	288.33	3773.00	0.00	296.97	3939.00	0.00	S
bridge-1-cont-2s	0.18	1.00	0.00	0.18	1.00	0.00	S
bridge-1-cont-act-2s	10.63	26.00	0.00	10.20	26.00	0.00	nH
bridge-2	105.32	733.00	0.00	134.94	1158.00	0.00	S
bridge-2-act	284.27	1124.00	0.00	288.50	1177.00	0.00	S
bridge-2-act-int	7200.07	112487.40	4.53	7200.03	82811.60	–	S
bridge-2-2s	7200.08	43287.20	42.14	7200.10	39747.60	907.06	S
bridge-2-int	36.62	645.00	0.00	40.45	765.00	0.00	S
bridge-2-mV-int	17.06	302.00	0.00	18.35	327.00	0.00	S
bridge-2-act-mV-int	157.85	918.00	0.00	7200.08	43084.00	–	S
bridge-2-act-mV	7.55	29.00	0.00	1902.13	13018.00	0.00	S
bridge-2-cont	0.08	1.00	0.00	0.08	1.00	0.00	S
bridge-2-cont-act	0.83	3.00	0.00	0.81	3.00	0.00	nH
bridge-2-mV	69.18	1015.00	0.00	76.09	1096.00	0.00	S
bridge-2-act-2s	7200.16	13911.00	–	7200.28	14171.40	–	–
bridge-2-act-2s-int	7200.12	35691.00	105673.87	7200.12	34997.50	–	S
bridge-2-act-2s-mV	1437.06	3728.00	0.00	6336.03	24501.00	0.00	S
bridge-2-2s-mV	1349.46	13544.00	0.00	1334.01	13544.00	0.00	nH
bridge-2-2s-int	431.15	3964.00	0.00	434.50	4031.00	0.00	S
bridge-2-2s-mV-int	229.01	2254.00	0.00	259.32	2551.00	0.00	S
bridge-2-cont-2s	0.17	1.00	0.00	0.17	1.00	0.00	S
bridge-2-cont-act-2s	10.29	26.00	0.00	10.34	26.00	0.00	S
bridge-2-act-2s-mV-int	3348.08	9062.00	0.00	7200.24	19941.00	–	S
bridge-3	40.62	362.00	0.00	44.21	415.00	0.00	S
bridge-3-act	190.01	712.00	0.00	184.18	724.00	0.00	nH
bridge-3-act-int	951.00	10108.00	0.00	7200.06	71202.00	–	S
bridge-3-2s	1861.59	10533.00	0.00	2526.70	14388.00	0.00	S
bridge-3-int	68.91	1149.00	0.00	71.79	1220.00	0.00	S
bridge-3-mV-int	4.64	89.00	0.00	6.38	123.00	0.00	S
bridge-3-act-mV-int	263.65	1570.00	0.00	7200.09	43322.20	–	S
bridge-3-act-mV	53.05	247.00	0.00	7200.07	40837.00	13.74	S
bridge-3-cont	0.10	1.00	0.00	0.10	1.00	0.00	S
bridge-3-cont-act	0.86	3.00	0.00	0.87	3.00	0.00	S
bridge-3-mV	86.10	1014.00	0.00	90.90	1071.00	0.00	S
bridge-3-act-2s	5658.01	11733.00	0.00	6223.10	12837.00	0.00	S
bridge-3-act-2s-int	7200.14	33115.80	–	7200.13	33016.00	–	–
bridge-3-act-2s-mV	1859.06	5119.00	0.00	7200.13	20692.25	–	S
bridge-3-2s-mV	2174.15	16846.00	0.00	2343.01	18040.00	0.00	S
bridge-3-2s-int	295.00	2537.00	0.00	300.62	2634.00	0.00	S
bridge-3-2s-mV-int	40.35	390.00	0.00	47.68	464.00	0.00	S
bridge-3-cont-2s	0.17	1.00	0.00	0.17	1.00	0.00	S
bridge-3-cont-act-2s	5.09	11.00	0.00	5.24	11.00	0.00	S

	solving as MISDP			without heuristic			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
bridge-3-act-2s-mV-int	2027.43	5821.00	0.00	7200.16	19829.00	–	S
bridge-4	4.99	79.00	0.00	7.08	132.00	0.00	S
bridge-4-act	20.78	114.00	0.00	22.73	136.00	0.00	S
bridge-4-2s	228.23	2299.00	0.00	265.70	2786.00	0.00	S
bridge-4-act-mV	12.62	81.00	0.00	1994.79	15646.00	0.00	S
bridge-4-cont	0.08	1.00	0.00	0.08	1.00	0.00	S
bridge-4-cont-act	0.74	3.00	0.00	0.75	3.00	0.00	S
bridge-4-mV	24.05	555.00	0.00	26.51	618.00	0.00	S
bridge-4-act-2s	835.05	2563.00	0.00	852.55	2563.00	0.00	S
bridge-4-act-2s-mV	840.77	3072.00	0.00	3808.52	14780.00	0.00	S
bridge-4-2s-mV	198.07	2592.00	0.00	251.05	3289.00	0.00	S
bridge-4-cont-2s	0.17	1.00	0.00	0.17	1.00	0.00	S
bridge-4-cont-act-2s	9.49	21.00	0.00	9.84	21.00	0.00	S
bridge-5	102.84	733.00	0.00	133.85	1158.00	0.00	S
bridge-5-act	293.04	1179.00	0.00	292.69	1179.00	0.00	nH
bridge-5-act-int	7200.05	106126.60	30.48	7200.06	80355.20	–	S
bridge-5-2s	7200.07	43462.60	42.12	7200.13	39869.00	906.85	S
bridge-5-int	36.22	645.00	0.00	40.83	765.00	0.00	S
bridge-5-mV-int	16.81	302.00	0.00	18.37	327.00	0.00	S
bridge-5-act-mV-int	2042.50	13175.00	0.00	7200.18	43057.80	–	S
bridge-5-act-mV	7.04	27.00	0.00	2309.51	14623.00	0.00	S
bridge-5-cont	0.08	1.00	0.00	0.08	1.00	0.00	S
bridge-5-cont-act	0.72	3.00	0.00	0.75	3.00	0.00	S
bridge-5-mV	68.17	1015.00	0.00	75.54	1096.00	0.00	S
bridge-5-act-2s	7200.34	13934.60	–	7200.23	14070.80	–	–
bridge-5-act-2s-int	7200.16	33020.80	99341.99	7200.09	31797.50	–	S
bridge-5-act-2s-mV	1961.70	5686.00	0.00	7200.20	25463.00	1.05	S
bridge-5-2scen-mV	1346.56	13544.00	0.00	1337.17	13544.00	0.00	nH
bridge-5-2s-int	429.55	3964.00	0.00	433.91	4031.00	0.00	S
bridge-5-2s-mV-int	227.79	2254.00	0.00	259.10	2551.00	0.00	S
bridge-5-cont-2s	0.17	1.00	0.00	0.18	1.00	0.00	S
bridge-5-cont-act-2s	9.31	21.00	0.00	9.72	21.00	0.00	S
bridge-5-act-2s-mV-int	3601.90	10558.00	0.00	7200.26	19076.50	–	S
bridge-6	40.33	362.00	0.00	44.44	415.00	0.00	S
bridge-6-act	209.19	800.00	0.00	212.02	800.00	0.00	S
bridge-6-act-int	7200.02	104381.60	19.52	7200.06	66028.80	–	S
bridge-6-2s	1814.74	10533.00	0.00	2523.85	14388.00	0.00	S
bridge-6-int	68.87	1149.00	0.00	72.09	1220.00	0.00	S
bridge-6-mV-int	16.31	337.00	0.00	19.55	400.00	0.00	S
bridge-6-act-mV-int	1146.08	7105.00	0.00	7200.12	43208.80	–	S
bridge-6-act-mV	124.55	584.00	0.00	7200.06	40219.40	–	S
bridge-6-cont	0.10	1.00	0.00	0.10	1.00	0.00	S
bridge-6-cont-act	0.61	2.00	0.00	0.64	2.00	0.00	S

	solving as MISDP			without heuristic			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
bridge-6-mV	43.68	501.00	0.00	47.47	548.00	0.00	S
bridge-6-act-2s	6195.92	13265.00	0.00	6727.72	14479.00	0.00	S
bridge-6-act-2s-int	7200.12	30386.40	–	7200.03	29885.75	–	–
bridge-6-act-2s-mV	1721.03	4825.00	0.00	7200.18	20334.75	–	S
bridge-6-2s-mV	1323.15	10086.00	0.00	1453.11	10890.00	0.00	S
bridge-6-2s-int	289.64	2537.00	0.00	299.63	2634.00	0.00	S
bridge-6-2s-mV-int	97.66	1009.00	0.00	106.53	1092.00	0.00	S
bridge-6-cont-2s	0.17	1.00	0.00	0.17	1.00	0.00	S
bridge-6-cont-act-2s	3.45	7.00	0.00	3.51	7.00	0.00	S
bridge-6-act-2s-mV-int	1953.60	5455.00	0.00	7200.16	19462.00	–	S
bridge-7	1432.54	2504.00	0.00	1683.93	3520.00	0.00	S
bridge-7-act	5908.73	4164.00	0.00	5841.99	4455.00	0.00	nH
bridge-7-act-int	7200.02	33381.80	118.24	7200.24	30518.00	–	S
bridge-7-2s	7200.43	8118.20	–	7200.22	8013.20	–	–
bridge-7-int	12.44	118.00	0.00	12.82	135.00	0.00	S
bridge-7-mV-int	1.29	13.00	0.00	1.41	14.00	0.00	S
bridge-7-act-mV-int	255.21	780.00	0.00	843.02	4210.00	0.00	S
bridge-7-act-mV	311.16	260.00	0.00	312.72	267.00	0.00	S
bridge-7-cont	0.16	1.00	0.00	0.17	1.00	0.00	S
bridge-7-cont-act	8.33	15.00	0.00	8.41	15.00	0.00	S
bridge-7-mV	13.37	38.00	0.00	20.86	49.00	0.00	S
bridge-7-act-2s	7201.90	2465.60	–	7201.14	2335.00	–	–
bridge-7-act-2s-int	7200.22	17251.00	205.52	7200.19	16759.00	–	S
bridge-7-act-2s-mV	7201.31	3583.40	–	7200.61	3696.50	–	–
bridge-7-2s-mV	536.99	1358.00	0.00	646.36	1484.00	0.00	S
bridge-7-2s-int	315.31	1156.00	0.00	329.92	1228.00	0.00	S
bridge-7-2s-mV-int	69.62	414.00	0.00	86.89	486.00	0.00	S
bridge-7-cont-2s	0.33	1.00	0.00	0.33	1.00	0.00	S
bridge-7-cont-act-2s	31.18	25.00	0.00	30.27	25.00	0.00	nH
bridge-7-act-2s-mV-int	1176.80	1246.00	0.00	7200.16	17316.00	–	S
bridge-8	7201.01	4361.40	–	7200.47	4664.60	–	–
bridge-8-act	7201.87	1313.80	–	7202.03	1411.20	–	–
bridge-8-act-int	7200.48	8372.00	–	7200.54	8143.80	–	–
bridge-8-2s	7201.54	2999.20	–	7200.94	3061.40	–	–
bridge-8-int	1568.68	4834.00	0.00	1636.89	5354.00	0.00	S
bridge-8-mV-int	64.08	324.00	0.00	70.72	365.00	0.00	S
bridge-8-act-mV-int	1840.51	1815.00	0.00	7200.15	8002.60	–	S
bridge-8-act-mV	7201.33	2186.00	–	7200.96	2263.60	–	–
bridge-8-cont	0.30	1.00	0.00	0.29	1.00	0.00	nH
bridge-8-cont-act	15.20	11.00	0.00	14.39	11.00	0.00	nH
bridge-8-mV	2010.62	2837.00	0.00	2217.31	3085.00	0.00	S
bridge-8-act-2s	7205.50	720.60	–	7208.01	691.60	–	–
bridge-8-act-2s-int	7200.95	4410.60	–	7201.26	4034.00	–	–

	solving as MISDP			without heuristic			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
bridge-8-act-2s-mV	7203.99	1125.20	–	7203.12	1132.75	–	–
bridge-8-2s-mV	7200.36	7087.20	4.90	7200.68	5928.25	–	S
bridge-8-2s-int	2889.06	5038.00	0.00	3106.06	5509.00	0.00	S
bridge-8-2s-mV-int	115.97	328.00	0.00	129.16	370.00	0.00	S
bridge-8-cont-2s	0.51	1.00	0.00	0.52	1.00	0.00	S
bridge-8-cont-act-2s	34.76	11.00	0.00	36.33	11.00	0.00	S
bridge-8-act-2s-mV-int	7201.32	3035.00	0.94	7201.45	2974.75	–	S
bridge-9	66.65	323.00	0.00	76.64	435.00	0.00	S
bridge-9-act	388.64	1070.00	0.00	363.05	1001.00	0.00	nH
bridge-9-act-int	5125.62	59718.00	0.00	7200.05	65900.20	–	S
bridge-9-2s	245.43	652.00	0.00	378.53	1249.00	0.00	S
bridge-9-int	57.17	1088.00	0.00	69.09	1206.00	0.00	S
bridge-9-mV-int	16.94	545.00	0.00	16.66	545.00	0.00	nH
bridge-9-act-mV-int	255.01	1867.00	0.00	404.55	2907.00	0.00	S
bridge-9-act-mV	74.36	290.00	0.00	284.25	915.00	0.00	S
bridge-9-cont	0.10	1.00	0.00	0.10	1.00	0.00	S
bridge-9-cont-act	13.23	82.00	0.00	13.02	82.00	0.00	nH
bridge-9-mV	46.71	498.00	0.00	45.36	498.00	0.00	nH
bridge-9-act-2s	807.06	939.00	0.00	880.19	1180.00	0.00	S
bridge-9-act-2s-int	548.16	3220.00	0.00	7200.07	58127.25	–	S
bridge-9-act-2s-mV	63.80	147.00	0.00	254.04	425.00	0.00	S
bridge-9-2s-mV	36.98	255.00	0.00	37.05	255.00	0.00	S
bridge-9-2s-int	182.06	1579.00	0.00	464.83	5745.00	0.00	S
bridge-9-2s-mV-int	18.44	319.00	0.00	18.31	319.00	0.00	nH
bridge-9-cont-2s	0.18	1.00	0.00	0.17	1.00	0.00	nH
bridge-9-cont-act-2s	22.48	79.00	0.00	22.73	79.00	0.00	S
bridge-9-act-2s-mV-int	517.27	1495.00	0.00	514.44	1511.00	0.00	nH
bridge-10	1102.61	6642.00	0.00	1081.06	6742.00	0.00	nH
bridge-10-act	720.74	861.00	0.00	968.78	1791.00	0.00	S
bridge-10-act-int	7200.10	27943.00	49.87	7200.19	22660.60	–	S
bridge-10-2s	7189.18	18789.00	0.02	7200.18	18738.00	0.18	S
bridge-10-int	49.82	455.00	0.00	48.95	455.00	0.00	nH
bridge-10-mV-int	130.01	2929.00	0.00	131.47	2958.00	0.00	S
bridge-10-act-mV-int	531.97	1819.00	0.00	1169.86	5204.00	0.00	S
bridge-10-act-mV	7200.18	45895.80	51.60	7200.07	45856.40	–	S
bridge-10-cont	0.14	1.00	0.00	0.13	1.00	0.00	nH
bridge-10-cont-act	0.46	1.00	0.00	0.44	1.00	0.00	nH
bridge-10-mV	7200.03	81793.00	33.60	7200.06	61355.40	214.52	S
bridge-10-act-2s	7200.93	4566.00	203.30	7201.06	4227.00	–	S
bridge-10-act-2s-int	7200.10	12193.20	–	7200.26	12056.25	–	–
bridge-10-act-2s-mV	7200.14	25639.40	–	7200.14	26216.50	–	–
bridge-10-2s-mV	7200.07	34747.00	–	7200.12	34258.25	–	–
bridge-10-2s-int	7200.07	48638.80	16.51	7200.07	46690.50	4017.47	S

	solving as MISDP			without heuristic			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
bridge-10-2s-mV-int	1339.31	10334.00	0.00	1354.79	10334.00	0.00	S
bridge-10-cont-2s	0.19	1.00	0.00	0.20	1.00	0.00	S
bridge-10-cont-act-2s	6.12	6.00	0.00	6.04	6.00	0.00	nH
bridge-10-act-2s-mV-int	7200.34	12816.60	1892.98	7200.19	14099.25	–	S
bridge-big	1792.50	1228.00	0.00	7200.83	4854.20	–	S
bridge-big-act	7204.64	865.60	–	7203.62	875.20	–	–
bridge-big-act-int	7202.14	3016.60	–	7201.35	3387.20	–	–
bridge-big-2s	7201.16	3132.80	–	7201.16	3221.20	–	–
bridge-big-int	7200.22	14620.60	–	7200.29	14332.60	–	–
bridge-big-cont	0.54	1.00	0.00	0.51	1.00	0.00	nH
bridge-big-cont-act	40.67	12.00	0.00	36.45	12.00	0.00	nH
bridge-big-act-2s	7209.36	430.80	–	7208.83	445.80	–	–
bridge-big-act-2s-int	7203.39	1114.20	–	7202.74	1330.20	–	–
bridge-big-act-2s-mV	7203.51	841.80	–	7202.71	1026.00	–	–
bridge-big-2s-mV	1.62	1.00	–	1.69	1.00	–	–
bridge-big-2s-int	7200.50	8395.20	–	7200.36	8132.20	–	–
bridge-big-2s-mV-int	0.55	1.00	–	0.57	1.00	–	–
bridge-big-cont-2s	0.89	1.00	0.00	0.89	1.00	0.00	S
bridge-big-cont-act-2s	7201.57	2741.80	0.01	7199.61	2817.20	0.00	nH
bridge-big-act-2s-mV-int	7202.58	2581.20	–	7201.87	2542.60	–	–
canti-1	23.01	620.00	0.00	29.96	892.00	0.00	S
canti-1-act	106.06	723.00	0.00	174.47	1153.00	0.00	S
canti-1-2s	43.88	505.00	0.00	50.89	606.00	0.00	S
canti-1-act-mV	3.34	23.00	0.00	7200.06	76117.00	6.28	S
canti-1-cont	0.08	1.00	0.00	0.08	1.00	0.00	S
canti-1-cont-act	0.99	5.00	0.00	0.98	5.00	0.00	nH
canti-1-mV	2.37	62.00	0.00	3.13	82.00	0.00	S
canti-1-act-2s	177.99	655.00	0.00	181.29	655.00	0.00	S
canti-1-cont-2s	0.13	1.00	0.00	0.13	1.00	0.00	S
canti-1-cont-act-2s	0.36	1.00	0.00	0.40	1.00	0.00	S
canti-1-m-act-mV	9.57	73.00	0.00	683.28	5774.00	0.00	S
canti-1-m-mV	3.62	103.00	0.00	4.57	123.00	0.00	S
canti-1-m-act-2s-mV	0.30	1.00	–	0.34	1.00	–	–
canti-1-m-2s-mV	0.08	1.00	–	0.09	1.00	–	–
canti-2	60.07	698.00	0.00	63.44	777.00	0.00	S
canti-2-act	182.73	1629.00	0.00	188.59	1629.00	0.00	S
canti-2-act-int	185.98	2695.00	0.00	7200.03	85305.20	1720.99	S
canti-2-2s	669.06	4754.00	0.00	690.41	4849.00	0.00	S
canti-2-int	14.79	339.00	0.00	15.12	352.00	0.00	S
canti-2-mV-int	2.39	58.00	0.00	3.18	75.00	0.00	S
canti-2-act-mV-int	18.54	130.00	0.00	695.43	4529.00	0.00	S
canti-2-act-mV	9.33	47.00	0.00	1044.02	7226.00	0.00	S
canti-2-cont	0.07	1.00	0.00	0.08	1.00	0.00	S

	solving as MISDP			without heuristic			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
canti-2-cont-act	0.95	5.00	0.00	1.00	5.00	0.00	S
canti-2-mV	19.45	343.00	0.00	28.54	455.00	0.00	S
canti-2-act-2s	1906.94	5111.00	0.00	2003.68	5111.00	0.00	S
canti-2-act-2s-int	7200.12	41560.60	1315.22	7200.12	42579.50	–	S
canti-2-act-2s-mV	1388.02	4528.00	0.00	4139.22	12036.00	0.00	S
canti-2-2s-mV	0.15	1.00	–	0.16	1.00	–	–
canti-2-2s-int	816.04	14020.00	0.00	887.02	14875.00	0.00	S
canti-2-2s-mV-int	0.08	1.00	–	0.09	1.00	–	–
canti-2-cont-2s	0.12	1.00	0.00	0.12	1.00	0.00	S
canti-2-cont-act-2s	0.37	1.00	0.00	0.38	1.00	0.00	S
canti-2-act-2s-mV-int	14.90	37.00	0.00	3380.69	9051.00	0.00	S
canti-2-m-mV-int	1.93	54.00	0.00	3.23	74.00	0.00	S
canti-2-m-act-mV-int	28.37	221.00	0.00	7200.10	59788.60	3.67	S
canti-2-m-act-mV	28.81	166.00	0.00	612.03	3727.00	0.00	S
canti-2-m-mV	6.16	102.00	0.00	8.88	118.00	0.00	S
canti-2-m-act-2s-mV	333.06	923.00	0.00	7200.14	21004.20	–	S
canti-2-m-2s-mV	49.13	385.00	0.00	67.95	516.00	0.00	S
canti-2-m-act-2s-mV-int	166.71	573.00	0.00	3585.56	11463.00	0.00	S
canti-3	124.15	1121.00	0.00	143.96	1371.00	0.00	S
canti-3-act	256.42	993.00	0.00	265.58	993.00	0.00	S
canti-3-act-int	242.99	1675.00	0.00	7200.04	61270.60	886.71	S
canti-3-2s	455.97	2831.00	0.00	536.50	3427.00	0.00	S
canti-3-int	11.14	207.00	0.00	13.68	319.00	0.00	S
canti-3-mV-int	1.57	34.00	0.00	7.84	194.00	0.00	S
canti-3-act-mV-int	23.18	159.00	0.00	7200.10	46609.80	–	S
canti-3-act-mV	7.27	32.00	0.00	7200.04	53503.00	5.56	S
canti-3-cont	0.08	1.00	0.00	0.08	1.00	0.00	S
canti-3-cont-act	0.95	5.00	0.00	1.00	5.00	0.00	S
canti-3-mV	40.15	569.00	0.00	57.12	781.00	0.00	S
canti-3-act-2s	1505.96	4274.00	0.00	1584.97	4274.00	0.00	S
canti-3-act-2s-int	7200.08	45527.40	49.39	7200.13	41022.25	–	S
canti-3-2s-int	888.54	10700.00	0.00	946.47	11425.00	0.00	S
canti-3-2s-mV-int	0.09	1.00	–	0.09	1.00	–	–
canti-3-cont-2s	0.13	1.00	0.00	0.13	1.00	0.00	S
canti-3-cont-act-2s	1.06	2.00	0.00	1.14	2.00	0.00	S
canti-3-act-2s-mV-int	0.25	1.00	–	0.27	1.00	–	–
canti-3-m-mV-int	3.65	97.00	0.00	6.90	184.00	0.00	S
canti-3-m-act-mV-int	19.49	145.00	0.00	3387.09	22634.00	0.00	S
canti-3-m-act-mV	9.64	52.00	0.00	379.51	3024.00	0.00	S
canti-3-m-mV	8.30	120.00	0.00	10.82	154.00	0.00	S
canti-3-m-act-2s-mV	49.27	121.00	0.00	6868.89	20610.00	0.00	S
canti-3-m-2s-mV	29.16	216.00	0.00	39.82	289.00	0.00	S
canti-3-m-act-2s-mV-int	45.01	149.00	0.00	7200.18	21778.20	0.46	S

	solving as MISDP			without heuristic			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
canti-4	22.82	620.00	0.00	29.82	892.00	0.00	S
canti-4-act	58.65	360.00	0.00	89.06	516.00	0.00	S
canti-4-2s	43.86	505.00	0.00	51.19	606.00	0.00	S
canti-4-act-mV	11.35	86.00	0.00	351.45	2803.00	0.00	S
canti-4-cont	0.07	1.00	0.00	0.08	1.00	0.00	S
canti-4-cont-act	0.19	1.00	0.00	0.19	1.00	0.00	S
canti-4-mV	10.10	267.00	0.00	13.18	349.00	0.00	S
canti-4-act-2s	151.59	523.00	0.00	185.90	634.00	0.00	S
canti-4-cont-2s	0.12	1.00	0.00	0.13	1.00	0.00	S
canti-4-cont-act-2s	0.37	1.00	0.00	0.41	1.00	0.00	S
canti-4-m-act-mV	16.41	131.00	0.00	1141.98	10544.00	0.00	S
canti-4-m-mV	3.60	103.00	0.00	4.51	123.00	0.00	S
canti-5	60.15	698.00	0.00	62.88	777.00	0.00	S
canti-5-act	191.21	1655.00	0.00	194.91	1655.00	0.00	S
canti-5-act-int	1865.97	25536.00	0.00	7200.03	95888.20	–	S
canti-5-2s	666.57	4754.00	0.00	689.91	4849.00	0.00	S
canti-5-int	14.82	339.00	0.00	15.15	352.00	0.00	S
canti-5-mV-int	2.12	48.00	0.00	2.73	62.00	0.00	S
canti-5-act-mV-int	121.63	880.00	0.00	7200.03	46931.00	–	S
canti-5-act-mV	7.58	37.00	0.00	7200.07	60929.80	1.30	S
canti-5-cont	0.08	1.00	0.00	0.08	1.00	0.00	S
canti-5-cont-act	0.19	1.00	0.00	0.21	1.00	0.00	S
canti-5-mV	5.69	88.00	0.00	8.48	122.00	0.00	S
canti-5-act-2s	1527.60	4151.00	0.00	1639.65	4260.00	0.00	S
canti-5-act-2s-int	7200.05	44512.40	2434.12	7200.04	44463.00	–	S
canti-5-act-2s-mV	135.63	286.00	0.00	6707.97	21351.00	0.00	S
canti-5-2s-mV	0.16	1.00	–	0.17	1.00	–	–
canti-5-2s-int	816.87	14020.00	0.00	890.64	14875.00	0.00	S
canti-5-2s-mV-int	0.09	1.00	–	0.09	1.00	–	–
canti-5-cont-2s	0.13	1.00	0.00	0.13	1.00	0.00	S
canti-5-cont-act-2s	0.39	1.00	0.00	0.40	1.00	0.00	S
canti-5-act-2s-mV-int	7200.10	21183.20	6.41	7200.21	20236.00	–	S
canti-5-m-mV-int	1.94	54.00	0.00	3.22	74.00	0.00	S
canti-5-m-act-mV-int	35.03	287.00	0.00	1404.58	13033.00	0.00	S
canti-5-m-act-mV	24.46	196.00	0.00	904.02	5681.00	0.00	S
canti-5-m-mV	6.18	102.00	0.00	8.82	118.00	0.00	S
canti-5-m-act-2s-mV	192.23	528.00	0.00	7200.09	21549.20	–	S
canti-5-m-2s-mV	49.14	385.00	0.00	67.25	516.00	0.00	S
canti-5-m-act-2s-mV-int	243.41	815.00	0.00	7200.11	23141.60	2.57	S
canti-6	123.61	1121.00	0.00	144.11	1371.00	0.00	S
canti-6-act	212.51	781.00	0.00	220.64	795.00	0.00	S
canti-6-act-int	2343.84	24420.00	0.00	7200.04	59571.60	–	S
canti-6-2s	456.76	2831.00	0.00	533.75	3427.00	0.00	S

	solving as MISDP			without heuristic			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
canti-6-int	11.17	207.00	0.00	13.72	319.00	0.00	S
canti-6-mV-int	1.67	38.00	0.00	2.11	48.00	0.00	S
canti-6-act-mV-int	16.28	112.00	0.00	4101.91	26448.00	0.00	S
canti-6-act-mV	55.89	301.00	0.00	7200.04	46444.60	9.13	S
canti-6-cont	0.07	1.00	0.00	0.08	1.00	0.00	S
canti-6-cont-act	0.19	1.00	0.00	0.20	1.00	0.00	S
canti-6-mV	81.38	1293.00	0.00	103.50	1547.00	0.00	S
canti-6-act-2s	1073.52	2517.00	0.00	1534.15	4041.00	0.00	S
canti-6-act-2s-int	7200.09	42176.20	83.96	7200.06	38074.00	–	S
canti-6-2s-int	892.59	10700.00	0.00	941.43	11425.00	0.00	S
canti-6-cont-2s	0.12	1.00	0.00	0.13	1.00	0.00	S
canti-6-cont-act-2s	0.38	1.00	0.00	0.39	1.00	0.00	S
canti-6-m-mV-int	3.65	97.00	0.00	6.89	184.00	0.00	S
canti-6-m-act-mV-int	4.68	35.00	0.00	5572.80	38106.00	0.00	S
canti-6-m-act-mV	13.45	72.00	0.00	610.11	4824.00	0.00	S
canti-6-m-mV	8.31	120.00	0.00	10.87	154.00	0.00	S
canti-6-m-act-2s-mV	70.49	178.00	0.00	7200.21	21112.40	–	S
canti-6-m-2s-mV	29.23	216.00	0.00	39.67	289.00	0.00	S
canti-6-m-act-2s-mV-int	89.84	271.00	0.00	6241.48	18606.00	0.00	S
canti-7	605.74	1517.00	0.00	647.95	1692.00	0.00	S
canti-7-act	658.47	495.00	0.00	884.80	737.00	0.00	S
canti-7-act-int	1373.81	5119.00	0.00	7200.17	20471.40	–	S
canti-7-2s	7200.21	11006.60	3.58	7200.48	10025.71	–	S
canti-7-int	32.57	219.00	0.00	40.00	347.00	0.00	S
canti-7-mV-int	32.86	500.00	0.00	38.74	546.00	0.00	S
canti-7-act-mV-int	882.18	2403.00	0.00	2804.67	9703.00	0.00	S
canti-7-act-mV	7200.46	10477.60	–	7200.44	9545.20	–	–
canti-7-cont	0.16	1.00	0.00	0.17	1.00	0.00	S
canti-7-cont-act	1.83	3.00	0.00	2.01	3.00	0.00	S
canti-7-mV	3457.21	37477.00	0.00	7200.07	32494.40	32.44	S
canti-7-act-2s	7201.61	3137.80	–	7201.23	2761.00	–	–
canti-7-act-2s-int	7200.35	13857.20	–	7200.37	13058.57	–	–
canti-7-act-2s-mV	7200.53	5361.40	–	7200.79	4998.57	–	–
canti-7-2s-mV	7200.12	23404.40	24.81	7200.17	19372.14	–	S
canti-7-2s-int	1240.38	5460.00	0.00	1283.04	5616.00	0.00	S
canti-7-2s-mV-int	9.02	93.00	0.00	45.25	310.00	0.00	S
canti-7-cont-2s	0.24	1.00	0.00	0.25	1.00	0.00	S
canti-7-cont-act-2s	6.01	5.00	0.00	6.90	5.00	0.00	S
canti-7-act-2s-mV-int	7200.26	8583.60	1921.40	7200.26	8418.71	–	S
canti-8	64.53	302.00	0.00	143.08	711.00	0.00	S
canti-8-act	143.33	293.00	0.00	273.87	661.00	0.00	S
canti-8-act-mV	22.83	91.00	0.00	52.57	157.00	0.00	S
canti-8-cont	0.07	1.00	0.00	0.06	1.00	0.00	nH

	solving as MISDP			without heuristic			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
canti-8-cont-act	0.91	3.00	0.00	0.89	3.00	0.00	nH
canti-8-mV	13.76	122.00	0.00	23.70	178.00	0.00	S
canti-9	254.01	481.00	0.00	288.71	700.00	0.00	S
canti-9-act	1703.01	1189.00	0.00	7200.80	4865.00	–	S
canti-9-act-int	7200.10	23209.80	363497.06	7200.22	20785.60	–	S
canti-9-2s	7200.44	8963.40	553.08	7200.46	8407.86	554.10	S
canti-9-int	1685.27	12650.00	0.00	1755.88	12880.00	0.00	S
canti-9-mV-int	99.65	1909.00	0.00	100.71	1910.00	0.00	S
canti-9-act-mV-int	7200.03	40585.40	128.56	7200.11	38860.40	–	S
canti-9-act-mV	7200.81	9526.60	–	7200.38	8819.60	–	–
canti-9-cont	0.18	1.00	0.00	0.17	1.00	0.00	nH
canti-9-cont-act	0.60	1.00	0.00	0.66	1.00	0.00	S
canti-9-mV	3367.77	19632.00	0.00	5070.35	26591.00	0.00	S
canti-9-act-2s	7201.73	2481.20	–	7201.56	2214.14	–	–
canti-9-act-2s-int	7200.51	9385.80	–	7200.24	8919.71	–	–
canti-9-act-2s-mV	7200.99	4291.60	–	7200.83	3864.71	–	–
canti-9-2s-mV	7200.16	17481.80	51.44	7200.31	15661.86	–	S
canti-9-2s-int	7200.13	27555.00	–	7200.14	27338.71	–	–
canti-9-2s-mV-int	54.41	355.00	0.00	85.62	493.00	0.00	S
canti-9-cont-2s	0.32	1.00	0.00	0.32	1.00	0.00	S
canti-9-cont-act-2s	1.19	1.00	0.00	1.35	1.00	0.00	S
canti-9-act-2s-mV-int	7200.33	8032.80	–	7200.58	7511.14	–	–
canti-10	192.92	409.00	0.00	986.13	2712.00	0.00	S
canti-10-act	888.75	1093.00	0.00	2663.99	3100.00	0.00	S
canti-10-act-int	939.82	4054.00	0.00	7200.22	24836.40	–	S
canti-10-2s	7200.21	12704.80	421.00	7200.29	12623.29	421.05	S
canti-10-int	6.41	46.00	0.00	15.99	211.00	0.00	S
canti-10-mV-int	51.73	823.00	0.00	77.06	1192.00	0.00	S
canti-10-act-mV-int	869.97	3356.00	0.00	1504.91	6456.00	0.00	S
canti-10-act-mV	7200.09	38623.20	–	7200.10	39296.40	–	–
canti-10-cont	0.14	1.00	0.00	0.15	1.00	0.00	S
canti-10-cont-act	3.71	8.00	0.00	3.94	8.00	0.00	S
canti-10-mV	2968.92	35012.00	0.00	7200.05	49142.80	–	S
canti-10-act-2s	7200.78	4654.60	475.07	7200.95	3942.43	475.68	S
canti-10-act-2s-int	7200.11	25469.00	144.82	7200.14	24388.57	–	S
canti-10-act-2s-mV	7200.60	6881.40	–	7200.49	6194.29	–	–
canti-10-2s-mV	7200.17	25858.20	–	7200.11	25193.57	–	–
canti-10-2s-int	2382.31	12917.00	0.00	2702.72	15006.00	0.00	S
canti-10-2s-mV-int	1.82	20.00	0.00	15.82	119.00	0.00	S
canti-10-cont-2s	0.21	1.00	0.00	0.22	1.00	0.00	S
canti-10-cont-act-2s	18.94	25.00	0.00	22.77	25.00	0.00	S
canti-10-act-2s-mV-int	7200.31	13706.60	2593.77	7200.27	12884.29	–	S
canti-big	1096.81	1273.00	0.00	1253.32	1590.00	0.00	S

	solving as MISDP			without heuristic			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
canti-big-act	5409.97	2225.00	0.00	5972.22	2225.00	0.00	S
canti-big-act-int	7200.36	11657.80	79.97	7200.65	10002.80	–	S
canti-big-2s	7201.09	3618.80	–	7200.99	3435.14	–	–
canti-big-int	118.57	432.00	0.00	155.05	807.00	0.00	S
canti-big-mV-int	80.54	693.00	0.00	130.11	931.00	0.00	S
canti-big-act-mV-int	387.04	511.00	0.00	7200.29	22467.20	54.56	S
canti-big-act-mV	7201.02	3169.60	–	7201.13	2818.20	–	–
canti-big-cont	0.31	1.00	0.00	0.32	1.00	0.00	S
canti-big-cont-act	4.00	2.00	0.00	4.25	2.00	0.00	S
canti-big-mV	7200.09	28768.40	3.82	7200.37	11109.00	–	S
canti-big-act-2s	7204.02	783.60	–	7202.90	805.14	–	–
canti-big-act-2s-int	7200.93	3276.80	–	7201.38	3157.00	–	–
canti-big-act-2s-mV	7201.39	1520.00	–	7204.57	1426.29	–	–
canti-big-2s-mV	7200.50	7355.60	–	7200.44	7074.29	–	–
canti-big-2s-int	7200.15	15132.20	–	7200.23	14895.71	–	–
canti-big-2s-mV-int	6.40	29.00	0.00	28.39	105.00	0.00	S
canti-big-cont-2s	0.51	1.00	0.00	0.52	1.00	0.00	S
canti-big-cont-act-2s	14.68	5.00	0.00	17.14	5.00	0.00	S
canti-big-act-2s-mV-int	7201.44	3427.60	–	7201.43	3224.86	–	–

Table A.4. – Comparing the solving process with and without the rounding heuristic for pure MISDP branch-and-bound.

A.1.2. A note on the solving behavior

Tables A.5 and A.6 give an idea of how good the SDP relaxations in the root node are. Additionally, they show the quality of the first solution and where the optimal solution is found. The last column presents how many percent of the nodes were processed after the optimal solution was found to prove optimality. We do not show all the results in one table because there are big differences between the truss formulation where the compliance in minimized and the model which minimizes the volume of the truss.

	root	first	best sol.	best sol.		nodes after
	solution	solution	found at	found at	overall	best solution
instance	gap [%]	gap [%]	depth	node	nodes	[%]
bridge-1	2.40	0.00	20	79	79	0.00
bridge-1-act	2.60	0.00	40	122	130	6.15
bridge-2	6.60	0.00	41	724	733	1.23
bridge-2-act	7.00	21.92	87	1064	1124	5.34
bridge-2-int	15.00	0.05	24	627	645	2.79
bridge-3	3.60	0.00	66	339	362	6.35
bridge-3-act	3.90	21.00	85	699	712	1.83

instance	root solution gap [%]	first solution gap [%]	best sol. found at depth	best sol. found at node	overall nodes	nodes after best solution [%]
bridge-3-act-int	13.10	105.36	28	10108	10108	0.00
bridge-3-int	11.10	60.91	40	1134	1149	1.31
bridge-4	2.40	0.00	20	79	79	0.00
bridge-4-act	3.00	0.00	28	113	114	0.88
bridge-5	6.60	0.00	41	724	733	1.23
bridge-5-act	7.10	0.00	86	1154	1179	2.12
bridge-5-int	15.00	0.05	24	627	645	2.79
bridge-6	3.60	0.00	66	339	362	6.35
bridge-6-act	4.10	37.89	84	763	800	4.63
bridge-6-int	11.10	60.91	40	1134	1149	1.31
bridge-7	3.90	0.00	106	2423	2504	3.23
bridge-7-act	3.40	0.27	156	4164	4164	0.00
bridge-7-int	5.00	12.99	29	117	118	0.85
bridge-8-int	11.10	0.00	70	4821	4834	0.27
bridge-9	5.90	> 10000	34	302	323	6.50
bridge-9-act	6.10	179.12	83	1070	1070	0.00
bridge-9-act-int	32.90	> 10000	77	59109	59718	1.02
bridge-9-int	16.70	> 10000	11	1064	1088	2.21
bridge-10	6.10	> 10000	72	6631	6642	0.17
bridge-10-act	3.90	9.04	84	858	861	0.35
bridge-10-int	20.00	0.00	10	276	455	39.34
bridge-big	2.50	0.00	83	1148	1228	6.51
canti-1	11.30	> 10000	25	619	620	0.16
canti-1-act	10.00	0.01	16	621	723	14.11
canti-2	15.20	114.47	48	698	698	0.00
canti-2-act	11.00	31.81	88	1615	1629	0.86
canti-2-act-int	38.70	> 10000	10	670	2695	75.14
canti-2-int	24.20	45.46	37	337	339	0.59
canti-3	7.00	> 10000	46	1121	1121	0.00
canti-3-act	11.10	33.34	54	962	993	3.12
canti-3-act-int	16.10	303.76	9	1453	1675	13.25
canti-3-int	5.70	0.00	30	207	207	0.00
canti-4	11.30	> 10000	25	619	620	0.16
canti-4-act	10.00	0.00	15	240	360	33.33
canti-5	15.20	114.47	48	698	698	0.00
canti-5-act	10.80	42.44	86	1018	1655	38.49
canti-5-act-int	56.00	3406.96	14	17220	25536	32.57
canti-5-int	24.20	45.46	37	337	339	0.59
canti-6	7.00	> 10000	46	1121	1121	0.00
canti-6-act	9.50	34.66	38	781	781	0.00
canti-6-act-int	23.90	21.47	32	24417	24420	0.01
canti-6-int	5.70	0.00	30	207	207	0.00

instance	root solution gap [%]	first solution gap [%]	best sol. found at depth	best sol. found at node	overall nodes	nodes after best solution [%]
canti-7	3.70	4.90	135	1511	1517	0.40
canti-7-act	0.70	160.45	27	480	495	3.03
canti-7-act-int	7.30	29.02	34	5119	5119	0.00
canti-7-int	4.40	0.00	48	219	219	0.00
canti-8	0.30	3.57	15	170	302	43.71
canti-8-act	0.00	0.00	15	175	293	40.27
canti-9	2.30	0.00	131	481	481	0.00
canti-9-act	5.30	462.93	176	1189	1189	0.00
canti-9-int	47.30	148.00	64	12649	12650	0.01
canti-10	0.80	> 10000	64	409	409	0.00
canti-10-act	1.50	13.12	208	1093	1093	0.00
canti-10-act-int	4.90	34.66	36	4054	4054	0.00
canti-10-int	2.40	0.00	13	46	46	0.00
canti-big	1.90	92.96	170	1273	1273	0.00
canti-big-act	2.30	91.83	323	2225	2225	0.00
canti-big-int	3.70	0.00	64	431	432	0.23
lit-as-1	11.00	0.00	10	83	85	2.35
lit-as-1-act	12.60	0.00	8	73	79	7.59
lit-as-5-big-act-int	5.60	0.00	35	2735	4463	38.72
lit-as-5-big-int	5.60	0.00	35	2735	4463	38.72
lit-as-6	10.20	0.00	12	1056	1983	46.75
lit-as-6-act	10.20	0.00	12	1056	1983	46.75
lit-s-1	12.00	2257.52	58	21579	21610	0.14

Table A.5. – Information about the quality of the first solution and the root relaxation of the truss instances, when solving using the MISDP branch-and-bound approach and minimizing the compliance.

instance	root solution gap [%]	first solution gap [%]	best sol. found at depth	best sol. found at node	overall nodes	nodes after best solution [%]
bridge-1-mV	9.90	263.37	6	15	847	98.23
bridge-2-mV	15.10	39.31	35	148	1015	85.42
bridge-3-mV	7.80	0.00	16	66	1014	93.49
bridge-4-mV	8.80	263.37	6	15	555	97.30
bridge-5-mV	15.10	39.31	35	148	1015	85.42
bridge-6-mV	6.80	9.76	15	59	501	88.22
bridge-7-mV	14.30	35.36	9	37	38	2.63
bridge-8-mV	12.00	29.22	54	1249	2837	55.97
bridge-9-mV	163.00	0.00	17	338	498	32.13
canti-1-mV	5.70	254.57	0	1	62	98.39

instance	root solution gap [%]	first solution gap [%]	best sol. found at depth	best sol. found at node	overall nodes	nodes after best solution [%]
canti-1-m-mV	17.30	494.90	5	44	103	57.28
canti-2-mV	18.00	0.00	2	7	343	97.96
canti-2-m-mV	24.00	0.00	0	1	102	99.02
canti-3-mV	9.60	12.49	17	206	569	63.80
canti-3-m-mV	19.70	21.24	5	55	120	54.17
canti-4-mV	10.40	206.81	10	86	267	67.79
canti-4-m-mV	17.30	494.90	5	44	103	57.28
canti-5-mV	12.00	0.00	3	8	88	90.91
canti-5-m-mV	24.00	0.00	0	1	102	99.02
canti-6-mV	10.60	9.53	39	346	1293	73.24
canti-6-m-mV	19.70	21.24	5	55	120	54.17
canti-7-mV	82.00	6.40	94	11127	37477	70.31
canti-8-mV	36.00	47.21	26	77	122	36.89
canti-9-mV	82.00	71.61	22	11096	19632	43.48
canti-10-mV	121.00	0.00	106	13567	35012	61.25
lit-as-1-mV	18.00	372.28	4	8	51	84.31
lit-as-5-mV	24.00	6482.68	16	296	3636	91.86
lit-as-6-mV	13.20	1226.66	19	2491	5558	55.18
lit-s-1-mV	18.00	2748.84	8	64	83	22.89
bridge-2-mV-int	5.00	943.69	14	52	302	82.78
bridge-3-mV-int	2.00	671.09	11	29	89	67.42
bridge-5-mV-int	5.00	943.69	14	52	302	82.78
bridge-6-mV-int	2.90	658.88	6	21	337	93.77
bridge-7-mV-int	3.10	7029.77	2	7	13	46.15
bridge-8-mV-int	12.80	8127.46	23	202	324	37.65
bridge-9-mV-int	229.10	6812.29	9	247	545	54.68
bridge-10-mV-int	308.20	> 10000	16	2929	2929	0.00
canti-2-mV-int	2.90	1299.50	5	12	58	79.31
canti-2-m-mV-int	9.00	2353.87	6	24	54	55.56
canti-3-mV-int	1.50	771.24	2	5	34	85.29
canti-3-m-mV-int	7.60	1775.26	12	96	97	1.03
canti-5-mV-int	2.20	1230.50	4	16	48	66.67
canti-5-m-mV-int	9.00	2353.87	6	24	54	55.56
canti-6-mV-int	1.60	694.65	4	10	38	73.68
canti-6-m-mV-int	7.60	1775.26	12	96	97	1.03
canti-7-mV-int	98.20	> 10000	25	379	500	24.20
canti-9-mV-int	92.60	7327.43	14	125	1909	93.45
canti-10-mV-int	125.70	> 10000	7	765	823	7.05
canti-big-mV-int	81.80	> 10000	6	125	693	81.96
lit-as-5-big-mV-int	36.00	> 10000	13	1033	1762	41.37
lit-as-6-big-mV-int	7.00	5420.30	38	1710	1710	0.00
bridge-1-act-mV	14.00	36.39	43	2576	2576	0.00

instance	root solution gap [%]	first solution gap [%]	best sol. found at depth	best sol. found at node	overall nodes	nodes after best solution [%]
bridge-2-act-mV-int	6.60	7.03	20	111	918	87.91
bridge-2-act-mV	0.80	29.60	10	29	29	0.00
bridge-3-act-mV-int	4.10	15.90	22	1211	1570	22.87
bridge-3-act-mV	4.20	13.11	61	247	247	0.00
bridge-4-act-mV	6.60	0.00	11	31	81	61.73
bridge-5-act-mV-int	10.20	16.84	19	4056	13175	69.21
bridge-5-act-mV	2.60	0.00	9	27	27	0.00
bridge-6-act-mV-int	5.20	9.91	13	4000	7105	43.70
bridge-6-act-mV	6.00	9.07	54	584	584	0.00
bridge-7-act-mV-int	15.20	39.20	13	604	780	22.56
bridge-7-act-mV	22.60	0.00	9	260	260	0.00
bridge-8-act-mV-int	17.30	12.22	27	851	1815	53.11
bridge-9-act-mV-int	273.80	0.00	11	1317	1867	29.46
bridge-9-act-mV	165.70	0.00	5	183	290	36.90
bridge-10-act-mV-int	262.90	86.29	74	1819	1819	0.00
canti-1-act-mV	3.40	6.68	3	7	23	69.57
canti-1-m-act-mV	22.30	20.48	15	73	73	0.00
canti-2-act-mV-int	7.20	17.41	11	105	130	19.23
canti-2-act-mV	6.30	34.02	11	47	47	0.00
canti-2-m-act-mV-int	23.10	0.00	7	22	221	90.05
canti-2-m-act-mV	24.60	24.47	7	166	166	0.00
canti-3-act-mV-int	4.50	2.89	3	60	159	62.26
canti-3-act-mV	0.70	0.00	12	32	32	0.00
canti-3-m-act-mV-int	11.00	39.43	8	56	145	61.38
canti-3-m-act-mV	12.90	4.00	9	52	52	0.00
canti-4-act-mV	12.10	0.00	1	2	86	97.67
canti-4-m-act-mV	28.50	0.00	13	131	131	0.00
canti-5-act-mV-int	10.00	15.07	15	638	880	27.50
canti-5-act-mV	5.60	0.00	9	37	37	0.00
canti-5-m-act-mV-int	30.00	0.00	7	21	287	92.68
canti-5-m-act-mV	29.00	24.47	5	51	196	73.98
canti-6-act-mV-int	4.00	5.04	10	39	112	65.18
canti-6-act-mV	8.80	0.00	12	198	301	34.22
canti-6-m-act-mV-int	5.60	12.11	8	21	35	40.00
canti-6-m-act-mV	17.50	0.00	0	1	72	98.61
canti-7-act-mV-int	128.50	59.90	57	2403	2403	0.00
canti-8-act-mV	40.10	0.00	8	38	91	58.24
canti-10-act-mV-int	99.80	5315.83	34	3356	3356	0.00
canti-big-act-mV-int	72.10	0.00	55	266	511	47.95
lit-as-1-act-mV	37.70	0.00	5	10	233	95.71
lit-as-5-big-act-mV-int	36.00	> 10000	13	1033	1762	41.37
lit-as-6-act-mV	13.20	1226.66	19	2491	5558	55.18

instance	root solution gap [%]	first solution gap [%]	best sol. found at depth	best sol. found at node	overall nodes	nodes after best solution [%]
lit-as-6-big-act-mV-int	7.00	5420.30	38	1710	1710	0.00
lit-s-1-act-mV	24.90	0.00	8	80	106	24.53

Table A.6. – Information about the quality of the first solution and the root relaxation of the truss instances, when solving using the MISDP branch-and-bound approach and minimizing the volume.

A.1.3. Comparing the models – volume minimization and discrete bars

We compared the different models for Truss Topology Design. In Table A.7 we present the same comparison for two different objective functions: minimizing the volume or minimizing the compliance. In this table the last column also shows which model can be solved faster: 'V' stands for the model that minimizes the volume and with 'C' we refer to the model that minimizes the compliance.

	minimizing the compliance			minimizing the volume			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
bridge-1	5.04	79.00	0.00	35.94	847.00	0.00	C
bridge-2	105.32	733.00	0.00	69.18	1015.00	0.00	V
bridge-3	40.62	362.00	0.00	86.10	1014.00	0.00	C
bridge-4	4.99	79.00	0.00	24.05	555.00	0.00	C
bridge-5	102.84	733.00	0.00	68.17	1015.00	0.00	V
bridge-6	40.33	362.00	0.00	43.68	501.00	0.00	C
bridge-7	1432.54	2504.00	0.00	13.37	38.00	0.00	V
bridge-8	7201.01	4361.40	–	2010.62	2837.00	0.00	V
bridge-9	66.65	323.00	0.00	46.71	498.00	0.00	V
bridge-10	1102.61	6642.00	0.00	7200.03	81793.00	33.60	C
canti-1	23.01	620.00	0.00	3.62	103.00	0.00	V
canti-2	60.07	698.00	0.00	6.16	102.00	0.00	V
canti-3	124.15	1121.00	0.00	8.30	120.00	0.00	V
canti-4	22.82	620.00	0.00	3.60	103.00	0.00	V
canti-5	60.15	698.00	0.00	6.18	102.00	0.00	V
canti-6	123.61	1121.00	0.00	8.31	120.00	0.00	V
canti-7	605.74	1517.00	0.00	3457.21	37477.00	0.00	C
canti-8	64.53	302.00	0.00	13.76	122.00	0.00	V
canti-9	254.01	481.00	0.00	3367.77	19632.00	0.00	C
canti-10	192.92	409.00	0.00	2968.92	35012.00	0.00	C
canti-big	1096.81	1273.00	0.00	7200.09	28768.40	3.82	C
lit-as-1	1.68	85.00	0.00	0.69	51.00	0.00	V

	minimizing the compliance			minimizing the volume			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
lit-as-2	7203.22	1205.60	–	7202.52	1821.40	2285.68	V
lit-as-2-big	7385.51	12.60	–	7329.79	27.20	–	–
lit-as-5	7200.69	7242.00	–	1931.00	3636.00	0.00	V
lit-as-5-big	7240.45	121.60	–	7216.83	242.40	–	–
lit-as-6	898.80	1983.00	0.00	1695.02	5558.00	0.00	C
lit-as-6-big	7202.64	1171.00	–	7201.96	1967.20	–	–
lit-s-1	3272.54	21610.00	0.00	6.55	83.00	0.00	V
bridge-1-act	20.72	130.00	0.00	355.10	2576.00	0.00	C
bridge-2-act	284.27	1124.00	0.00	7.55	29.00	0.00	V
bridge-3-act	190.01	712.00	0.00	53.05	247.00	0.00	V
bridge-4-act	20.78	114.00	0.00	12.62	81.00	0.00	V
bridge-5-act	293.04	1179.00	0.00	7.04	27.00	0.00	V
bridge-6-act	209.19	800.00	0.00	124.55	584.00	0.00	V
bridge-7-act	5908.73	4164.00	0.00	311.16	260.00	0.00	V
bridge-8-act	7201.87	1313.80	–	7201.33	2186.00	–	–
bridge-9-act	388.64	1070.00	0.00	74.36	290.00	0.00	V
bridge-10-act	720.74	861.00	0.00	7200.18	45895.80	51.60	C
canti-1-act	106.06	723.00	0.00	9.57	73.00	0.00	V
canti-2-act	182.73	1629.00	0.00	28.81	166.00	0.00	V
canti-3-act	256.42	993.00	0.00	9.64	52.00	0.00	V
canti-4-act	58.65	360.00	0.00	16.41	131.00	0.00	V
canti-5-act	191.21	1655.00	0.00	24.46	196.00	0.00	V
canti-6-act	212.51	781.00	0.00	13.45	72.00	0.00	V
canti-7-act	658.47	495.00	0.00	7200.46	10477.60	–	C
canti-8-act	143.33	293.00	0.00	22.83	91.00	0.00	V
canti-9-act	1703.01	1189.00	0.00	7200.81	9526.60	–	C
canti-10-act	888.75	1093.00	0.00	7200.09	38623.20	–	C
canti-big-act	5409.97	2225.00	0.00	7201.02	3169.60	–	C
lit-as-1-act	5.03	79.00	0.00	8.55	233.00	0.00	C
lit-as-2-act	7246.01	46.20	–	7238.32	95.60	–	–
lit-as-2-big-act	8169.05	5.00	–			–	–
lit-as-5-act	7207.23	498.20	–	7085.71	1456.40	0.09	V
lit-as-5-big-act	7224.05	109.20	–	7215.69	248.60	–	–
lit-as-6-act	898.11	1983.00	0.00	1711.35	5558.00	0.00	C
lit-as-6-big-act	7202.62	1189.60	–	7202.36	1936.40	–	–
lit-s-1-act	7200.27	11110.00	2063.86	46.05	106.00	0.00	V
bridge-2-int	105.32	733.00	0.00	17.06	302.00	0.00	V
bridge-3-int	40.62	362.00	0.00	4.64	89.00	0.00	V
bridge-5-int	102.84	733.00	0.00	16.81	302.00	0.00	V
bridge-6-int	40.33	362.00	0.00	16.31	337.00	0.00	V
bridge-7-int	1432.54	2504.00	0.00	1.29	13.00	0.00	V
bridge-8-int	7201.01	4361.40	–	64.08	324.00	0.00	V
bridge-9-int	66.65	323.00	0.00	16.94	545.00	0.00	V

	minimizing the compliance			minimizing the volume			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
bridge-10-int	1102.61	6642.00	0.00	130.01	2929.00	0.00	V
canti-2-int	60.07	698.00	0.00	1.93	54.00	0.00	V
canti-3-int	124.15	1121.00	0.00	3.65	97.00	0.00	V
canti-5-int	60.15	698.00	0.00	1.94	54.00	0.00	V
canti-6-int	123.61	1121.00	0.00	3.65	97.00	0.00	V
canti-7-int	605.74	1517.00	0.00	32.86	500.00	0.00	V
canti-8-int	64.53	302.00	0.00	4.06	109.00	0.00	V
canti-9-int	254.01	481.00	0.00	99.65	1909.00	0.00	V
canti-10-int	192.92	409.00	0.00	51.73	823.00	0.00	V
canti-big-int	1096.81	1273.00	0.00	80.54	693.00	0.00	V
lit-as-2-big-int	7385.51	12.60	–	7202.33	1756.60	2.04	V
lit-as-5-big-int	7240.45	121.60	–	883.51	1762.00	0.00	V
lit-as-6-big-int	7202.64	1171.00	–	571.37	1710.00	0.00	V
bridge-2-act-int	284.27	1124.00	0.00	157.85	918.00	0.00	V
bridge-3-act-int	190.01	712.00	0.00	263.65	1570.00	0.00	C
bridge-5-act-int	293.04	1179.00	0.00	2042.50	13175.00	0.00	C
bridge-6-act-int	209.19	800.00	0.00	1146.08	7105.00	0.00	C
bridge-7-act-int	5908.73	4164.00	0.00	255.21	780.00	0.00	V
bridge-8-act-int	7201.87	1313.80	–	1840.51	1815.00	0.00	V
bridge-9-act-int	388.64	1070.00	0.00	255.01	1867.00	0.00	V
bridge-10-act-int	720.74	861.00	0.00	531.97	1819.00	0.00	V
canti-2-act-int	182.73	1629.00	0.00	28.37	221.00	0.00	V
canti-3-act-int	256.42	993.00	0.00	19.49	145.00	0.00	V
canti-5-act-int	191.21	1655.00	0.00	35.03	287.00	0.00	V
canti-6-act-int	212.51	781.00	0.00	4.68	35.00	0.00	V
canti-7-act-int	658.47	495.00	0.00	882.18	2403.00	0.00	C
canti-8-act-int	143.33	293.00	0.00	37.90	239.00	0.00	V
canti-9-act-int	1703.01	1189.00	0.00	7200.03	40585.40	128.56	C
canti-10-act-int	888.75	1093.00	0.00	869.97	3356.00	0.00	V
canti-big-act-int	5409.97	2225.00	0.00	387.04	511.00	0.00	V
lit-as-2-big-act-int	8169.05	5.00	–	7249.71	93.80	–	–
lit-as-5-big-act-int	7224.05	109.20	–	898.65	1762.00	0.00	V
lit-as-6-big-act-int	7202.62	1189.60	–	578.36	1710.00	0.00	V
bridge-1-2scen	232.17	2299.00	0.00	288.33	3773.00	0.00	C
bridge-2-2scen	7200.08	43287.20	42.14	1349.46	13544.00	0.00	V
bridge-3-2scen	1861.59	10533.00	0.00	2174.15	16846.00	0.00	C
bridge-4-2scen	228.23	2299.00	0.00	198.07	2592.00	0.00	V
bridge-5-2scen	7200.07	43462.60	42.12	1346.56	13544.00	0.00	V
bridge-6-2scen	1814.74	10533.00	0.00	1323.15	10086.00	0.00	V
bridge-7-2scen	7200.43	8118.20	–	536.99	1358.00	0.00	V
bridge-8-2scen	7201.54	2999.20	–	7200.36	7087.20	4.90	V
bridge-9-2scen	245.43	652.00	0.00	36.98	255.00	0.00	V
bridge-10-2scen	7189.18	18789.00	0.02	7200.07	34747.00	–	C

	minimizing the compliance			minimizing the volume			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
canti-1-2scen	43.88	505.00	0.00	0.08	1.00	infeasible	–
canti-2-2scen	669.06	4754.00	0.00	49.13	385.00	0.00	V
canti-3-2scen	455.97	2831.00	0.00	29.16	216.00	0.00	V
canti-4-2scen	43.86	505.00	0.00	0.08	1.00	infeasible	–
canti-5-2scen	666.57	4754.00	0.00	49.14	385.00	0.00	V
canti-6-2scen	456.76	2831.00	0.00	29.23	216.00	0.00	V
canti-7-2scen	7200.21	11006.60	3.58	7200.12	23404.40	24.81	C
canti-9-2scen	7200.44	8963.40	553.08	7200.16	17481.80	51.44	V
canti-10-2scen	7200.21	12704.80	421.00	7200.17	25858.20	–	C
canti-big-2scen	7201.09	3618.80	–	7200.50	7355.60	–	–
bridge-1-act-2scen	797.30	2158.00	0.00	696.68	2436.00	0.00	V
bridge-2-act-2scen	7200.16	13911.00	–	1437.06	3728.00	0.00	V
bridge-3-act-2scen	5658.01	11733.00	0.00	1859.06	5119.00	0.00	V
bridge-4-act-2scen	835.05	2563.00	0.00	840.77	3072.00	0.00	C
bridge-5-act-2scen	7200.34	13934.60	–	1961.70	5686.00	0.00	V
bridge-6-act-2scen	6195.92	13265.00	0.00	1721.03	4825.00	0.00	V
bridge-7-act-2scen	7201.90	2465.60	–	7201.31	3583.40	–	–
bridge-8-act-2scen	7205.50	720.60	–	7203.99	1125.20	–	–
bridge-9-act-2scen	807.06	939.00	0.00	63.80	147.00	0.00	V
bridge-10-act-2scen	7200.93	4566.00	203.30	7200.14	25639.40	–	C
canti-1-act-2scen	177.99	655.00	0.00	0.30	1.00	infeasible	–
canti-2-act-2scen	1906.94	5111.00	0.00	333.06	923.00	0.00	V
canti-3-act-2scen	1505.96	4274.00	0.00	49.27	121.00	0.00	V
canti-4-act-2scen	151.59	523.00	0.00	0.29	1.00	infeasible	–
canti-5-act-2scen	1527.60	4151.00	0.00	192.23	528.00	0.00	V
canti-6-act-2scen	1073.52	2517.00	0.00	70.49	178.00	0.00	V
canti-7-act-2scen	7201.61	3137.80	–	7200.53	5361.40	–	–
canti-9-act-2scen	7201.73	2481.20	–	7200.99	4291.60	–	–
canti-10-act-2scen	7200.78	4654.60	475.07	7200.60	6881.40	–	C
canti-big-act-2scen	7204.02	783.60	–	7201.39	1520.00	–	–
bridge-2-2scen-int	7200.08	43287.20	42.14	229.01	2254.00	0.00	V
bridge-3-2scen-int	1861.59	10533.00	0.00	40.35	390.00	0.00	V
bridge-5-2scen-int	7200.07	43462.60	42.12	227.79	2254.00	0.00	V
bridge-6-2scen-int	1814.74	10533.00	0.00	97.66	1009.00	0.00	V
bridge-7-2scen-int	7200.43	8118.20	–	69.62	414.00	0.00	V
bridge-8-2scen-int	7201.54	2999.20	–	115.97	328.00	0.00	V
bridge-9-2scen-int	245.43	652.00	0.00	18.44	319.00	0.00	V
bridge-10-2scen-int	7189.18	18789.00	0.02	1339.31	10334.00	0.00	V
canti-7-2scen-int	7200.21	11006.60	3.58	9.02	93.00	0.00	V
canti-9-2scen-int	7200.44	8963.40	553.08	54.41	355.00	0.00	V
canti-10-2scen-int	7200.21	12704.80	421.00	1.82	20.00	0.00	V
canti-big-2scen-int	7201.09	3618.80	–	6.40	29.00	0.00	V
bridge-2-act-2scen-int	7200.16	13911.00	–	3348.08	9062.00	0.00	V

	minimizing the compliance			minimizing the volume			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
bridge-3-act-2scen-int	5658.01	11733.00	0.00	2027.43	5821.00	0.00	V
bridge-5-act-2scen-int	7200.34	13934.60	–	3601.90	10558.00	0.00	V
bridge-6-act-2scen-int	6195.92	13265.00	0.00	1953.60	5455.00	0.00	V
bridge-7-act-2scen-int	7201.90	2465.60	–	1176.80	1246.00	0.00	V
bridge-8-act-2scen-int	7205.50	720.60	–	7201.32	3035.00	0.94	V
bridge-9-act-2scen-int	807.06	939.00	0.00	517.27	1495.00	0.00	V
bridge-10-act-2scen-int	7200.93	4566.00	203.30	7200.34	12816.60	1892.98	C
canti-2-act-2scen-int	1906.94	5111.00	0.00	166.71	573.00	0.00	V
canti-3-act-2scen-int	1505.96	4274.00	0.00	45.01	149.00	0.00	V
canti-5-act-2scen-int	1527.60	4151.00	0.00	243.41	815.00	0.00	V
canti-6-act-2scen-int	1073.52	2517.00	0.00	89.84	271.00	0.00	V
canti-7-act-2scen-int	7201.61	3137.80	–	7200.26	8583.60	1921.40	V
canti-9-act-2scen-int	7201.73	2481.20	–	7200.33	8032.80	–	–
canti-10-act-2scen-int	7200.78	4654.60	475.07	7200.31	13706.60	2593.77	C
canti-big-act-2scen-int	7204.02	783.60	–	7201.44	3427.60	–	–

Table A.7. – Comparing two different objective functions for truss topology design: minimizing the compliance and minimizing the volume.

Table A.8 shows the differences between modeling discrete cross-sectional areas using binary and integer variables. The last column again shows which model can be solved faster: '*B*' stands for the model with binary variables and '*I*' represents the model with integer variables.

	solving with binary variables			solving with integer variables			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
bridge-2	105.32	733.00	0.00	36.62	645.00	0.00	I
bridge-3	40.62	362.00	0.00	68.91	1149.00	0.00	B
bridge-5	102.84	733.00	0.00	36.22	645.00	0.00	I
bridge-6	40.33	362.00	0.00	68.87	1149.00	0.00	B
bridge-7	1432.54	2504.00	0.00	12.44	118.00	0.00	I
bridge-8	7201.01	4361.40	–	1568.68	4834.00	0.00	I
bridge-9	66.65	323.00	0.00	57.17	1088.00	0.00	I
bridge-10	1102.61	6642.00	0.00	49.82	455.00	0.00	I
bridge-big	1792.50	1228.00	0.00	7200.22	14620.60	–	B
canti-2	60.07	698.00	0.00	14.79	339.00	0.00	I
canti-3	124.15	1121.00	0.00	11.14	207.00	0.00	I
canti-5	60.15	698.00	0.00	14.82	339.00	0.00	I
canti-6	123.61	1121.00	0.00	11.17	207.00	0.00	I
canti-7	605.74	1517.00	0.00	32.57	219.00	0.00	I
canti-8	64.53	302.00	0.00	7.20	109.00	0.00	I
canti-9	254.01	481.00	0.00	1685.27	12650.00	0.00	B

	solving with binary variables			solving with integer variables			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
canti-10	192.92	409.00	0.00	6.41	46.00	0.00	I
canti-big	1096.81	1273.00	0.00	118.57	432.00	0.00	I
lit-as-2-big	7385.51	12.60	–	7203.79	1085.40	–	–
lit-as-5-big	7240.45	121.60	–	4450.68	4463.00	0.00	I
lit-as-6-big	7202.64	1171.00	–	7200.21	15042.40	0.10	I
bridge-2-act	284.27	1124.00	0.00	7200.07	112487.40	4.53	B
bridge-3-act	190.01	712.00	0.00	951.00	10108.00	0.00	B
bridge-5-act	293.04	1179.00	0.00	7200.05	106126.60	30.48	B
bridge-6-act	209.19	800.00	0.00	7200.02	104381.60	19.52	B
bridge-7-act	5908.73	4164.00	0.00	7200.02	33381.80	118.24	B
bridge-8-act	7201.87	1313.80	–	7200.48	8372.00	–	–
bridge-9-act	388.64	1070.00	0.00	5125.62	59718.00	0.00	B
bridge-10-act	720.74	861.00	0.00	7200.10	27943.00	49.87	B
bridge-big-act	7204.64	865.60	–	7202.14	3016.60	–	–
canti-2-act	182.73	1629.00	0.00	185.98	2695.00	0.00	B
canti-3-act	256.42	993.00	0.00	242.99	1675.00	0.00	I
canti-5-act	191.21	1655.00	0.00	1865.97	25536.00	0.00	B
canti-6-act	212.51	781.00	0.00	2343.84	24420.00	0.00	B
canti-7-act	658.47	495.00	0.00	1373.81	5119.00	0.00	B
canti-8-act	143.33	293.00	0.00	5544.24	59381.00	0.00	B
canti-9-act	1703.01	1189.00	0.00	7200.10	23209.80	363497.06	B
canti-10-act	888.75	1093.00	0.00	939.82	4054.00	0.00	B
canti-big-act	5409.97	2225.00	0.00	7200.36	11657.80	79.97	B
lit-as-2-big-act	8169.05	5.00	–	7226.34	82.60	–	–
lit-as-5-big-act	7224.05	109.20	–	4323.24	4463.00	0.00	I
lit-as-6-big-act	7202.62	1189.60	–	7200.28	15270.40	0.09	I
bridge-2-mV	69.18	1015.00	0.00	17.06	302.00	0.00	I
bridge-3-mV	86.10	1014.00	0.00	4.64	89.00	0.00	I
bridge-5-mV	68.17	1015.00	0.00	16.81	302.00	0.00	I
bridge-6-mV	43.68	501.00	0.00	16.31	337.00	0.00	I
bridge-7-mV	13.37	38.00	0.00	1.29	13.00	0.00	I
bridge-8-mV	2010.62	2837.00	0.00	64.08	324.00	0.00	I
bridge-9-mV	46.71	498.00	0.00	16.94	545.00	0.00	I
bridge-10-mV	7200.03	81793.00	33.60	130.01	2929.00	0.00	I
canti-2-mV	6.16	102.00	0.00	1.93	54.00	0.00	I
canti-3-mV	8.30	120.00	0.00	3.65	97.00	0.00	I
canti-5-mV	6.18	102.00	0.00	1.94	54.00	0.00	I
canti-6-mV	8.31	120.00	0.00	3.65	97.00	0.00	I
canti-7-mV	3457.21	37477.00	0.00	32.86	500.00	0.00	I
canti-8-mV	13.76	122.00	0.00	4.06	109.00	0.00	I
canti-9-mV	3367.77	19632.00	0.00	99.65	1909.00	0.00	I
canti-10-mV	2968.92	35012.00	0.00	51.73	823.00	0.00	I
canti-big-mV	7200.09	28768.40	3.82	80.54	693.00	0.00	I

	solving with binary variables			solving with integer variables			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
bridge-2-act-mV	7.55	29.00	0.00	157.85	918.00	0.00	B
bridge-3-act-mV	53.05	247.00	0.00	263.65	1570.00	0.00	B
bridge-5-act-mV	7.04	27.00	0.00	2042.50	13175.00	0.00	B
bridge-6-act-mV	124.55	584.00	0.00	1146.08	7105.00	0.00	B
bridge-7-act-mV	311.16	260.00	0.00	255.21	780.00	0.00	I
bridge-8-act-mV	7201.33	2186.00	–	1840.51	1815.00	0.00	I
bridge-9-act-mV	74.36	290.00	0.00	255.01	1867.00	0.00	B
bridge-10-act-mV	7200.18	45895.80	51.60	531.97	1819.00	0.00	I
canti-2-act-mV	28.81	166.00	0.00	28.37	221.00	0.00	I
canti-3-act-mV	9.64	52.00	0.00	19.49	145.00	0.00	B
canti-5-act-mV	24.46	196.00	0.00	35.03	287.00	0.00	B
canti-6-act-mV	13.45	72.00	0.00	4.68	35.00	0.00	I
canti-7-act-mV	7200.46	10477.60	–	882.18	2403.00	0.00	I
canti-8-act-mV	22.83	91.00	0.00	37.90	239.00	0.00	B
canti-9-act-mV	7200.81	9526.60	–	7200.03	40585.40	128.56	I
canti-10-act-mV	7200.09	38623.20	–	869.97	3356.00	0.00	I
canti-big-act-mV	7201.02	3169.60	–	387.04	511.00	0.00	I
lit-as-5-big-act-mV	7215.69	248.60	–	898.65	1762.00	0.00	I
lit-as-6-big-act-mV	7202.36	1936.40	–	578.36	1710.00	0.00	I
bridge-2-2scen	7200.08	43287.20	42.14	431.15	3964.00	0.00	I
bridge-3-2scen	1861.59	10533.00	0.00	295.00	2537.00	0.00	I
bridge-5-2scen	7200.07	43462.60	42.12	429.55	3964.00	0.00	I
bridge-6-2scen	1814.74	10533.00	0.00	289.64	2537.00	0.00	I
bridge-7-2scen	7200.43	8118.20	–	315.31	1156.00	0.00	I
bridge-8-2scen	7201.54	2999.20	–	2889.06	5038.00	0.00	I
bridge-9-2scen	245.43	652.00	0.00	182.06	1579.00	0.00	I
bridge-10-2scen	7189.18	18789.00	0.02	7200.07	48638.80	16.51	B
bridge-big-2scen	7201.16	3132.80	–	7200.50	8395.20	–	–
canti-2-2scen	669.06	4754.00	0.00	816.04	14020.00	0.00	B
canti-3-2scen	455.97	2831.00	0.00	888.54	10700.00	0.00	B
canti-5-2scen	666.57	4754.00	0.00	816.87	14020.00	0.00	B
canti-6-2scen	456.76	2831.00	0.00	892.59	10700.00	0.00	B
canti-7-2scen	7200.21	11006.60	3.58	1240.38	5460.00	0.00	I
canti-9-2scen	7200.44	8963.40	553.08	7200.13	27555.00	–	B
canti-10-2scen	7200.21	12704.80	421.00	2382.31	12917.00	0.00	I
canti-big-2scen	7201.09	3618.80	–	7200.15	15132.20	–	–
bridge-2-act-2scen	7200.16	13911.00	–	7200.12	35691.00	105673.87	I
bridge-3-act-2scen	5658.01	11733.00	0.00	7200.14	33115.80	–	B
bridge-5-act-2scen	7200.34	13934.60	–	7200.16	33020.80	99341.99	I
bridge-6-act-2scen	6195.92	13265.00	0.00	7200.12	30386.40	–	B
bridge-7-act-2scen	7201.90	2465.60	–	7200.22	17251.00	205.52	I
bridge-8-act-2scen	7205.50	720.60	–	7200.95	4410.60	–	–
bridge-9-act-2scen	807.06	939.00	0.00	548.16	3220.00	0.00	I

	solving with binary variables			solving with integer variables			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
bridge-10-act-2scen	7200.93	4566.00	203.30	7200.10	12193.20	–	B
bridge-big-act-2scen	7209.36	430.80	–	7203.39	1114.20	–	–
canti-2-act-2scen	1906.94	5111.00	0.00	7200.12	41560.60	1315.22	B
canti-3-act-2scen	1505.96	4274.00	0.00	7200.08	45527.40	49.39	B
canti-5-act-2scen	1527.60	4151.00	0.00	7200.05	44512.40	2434.12	B
canti-6-act-2scen	1073.52	2517.00	0.00	7200.09	42176.20	83.96	B
canti-7-act-2scen	7201.61	3137.80	–	7200.35	13857.20	–	–
canti-9-act-2scen	7201.73	2481.20	–	7200.51	9385.80	–	–
bridge-2-2scen	7200.08	43287.20	42.14	431.15	3964.00	0.00	I
bridge-3-2scen-mV	2174.15	16846.00	0.00	40.35	390.00	0.00	I
bridge-5-2scen-mV	1346.56	13544.00	0.00	227.79	2254.00	0.00	I
bridge-6-2scen-mV	1323.15	10086.00	0.00	97.66	1009.00	0.00	I
bridge-7-2scen-mV	536.99	1358.00	0.00	69.62	414.00	0.00	I
bridge-8-2scen-mV	7200.36	7087.20	4.90	115.97	328.00	0.00	I
bridge-9-2scen-mV	36.98	255.00	0.00	18.44	319.00	0.00	I
bridge-10-2scen-mV	7200.07	34747.00	–	1339.31	10334.00	0.00	I
canti-7-2scen-mV	7200.12	23404.40	24.81	9.02	93.00	0.00	I
canti-9-2scen-mV	7200.16	17481.80	51.44	54.41	355.00	0.00	I
canti-10-2scen-mV	7200.17	25858.20	–	1.82	20.00	0.00	I
canti-big-2scen-mV	7200.50	7355.60	–	6.40	29.00	0.00	I
bridge-2-act-2scen-mV	1437.06	3728.00	0.00	3348.08	9062.00	0.00	B
bridge-3-act-2scen-mV	1859.06	5119.00	0.00	2027.43	5821.00	0.00	B
bridge-5-act-2scen-mV	1961.70	5686.00	0.00	3601.90	10558.00	0.00	B
bridge-6-act-2scen-mV	1721.03	4825.00	0.00	1953.60	5455.00	0.00	B
bridge-7-act-2scen-mV	7201.31	3583.40	–	1176.80	1246.00	0.00	I
bridge-8-act-2scen-mV	7203.99	1125.20	–	7201.32	3035.00	0.94	I
bridge-9-act-2scen-mV	63.80	147.00	0.00	517.27	1495.00	0.00	B
bridge-10-act-2scen-mV	7200.14	25639.40	–	7200.34	12816.60	1892.98	I
canti-2-act-2scen-mV	333.06	923.00	0.00	166.71	573.00	0.00	I
canti-3-act-2scen-mV	49.27	121.00	0.00	45.01	149.00	0.00	I
canti-5-act-2scen-mV	192.23	528.00	0.00	243.41	815.00	0.00	B
canti-6-act-2scen-mV	70.49	178.00	0.00	89.84	271.00	0.00	B
canti-7-act-2scen-mV	7200.53	5361.40	–	7200.26	8583.60	1921.40	I
canti-9-act-2scen-mV	7200.99	4291.60	–	7200.33	8032.80	–	–
canti-10-act-2scen-mV	7200.60	6881.40	–	7200.31	13706.60	2593.77	I
canti-big-act-2scen-mV	7201.39	1520.00	–	7201.44	3427.60	–	–

Table A.8. – Comparison of the different models for discrete cross-sectional areas: binary or integer variables.

A.1.4. Comparing the models – MIP and MISDP

Not all extensions we presented in Chapter 5 can also be modeled within the MIP formulation, hence we need to look at a smaller test set when comparing MIP and MISDP. We tested 104 single load instances and present the results in Tables A.9 and A.10. The results for multiple loads are shown in the following section. The tables are of the same kind as before with a last column for showing which model was the best: '*M*' stands for the MIP and '*S*' for the MISDP.

	solving as MISDP			solving as MIP			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
bridge-1	5.04	79.00	0.00	1565.34	481242.00	0.01	S
bridge-1-act	20.72	130.00	0.00	7200.01	1235610.00	1.19	S
bridge-2	105.32	733.00	0.00	7200.01	3995826.20	53.07	S
bridge-2-act	284.27	1124.00	0.00	7200.01	819100.00	56.54	S
bridge-3	40.62	362.00	0.00	7200.01	1861801.20	44.47	S
bridge-3-act	190.01	712.00	0.00	7200.02	1256853.20	46.20	S
bridge-4	4.99	79.00	0.00	1450.93	481242.00	0.01	S
bridge-4-act	20.78	114.00	0.00	2734.56	414885.00	0.01	S
bridge-5	102.84	733.00	0.00	7200.01	4033469.40	53.72	S
bridge-5-act	293.04	1179.00	0.00	7200.01	724078.67	61.16	S
bridge-6	40.33	362.00	0.00	7200.01	1948596.20	45.62	S
bridge-6-act	209.19	800.00	0.00	7200.01	1033619.20	43.55	S
bridge-7	1432.54	2504.00	0.00	7200.01	3434977.60	76.18	S
bridge-7-act	5908.73	4164.00	0.00	7200.01	1838213.80	75.32	S
bridge-9	66.65	323.00	0.00	7200.01	1916343.20	54.61	S
bridge-9-act	388.64	1070.00	0.00	7200.01	1204261.60	58.24	S
bridge-10	1102.61	6642.00	0.00	7200.01	3176919.20	80.61	S
bridge-10-act	720.74	861.00	0.00	7200.02	826800.60	82.31	S
bridge-big	1792.50	1228.00	0.00	348.68	46800.00	–	S
canti-1	23.01	620.00	0.00	84.39	34887.00	0.01	S
canti-1-act	106.06	723.00	0.00	214.16	46889.00	0.01	S
canti-2	60.07	698.00	0.00	3404.95	3769459.00	0.01	S
canti-2-act	182.73	1629.00	0.00	7200.01	736087.60	38.73	S
canti-3	124.15	1121.00	0.00	7200.01	1746646.60	34.00	S
canti-3-act	256.42	993.00	0.00	7200.01	925988.80	25.20	S
canti-4	22.82	620.00	0.00	79.53	34887.00	0.01	S
canti-4-act	58.65	360.00	0.00	241.24	60354.00	0.01	S
canti-5	60.15	698.00	0.00	3371.27	3769459.00	0.01	S
canti-5-act	191.21	1655.00	0.00	7200.02	641635.60	41.39	S
canti-6	123.61	1121.00	0.00	7200.01	1725921.60	33.27	S
canti-6-act	212.51	781.00	0.00	7200.02	573254.60	36.08	S
canti-7	605.74	1517.00	0.00	7200.01	2738881.75	76.90	S
canti-7-act	658.47	495.00	0.00	7200.01	883536.40	78.87	S
canti-8	64.53	302.00	0.00	7200.06	539359.80	0.14	S

	solving as MISDP			solving as MIP			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
canti-8-act	143.33	293.00	0.00	7200.05	574799.60	0.13	S
canti-9	254.01	481.00	0.00	7200.01	2980492.50	73.06	S
canti-9-act	1703.01	1189.00	0.00	7200.01	982240.40	75.35	S
canti-10	192.92	409.00	0.00	7200.01	2933699.75	57.86	S
canti-10-act	888.75	1093.00	0.00	7200.01	828881.80	56.41	S
canti-big	1096.81	1273.00	0.00	7200.01	2251844.25	85.04	S
canti-big-act	5409.97	2225.00	0.00	7200.00	1120412.60	83.44	S
lit-as-1	1.68	85.00	0.00	0.81	1619.00	0.00	M
lit-as-1-act	5.03	79.00	0.00	0.83	1619.00	0.00	M
lit-as-6	898.80	1983.00	0.00	5.96	0.00	0.00	M
lit-as-6-act	898.11	1983.00	0.00	5.86	0.00	0.00	M
lit-s-1	3272.54	21610.00	0.00	7200.02	974901.20	66.99	S
bridge-1-act-mV	355.10	2576.00	0.00	1.30	1278.00	0.00	M
bridge-1-mV	35.94	847.00	0.00	0.38	507.00	0.00	M
bridge-2-act-mV	7.55	29.00	0.00	2.83	1948.00	0.00	M
bridge-2-mV	69.18	1015.00	0.00	2.84	3402.00	0.00	M
bridge-3-act-mV	53.05	247.00	0.00	2.83	1948.00	0.00	M
bridge-3-mV	86.10	1014.00	0.00	2.83	3402.00	0.00	M
bridge-4-act-mV	12.62	81.00	0.00	2.46	1375.00	0.01	M
bridge-4-mV	24.05	555.00	0.00	0.38	507.00	0.00	M
bridge-5-act-mV	7.04	27.00	0.00	7.97	3949.00	0.01	S
bridge-5-mV	68.17	1015.00	0.00	2.79	3402.00	0.00	M
bridge-6-act-mV	124.55	584.00	0.00	7.99	3949.00	0.01	M
bridge-6-mV	43.68	501.00	0.00	2.84	3402.00	0.00	M
bridge-7-act-mV	311.16	260.00	0.00	12.75	2578.00	0.00	M
bridge-7-mV	13.37	38.00	0.00	4.73	1776.00	0.00	M
bridge-8-mV	2010.62	2837.00	0.00	7200.00	1677617.60	43.32	S
bridge-9-act-mV	74.36	290.00	0.00	1.68	829.00	0.00	M
bridge-9-mV	46.71	498.00	0.00	0.19	142.00	0.00	M
bridge-10-act-mV	7200.18	45895.80	51.60	143.22	56589.00	0.01	M
bridge-10-mV	7200.03	81793.00	33.60	66.95	46863.00	0.00	M
canti-1-act-mV	3.34	23.00	0.00	3.98	4111.00	0.00	S
canti-1-mV	2.37	62.00	0.00	0.68	1863.00	0.00	M
canti-1-act-mV	9.57	73.00	0.00	3.90	4111.00	0.00	M
canti-1-mV	3.62	103.00	0.00	0.67	1863.00	0.00	M
canti-2-act-mV	9.33	47.00	0.00	5.83	4118.00	0.00	M
canti-2-mV	19.45	343.00	0.00	2.06	2480.00	0.00	M
canti-2-act-mV	28.81	166.00	0.00	5.77	4118.00	0.00	M
canti-2-mV	6.16	102.00	0.00	2.07	2480.00	0.00	M
canti-3-act-mV	7.27	32.00	0.00	5.85	4118.00	0.00	M
canti-3-mV	40.15	569.00	0.00	2.10	2480.00	0.00	M
canti-3-act-mV	9.64	52.00	0.00	5.76	4118.00	0.00	M
canti-3-mV	8.30	120.00	0.00	2.06	2480.00	0.00	M

	solving as MISDP			solving as MIP			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
canti-4-act-mV	11.35	86.00	0.00	2.01	2475.00	0.00	M
canti-4-mV	10.10	267.00	0.00	0.69	1863.00	0.00	M
canti-4-act-mV	16.41	131.00	0.00	1.94	2475.00	0.00	M
canti-4-mV	3.60	103.00	0.00	0.67	1863.00	0.00	M
canti-5-act-mV	7.58	37.00	0.00	4.59	3683.00	0.01	M
canti-5-mV	5.69	88.00	0.00	2.12	2480.00	0.00	M
canti-5-act-mV	24.46	196.00	0.00	4.51	3683.00	0.01	M
canti-5-mV	6.18	102.00	0.00	2.06	2480.00	0.00	M
canti-6-act-mV	55.89	301.00	0.00	4.57	3683.00	0.01	M
canti-6-mV	81.38	1293.00	0.00	2.14	2480.00	0.00	M
canti-6-act-mV	13.45	72.00	0.00	4.47	3683.00	0.01	M
canti-6-mV	8.31	120.00	0.00	2.05	2480.00	0.00	M
canti-7-act-mV	7200.46	10477.60	–	902.51	256354.00	0.01	M
canti-7-mV	3457.21	37477.00	0.00	187.76	106988.00	0.01	M
canti-8-act-mV	22.83	91.00	0.00	0.58	918.00	0.00	M
canti-8-mV	13.76	122.00	0.00	0.62	918.00	0.00	M
canti-9-act-mV	7200.81	9526.60	–	2137.02	559311.00	0.01	M
canti-9-mV	3367.77	19632.00	0.00	657.35	354823.00	0.01	M
canti-10-act-mV	7200.09	38623.20	–	86.59	34390.00	0.00	M
canti-10-mV	2968.92	35012.00	0.00	18.87	14699.00	0.00	M
canti-big-act-mV	7201.02	3169.60	–	3841.72	542136.00	0.01	M
canti-big-mV	7200.09	28768.40	3.82	666.29	167430.00	0.01	M
lit-as-1-act-mV	8.55	233.00	0.00	0.61	2062.00	0.00	M
lit-as-1-mV	0.69	51.00	0.00	0.59	2062.00	0.00	M
lit-as-5-mV	1931.00	3636.00	0.00	7200.01	1707782.60	34.55	S
lit-as-6-act-mV	1711.35	5558.00	0.00	7200.01	14436.00	–	S
lit-as-6-mV	1695.02	5558.00	0.00	7200.01	14218.60	–	S
lit-s-1-act-mV	46.05	106.00	0.00	6.86	6765.00	0.00	M
lit-s-1-mV	6.55	83.00	0.00	7.29	6765.00	0.00	S

Table A.9. – Comparing the MIP and the MISDP model for the instances where at least one of the models was able to solve the problem to optimality within the time limit of two hours.

	solving as MISDP			solving as MIP		
instance	time	nodes	gap [%]	time	nodes	gap [%]
bridge-8	7201.01	4361.40	–	7200.01	1877770.20	89.59
bridge-8-act	7201.87	1313.80	–	1883.15	181138.00	–
bridge-big-act	7204.64	865.60	–	7200.01	1024065.80	77.90
lit-as-2	7203.22	1205.60	–	6515.08	18791.80	–
lit-as-2-act	7246.01	46.20	–	6855.77	18548.80	–
lit-as-2-big	7385.51	12.60	–	7199.98	69346.60	–

	solving as MISDP			solving as MIP		
instance	time	nodes	gap [%]	time	nodes	gap [%]
lit-as-2-big-act	8169.05	5.00	–	7199.98	73123.20	–
lit-as-5	7200.69	7242.00	–	7200.07	391282.80	85.71
lit-as-5-act	7207.23	498.20	–	7200.03	411888.40	87.09
lit-as-5-big	7240.45	121.60	–	7200.00	4311.00	99.20
lit-as-5-big-act	7224.05	109.20	–	7200.00	3913.20	99.20
lit-as-6-big	7202.64	1171.00	–	7200.08	2952.20	94.65
lit-as-6-big-act	7202.62	1189.60	–	7200.07	3244.25	94.52
lit-s-1-act	7200.27	11110.00	2063.86	7200.01	944504.20	66.20
bridge-8-act-mV	7201.33	2186.00	–	7200.00	1044849.80	51.88
lit-as-2-act-mV	7238.32	95.60	–	7199.99	331254.75	95.61
lit-as-2-mV	7202.52	1821.40	2285.68	7199.99	330550.00	95.51
lit-as-2-big-mV	7329.79	27.20	–	7199.93	123304.00	96.27
lit-as-5-act-mV	7085.71	1456.40	0.09	7200.01	1730289.00	34.89
lit-as-5-big-act-mV	7215.69	248.60	–	7199.98	545305.25	51.89
lit-as-5-big-mV	7216.83	242.40	–	7199.98	545489.00	55.87
lit-as-6-big-act-mV	7202.36	1936.40	–	7200.16	322.50	98.88
lit-as-6-big-mV	7201.96	1967.20	–	7200.11	409.40	93.92

Table A.10. – Comparing the MIP and the MISDP model for the instances where both models reached the timelimit of two hours.

A.1.5. Stability constraints

We consider three different types of stability constraints: multiple loads, actuator positioning, and vibrations.

Multiple loads

Computing the optimal topology of trusses under multiple loads is not that easy. This can be seen in Table A.11. There are many instances where the MIP aborted due to numerical trouble. All other instances it was not able to solve.

	solving as MISDP			solving as MIP			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
bridge-1-3scen	345.54	2300.00	0.00	7200.01	183224.40	18.63	S
bridge-1-2scen	232.17	2299.00	0.00	7200.01	334152.80	18.35	S
bridge-1-act-2scen	797.30	2158.00	0.00	7200.01	110970.33	20.03	S
bridge-1-act-2scen-mV	696.68	2436.00	0.00	–	–	–	S
bridge-1-2scen-mV	288.33	3773.00	0.00	–	–	–	S
bridge-2-3scen	7200.14	28564.80	–	7200.01	64589.40	–	S
bridge-2-2scen	7200.08	43287.20	–	7200.01	185690.40	–	S
bridge-2-act-2scen	7200.16	13911.00	–	–	–	–	S

	solving as MISDP			solving as MIP			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
bridge-2-act-2scen-mV	1437.06	3728.00	0.00	–	–	–	S
bridge-2-2scen-mV	1349.46	13544.00	0.00	–	–	–	S
bridge-3-3scen	2670.00	10531.00	0.00	7200.01	94441.40	100.00	S
bridge-3-2scen	1861.59	10533.00	0.00	–	–	–	S
bridge-3-act-2scen	5658.01	11733.00	0.00	7200.01	129543.33	100.00	S
bridge-3-act-2scen-mV	1859.06	5119.00	0.00	–	–	–	S
bridge-3-2scen-mV	2174.15	16846.00	0.00	–	–	–	S
bridge-4-3scen	343.13	2300.00	0.00	7200.01	187769.80	19.08	S
bridge-4-2scen	228.23	2299.00	0.00	7200.01	333132.80	18.36	S
bridge-4-act-2scen	835.05	2563.00	0.00	7200.01	101006.83	18.87	S
bridge-4-act-2scen-mV	840.77	3072.00	0.00	–	–	–	S
bridge-4-2scen-mV	198.07	2592.00	0.00	–	–	–	S
bridge-5-3scen	7200.12	28478.80	–	7200.01	66658.00	–	S
bridge-5-2scen	7200.07	43462.60	–	7200.01	185630.60	–	S
bridge-5-act-2scen	7200.34	13934.60	–	–	–	–	S
bridge-5-act-2scen-mV	1961.70	5686.00	0.00	–	–	–	S
bridge-5-2scen-mV	1346.56	13544.00	0.00	–	–	–	S
bridge-6-3scen	2667.76	10531.00	0.00	7200.01	94984.80	100.00	S
bridge-6-2scen	1814.74	10533.00	0.00	–	–	–	S
bridge-6-act-2scen	6195.92	13265.00	0.00	7200.01	102860.67	100.00	S
bridge-6-act-2scen-mV	1721.03	4825.00	0.00	–	–	–	S
bridge-6-2scen-mV	1323.15	10086.00	0.00	–	–	–	S
bridge-7-3scen	7200.36	6987.20	–	7200.01	9482.20	–	S
bridge-7-2scen	7200.43	8118.20	–	7200.00	28811.00	–	S
bridge-7-act-2scen	7201.90	2465.60	–	7200.00	38306.67	–	S
bridge-7-act-2scen-mV	7201.31	3583.40	–	–	–	–	S
bridge-7-2scen-mV	536.99	1358.00	0.00	–	–	–	S
bridge-8-3scen	7200.90	2718.80	–	7199.99	16376.80	–	S
bridge-8-2scen	7201.54	2999.20	–	7200.00	26798.40	–	S
bridge-8-act-2scen	7205.50	720.60	–	–	–	–	S
bridge-8-act-2scen-mV	7203.99	1125.20	–	–	–	–	S
bridge-8-2scen-mV	7200.36	7087.20	–	–	–	–	S
bridge-9-3scen	3747.72	9217.00	0.00	7200.01	56488.20	100.00	S
bridge-9-2scen	245.43	652.00	0.00	7200.01	93841.50	77.55	S
bridge-9-act-2scen	807.06	939.00	0.00	–	–	–	S
bridge-9-act-2scen-mV	63.80	147.00	0.00	–	–	–	S
bridge-9-2scen-mV	36.98	255.00	0.00	–	–	–	S
bridge-10-3scen	7200.29	11057.80	–	7200.00	18244.00	–	S
bridge-10-2scen	7189.18	18789.00	–	7200.01	69962.40	–	S
bridge-10-act-2scen	7200.93	4566.00	–	7200.01	21967.67	–	S
bridge-10-act-2scen-mV	7200.14	25639.40	–	–	–	–	S
bridge-10-2scen-mV	7200.07	34747.00	–	–	–	–	S
canti-1-3scen	534.35	4719.00	0.00	7200.01	352081.40	76.43	S

	solving as MISDP			solving as MIP			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
canti-1-2scen	43.88	505.00	0.00	7200.01	321920.80	19.40	S
canti-1-act-2scen	177.99	655.00	0.00	7200.01	412757.50	73.41	S
canti-1-m-act-2scen-mV	0.30	1.00	–	–	–	–	S
canti-1-m-2scen-mV	0.08	1.00	–	–	–	–	S
canti-2-3scen	1767.22	10880.00	0.00	–	–	–	S
canti-2-2scen	669.06	4754.00	0.00	–	–	–	S
canti-2-act-2scen	1906.94	5111.00	0.00	–	–	–	S
canti-2-m-act-2scen-mV	333.06	923.00	0.00	–	–	–	S
canti-2-m-2scen-mV	49.13	385.00	0.00	–	–	–	S
canti-3-3scen	1062.21	4736.00	0.00	7200.01	110184.40	100.00	S
canti-3-2scen	455.97	2831.00	0.00	–	–	–	S
canti-3-act-2scen	1505.96	4274.00	0.00	7200.01	120770.17	100.00	S
canti-3-m-act-2scen-mV	49.27	121.00	0.00	–	–	–	S
canti-3-m-2scen-mV	29.16	216.00	0.00	–	–	–	S
canti-4-3scen	535.78	4719.00	0.00	7200.01	344809.00	76.39	S
canti-4-2scen	43.86	505.00	0.00	7092.60	321454.40	13.45	S
canti-4-act-2scen	151.59	523.00	0.00	7200.01	465156.50	73.90	S
canti-4-m-act-2scen-mV	0.29	1.00	–	–	–	–	S
canti-4-m-2scen-mV	0.08	1.00	–	–	–	–	S
canti-5-3scen	1772.46	10880.00	0.00	–	–	–	S
canti-5-2scen	666.57	4754.00	0.00	–	–	–	S
canti-5-act-2scen	1527.60	4151.00	0.00	–	–	–	S
canti-5-m-act-2scen-mV	192.23	528.00	0.00	–	–	–	S
canti-5-m-2scen-mV	49.14	385.00	0.00	–	–	–	S
canti-6-3scen	1070.28	4736.00	0.00	7200.01	111308.80	100.00	S
canti-6-2scen	456.76	2831.00	0.00	–	–	–	S
canti-6-act-2scen	1073.52	2517.00	0.00	7200.01	116473.17	100.00	S
canti-6-m-act-2scen-mV	70.49	178.00	0.00	–	–	–	S
canti-6-m-2scen-mV	29.23	216.00	0.00	–	–	–	S
canti-7-3scen	7200.57	7887.00	–	7200.00	19336.60	–	S
canti-7-2scen	7200.21	11006.60	–	7200.01	50469.40	–	S
canti-7-act-2scen	7201.61	3137.80	–	–	–	–	S
canti-7-act-2scen-mV	7200.53	5361.40	–	–	–	–	S
canti-7-2scen-mV	7200.12	23404.40	–	–	–	–	S
canti-9-3scen	7200.38	6472.40	–	7200.01	29571.80	–	S
canti-9-2scen	7200.44	8963.40	–	7200.01	27386.00	–	S
canti-9-act-2scen	7201.73	2481.20	–	7200.00	47544.50	–	S
canti-9-act-2scen-mV	7200.99	4291.60	–	–	–	–	S
canti-9-2scen-mV	7200.16	17481.80	–	–	–	–	S
canti-10-3scen	7200.39	10192.80	–	7200.00	34682.80	–	S
canti-10-2scen	7200.21	12704.80	–	–	–	–	S
canti-10-act-2scen	7200.78	4654.60	–	7200.00	65428.17	–	S
canti-10-act-2scen-mV	7200.60	6881.40	–	–	–	–	S

	solving as MISDP			solving as MIP			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
canti-10-2scen-mV	7200.17	25858.20	–	–	–	–	S
canti-big-3scen	7201.23	3037.60	–	7200.00	13843.20	–	S
canti-big-2scen	7201.09	3618.80	–	7200.02	33119.80	–	S
canti-big-act-2scen	7204.02	783.60	–	7200.02	4908.50	–	S
canti-big-act-2scen-mV	7201.39	1520.00	–	–	–	–	S
canti-big-2scen-mV	7200.50	7355.60	–	–	–	–	S

Table A.11. – Comparing the MIP and the MISDP model for the truss instances with more than one scneario.

Positioning actuators

	solving as MISDP			solving as MIP			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
bridge-1-act	20.72	130.00	0.00	7200.01	1235610.00	1.19	S
bridge-1-act-mV	355.10	2576.00	0.00	1.30	1278.00	0.00	M
bridge-1-act-2scen	797.30	2158.00	0.00	7200.01	110970.33	20.03	S
bridge-1-act-2scen-mV	696.68	2436.00	0.00	–	–	–	S
bridge-2-act	284.27	1124.00	0.00	7200.01	819100.00	56.54	S
bridge-2-act-mV	7.55	29.00	0.00	2.83	1948.00	0.00	M
bridge-2-act-2scen	7200.16	13911.00	–	–	–	–	–
bridge-2-act-2scen-mV	1437.06	3728.00	0.00	–	–	–	S
bridge-3-act	190.01	712.00	0.00	7200.02	1256853.20	46.20	S
bridge-3-act-mV	53.05	247.00	0.00	2.83	1948.00	0.00	M
bridge-3-act-2scen	5658.01	11733.00	0.00	7200.01	129543.33	100.00	S
bridge-3-act-2scen-mV	1859.06	5119.00	0.00	–	–	–	S
bridge-4-act	20.78	114.00	0.00	2734.56	414885.00	0.01	S
bridge-4-act-mV	12.62	81.00	0.00	2.46	1375.00	0.01	M
bridge-4-act-2scen	835.05	2563.00	0.00	7200.01	101006.83	18.87	S
bridge-4-act-2scen-mV	840.77	3072.00	0.00	–	–	–	S
bridge-5-act	293.04	1179.00	0.00	7200.01	724078.67	61.16	S
bridge-5-act-mV	7.04	27.00	0.00	7.97	3949.00	0.01	S
bridge-5-act-2scen	7200.34	13934.60	–	–	–	–	–
bridge-5-act-2scen-mV	1961.70	5686.00	0.00	–	–	–	S
bridge-6-act	209.19	800.00	0.00	7200.01	1033619.20	43.55	S
bridge-6-act-mV	124.55	584.00	0.00	7.99	3949.00	0.01	M
bridge-6-act-2scen	6195.92	13265.00	0.00	7200.01	102860.67	100.00	S
bridge-6-act-2scen-mV	1721.03	4825.00	0.00	–	–	–	S
bridge-7-act	5908.73	4164.00	0.00	7200.01	1838213.80	75.32	S
bridge-7-act-mV	311.16	260.00	0.00	12.75	2578.00	0.00	M
bridge-7-act-2scen	7201.90	2465.60	–	7200.00	38306.67	–	–

	solving as MISDP			solving as MIP			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
bridge-7-act-2scen-mV	7201.31	3583.40	–	–	–	–	–
bridge-8-act	7201.87	1313.80	–	1883.15	181138.00	–	–
bridge-8-act-mV	7201.33	2186.00	–	7200.00	1044849.80	–	–
bridge-8-act-2scen	7205.50	720.60	–	–	–	–	–
bridge-8-act-2scen-mV	7203.99	1125.20	–	–	–	–	–
bridge-9-act	388.64	1070.00	0.00	7200.01	1204261.60	58.24	S
bridge-9-act-mV	74.36	290.00	0.00	1.68	829.00	0.00	M
bridge-9-act-2scen	807.06	939.00	0.00	–	–	–	S
bridge-9-act-2scen-mV	63.80	147.00	0.00	–	–	–	S
bridge-10-act	720.74	861.00	0.00	7200.02	826800.60	82.31	S
bridge-10-act-mV	7200.18	45895.80	51.60	143.22	56589.00	0.01	M
bridge-10-act-2scen	7200.93	4566.00	–	7200.01	21967.67	–	–
bridge-10-act-2scen-mV	7200.14	25639.40	–	–	–	–	–
canti-1-act	106.06	723.00	0.00	214.16	46889.00	0.01	S
canti-1-act-2scen	177.99	655.00	0.00	7200.01	412757.50	73.41	S
canti-1-m-act-mV	9.57	73.00	0.00	3.90	4111.00	0.00	M
canti-2-act	182.73	1629.00	0.00	7200.01	736087.60	38.73	S
canti-2-act-2scen	1906.94	5111.00	0.00	–	–	–	S
canti-2-m-act-mV	28.81	166.00	0.00	5.77	4118.00	0.00	M
canti-2-m-act-2scen-mV	333.06	923.00	0.00	–	–	–	S
canti-3-act	256.42	993.00	0.00	7200.01	925988.80	25.20	S
canti-3-act-2scen	1505.96	4274.00	0.00	7200.01	120770.17	100.00	S
canti-3-m-act-mV	9.64	52.00	0.00	5.76	4118.00	0.00	M
canti-3-m-act-2scen-mV	49.27	121.00	0.00	–	–	–	S
canti-4-act	58.65	360.00	0.00	241.24	60354.00	0.01	S
canti-4-act-2scen	151.59	523.00	0.00	7200.01	465156.50	73.90	S
canti-4-m-act-mV	16.41	131.00	0.00	1.94	2475.00	0.00	M
canti-5-act	191.21	1655.00	0.00	7200.02	641635.60	41.39	S
canti-5-act-2scen	1527.60	4151.00	0.00	–	–	–	S
canti-5-m-act-mV	24.46	196.00	0.00	4.51	3683.00	0.01	M
canti-5-m-act-2scen-mV	192.23	528.00	0.00	–	–	–	S
canti-6-act	212.51	781.00	0.00	7200.02	573254.60	36.08	S
canti-6-act-2scen	1073.52	2517.00	0.00	7200.01	116473.17	100.00	S
canti-6-m-act-mV	13.45	72.00	0.00	4.47	3683.00	0.01	M
canti-6-m-act-2scen-mV	70.49	178.00	0.00	–	–	–	S
canti-7-act	658.47	495.00	0.00	7200.01	883536.40	78.87	S
canti-7-act-mV	7200.46	10477.60	–	902.51	256354.00	0.01	M
canti-7-act-2scen	7201.61	3137.80	–	–	–	–	–
canti-7-act-2scen-mV	7200.53	5361.40	–	–	–	–	–
canti-9-act	1703.01	1189.00	0.00	7200.01	982240.40	75.35	S
canti-9-act-mV	7200.81	9526.60	–	2137.02	559311.00	0.01	M
canti-9-act-2scen	7201.73	2481.20	–	7200.00	47544.50	–	–
canti-9-act-2scen-mV	7200.99	4291.60	–	–	–	–	–

	solving as MISDP			solving as MIP			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
canti-10-act	888.75	1093.00	0.00	7200.01	828881.80	56.41	S
canti-10-act-mV	7200.09	38623.20	–	86.59	34390.00	0.00	M
canti-10-act-2scen	7200.78	4654.60	–	7200.00	65428.17	–	–
canti-10-act-2scen-mV	7200.60	6881.40	–	–	–	–	–
canti-big-act	5409.97	2225.00	0.00	7200.00	1120412.60	83.44	S
canti-big-act-mV	7201.02	3169.60	–	3841.72	542136.00	0.01	M
canti-big-act-2scen	7204.02	783.60	–	7200.02	4908.50	–	–
canti-big-act-2scen-mV	7201.39	1520.00	–	–	–	–	–

Table A.12. – Comparing MIP and MISDP model for the instances with actuator positioning.

Vibrations

The detailed results for using vibration constraints compared to the model without considering vibrations are presented in Table A.13. In this table we also present the value of the optimal solution because considering vibrations can change this value. A comparison to the MIP is not possible because vibrations cannot be modeled using only linear constraints.

	using vibration constraints				without considering vibrations			
			gap	optimal			gap	optimal
instance	time	nodes	[%]	value	time	nodes	[%]	value
bridge-1	215.15	2446.00	0.00	1.65	7.00	132.00	0.00	1.55
bridge-1-mV	117.43	1601.00	0.00	23.61	38.78	932.00	0.00	23.43
bridge-1-cont	0.15	1.00	0.00	1.05	0.08	1.00	0.00	1.04
bridge-1-act	624.36	2713.00	0.00	1.55	20.33	130.00	0.00	1.44
bridge-1-2scen	363.94	2691.00	0.00	1.67	265.19	2786.00	0.00	1.67
bridge-2	3526.60	21329.00	0.00	1.42	134.94	1158.00	0.00	1.17
bridge-2-mV	609.77	6081.00	0.00	23.96	75.93	1096.00	0.00	22.90
bridge-2-cont	0.15	1.00	0.00	1.05	0.08	1.00	0.00	1.04
bridge-2-act	5775.76	18038.00	0.00	1.31	288.50	1177.00	0.00	1.10
bridge-2-2scen	7200.18	28077.00	–	–	7200.08	39900.60	906.85	15.48
bridge-3	580.39	3216.00	0.00	0.88	44.21	415.00	0.00	0.85
bridge-3-mV	2111.13	16959.00	0.00	34.44	90.72	1071.00	0.00	33.14
bridge-3-cont	0.16	1.00	0.00	0.69	0.10	1.00	0.00	0.68
bridge-3-act	1069.90	3330.00	0.00	0.81	184.18	724.00	0.00	0.80
bridge-3-2scen	3522.83	14156.00	0.00	0.92	2519.36	14388.00	0.00	0.92
bridge-4	214.01	2446.00	0.00	1.65	7.08	132.00	0.00	1.55
bridge-4-mV	100.20	1365.00	0.00	23.79	26.46	618.00	0.00	23.43
bridge-4-cont	0.15	1.00	0.00	1.05	0.08	1.00	0.00	1.04
bridge-4-act	630.17	2731.00	0.00	1.52	22.73	136.00	0.00	1.42
bridge-4-2scen	363.75	2691.00	0.00	1.67	264.95	2786.00	0.00	1.67
bridge-5	3526.70	21329.00	0.00	1.42	133.85	1158.00	0.00	1.17
bridge-5-mV	608.39	6081.00	0.00	23.96	75.48	1096.00	0.00	22.90

	using vibration constraints				without considering vibrations			
			gap	optimal			gap	optimal
instance	time	nodes	[%]	value	time	nodes	[%]	value
bridge-5-cont	0.16	1.00	0.00	1.05	0.08	1.00	0.00	1.04
bridge-5-act	5598.70	17834.00	0.00	1.29	292.69	1179.00	0.00	1.08
bridge-5-2scen	7200.17	28118.80	–	–	7200.13	39956.60	906.73	15.48
bridge-6	580.32	3216.00	0.00	0.88	44.44	415.00	0.00	0.85
bridge-6-mV	1652.06	13230.00	0.00	34.61	47.41	548.00	0.00	33.14
bridge-6-cont	0.16	1.00	0.00	0.69	0.10	1.00	0.00	0.68
bridge-6-act	740.20	2216.00	0.00	0.79	212.02	800.00	0.00	0.78
bridge-6-2scen	3516.71	14156.00	0.00	0.92	2522.14	14388.00	0.00	0.92
bridge-7	7200.53	9024.20	–	–	1683.93	3520.00	0.00	0.64
bridge-7-mV	1605.62	5022.00	0.00	14.90	20.67	49.00	0.00	12.49
bridge-7-cont	0.36	1.00	0.00	0.51	0.17	1.00	0.00	0.51
bridge-7-act	7201.34	4020.00	–	–	5841.99	4455.00	0.00	0.58
bridge-7-2scen	7200.50	6266.00	–	–	7200.27	8095.80	–	–
bridge-8	7200.70	3555.00	–	–	7200.47	4664.60	–	–
bridge-8-mV	7200.38	6576.80	–	–	2204.14	3085.00	0.00	19.31
bridge-8-cont	0.51	1.00	0.00	1.82	0.29	1.00	0.00	1.82
bridge-8-act	7202.61	1382.40	–	–	7202.03	1411.20	–	–
bridge-8-2scen	7200.99	2639.60	–	–	7201.22	3079.00	–	–
bridge-9	56.62	163.00	0.00	0.17	76.64	435.00	0.00	0.17
bridge-9-mV	70.56	485.00	0.00	4.83	45.28	498.00	0.00	4.83
bridge-9-cont	0.16	1.00	0.00	0.13	0.10	1.00	0.00	0.13
bridge-9-act	463.96	1008.00	0.00	0.14	363.05	1001.00	0.00	0.14
bridge-9-2scen	531.70	1286.00	0.00	0.21	376.73	1249.00	0.00	0.21
bridge-10	1520.34	5371.00	0.00	0.24	1081.06	6742.00	0.00	0.24
bridge-10-mV	7200.06	28065.00	192.61	20.62	7200.06	61409.83	214.51	20.62
bridge-10-cont	0.24	1.00	0.00	0.22	0.13	1.00	0.00	0.22
bridge-10-act	1292.84	1810.00	0.00	0.16	968.78	1791.00	0.00	0.16
bridge-10-2scen	6308.62	10734.00	0.00	0.56	7200.18	18818.80	0.18	0.56
bridge-big	7201.04	3460.40	–	–	7200.83	4854.20	–	–
bridge-big-cont	0.78	1.00	0.00	2.87	0.51	1.00	0.00	2.87
bridge-big-act	7205.44	905.40	–	–	7203.62	875.20	–	–
bridge-big-2scen	7201.46	2558.00	–	–	7201.16	3221.20	–	–
canti-1-m	70.46	893.00	0.00	0.34	29.96	892.00	0.00	0.34
canti-1-mV	5.92	86.00	0.00	19.45	4.57	123.00	0.00	13.97
canti-1-cont	0.13	1.00	0.00	0.23	0.08	1.00	0.00	0.23
canti-1-act	219.74	1028.00	0.00	0.26	174.47	1153.00	0.00	0.26
canti-1-2scen	72.99	595.00	0.00	1.43	50.95	606.00	0.00	1.43
canti-2-m	103.23	701.00	0.00	0.28	63.44	777.00	0.00	0.28
canti-2-mV	17.53	154.00	0.00	15.65	9.30	124.00	0.00	12.74
canti-2-cont	0.12	1.00	0.00	0.23	0.08	1.00	0.00	0.23
canti-2-act	219.58	1083.00	0.00	0.21	188.59	1629.00	0.00	0.21
canti-2-2scen	1022.01	4975.00	0.00	1.43	688.81	4849.00	0.00	1.43

	using vibration constraints				without considering vibrations			
			gap	optimal			gap	optimal
instance	time	nodes	[%]	value	time	nodes	[%]	value
canti-3-m	284.06	1550.00	0.00	0.19	143.96	1371.00	0.00	0.19
canti-3-mV	6.36	51.00	0.00	21.65	9.72	136.83	0.00	16.65
canti-3-cont	0.13	1.00	0.00	0.16	0.08	1.00	0.00	0.16
canti-3-act	483.64	1495.00	0.00	0.15	265.58	993.00	0.00	0.15
canti-3-2scen	743.96	3231.00	0.00	0.86	535.94	3427.00	0.00	0.86
canti-4-m	70.10	893.00	0.00	0.34	29.82	892.00	0.00	0.34
canti-4-mV	5.93	86.00	0.00	19.45	4.33	116.83	0.00	14.89
canti-4-cont	0.13	1.00	0.00	0.23	0.08	1.00	0.00	0.23
canti-4-act	134.73	575.00	0.00	0.24	89.06	516.00	0.00	0.24
canti-4-2scen	72.64	595.00	0.00	1.43	50.97	606.00	0.00	1.43
canti-5-m	103.16	701.00	0.00	0.28	62.88	777.00	0.00	0.28
canti-5-mV	17.43	154.00	0.00	15.65	9.17	124.00	0.00	12.74
canti-5-cont	0.13	1.00	0.00	0.23	0.08	1.00	0.00	0.23
canti-5-act	235.15	1149.00	0.00	0.20	194.91	1655.00	0.00	0.20
canti-5-2scen	1019.45	4975.00	0.00	1.43	690.03	4849.00	0.00	1.43
canti-6-m	282.83	1550.00	0.00	0.19	144.11	1371.00	0.00	0.19
canti-6-mV	6.39	51.00	0.00	21.65	9.75	136.83	0.00	16.65
canti-6-cont	0.13	1.00	0.00	0.16	0.08	1.00	0.00	0.16
canti-6-act	372.77	1086.00	0.00	0.14	220.64	795.00	0.00	0.14
canti-6-2scen	745.07	3231.00	0.00	0.86	532.63	3427.00	0.00	0.86
canti-7	672.25	1040.00	0.00	0.43	645.81	1692.00	0.00	0.43
canti-7-mV	7200.22	20339.00	–	–	7200.06	32501.67	32.44	12.49
canti-7-cont	0.25	1.00	0.00	0.40	0.17	1.00	0.00	0.40
canti-7-act	1038.45	738.00	0.00	0.34	885.72	737.00	0.00	0.34
canti-7-2scen	7200.49	7978.60	–	–	7200.50	10078.63	–	–
canti-9	7200.30	12501.40	8.76	0.68	291.40	700.00	0.00	0.59
canti-9-mV	7200.12	21590.80	21.07	12.93	5095.96	26591.00	0.00	10.93
canti-9-cont	0.31	1.00	0.00	0.57	0.17	1.00	0.00	0.57
canti-9-act	7200.97	4514.20	–	–	7200.82	4849.00	–	–
canti-9-2scen	7200.50	6470.80	–	–	7200.49	8449.63	554.03	4.82
canti-10	1365.84	2560.00	0.00	0.19	982.20	2712.00	0.00	0.19
canti-10-mV	7200.15	27974.00	–	–	7200.05	49350.67	–	–
canti-10-cont	0.26	1.00	0.00	0.14	0.15	1.00	0.00	0.14
canti-10-act	2979.77	2887.00	0.00	0.16	2650.48	3100.00	0.00	0.16
canti-10-2scen	7200.45	9792.20	–	–	7200.26	12621.88	421.04	1.77
canti-big	2515.46	2795.00	0.00	0.37	1257.19	1590.00	0.00	0.37
canti-big-cont	0.53	1.00	0.00	0.35	0.32	1.00	0.00	0.35
canti-big-act	6756.89	2277.00	0.00	0.31	5898.78	2225.00	0.00	0.31

Table A.13. – Comparing solving times and solution values for trusses with and without vibration constraints.

A.1.6. Statistics

Finally we present the different tables for the statistics of our five solving runs. For each model, the MIP and the MISDP, we have three different tables, one for time, nodes, and gap respectively. In each table we show the average, the minimum and maximum and the range between minimum and maximum. Additionally, we present the percentage value of the range with respect to the maximum. We show these results for the MIP in Tables A.14, A.15, and A.16. The MISDP runs are presented in Tables A.17, A.18, and A.19.

instance	average time	min. time	max. time	range	[%]
bridge-1	1565.34	1358.26	2089.22	730.96	34.99
bridge-1-act-mV	1.30	1.25	1.34	0.09	6.72
bridge-1-mV	0.38	0.37	0.40	0.03	7.50
bridge-2-act-mV	2.83	2.72	2.93	0.21	7.17
bridge-2-mV	2.84	2.79	2.94	0.15	5.10
bridge-3-act-mV	2.83	2.73	2.93	0.20	6.83
bridge-3-mV	2.83	2.70	2.93	0.23	7.85
bridge-4	1450.93	1177.57	1873.90	696.33	37.16
bridge-4-act	2734.56	2491.59	3225.42	733.83	22.75
bridge-4-act-mV	2.46	2.40	2.52	0.12	4.76
bridge-4-mV	0.38	0.35	0.40	0.05	12.50
bridge-5-act-mV	7.97	7.57	8.43	0.86	10.20
bridge-5-mV	2.79	2.68	2.91	0.23	7.90
bridge-6-act-mV	7.99	7.84	8.34	0.50	6.00
bridge-6-mV	2.84	2.72	2.93	0.21	7.17
bridge-7-act-mV	12.75	12.08	13.68	1.60	11.70
bridge-7-mV	4.73	4.36	5.06	0.70	13.83
bridge-8-act	1883.15	1774.33	2037.50	263.17	12.92
bridge-9-act-mV	1.68	1.56	1.87	0.31	16.58
bridge-9-mV	0.19	0.17	0.20	0.03	15.00
bridge-10-act-mV	143.22	137.10	150.41	13.31	8.85
bridge-10-mV	66.95	64.40	68.80	4.40	6.40
bridge-big	348.68	298.48	454.21	155.73	34.29
canti-1	84.39	63.37	120.38	57.01	47.36
canti-1-act	214.16	171.79	243.53	71.74	29.46
canti-1-act-mV	3.98	3.90	4.04	0.14	3.47
canti-1-mV	0.68	0.64	0.71	0.07	9.86
canti-1-m-act-mV	3.90	3.84	3.98	0.14	3.52
canti-1-m-mV	0.67	0.65	0.70	0.05	7.14
canti-2	3404.95	3259.58	3702.06	442.48	11.95
canti-2-act-mV	5.83	5.62	6.00	0.38	6.33
canti-2-mV	2.06	1.97	2.14	0.17	7.94
canti-2-m-act-mV	5.77	5.60	5.98	0.38	6.35
canti-2-m-mV	2.07	2.02	2.14	0.12	5.61
canti-3-act-mV	5.85	5.66	6.02	0.36	5.98

instance	average time	min. time	max. time	range	[%]
canti-3-mV	2.10	1.99	2.20	0.21	9.55
canti-3-m-act-mV	5.76	5.56	6.01	0.45	7.49
canti-3-m-mV	2.06	2.02	2.11	0.09	4.27
canti-4	79.53	62.77	119.00	56.23	47.25
canti-4-act	241.24	202.57	271.03	68.46	25.26
canti-4-2scen	7092.60	6764.25	7200.01	435.76	6.05
canti-4-act-mV	2.01	1.93	2.07	0.14	6.76
canti-4-mV	0.69	0.66	0.71	0.05	7.04
canti-4-m-act-mV	1.94	1.89	2.02	0.13	6.44
canti-4-m-mV	0.67	0.64	0.70	0.06	8.57
canti-5	3371.27	3300.42	3497.26	196.84	5.63
canti-5-act-mV	4.59	4.50	4.69	0.19	4.05
canti-5-mV	2.12	1.97	2.25	0.28	12.44
canti-5-m-act-mV	4.51	4.36	4.66	0.30	6.44
canti-5-m-mV	2.06	2.01	2.14	0.13	6.07
canti-6-act-mV	4.57	4.44	4.73	0.29	6.13
canti-6-mV	2.14	2.01	2.27	0.26	11.45
canti-6-m-act-mV	4.47	4.37	4.59	0.22	4.79
canti-6-m-mV	2.05	2.02	2.14	0.12	5.61
canti-7-act-mV	902.51	852.03	968.45	116.42	12.02
canti-7-mV	187.76	178.01	197.40	19.39	9.82
canti-8-act-mV	0.58	0.57	0.60	0.03	5.00
canti-8-mV	0.62	0.56	0.72	0.16	22.22
canti-9-act-mV	2137.02	2011.75	2269.39	257.64	11.35
canti-9-mV	657.35	648.64	666.23	17.59	2.64
canti-10-act-mV	86.59	83.98	87.73	3.75	4.27
canti-10-mV	18.87	18.21	19.50	1.29	6.62
canti-big-act-mV	3841.72	3734.79	3996.85	262.06	6.56
canti-big-mV	666.29	650.00	678.27	28.27	4.17
lit-as-1	0.81	0.76	0.89	0.13	14.61
lit-as-1-act	0.83	0.75	0.89	0.14	15.73
lit-as-1-act-mV	0.61	0.59	0.63	0.04	6.35
lit-as-1-mV	0.59	0.56	0.61	0.05	8.20
lit-as-2	6515.08	5564.31	7200.01	1635.70	22.72
lit-as-2-act	6855.77	6383.47	7200.01	816.54	11.34
lit-as-6	5.96	5.52	6.30	0.78	12.38
lit-as-6-act	5.86	5.62	6.00	0.38	6.33
lit-s-1-act-mV	6.86	6.72	7.03	0.31	4.41
lit-s-1-mV	7.29	6.72	7.73	1.01	13.07

Table A.14. – Statistics about the solving processes for the MIP for the instances that where solved to optimality.

	nodes				
instance	average	min.	max.	range	[%]
bridge-1-act	1235610.00	1049154	1404272	355118	25.29
bridge-1-3scen	183224.40	172507	191177	18670	9.77
bridge-1-2scen	334152.80	328716	340588	11872	3.49
bridge-1-act-2scen	110970.33	107700	113355	5655	4.99
bridge-2	3995826.20	3510500	4184100	673600	16.10
bridge-2-act	819100.00	754090	895619	141529	15.80
bridge-2-3scen	64589.40	60700	67163	6463	9.62
bridge-2-2scen	185690.40	180070	189284	9214	4.87
bridge-3	1861801.20	1550984	1955903	404919	20.70
bridge-3-act	1256853.20	1205500	1295534	90034	6.95
bridge-3-3scen	94441.40	93664	95825	2161	2.26
bridge-3-act-2scen	129543.33	124097	134460	10363	7.71
bridge-4-3scen	187769.80	183073	190048	6975	3.67
bridge-4-2scen	333132.80	322940	339673	16733	4.93
bridge-4-act-2scen	101006.83	98215	102218	4003	3.92
bridge-5	4033469.40	3565733	4277454	711721	16.64
bridge-5-act	724078.67	696173	753436	57263	7.60
bridge-5-3scen	66658.00	65419	67677	2258	3.34
bridge-5-2scen	185630.60	181874	188629	6755	3.58
bridge-6	1948596.20	1697163	2155609	458446	21.27
bridge-6-act	1033619.20	988659	1086891	98232	9.04
bridge-6-3scen	94984.80	93781	96300	2519	2.62
bridge-6-act-2scen	102860.67	100722	106333	5611	5.28
bridge-7	3434977.60	3089528	3642800	553272	15.19
bridge-7-act	1838213.80	1744730	1918059	173329	9.04
bridge-7-2scen	28811.00	26600	30592	3992	13.05
bridge-7-act-2scen	38306.67	37570	38988	1418	3.64
bridge-8	1877770.20	1616479	2011702	395223	19.65
bridge-8-act-mV	1044849.80	1022648	1075541	52893	4.92
bridge-8-mV	1677617.60	1603282	1722096	118814	6.90
bridge-9	1916343.20	1732033	1990041	258008	12.96
bridge-9-act	1204261.60	1151398	1231100	79702	6.47
bridge-9-3scen	56488.20	54987	58474	3487	5.96
bridge-9-2scen	93841.50	92193	95130	2937	3.09
bridge-10	3176919.20	2910039	3317188	407149	12.27
bridge-10-act	826800.60	766779	865178	98399	11.37
bridge-10-3scen	18244.00	18107	18373	266	1.45
bridge-10-act-2scen	21967.67	20387	22915	2528	11.03
bridge-big-act	1024065.80	943104	1101100	157996	14.35
bridge-big-act-2scen	2701.50	2300	3017	717	23.77
canti-1-3scen	352081.40	342462	358142	15680	4.38
canti-1-2scen	321920.80	316778	328612	11834	3.60
canti-1-act-2scen	412757.50	400700	420138	19438	4.63

	nodes				
instance	average	min.	max.	range	[%]
canti-2-act	736087.60	659309	828242	168933	20.40
canti-3	1746646.60	1241000	1995421	754421	37.81
canti-3-act	925988.80	867861	985778	117917	11.96
canti-3-3scen	110184.40	107500	112858	5358	4.75
canti-3-act-2scen	120770.17	115900	123873	7973	6.44
canti-4-3scen	344809.00	335876	355329	19453	5.47
canti-4-act-2scen	465156.50	452160	490080	37920	7.74
canti-5-act	641635.60	565999	700780	134781	19.23
canti-6	1725921.60	1294483	1875703	581220	30.99
canti-6-act	573254.60	509472	640532	131060	20.46
canti-6-3scen	111308.80	108000	115357	7357	6.38
canti-6-act-2scen	116473.17	113600	118880	5280	4.44
canti-7	2738881.75	2527931	2877810	349879	12.16
canti-7-act	883536.40	860127	911637	51510	5.65
canti-7-3scen	19336.60	18417	20381	1964	9.64
canti-7-2scen	50469.40	47742	52509	4767	9.08
canti-8	539359.80	484820	571376	86556	15.15
canti-8-act	574799.60	533100	604300	71200	11.78
canti-9	2980492.50	2751701	3112936	361235	11.60
canti-9-act	982240.40	962367	1004000	41633	4.15
canti-9-3scen	29571.80	28500	30135	1635	5.43
canti-9-2scen	27386.00	25280	29811	4531	15.20
canti-10	2933699.75	2504600	3202200	697600	21.79
canti-10-act	828881.80	782105	854041	71936	8.42
canti-10-3scen	34682.80	34221	35064	843	2.40
canti-10-act-2scen	65428.17	63868	67038	3170	4.73
canti-big	2251844.25	2059055	2363200	304145	12.87
canti-big-act	1120412.60	1007247	1226140	218893	17.85
canti-big-2scen	33119.80	31100	35235	4135	11.74
canti-big-act-2scen	4908.50	4443	5317	874	16.44
lit-as-2-act-mV	331254.75	326905	335761	8856	2.64
lit-as-2-mV	330550.00	320867	342047	21180	6.19
lit-as-2-big-act-mV	122778.50	117566	128076	10510	8.21
lit-as-2-big-mV	123304.00	110108	135706	25598	18.86
lit-as-5	391282.80	332800	442761	109961	24.84
lit-as-5-act	411888.40	350344	436200	85856	19.68
lit-as-5-act-mV	1730289.00	1714074	1753126	39052	2.23
lit-as-5-mV	1707782.60	1656121	1785727	129606	7.26
lit-as-5-big	4311.00	3293	5095	1802	35.37
lit-as-5-big-act	3913.20	3081	4278	1197	27.98
lit-as-5-big-act-mV	545305.25	514323	568346	54023	9.51
lit-as-5-big-mV	545489.00	494019	566050	72031	12.73
lit-as-6-big	2952.20	2731	3226	495	15.34

	nodes				
instance	average	min.	max.	range	[%]
lit-as-6-big-act	3244.25	2873	3464	591	17.06
lit-as-6-big-act-mV	322.50	320	325	5	1.54
lit-as-6-big-mV	409.40	322	500	178	35.60
lit-s-1	974901.20	834469	1107400	272931	24.65
lit-s-1-act	944504.20	762107	1075280	313173	29.12

Table A.15. – Some statistics for the nodes about the solving processes for the MIP for the instances that reached the time limit.

	gap [%]				
instance	aver.	min.	max.	range	[%]
bridge-1-act	1.19	0.18	2.23	2.05	91.93
bridge-1-3scen	18.63	16.71	19.42	2.71	13.95
bridge-1-2scen	18.35	18.32	18.38	0.06	0.33
bridge-1-act-2scen	20.03	17.15	23.85	6.70	28.09
bridge-2	53.07	52.35	54.19	1.84	3.40
bridge-2-act	56.54	54.92	58.05	3.13	5.39
bridge-2-3scen	75.36	73.47	78.85	5.38	6.82
bridge-2-2scen	45.45	34.65	53.03	18.38	34.66
bridge-3	44.47	43.19	45.66	2.47	5.41
bridge-3-act	46.20	46.08	46.37	0.29	0.63
bridge-3-3scen	100.00	100.00	100.00	0.00	0.00
bridge-3-act-2scen	100.00	100.00	100.00	0.00	0.00
bridge-4-3scen	19.08	18.32	19.42	1.10	5.66
bridge-4-2scen	18.36	18.32	18.40	0.08	0.43
bridge-4-act-2scen	18.87	17.19	20.42	3.23	15.82
bridge-5	53.72	52.97	54.48	1.51	2.77
bridge-5-act	61.16	60.99	61.33	0.34	0.55
bridge-5-3scen	76.72	75.79	78.85	3.06	3.88
bridge-5-2scen	51.34	44.57	53.04	8.47	15.97
bridge-6	45.62	43.92	46.75	2.83	6.05
bridge-6-act	43.55	42.34	44.65	2.31	5.17
bridge-6-3scen	100.00	100.00	100.00	0.00	0.00
bridge-6-act-2scen	100.00	100.00	100.00	0.00	0.00
bridge-7	76.18	74.24	78.21	3.97	5.08
bridge-7-act	75.32	74.65	76.65	2.00	2.61
bridge-7-2scen	86.45	84.41	87.51	3.10	3.54
bridge-7-act-2scen	100.00	100.00	100.00	0.00	0.00
bridge-8	89.59	86.75	92.95	6.20	6.67
bridge-8-act-mV	51.88	48.93	62.17	13.24	21.30
bridge-8-mV	43.32	41.92	48.14	6.22	12.92
bridge-9	54.61	53.27	56.12	2.85	5.08

	gap [%]				
instance	aver.	min.	max.	range	[%]
bridge-9-act	58.24	54.88	61.67	6.79	11.01
bridge-9-3scen	100.00	100.00	100.00	0.00	0.00
bridge-9-2scen	77.55	71.77	81.78	10.01	12.24
bridge-10	80.61	80.44	80.81	0.37	0.46
bridge-10-act	82.31	81.13	82.96	1.83	2.21
bridge-10-3scen	93.14	91.67	98.62	6.95	7.05
bridge-10-act-2scen	90.09	84.77	92.40	7.63	8.26
bridge-big-act	77.90	75.77	82.54	6.77	8.20
bridge-big-act-2scen	89.35	88.75	89.64	0.89	0.99
canti-1-3scen	76.43	75.96	77.60	1.64	2.11
canti-1-2scen	19.40	16.82	21.04	4.22	20.06
canti-1-act-2scen	73.41	72.85	74.33	1.48	1.99
canti-2-act	38.73	34.04	42.46	8.42	19.83
canti-3	34.00	32.04	37.24	5.20	13.96
canti-3-act	25.20	21.81	27.04	5.23	19.34
canti-3-3scen	100.00	100.00	100.00	0.00	0.00
canti-3-act-2scen	100.00	100.00	100.00	0.00	0.00
canti-4-3scen	76.39	75.98	76.89	0.91	1.18
canti-4-act-2scen	73.90	73.11	76.14	3.03	3.98
canti-5-act	41.39	40.84	42.13	1.29	3.06
canti-6	33.27	28.64	37.22	8.58	23.05
canti-6-act	36.08	35.59	36.58	0.99	2.71
canti-6-3scen	100.00	100.00	100.00	0.00	0.00
canti-6-act-2scen	100.00	100.00	100.00	0.00	0.00
canti-7	76.90	76.63	77.04	0.41	0.53
canti-7-act	78.87	78.04	79.68	1.64	2.06
canti-7-3scen	94.31	91.67	98.07	6.40	6.53
canti-7-2scen	89.97	88.84	91.36	2.52	2.76
canti-8	0.14	0.14	0.15	0.01	6.67
canti-8-act	0.13	0.13	0.14	0.01	7.14
canti-9	73.06	72.87	73.19	0.32	0.44
canti-9-act	75.35	75.29	75.40	0.11	0.15
canti-9-3scen	100.00	100.00	100.00	0.00	0.00
canti-9-2scen	93.70	88.59	95.07	6.48	6.82
canti-10	57.86	57.07	58.84	1.77	3.01
canti-10-act	56.41	55.56	58.39	2.83	4.85
canti-10-3scen	100.00	100.00	100.00	0.00	0.00
canti-10-act-2scen	100.00	100.00	100.00	0.00	0.00
canti-big	85.04	84.83	85.24	0.41	0.48
canti-big-act	83.44	82.29	85.27	2.98	3.49
canti-big-2scen	100.00	100.00	100.00	0.00	0.00
canti-big-act-2scen	93.86	92.98	94.72	1.74	1.84
lit-as-2-act-mV	95.61	95.42	95.72	0.30	0.31

	gap [%]				
instance	aver.	min.	max.	range	[%]
lit-as-2-mV	95.51	95.31	95.72	0.41	0.43
lit-as-2-big-act-mV	96.29	95.64	96.61	0.97	1.00
lit-as-2-big-mV	96.27	96.02	96.50	0.48	0.50
lit-as-5	85.71	83.99	87.47	3.48	3.98
lit-as-5-act	87.09	85.62	88.53	2.91	3.29
lit-as-5-act-mV	34.89	32.24	36.60	4.36	11.91
lit-as-5-mV	34.55	32.36	36.17	3.81	10.53
lit-as-5-big	99.20	99.20	99.20	0.00	0.00
lit-as-5-big-act	99.20	99.20	99.20	0.00	0.00
lit-as-5-big-act-mV	51.89	51.72	52.17	0.45	0.86
lit-as-5-big-mV	55.87	53.39	65.40	12.01	18.36
lit-as-6-big	94.65	94.39	94.85	0.46	0.48
lit-as-6-big-act	94.52	94.31	94.85	0.54	0.57
lit-as-6-big-act-mV	98.88	98.88	98.88	0.00	0.00
lit-as-6-big-mV	93.92	90.36	99.09	8.73	8.81
lit-s-1	66.99	64.67	69.50	4.83	6.95
lit-s-1-act	66.20	63.41	69.71	6.30	9.04

Table A.16. – Some statistics for the gap about the solving processes for the MIP for the instances that reached the time limit.

instance	average time	min. time	max. time	range	[%]
bridge-1	5.04	5.01	5.06	0.05	0.99
bridge-1-act	20.72	20.44	21.12	0.68	3.22
bridge-1-3scen	345.54	343.49	347.00	3.51	1.01
bridge-1-2scen	232.17	228.73	234.10	5.37	2.29
bridge-1-act-mV	355.10	348.35	358.82	10.47	2.92
bridge-1-cont	0.08	0.07	0.09	0.02	22.22
bridge-1-cont-act	0.82	0.81	0.83	0.02	2.41
bridge-1-mV	35.94	35.68	36.36	0.68	1.87
bridge-1-act-2scen	797.30	755.37	819.79	64.42	7.86
bridge-1-act-2scen-mV	696.68	667.07	708.46	41.39	5.84
bridge-1-2scen-mV	288.33	286.11	290.71	4.60	1.58
bridge-1-cont-2scen	0.18	0.17	0.19	0.02	10.53
bridge-1-cont-act-2scen	10.63	10.34	10.89	0.55	5.05
bridge-2	105.32	104.56	105.68	1.12	1.06
bridge-2-act	284.27	281.01	288.00	6.99	2.43
bridge-2-act-3scen-mV-int	4906.39	4783.54	5045.27	261.73	5.19
bridge-2-int	36.62	36.23	37.02	0.79	2.13
bridge-2-mV-int	17.06	16.86	17.28	0.42	2.43
bridge-2-act-mV-int	157.85	157.06	159.56	2.50	1.57
bridge-2-act-mV	7.55	7.46	7.63	0.17	2.23

instance	average time	min. time	max. time	range	[%]
bridge-2-cont	0.08	0.07	0.09	0.02	22.22
bridge-2-cont-act	0.83	0.81	0.86	0.05	5.81
bridge-2-mV	69.18	68.86	69.77	0.91	1.30
bridge-2-act-2scen-mV	1437.06	1418.01	1445.00	26.99	1.87
bridge-2-2scen-mV	1349.46	1328.98	1372.58	43.60	3.18
bridge-2-2scen-int	431.15	425.09	436.12	11.03	2.53
bridge-2-2scen-mV-int	229.01	227.31	231.53	4.22	1.82
bridge-2-cont-2scen	0.17	0.16	0.17	0.01	5.88
bridge-2-cont-act-2scen	10.29	9.95	10.52	0.57	5.42
bridge-2-act-2scen-mV-int	3348.08	3206.09	3428.84	222.75	6.50
bridge-3	40.62	40.12	40.82	0.70	1.71
bridge-3-act	190.01	186.07	193.22	7.15	3.70
bridge-3-act-int	951.00	943.88	958.31	14.43	1.51
bridge-3-3scen	2670.00	2649.43	2680.30	30.87	1.15
bridge-3-2scen	1861.59	1832.51	1877.38	44.87	2.39
bridge-3-act-3scen-mV-int	3195.60	3155.24	3225.39	70.15	2.17
bridge-3-int	68.91	68.60	69.71	1.11	1.59
bridge-3-mV-int	4.64	4.60	4.70	0.10	2.13
bridge-3-act-mV-int	263.65	261.00	265.80	4.80	1.81
bridge-3-act-mV	53.05	51.65	54.10	2.45	4.53
bridge-3-cont	0.10	0.09	0.11	0.02	18.18
bridge-3-cont-act	0.86	0.84	0.87	0.03	3.45
bridge-3-mV	86.10	85.85	86.74	0.89	1.03
bridge-3-act-2scen	5658.01	5575.73	5721.34	145.61	2.55
bridge-3-act-2scen-mV	1859.06	1844.34	1881.99	37.65	2.00
bridge-3-2scen-mV	2174.15	2159.92	2184.97	25.05	1.15
bridge-3-2scen-int	295.00	292.75	298.40	5.65	1.89
bridge-3-2scen-mV-int	40.35	40.07	40.74	0.67	1.64
bridge-3-cont-2scen	0.17	0.15	0.18	0.03	16.67
bridge-3-cont-act-2scen	5.09	5.00	5.16	0.16	3.10
bridge-3-act-2scen-mV-int	2027.43	2006.30	2040.75	34.45	1.69
bridge-4	4.99	4.94	5.02	0.08	1.59
bridge-4-act	20.78	20.40	21.33	0.93	4.36
bridge-4-3scen	343.13	342.19	344.71	2.52	0.73
bridge-4-2scen	228.23	226.79	229.05	2.26	0.99
bridge-4-act-mV	12.62	12.44	12.97	0.53	4.09
bridge-4-cont	0.08	0.07	0.09	0.02	22.22
bridge-4-cont-act	0.74	0.70	0.75	0.05	6.67
bridge-4-mV	24.05	23.90	24.20	0.30	1.24
bridge-4-act-2scen	835.05	828.13	842.19	14.06	1.67
bridge-4-act-2scen-mV	840.77	822.38	859.83	37.45	4.36
bridge-4-2scen-mV	198.07	196.81	200.12	3.31	1.65
bridge-4-cont-2scen	0.17	0.17	0.18	0.01	5.56
bridge-4-cont-act-2scen	9.49	9.43	9.53	0.10	1.05

instance	average time	min. time	max. time	range	[%]
bridge-5	102.84	102.25	103.39	1.14	1.10
bridge-5-act	293.04	291.88	294.51	2.63	0.89
bridge-5-act-3scen-mV-int	5708.07	5620.59	5811.29	190.70	3.28
bridge-5-int	36.22	36.09	36.42	0.33	0.91
bridge-5-mV-int	16.81	16.71	16.86	0.15	0.89
bridge-5-act-mV-int	2042.50	2003.31	2081.67	78.36	3.76
bridge-5-act-mV	7.04	6.99	7.13	0.14	1.96
bridge-5-cont	0.08	0.06	0.08	0.02	25.00
bridge-5-cont-act	0.72	0.71	0.73	0.02	2.74
bridge-5-mV	68.17	67.37	68.66	1.29	1.88
bridge-5-act-2scen-mV	1961.70	1949.70	1977.95	28.25	1.43
bridge-5-2scen-mV	1346.56	1331.65	1359.19	27.54	2.03
bridge-5-2scen-int	429.55	426.25	433.25	7.00	1.62
bridge-5-2scen-mV-int	227.79	224.27	230.32	6.05	2.63
bridge-5-cont-2scen	0.17	0.16	0.18	0.02	11.11
bridge-5-cont-act-2scen	9.31	9.22	9.44	0.22	2.33
bridge-5-act-2scen-mV-int	3601.90	3581.79	3637.75	55.96	1.54
bridge-6	40.33	40.19	40.58	0.39	0.96
bridge-6-act	209.19	207.21	210.57	3.36	1.60
bridge-6-3scen	2667.76	2647.76	2685.99	38.23	1.42
bridge-6-2scen	1814.74	1802.68	1827.64	24.96	1.37
bridge-6-act-3scen-mV-int	2964.04	2921.17	3033.37	112.20	3.70
bridge-6-int	68.87	68.42	69.43	1.01	1.45
bridge-6-mV-int	16.31	16.11	16.40	0.29	1.77
bridge-6-act-mV-int	1146.08	1124.51	1154.55	30.04	2.60
bridge-6-act-mV	124.55	123.59	124.84	1.25	1.00
bridge-6-cont	0.10	0.09	0.11	0.02	18.18
bridge-6-cont-act	0.61	0.58	0.63	0.05	7.94
bridge-6-mV	43.68	43.41	43.86	0.45	1.03
bridge-6-act-2scen	6195.92	6133.50	6397.20	263.70	4.12
bridge-6-act-2scen-mV	1721.03	1695.35	1760.33	64.98	3.69
bridge-6-2scen-mV	1323.15	1311.31	1342.61	31.30	2.33
bridge-6-2scen-int	289.64	287.43	290.68	3.25	1.12
bridge-6-2scen-mV-int	97.66	96.87	98.39	1.52	1.54
bridge-6-cont-2scen	0.17	0.16	0.17	0.01	5.88
bridge-6-cont-act-2scen	3.45	3.41	3.53	0.12	3.40
bridge-6-act-2scen-mV-int	1953.60	1918.02	2018.14	100.12	4.96
bridge-7	1432.54	1388.86	1464.06	75.20	5.14
bridge-7-act	5908.73	5730.60	6061.70	331.10	5.46
bridge-7-act-3scen-mV-int	2664.79	2509.70	2724.99	215.29	7.90
bridge-7-int	12.44	12.32	12.50	0.18	1.44
bridge-7-mV-int	1.29	1.28	1.31	0.03	2.29
bridge-7-act-mV-int	255.21	249.50	260.75	11.25	4.31
bridge-7-act-mV	311.16	300.06	319.92	19.86	6.21

instance	average time	min. time	max. time	range	[%]
bridge-7-cont	0.16	0.15	0.17	0.02	11.76
bridge-7-cont-act	8.33	7.89	8.61	0.72	8.36
bridge-7-mV	13.37	13.15	13.55	0.40	2.95
bridge-7-2scen-mV	536.99	529.10	549.76	20.66	3.76
bridge-7-2scen-int	315.31	312.35	317.79	5.44	1.71
bridge-7-2scen-mV-int	69.62	69.02	69.99	0.97	1.39
bridge-7-cont-2scen	0.33	0.32	0.33	0.01	3.03
bridge-7-cont-act-2scen	31.18	30.18	32.51	2.33	7.17
bridge-7-act-2scen-mV-int	1176.80	1106.15	1223.00	116.85	9.55
bridge-8-int	1568.68	1542.18	1595.91	53.73	3.37
bridge-8-mV-int	64.08	63.47	64.66	1.19	1.84
bridge-8-act-mV-int	1840.51	1735.05	1904.88	169.83	8.92
bridge-8-cont	0.30	0.29	0.32	0.03	9.38
bridge-8-cont-act	15.20	14.33	15.66	1.33	8.49
bridge-8-mV	2010.62	1890.78	2096.96	206.18	9.83
bridge-8-2scen-int	2889.06	2860.48	2913.25	52.77	1.81
bridge-8-2scen-mV-int	115.97	114.45	117.12	2.67	2.28
bridge-8-cont-2scen	0.51	0.49	0.52	0.03	5.77
bridge-8-cont-act-2scen	34.76	33.61	36.03	2.42	6.72
bridge-9	66.65	65.86	67.39	1.53	2.27
bridge-9-act	388.64	383.48	396.43	12.95	3.27
bridge-9-act-int	5125.62	5095.00	5180.69	85.69	1.65
bridge-9-3scen	3747.72	3706.76	3795.03	88.27	2.33
bridge-9-2scen	245.43	240.89	249.44	8.55	3.43
bridge-9-act-3scen-mV-int	2148.21	2112.18	2172.40	60.22	2.77
bridge-9-int	57.17	56.64	57.76	1.12	1.94
bridge-9-mV-int	16.94	16.85	17.04	0.19	1.12
bridge-9-act-mV-int	255.01	253.22	256.45	3.23	1.26
bridge-9-act-mV	74.36	73.11	75.85	2.74	3.61
bridge-9-cont	0.10	0.09	0.11	0.02	18.18
bridge-9-cont-act	13.23	13.00	13.39	0.39	2.91
bridge-9-mV	46.71	46.30	47.20	0.90	1.91
bridge-9-act-2scen	807.06	792.70	818.34	25.64	3.13
bridge-9-act-2scen-int	548.16	542.83	555.21	12.38	2.23
bridge-9-act-2scen-mV	63.80	59.93	65.43	5.50	8.41
bridge-9-2scen-mV	36.98	36.05	37.93	1.88	4.96
bridge-9-2scen-int	182.06	180.64	184.49	3.85	2.09
bridge-9-2scen-mV-int	18.44	18.31	18.63	0.32	1.72
bridge-9-cont-2scen	0.18	0.16	0.19	0.03	15.79
bridge-9-cont-act-2scen	22.48	22.10	23.09	0.99	4.29
bridge-9-act-2scen-mV-int	517.27	511.88	525.49	13.61	2.59
bridge-10	1102.61	1084.55	1109.96	25.41	2.29
bridge-10-act	720.74	700.40	752.27	51.87	6.90
bridge-10-2scen	7189.18	7156.18	7200.07	43.89	0.61

instance	average time	min. time	max. time	range	[%]
bridge-10-int	49.82	49.64	50.01	0.37	0.74
bridge-10-mV-int	130.01	128.55	131.14	2.59	1.97
bridge-10-act-mV-int	531.97	524.89	548.35	23.46	4.28
bridge-10-cont	0.14	0.13	0.15	0.02	13.33
bridge-10-cont-act	0.46	0.41	0.50	0.09	18.00
bridge-10-2scen-mV-int	1339.31	1327.15	1351.53	24.38	1.80
bridge-10-cont-2scen	0.19	0.18	0.20	0.02	10.00
bridge-10-cont-act-2scen	6.12	5.97	6.36	0.39	6.13
bridge-big	1792.50	1732.24	1834.21	101.97	5.56
bridge-big-cont	0.54	0.51	0.55	0.04	7.27
bridge-big-cont-act	40.67	39.69	41.72	2.03	4.87
bridge-big-2scen-mV	1.62	1.59	1.68	0.09	5.36
bridge-big-2scen-mV-int	0.55	0.54	0.57	0.03	5.26
bridge-big-cont-2scen	0.89	0.87	0.90	0.03	3.33
canti-1	23.01	22.81	23.17	0.36	1.55
canti-1-act	106.06	105.27	107.38	2.11	1.96
canti-1-3scen	534.35	531.29	538.48	7.19	1.34
canti-1-2scen	43.88	43.85	43.96	0.11	0.25
canti-1-act-mV	3.34	3.28	3.41	0.13	3.81
canti-1-cont	0.08	0.07	0.09	0.02	22.22
canti-1-cont-act	0.99	0.97	1.02	0.05	4.90
canti-1-mV	2.37	2.33	2.39	0.06	2.51
canti-1-act-2scen	177.99	177.00	179.44	2.44	1.36
canti-1-act-2scen-mV	0.29	0.27	0.30	0.03	10.00
canti-1-2scen-mV	0.09	0.08	0.09	0.01	11.11
canti-1-cont-2scen	0.13	0.12	0.14	0.02	14.29
canti-1-cont-act-2scen	0.36	0.35	0.38	0.03	7.89
canti-1-m-act-mV	9.57	9.42	9.78	0.36	3.68
canti-1-m-mV	3.62	3.56	3.67	0.11	3.00
canti-1-m-act-2scen-mV	0.30	0.28	0.32	0.04	12.50
canti-1-m-2scen-mV	0.08	0.07	0.09	0.02	22.22
canti-2	60.07	59.53	60.71	1.18	1.94
canti-2-act	182.73	181.26	184.44	3.18	1.72
canti-2-act-int	185.98	184.82	187.39	2.57	1.37
canti-2-3scen	1767.22	1754.79	1777.51	22.72	1.28
canti-2-2scen	669.06	664.60	673.97	9.37	1.39
canti-2-act-3scen-mV-int	220.09	216.93	221.82	4.89	2.20
canti-2-int	14.79	14.64	14.93	0.29	1.94
canti-2-mV-int	2.39	2.37	2.40	0.03	1.25
canti-2-act-mV-int	18.54	18.50	18.58	0.08	0.43
canti-2-act-mV	9.33	9.14	9.53	0.39	4.09
canti-2-cont	0.07	0.06	0.08	0.02	25.00
canti-2-cont-act	0.95	0.94	0.95	0.01	1.05
canti-2-mV	19.45	19.27	19.64	0.37	1.88

instance	average time	min. time	max. time	range	[%]
canti-2-act-2scen	1906.94	1890.16	1916.36	26.20	1.37
canti-2-act-2scen-mV	1388.02	1374.67	1395.47	20.80	1.49
canti-2-2scen-mV	0.15	0.14	0.17	0.03	17.65
canti-2-2scen-int	816.04	812.91	820.87	7.96	0.97
canti-2-2scen-mV-int	0.08	0.07	0.09	0.02	22.22
canti-2-cont-2scen	0.12	0.11	0.13	0.02	15.38
canti-2-cont-act-2scen	0.37	0.35	0.38	0.03	7.89
canti-2-act-2scen-mV-int	14.90	14.73	15.15	0.42	2.77
canti-2-m-act-3scen-mV-int	357.34	344.18	370.75	26.57	7.17
canti-2-m-mV-int	1.93	1.90	1.95	0.05	2.56
canti-2-m-act-mV-int	28.37	27.77	28.80	1.03	3.58
canti-2-m-act-mV	28.81	28.35	29.18	0.83	2.84
canti-2-m-mV	6.16	6.08	6.24	0.16	2.56
canti-2-m-act-2scen-mV	333.06	322.35	343.18	20.83	6.07
canti-2-m-2scen-mV	49.13	48.70	49.73	1.03	2.07
canti-2-m-act-2scen-mV-int	166.71	163.27	168.40	5.13	3.05
canti-3	124.15	123.11	126.07	2.96	2.35
canti-3-act	256.42	251.98	259.18	7.20	2.78
canti-3-act-int	242.99	241.41	243.97	2.56	1.05
canti-3-3scen	1062.21	1053.17	1071.40	18.23	1.70
canti-3-2scen	455.97	449.61	458.81	9.20	2.01
canti-3-act-3scen-mV-int	0.40	0.39	0.41	0.02	4.88
canti-3-int	11.14	11.08	11.24	0.16	1.42
canti-3-mV-int	1.57	1.56	1.58	0.02	1.27
canti-3-act-mV-int	23.18	22.93	23.31	0.38	1.63
canti-3-act-mV	7.27	7.09	7.43	0.34	4.58
canti-3-cont	0.08	0.07	0.08	0.01	12.50
canti-3-cont-act	0.95	0.94	0.95	0.01	1.05
canti-3-mV	40.15	40.06	40.37	0.31	0.77
canti-3-act-2scen	1505.96	1481.85	1537.70	55.85	3.63
canti-3-act-2scen-mV	0.41	0.38	0.43	0.05	11.63
canti-3-2scen-mV	0.15	0.13	0.16	0.03	18.75
canti-3-2scen-int	888.54	880.81	900.51	19.70	2.19
canti-3-2scen-mV-int	0.09	0.08	0.09	0.01	11.11
canti-3-cont-2scen	0.13	0.12	0.13	0.01	7.69
canti-3-cont-act-2scen	1.06	1.04	1.08	0.04	3.70
canti-3-act-2scen-mV-int	0.25	0.23	0.27	0.04	14.81
canti-3-m-act-3scen-mV-int	103.17	101.70	104.25	2.55	2.45
canti-3-m-mV-int	3.65	3.61	3.69	0.08	2.17
canti-3-m-act-mV-int	19.49	19.25	19.91	0.66	3.31
canti-3-m-act-mV	9.64	9.43	9.83	0.40	4.07
canti-3-m-mV	8.30	8.23	8.40	0.17	2.02
canti-3-m-act-2scen-mV	49.27	48.30	50.36	2.06	4.09
canti-3-m-2scen-mV	29.16	28.89	29.58	0.69	2.33

instance	average time	min. time	max. time	range	[%]
canti-3-m-act-2scen-mV-int	45.01	44.18	45.95	1.77	3.85
canti-4	22.82	22.50	23.09	0.59	2.56
canti-4-act	58.65	58.45	58.82	0.37	0.63
canti-4-3scen	535.78	532.65	542.45	9.80	1.81
canti-4-2scen	43.86	43.58	44.24	0.66	1.49
canti-4-act-mV	11.35	11.25	11.42	0.17	1.49
canti-4-cont	0.07	0.07	0.07	0.00	0.00
canti-4-cont-act	0.19	0.18	0.20	0.02	10.00
canti-4-mV	10.10	10.04	10.19	0.15	1.47
canti-4-act-2scen	151.59	150.28	153.80	3.52	2.29
canti-4-act-2scen-mV	0.30	0.28	0.31	0.03	9.68
canti-4-2scen-mV	0.08	0.07	0.09	0.02	22.22
canti-4-cont-2scen	0.12	0.11	0.14	0.03	21.43
canti-4-cont-act-2scen	0.37	0.37	0.38	0.01	2.63
canti-4-m-act-mV	16.41	16.13	16.69	0.56	3.36
canti-4-m-mV	3.60	3.54	3.63	0.09	2.48
canti-4-m-act-2scen-mV	0.29	0.28	0.30	0.02	6.67
canti-4-m-2scen-mV	0.08	0.07	0.10	0.03	30.00
canti-5	60.15	59.85	60.43	0.58	0.96
canti-5-act	191.21	188.65	193.16	4.51	2.33
canti-5-act-int	1865.97	1855.87	1873.99	18.12	0.97
canti-5-3scen	1772.46	1765.39	1780.61	15.22	0.85
canti-5-2scen	666.57	662.34	669.26	6.92	1.03
canti-5-int	14.82	14.74	14.85	0.11	0.74
canti-5-mV-int	2.12	2.10	2.14	0.04	1.87
canti-5-act-mV-int	121.63	120.65	122.38	1.73	1.41
canti-5-act-mV	7.58	7.47	7.73	0.26	3.36
canti-5-cont	0.08	0.07	0.08	0.01	12.50
canti-5-cont-act	0.19	0.18	0.20	0.02	10.00
canti-5-mV	5.69	5.66	5.71	0.05	0.88
canti-5-act-2scen	1527.60	1515.99	1539.02	23.03	1.50
canti-5-act-2scen-mV	135.63	132.91	139.37	6.46	4.64
canti-5-2scen-mV	0.16	0.14	0.17	0.03	17.65
canti-5-2scen-int	816.87	809.71	820.78	11.07	1.35
canti-5-2scen-mV-int	0.09	0.08	0.10	0.02	20.00
canti-5-cont-2scen	0.13	0.12	0.13	0.01	7.69
canti-5-cont-act-2scen	0.39	0.37	0.41	0.04	9.76
canti-5-m-act-3scen-mV-int	1082.54	1056.51	1101.49	44.98	4.08
canti-5-m-mV-int	1.94	1.90	1.96	0.06	3.06
canti-5-m-act-mV-int	35.03	34.68	35.76	1.08	3.02
canti-5-m-act-mV	24.46	24.12	25.12	1.00	3.98
canti-5-m-mV	6.18	6.11	6.26	0.15	2.40
canti-5-m-act-2scen-mV	192.23	187.27	198.54	11.27	5.68
canti-5-m-2scen-mV	49.14	48.80	49.53	0.73	1.47

instance	average time	min. time	max. time	range	[%]
canti-5-m-act-2scen-mV-int	243.41	241.24	246.88	5.64	2.28
canti-6	123.61	123.19	124.31	1.12	0.90
canti-6-act	212.51	208.89	216.33	7.44	3.44
canti-6-act-int	2343.84	2336.46	2350.40	13.94	0.59
canti-6-3scen	1070.28	1063.09	1085.89	22.80	2.10
canti-6-2scen	456.76	452.91	459.03	6.12	1.33
canti-6-act-3scen-mV-int	0.58	0.55	0.61	0.06	9.84
canti-6-int	11.17	11.13	11.20	0.07	0.62
canti-6-mV-int	1.67	1.65	1.68	0.03	1.79
canti-6-act-mV-int	16.28	16.06	16.40	0.34	2.07
canti-6-act-mV	55.89	55.24	56.16	0.92	1.64
canti-6-cont	0.07	0.06	0.08	0.02	25.00
canti-6-cont-act	0.19	0.18	0.20	0.02	10.00
canti-6-mV	81.38	81.10	81.82	0.72	0.88
canti-6-act-2scen	1073.52	1054.16	1085.42	31.26	2.88
canti-6-act-2scen-mV	0.42	0.41	0.44	0.03	6.82
canti-6-2scen-mV	0.16	0.15	0.17	0.02	11.76
canti-6-2scen-int	892.59	886.34	898.25	11.91	1.33
canti-6-2scen-mV-int	0.09	0.09	0.09	0.00	0.00
canti-6-cont-2scen	0.12	0.12	0.13	0.01	7.69
canti-6-cont-act-2scen	0.38	0.37	0.39	0.02	5.13
canti-6-act-2scen-mV-int	0.30	0.29	0.30	0.01	3.33
canti-6-m-act-3scen-mV-int	119.44	115.23	121.89	6.66	5.46
canti-6-m-mV-int	3.65	3.63	3.69	0.06	1.63
canti-6-m-act-mV-int	4.68	4.59	4.83	0.24	4.97
canti-6-m-act-mV	13.45	13.12	13.94	0.82	5.88
canti-6-m-mV	8.31	8.24	8.42	0.18	2.14
canti-6-m-act-2scen-mV	70.49	68.49	72.14	3.65	5.06
canti-6-m-2scen-mV	29.23	28.98	29.50	0.52	1.76
canti-6-m-act-2scen-mV-int	89.84	87.90	91.79	3.89	4.24
canti-7	605.74	593.72	629.13	35.41	5.63
canti-7-act	658.47	629.93	683.55	53.62	7.84
canti-7-act-int	1373.81	1363.89	1386.06	22.17	1.60
canti-7-int	32.57	32.37	32.77	0.40	1.22
canti-7-mV-int	32.86	32.38	33.11	0.73	2.20
canti-7-act-mV-int	882.18	876.23	891.31	15.08	1.69
canti-7-cont	0.16	0.15	0.18	0.03	16.67
canti-7-cont-act	1.83	1.79	1.87	0.08	4.28
canti-7-mV	3457.21	3434.68	3481.07	46.39	1.33
canti-7-2scen-int	1240.38	1233.60	1247.12	13.52	1.08
canti-7-2scen-mV-int	9.02	8.91	9.10	0.19	2.09
canti-7-cont-2scen	0.24	0.22	0.25	0.03	12.00
canti-7-cont-act-2scen	6.01	5.90	6.19	0.29	4.68
canti-8	64.53	64.21	64.98	0.77	1.18

instance	average time	min. time	max. time	range	[%]
canti-8-act	143.33	138.05	146.33	8.28	5.66
canti-8-act-mV	22.83	22.26	24.29	2.03	8.36
canti-8-cont	0.07	0.06	0.08	0.02	25.00
canti-8-cont-act	0.91	0.88	0.93	0.05	5.38
canti-8-mV	13.76	13.55	14.00	0.45	3.21
canti-9	254.01	251.36	255.82	4.46	1.74
canti-9-act	1703.01	1686.25	1726.87	40.62	2.35
canti-9-int	1685.27	1669.08	1705.18	36.10	2.12
canti-9-mV-int	99.65	99.26	100.34	1.08	1.08
canti-9-cont	0.18	0.17	0.19	0.02	10.53
canti-9-cont-act	0.60	0.58	0.62	0.04	6.45
canti-9-mV	3367.77	3340.57	3401.05	60.48	1.78
canti-9-2scen-mV-int	54.41	54.26	54.54	0.28	0.51
canti-9-cont-2scen	0.32	0.30	0.35	0.05	14.29
canti-9-cont-act-2scen	1.19	1.14	1.24	0.10	8.06
canti-10	192.92	190.91	197.04	6.13	3.11
canti-10-act	888.75	846.15	923.95	77.80	8.42
canti-10-act-int	939.82	926.05	947.79	21.74	2.29
canti-10-int	6.41	6.37	6.46	0.09	1.39
canti-10-mV-int	51.73	51.62	51.87	0.25	0.48
canti-10-act-mV-int	869.97	858.44	888.16	29.72	3.35
canti-10-cont	0.14	0.13	0.14	0.01	7.14
canti-10-cont-act	3.71	3.59	3.82	0.23	6.02
canti-10-mV	2968.92	2955.56	2979.25	23.69	0.80
canti-10-2scen-int	2382.31	2363.55	2400.28	36.73	1.53
canti-10-2scen-mV-int	1.82	1.81	1.84	0.03	1.63
canti-10-cont-2scen	0.21	0.20	0.22	0.02	9.09
canti-10-cont-act-2scen	18.94	18.44	19.52	1.08	5.53
canti-big	1096.81	1069.81	1112.56	42.75	3.84
canti-big-act	5409.97	5359.37	5464.90	105.53	1.93
canti-big-int	118.57	117.22	119.62	2.40	2.01
canti-big-mV-int	80.54	79.87	81.15	1.28	1.58
canti-big-act-mV-int	387.04	381.75	390.94	9.19	2.35
canti-big-cont	0.31	0.30	0.32	0.02	6.25
canti-big-cont-act	4.00	3.93	4.05	0.12	2.96
canti-big-2scen-mV-int	6.40	6.36	6.46	0.10	1.55
canti-big-cont-2scen	0.51	0.50	0.52	0.02	3.85
canti-big-cont-act-2scen	14.68	13.75	15.65	1.90	12.14
lit-as-1	1.68	1.66	1.69	0.03	1.78
lit-as-1-act	5.03	4.97	5.09	0.12	2.36
lit-as-1-act-mV	8.55	8.42	8.62	0.20	2.32
lit-as-1-cont	0.02	0.00	0.03	0.03	100.00
lit-as-1-cont-act	1.77	1.74	1.81	0.07	3.87
lit-as-1-mV	0.69	0.68	0.70	0.02	2.86

instance	average time	min. time	max. time	range	[%]
lit-as-2-cont	5.83	5.28	6.16	0.88	14.29
lit-as-2-big-cont	6.18	5.87	6.77	0.90	13.29
lit-as-5-act-mV	7085.71	6765.69	7204.47	438.78	6.09
lit-as-5-cont	1.01	0.96	1.06	0.10	9.43
lit-as-5-cont-act	10.60	9.24	11.77	2.53	21.50
lit-as-5-mV	1931.00	1892.00	1965.24	73.24	3.73
lit-as-5-big-act-int	4323.24	4242.83	4378.67	135.84	3.10
lit-as-5-big-int	4450.68	4366.13	4557.36	191.23	4.20
lit-as-5-big-mV-int	883.51	874.96	892.69	17.73	1.99
lit-as-5-big-act-mV-int	898.65	890.97	908.81	17.84	1.96
lit-as-5-big-cont	1.00	0.96	1.04	0.08	7.69
lit-as-5-big-cont-act	1.12	1.04	1.17	0.13	11.11
lit-as-6	898.80	890.85	901.48	10.63	1.18
lit-as-6-act	898.11	893.33	905.17	11.84	1.31
lit-as-6-act-mV	1711.35	1702.57	1725.93	23.36	1.35
lit-as-6-cont	0.50	0.50	0.51	0.01	1.96
lit-as-6-cont-act	0.52	0.51	0.53	0.02	3.77
lit-as-6-mV	1695.02	1687.44	1702.90	15.46	0.91
lit-as-6-big-mV-int	571.37	568.79	578.14	9.35	1.62
lit-as-6-big-act-mV-int	578.36	569.37	586.66	17.29	2.95
lit-as-6-big-cont	0.50	0.49	0.52	0.03	5.77
lit-as-6-big-cont-act	0.51	0.49	0.54	0.05	9.26
lit-s-1	3272.54	3252.94	3288.88	35.94	1.09
lit-s-1-act-mV	46.05	44.90	48.00	3.10	6.46
lit-s-1-cont	0.16	0.15	0.17	0.02	11.76
lit-s-1-mV	6.55	6.47	6.69	0.22	3.29

Table A.17. – Some statistics about the solving processes for the MISDP for the instances that where solved to optimality.

	nodes				
instance	average	min.	max.	range	[%]
bridge-2-act-int	112487.40	111030	113764	2734	2.40
bridge-2-3scen	28564.80	27980	28938	958	3.31
bridge-2-2scen	43287.20	42755	43739	984	2.25
bridge-2-act-2scen-int	35691.00	35143	36018	875	2.43
bridge-5-act-int	106126.60	105368	106778	1410	1.32
bridge-5-3scen	28478.80	28290	28667	377	1.32
bridge-5-2scen	43462.60	42971	44171	1200	2.72
bridge-5-act-2scen-int	33020.80	32663	33399	736	2.20
bridge-6-act-int	104381.60	103867	104675	808	0.77
bridge-7-act-int	33381.80	33127	33556	429	1.28
bridge-7-3scen	6987.20	6897	7123	226	3.17

	nodes				
instance	average	min.	max.	range	[%]
bridge-7-act-2scen-int	17251.00	17006	17408	402	2.31
bridge-8-2scen-mV	7087.20	6870	7328	458	6.25
bridge-8-act-2scen-mV-int	3035.00	2930	3118	188	6.03
bridge-10-act-int	27943.00	27512	28346	834	2.94
bridge-10-3scen	11057.80	10912	11136	224	2.01
bridge-10-act-mV	45895.80	45791	46009	218	0.47
bridge-10-mV	81793.00	81247	82187	940	1.14
bridge-10-act-2scen	4566.00	4430	4673	243	5.20
bridge-10-2scen-int	48638.80	48357	49112	755	1.54
bridge-10-act-2scen-mV-int	12816.60	12599	13133	534	4.07
canti-2-act-2scen-int	41560.60	41180	41903	723	1.73
canti-3-act-2scen-int	45527.40	45313	45745	432	0.94
canti-5-act-3scen-mV-int	13104.40	12985	13184	199	1.51
canti-5-act-2scen-int	44512.40	44232	44883	651	1.45
canti-5-act-2scen-mV-int	21183.20	20953	21395	442	2.07
canti-6-act-2scen-int	42176.20	41650	42612	962	2.26
canti-7-2scen	11006.60	10609	11197	588	5.25
canti-7-act-3scen-mV-int	5090.20	4910	5376	466	8.67
canti-7-2scen-mV	23404.40	23053	23658	605	2.56
canti-7-act-2scen-mV-int	8583.60	8135	8751	616	7.04
canti-9-act-int	23209.80	23075	23329	254	1.09
canti-9-3scen	6472.40	6257	6659	402	6.04
canti-9-2scen	8963.40	8709	9212	503	5.46
canti-9-act-mV-int	40585.40	39779	41165	1386	3.37
canti-9-2scen-mV	17481.80	17193	17653	460	2.61
canti-10-2scen	12704.80	12643	12827	184	1.43
canti-10-act-2scen	4654.60	4576	4837	261	5.40
canti-10-act-2scen-int	25469.00	25324	25568	244	0.95
canti-10-act-2scen-mV-int	13706.60	13602	13850	248	1.79
canti-big-act-int	11657.80	11469	11856	387	3.26
canti-big-mV	28768.40	28639	28862	223	0.77
lit-as-2-cont-act	104.80	96	116	20	17.24
lit-as-2-mV	1821.40	1772	1921	149	7.76
lit-as-2-big-mV-int	1756.60	1668	1819	151	8.30
lit-as-6-big-act-int	15270.40	15223	15296	73	0.48
lit-as-6-big-int	15042.40	14931	15284	353	2.31
lit-s-1-act	11110.00	11020	11189	169	1.51

Table A.18. – Some statistics for the nodes about the solving processes for the MISDP for the instances that reached the time limit.

	gap [%]				
instance	aver.	min.	max.	range	[%]
bridge-2-act-int	4.53	4.44	4.61	0.17	3.69
bridge-2-3scen	45.04	44.95	45.19	0.24	0.53
bridge-2-2scen	42.14	42.06	42.23	0.17	0.40
bridge-2-act-2scen-int	105673.87	105627.75	105777.04	149.29	0.14
bridge-5-act-int	30.48	30.44	30.52	0.08	0.26
bridge-5-3scen	45.07	45.01	45.13	0.12	0.27
bridge-5-2scen	42.12	42.01	42.20	0.19	0.45
bridge-5-act-2scen-int	99341.99	99313.01	99374.68	61.67	0.06
bridge-6-act-int	19.52	19.52	19.53	0.01	0.05
bridge-7-act-int	118.24	118.20	118.31	0.11	0.09
bridge-7-3scen	152.85	152.80	152.88	0.08	0.05
bridge-7-act-2scen-int	205.52	198.25	234.28	36.03	15.38
bridge-8-2scen-mV	4.90	4.86	4.95	0.09	1.82
bridge-8-act-2scen-mV-int	0.94	0.85	1.10	0.25	22.73
bridge-10-act-int	49.87	49.84	49.95	0.11	0.22
bridge-10-3scen	4668.82	4667.86	4670.13	2.27	0.05
bridge-10-act-mV	51.60	51.28	51.91	0.63	1.21
bridge-10-mV	33.60	33.48	33.71	0.23	0.68
bridge-10-act-2scen	203.30	203.15	203.45	0.30	0.15
bridge-10-2scen-int	16.51	16.41	16.59	0.18	1.08
bridge-10-act-2scen-mV-int	1892.98	1892.06	1896.00	3.94	0.21
canti-2-act-2scen-int	1315.22	1314.10	1316.61	2.51	0.19
canti-3-act-2scen-int	49.39	49.38	49.44	0.06	0.12
canti-5-act-3scen-mV-int	6.37	6.37	6.37	0.00	0.00
canti-5-act-2scen-int	2434.12	2433.79	2434.20	0.41	0.02
canti-5-act-2scen-mV-int	6.41	6.40	6.41	0.01	0.16
canti-6-act-2scen-int	83.96	83.80	84.20	0.40	0.48
canti-7-2scen	3.58	3.55	3.65	0.10	2.74
canti-7-act-3scen-mV-int	4272.15	4256.79	4283.68	26.89	0.63
canti-7-2scen-mV	24.81	24.73	24.94	0.21	0.84
canti-7-act-2scen-mV-int	1921.40	1920.97	1921.66	0.69	0.04
canti-9-act-int	363497.06	363462.93	363535.46	72.53	0.02
canti-9-3scen	57.77	57.63	57.91	0.28	0.48
canti-9-2scen	553.08	552.62	553.54	0.92	0.17
canti-9-act-mV-int	128.56	128.56	128.56	0.00	0.00
canti-9-2scen-mV	51.44	51.32	51.62	0.30	0.58
canti-10-2scen	421.00	420.89	421.03	0.14	0.03
canti-10-act-2scen	475.07	474.80	475.14	0.34	0.07
canti-10-act-2scen-int	144.82	144.81	144.85	0.04	0.03
canti-10-act-2scen-mV-int	2593.77	2591.34	2595.51	4.17	0.16
canti-big-act-int	79.97	79.94	80.02	0.08	0.10
canti-big-mV	3.82	3.77	3.87	0.10	2.58
lit-as-2-cont-act	0.04	0.04	0.04	0.00	0.00

	gap [%]				
instance	aver.	min.	max.	range	[%]
lit-as-2-mV	2285.68	2284.43	2286.12	1.69	0.07
lit-as-2-big-mV-int	2.04	2.02	2.07	0.05	2.42
lit-as-6-big-act-int	0.09	0.09	0.09	0.00	0.00
lit-as-6-big-int	0.10	0.09	0.10	0.01	10.00
lit-s-1-act	2063.86	2063.59	2064.17	0.58	0.03

Table A.19. – Some statistics for the gap about the solving processes for the MISDP for the instances that reached the time limit.

A.2 Results for the Maximum Cut Problems

This section contains the detailed tables for the computations of Chapter 7. For all computational results for the Maximum Cut Problem the solving time is always presented in seconds and the gap is in percent. Additionally, in most of the cases have a time limit of two hours. If the time limit is different from two hours we will explicitly state it.

A.2.1. The test set

We start with an overview of the test set. The first column of Tables A.20 shows the name of the instance, this first column will be identical in all the following tables in this chapter. In the second and third column the main characteristics are presented: the number of vertices and the number of edges in the graph. The following two columns give an idea on the maximum degree of the vertices. First, in column four we give the degree of the vertex with maximum degree. Then we show how many vertices exist with this degree. In the last four columns we present information about the edge weights. For each instance we state the minimum edge weight over all edges and how many edges with this weight exist. The same is shown for maximum edge weight in the last two columns.

			max.	# vert.	min.	# edges	max.	# edges
			degree	w. max.	edge	w. min.	edge	w. max.
name	vertices	edges	of vert.	degree	weight	weight	weight	weight
mc30-0	30	314	27	1	1	1	130	3
mc30-1	30	284	27	1	1	3	130	3
mc30-2	30	297	27	1	1	6	130	1
mc30-3	30	276	24	2	1	3	130	1
mc30-4	30	297	25	2	1	1	130	2
mc30-5	30	295	25	2	1	2	130	3
mc30-6	30	292	27	2	2	2	130	3
mc30-7	30	304	27	2	1	2	130	3
mc30-8	30	287	28	2	1	2	130	2
mc30-9	30	290	27	1	2	1	130	2
mc30-10	30	304	27	1	1	2	130	3
mc30-11	30	266	26	1	1	2	130	1
mc30-12	30	291	27	1	1	3	130	1
mc30-13	30	305	25	3	1	3	130	4
mc30-14	30	281	26	1	1	4	130	1
mc30-15	30	305	29	1	1	1	130	4
mc30-16	30	309	27	1	1	6	130	2
mc30-17	30	302	26	1	1	6	129	2
mc30-18	30	291	28	1	1	2	130	2

			max.	# vert.	min.	# edges	max.	# edges
			degree	w. max.	edge	w. min.	edge	w. max.
name	vertices	edges	of vert.	degree	weight	weight	weight	weight
mc30-19	30	269	28	1	2	2	129	3
mc35-0	35	424	32	1	1	1	130	5
mc35-1	35	394	33	1	1	1	130	1
mc35-2	35	383	32	1	1	1	130	3
mc35-3	35	386	32	1	1	2	130	7
mc35-4	35	475	33	1	1	4	130	2
mc35-5	35	425	31	4	1	4	130	4
mc35-6	35	335	27	1	1	5	130	1
mc35-7	35	379	32	1	1	1	130	3
mc35-8	35	396	31	1	1	3	130	2
mc35-9	35	407	30	1	1	3	130	2
mc35-10	35	389	33	2	1	4	130	1
mc35-11	35	431	32	1	1	1	130	5
mc35-12	35	449	34	1	1	5	130	4
mc35-13	35	376	29	1	1	1	130	3
mc35-14	35	438	33	1	1	3	130	6
mc35-15	35	447	32	1	1	2	130	7
mc35-16	35	393	31	1	1	3	130	4
mc35-17	35	360	31	1	1	4	130	7
mc35-18	35	394	30	1	1	2	130	6
mc35-19	35	402	31	1	1	5	130	3
mc40-0	40	575	38	1	1	4	130	4
mc40-1	40	503	36	1	1	6	130	4
mc40-2	40	491	35	1	1	6	130	4
mc40-3	40	537	38	1	1	3	130	4
mc40-4	40	540	38	1	1	4	130	3
mc40-5	40	513	37	1	1	4	129	4
mc40-6	40	508	36	2	1	5	130	5
mc40-7	40	520	37	1	1	6	130	5
mc40-8	40	541	36	1	1	4	130	2
mc40-9	40	483	35	1	1	3	130	6
mc40-10	40	608	39	1	1	2	130	5
mc40-11	40	561	37	1	1	3	130	4
mc40-12	40	496	37	1	1	6	130	4
mc40-13	40	527	36	1	1	3	130	9
mc40-14	40	523	35	1	1	3	130	9
mc40-15	40	561	36	1	1	2	130	7
mc40-16	40	539	36	2	1	1	130	8
mc40-17	40	541	36	1	1	6	130	7
mc40-18	40	549	38	1	1	6	130	3
mc40-19	40	563	38	1	1	1	130	3
mc45-0	45	750	43	1	1	10	130	5

name	vertices	edges	max. degree of vert.	# vert. w. max. degree	min. edge weight	# edges w. min. weight	max. edge weight	# edges w. max. weight
mc45-1	45	682	42	1	1	5	130	6
mc45-2	45	695	44	1	1	4	130	4
mc45-3	45	622	39	2	1	6	130	5
mc45-4	45	748	43	1	1	4	130	5
mc45-5	45	678	43	2	1	9	130	6
mc45-6	45	645	40	1	1	5	130	1
mc45-7	45	668	43	1	1	4	130	6
mc45-8	45	712	43	1	1	7	130	4
mc45-9	45	712	44	1	1	2	130	3
mc45-10	45	660	41	2	1	6	130	2
mc45-11	45	680	41	1	1	3	130	8
mc45-12	45	587	41	1	1	6	130	5
mc45-13	45	650	44	1	1	1	130	5
mc45-14	45	684	44	1	1	3	130	4
mc45-15	45	694	43	1	1	8	130	6
mc45-16	45	625	40	1	1	4	130	6
mc45-17	45	612	40	1	1	6	130	5
mc45-18	45	623	40	1	1	6	130	8
mc45-19	45	667	42	1	1	2	130	7
mc50-0	50	733	43	2	1	7	130	6
mc50-1	50	871	47	1	1	11	130	4
mc50-2	50	873	45	4	1	9	130	5
mc50-3	50	765	47	1	1	6	130	7
mc50-4	50	863	46	2	1	5	130	7
mc50-5	50	833	43	2	1	11	130	13
mc50-6	50	888	46	3	1	3	130	6
mc50-7	50	816	44	2	1	8	130	7
mc50-8	50	870	45	1	1	6	130	6
mc50-9	50	881	45	1	1	3	130	6
mc50-10	50	812	47	1	1	5	130	4
mc50-11	50	898	49	1	1	16	130	12
mc50-12	50	818	45	1	1	8	130	6
mc50-13	50	842	46	1	1	9	130	9
mc50-14	50	891	46	1	1	6	130	5
mc50-15	50	854	47	1	1	8	130	4
mc50-16	50	793	48	2	1	7	130	10
mc50-17	50	794	44	1	1	5	130	6
mc50-18	50	757	44	1	1	5	130	5
mc50-19	50	886	48	1	1	5	130	7

Table A.20. – Instance statistics for the randomly generated max-cut test set.

A.2.2. Comparing MIP and MISDP

Here we show the detailed results for the comparison of MIP and MISDP.

Remark A.2.1. *As we still only show arithmetic means it is possible that an instance with a solving time smaller than* 7200 *seconds is shown in Table A.22 where the instances are presented that reached the time limit. This is due to the fact that this instance was possibly solved one or more times and hit the time limit in the other runs.*

The Tables A.21, A.22, and A.23 will compare the two models concerning solving time, number of nodes needed in the branch-and-bound tree and the gap if the time limit is reached. The first column of each table is again the name of the instance. Then there are three columns for the corresponding values of the solving procedure using our MISDP software and three columns for the solving process of the MIP model. Additionally, we show which of the two formulations was better in the last column of every table. This column has the title *best* and entries '*M*' and '*S*'. '*M*' stands for the MIP and '*S*' for the MISDP.

	solving as MISDP			solving as MIP			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
mc30-0	911.55	709.00	0.00	157.37	310.00	0.00	M
mc30-1	94.86	66.00	0.00	19.14	7.00	0.00	M
mc30-2	95.61	72.00	0.00	93.66	91.00	0.00	M
mc30-3	191.90	143.00	0.00	55.56	32.00	0.00	M
mc30-4	248.53	194.00	0.00	126.77	165.00	0.00	M
mc30-5	137.20	106.00	0.00	77.37	76.00	0.00	M
mc30-6	249.41	208.00	0.00	78.14	81.00	0.00	M
mc30-7	582.86	464.00	0.00	170.35	222.00	0.00	M
mc30-8	101.63	71.00	0.00	45.44	52.00	0.00	M
mc30-9	204.62	160.00	0.00	51.31	47.00	0.00	M
mc30-10	162.12	122.00	0.00	74.81	108.00	0.00	M
mc30-11	180.88	145.00	0.00	60.51	54.00	0.00	M
mc30-12	549.00	453.00	0.00	84.85	129.00	0.00	M
mc30-13	82.90	57.00	0.00	66.07	42.00	0.00	M
mc30-14	383.51	307.00	0.00	45.30	24.00	0.00	M
mc30-15	391.07	323.00	0.00	157.45	272.00	0.00	M
mc30-16	218.20	180.00	0.00	69.60	67.00	0.00	M
mc30-17	84.71	65.00	0.00	70.86	64.00	0.00	M
mc30-18	287.26	199.00	0.00	75.60	56.00	0.00	M
mc30-19	600.39	446.00	0.00	38.86	45.00	0.00	M
mc35-0	1090.37	404.00	0.00	775.00	391.00	0.00	M
mc35-1	855.90	347.00	0.00	524.12	273.00	0.00	M
mc35-2	899.95	313.00	0.00	426.94	122.00	0.00	M
mc35-3	2070.39	854.00	0.00	404.60	188.00	0.00	M
mc35-4	3215.60	1245.00	0.00	1902.30	1882.00	0.00	M

	solving as MISDP			solving as MIP			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
mc35-5	1047.74	391.00	0.00	684.50	551.00	0.00	M
mc35-6	2083.98	767.00	0.00	163.82	17.00	0.00	M
mc35-7	389.11	129.00	0.00	295.03	107.00	0.00	M
mc35-8	1217.64	493.00	0.00	564.51	257.00	0.00	M
mc35-9	752.56	264.00	0.00	453.89	222.00	0.00	M
mc35-10	884.18	313.00	0.00	317.38	169.00	0.00	M
mc35-11	2501.49	977.00	0.00	988.71	617.00	0.00	M
mc35-12	953.90	352.00	0.00	1303.65	1296.00	0.00	S
mc35-13	2281.37	931.00	0.00	637.88	448.00	0.00	M
mc35-14	468.62	182.00	0.00	869.69	759.00	0.00	S
mc35-15	1158.87	393.00	0.00	1596.41	1323.00	0.00	S
mc35-16	573.46	203.00	0.00	322.35	183.00	0.00	M
mc35-17	210.48	74.00	0.00	343.09	120.00	0.00	S
mc35-18	805.69	283.00	0.00	623.96	369.00	0.00	M
mc35-19	1698.72	663.00	0.00	801.60	684.00	0.00	M
mc40-0	4675.02	904.00	0.00	7199.74	2292.80	2.72	S
mc40-1	4880.23	942.00	0.00	7199.73	1858.40	2.57	S
mc40-2	7202.90	1426.00	163.42	3960.25	1280.00	0.00	M
mc40-4	2232.79	406.00	0.00	2142.15	381.00	0.00	M
mc40-5	4348.02	828.00	0.00	5083.34	626.00	0.00	S
mc40-6	3916.87	734.00	0.00	1787.34	304.00	0.00	M
mc40-8	3523.96	743.00	0.00	4986.76	1608.00	0.01	S
mc40-9	2265.56	450.00	0.00	1575.11	272.00	0.00	M
mc40-10	3984.38	761.00	0.00	7199.76	1751.20	5.54	S
mc40-11	2465.58	449.00	0.00	6846.12	2044.60	1.17	S
mc40-12	4986.98	882.00	0.00	3821.84	919.00	0.00	M
mc40-14	3454.60	674.00	0.00	4416.41	1594.00	0.00	S
mc40-17	3552.70	684.00	0.00	6541.75	2910.00	0.00	S
mc40-19	5369.73	973.00	0.00	6985.74	2359.00	0.87	S
mc45-1	4473.02	427.00	0.00	7199.59	730.20	5.23	S
mc45-9	4751.70	412.00	0.00	7199.62	646.00	7.72	S
mc45-17	6702.10	594.00	0.00	7199.61	567.00	4.73	S
mc45-19	6403.76	623.00	0.00	7199.59	611.20	6.16	S

Table A.21. – Comparing the MIP and the MISDP model for the instances where at least one of the models was able to solve the problem to optimality within the time limit of two hours.

	solving as MISDP			solving as MIP			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
mc40-3	7202.71	1549.00	158.13	5879.97	2139.80	0.19	M
mc40-7	7202.93	1350.00	159.50	7199.71	2147.20	2.04	M

	solving as MISDP			solving as MIP			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
mc40-13	6649.80	1145.80	32.04	6971.72	2028.00	1.35	M
mc40-15	7202.06	1440.60	159.14	6527.66	2667.20	0.48	M
mc40-16	7202.41	1346.20	157.74	7199.77	1698.20	2.37	M
mc40-18	7203.31	1459.60	157.86	7199.77	2625.00	2.86	M
mc45-0	7205.84	680.80	153.85	7199.63	647.80	10.76	M
mc45-2	7204.59	734.80	157.10	7199.56	625.60	8.00	M
mc45-3	7204.64	825.40	158.76	7199.62	667.40	8.40	M
mc45-4	7206.62	696.40	153.59	7199.63	821.40	7.93	M
mc45-5	7208.24	672.80	156.48	7199.64	712.00	5.81	M
mc45-6	7205.73	757.40	162.42	7199.60	391.80	6.19	M
mc45-7	7206.40	687.80	165.50	7199.49	445.80	4.36	M
mc45-8	7205.37	735.80	157.33	7199.54	654.60	5.85	M
mc45-10	7207.61	756.40	160.85	7199.63	408.60	5.35	M
mc45-11	5872.94	560.20	31.92	7199.62	505.60	5.48	M
mc45-12	7205.25	745.80	170.82	7199.61	491.80	3.29	M
mc45-13	7205.37	749.60	160.75	7199.60	695.00	4.72	M
mc45-14	7204.86	758.00	159.04	7199.59	788.40	7.69	M
mc45-15	7206.99	843.00	158.47	7199.60	505.20	7.89	M
mc45-16	7206.42	754.20	161.16	7199.56	595.80	5.82	M
mc45-18	6965.07	703.20	66.11	7199.58	617.00	4.34	M
mc50-0	7208.86	418.20	162.81	7199.40	295.40	9.09	M
mc50-1	7211.28	394.00	156.86	7199.48	254.40	10.97	M
mc50-2	7206.04	428.00	157.59	7199.35	383.20	12.66	M
mc50-3	7207.13	390.00	162.61	7199.41	186.40	10.70	M
mc50-4	7204.86	472.60	149.06	7199.41	112.40	15.73	M
mc50-5	7206.66	468.20	154.63	7199.48	115.40	15.10	M
mc50-6	7217.39	415.20	152.80	7199.39	175.20	13.80	M
mc50-7	7208.33	496.20	155.59	7199.38	71.40	16.76	M
mc50-8	7206.46	478.40	153.03	7199.30	90.00	14.46	M
mc50-9	7207.15	470.00	152.71	7199.33	208.40	14.26	M
mc50-10	7208.08	461.80	154.77	7199.29	100.60	15.56	M
mc50-11	7211.03	463.00	154.26	7199.36	196.60	12.16	M
mc50-12	7209.88	424.00	158.76	7199.29	174.20	13.76	M
mc50-13	7205.75	429.80	159.98	7199.43	264.40	9.80	M
mc50-14	7207.56	442.80	151.73	7199.43	325.80	12.51	M
mc50-15	7214.48	444.40	152.60	7199.44	268.00	11.15	M
mc50-16	7212.39	415.00	161.38	7199.44	125.00	12.16	M
mc50-17	7213.16	356.60	161.50	7199.33	89.40	13.83	M
mc50-18	7208.45	459.40	160.39	7199.41	202.20	10.31	M
mc50-19	7208.77	438.60	149.16	7199.49	292.00	12.36	M

Table A.22. – Comparing the MIP and the MISDP model for the instances where both models reached the time limit of two hours.

	solving as MISDP			solving as MIP			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
mc40-0	4647.80	904.00	0.00	14642.92	4136.00	0.24	S
mc40-1	5088.86	942.00	0.00	9265.75	2675.70	1.28	S
mc40-2	11115.32	1930.00	0.00	3960.25	1280.00	0.00	M
mc40-3	18003.14	3260.40	156.70	5879.97	2139.80	0.19	M
mc40-4	2364.98	406.00	0.00	2142.15	381.00	0.00	M
mc40-5	4701.24	828.00	0.00	5083.34	626.00	0.00	S
mc40-6	4114.69	734.00	0.00	1787.34	304.00	0.00	M
mc40-7	17430.68	2832.60	0.03	10009.41	3502.10	1.02	S
mc40-8	3607.37	743.00	0.00	4986.76	1608.00	0.01	S
mc40-9	2276.32	450.00	0.00	1575.11	272.00	0.00	M
mc40-10	4067.04	761.00	0.00	17999.71	5175.00	3.95	S
mc40-11	2457.33	449.00	0.00	6893.80	2224.80	0.58	S
mc40-12	5054.17	882.00	0.00	3821.84	919.00	0.00	M
mc40-13	6946.42	1173.00	0.00	6893.17	2233.50	0.67	S
mc40-14	3767.81	674.00	0.00	4416.41	1594.00	0.00	S
mc40-15	13347.76	2368.00	0.00	8267.43	2800.00	0.01	M
mc40-16	9378.49	1690.00	0.00	10207.50	2631.00	0.00	S
mc40-17	3937.01	684.00	0.00	7466.46	2910.00	0.00	S
mc40-18	16357.27	3171.00	31.32	17325.57	6371.40	1.19	M
mc40-19	5410.63	973.00	0.00	9338.19	2607.00	0.00	S
mc45-0	10069.66	932.00	0.00	17999.61	1654.60	8.09	S
mc45-1	4511.32	427.00	0.00	12247.99	1536.60	2.90	S
mc45-2	18004.32	1828.00	155.81	17999.63	1161.20	7.42	M
mc45-3	18005.45	1894.60	157.39	17999.60	1428.60	7.56	M
mc45-4	18006.45	1725.80	152.07	17999.57	1496.40	7.75	M
mc45-5	18005.43	1936.00	154.90	17999.61	1675.20	4.18	M
mc45-6	13670.48	1466.00	0.00	17995.32	1834.80	0.75	S
mc45-7	13227.84	1243.00	0.00	15867.69	2130.40	0.19	S
mc45-8	11838.21	1185.00	0.00	17999.60	2139.60	2.70	S
mc45-9	4453.05	412.00	0.00	17999.59	1626.80	6.00	S
mc45-10	18003.18	1931.00	159.22	16272.35	1791.20	0.16	M
mc45-11	5803.61	569.00	0.00	17342.75	1913.20	1.35	S
mc45-12	18007.29	1852.60	168.98	10452.09	940.90	1.65	M
mc45-13	14943.62	1459.00	0.00	17999.55	1714.40	3.09	S
mc45-14	18006.27	1873.60	157.66	17999.56	1538.40	5.50	M
mc45-15	18006.17	2045.60	156.93	17999.56	1736.20	4.84	M
mc45-16	18005.26	1940.80	159.62	17999.54	1654.00	4.40	M
mc45-17	7043.77	594.00	0.00	9277.37	827.50	2.37	S
mc45-18	7783.00	752.00	0.00	9743.35	1070.00	2.17	S
mc45-19	5882.37	623.00	0.00	17999.60	1977.60	4.42	S
mc50-0	18009.62	969.40	161.49	17999.37	669.60	6.50	M
mc50-1	18016.04	873.40	155.78	17999.38	749.60	6.85	M
mc50-2	18011.07	950.00	156.38	17999.39	1055.60	9.59	M

	solving as MISDP			solving as MIP			
instance	time	nodes	gap [%]	time	nodes	gap [%]	best
mc50-3	12632.35	601.00	0.00	17999.33	648.80	7.54	S
mc50-4	18010.01	1151.20	148.17	17999.37	341.20	13.44	M
mc50-5	18011.06	1177.20	153.44	17999.45	648.80	10.40	M
mc50-6	8317.30	460.00	0.00	17999.37	710.60	10.08	S
mc50-7	18011.79	1062.40	154.60	17999.38	296.60	12.58	M
mc50-8	18009.91	1067.00	152.09	17999.43	376.40	11.42	M
mc50-9	18006.08	1104.40	151.62	17999.43	810.80	12.77	M
mc50-10	18004.75	1073.60	153.70	17999.35	461.20	11.79	M
mc50-11	18009.60	1004.00	153.23	17999.35	846.20	9.92	M
mc50-12	18011.55	963.40	157.44	17999.44	839.80	8.78	M
mc50-13	18015.48	945.80	158.61	17999.44	911.20	7.78	M
mc50-14	18014.15	1014.60	150.72	17999.41	984.20	11.35	M
mc50-15	18007.69	1061.00	151.54	17999.48	847.20	9.06	M
mc50-16	14658.19	803.00	0.00	17999.36	663.20	6.76	S
mc50-17	18014.23	839.00	160.17	17999.33	418.00	7.93	M
mc50-18	18005.90	1117.00	159.11	17999.40	578.60	7.54	M
mc50-19	18010.25	1142.60	148.11	17999.37	716.60	11.02	M

Table A.23. – Comparing the MIP and the MISDP model with a time limit of five hours for the larger instances.

A.2.3. Differences of the Maximum Cut Problem and Truss Topology Design

In Table A.24 we present information about the quality of the root relaxation and the first solution. For problems from Truss Topology Design we already stated such a table (see Table A.5). These two tables demonstrate big differences in the two applications.

instance	root solution gap [%]	first solution gap [%]	best sol. found at depth	best sol. found at node	overall nodes	nodes after best solution [%]
mc30-0	4.40	161.27	17	709	709	0.00
mc30-1	2.80	193.36	21	66	66	0.00
mc30-2	1.80	173.75	20	72	72	0.00
mc30-3	4.40	184.15	22	143	143	0.00
mc30-4	4.60	170.21	18	193	194	0.52
mc30-5	4.70	182.43	21	103	106	2.83
mc30-6	4.50	174.61	20	190	208	8.65
mc30-7	5.20	164.32	18	464	464	0.00
mc30-8	3.00	186.20	21	70	71	1.41
mc30-9	3.00	179.14	23	158	160	1.25
mc30-10	4.00	175.50	9	122	122	0.00

instance	root solution gap [%]	first solution gap [%]	best sol. found at depth	best sol. found at node	overall nodes	nodes after best solution [%]
mc30-11	4.10	182.78	17	145	145	0.00
mc30-12	5.00	174.04	18	453	453	0.00
mc30-13	2.50	188.46	13	54	57	5.26
mc30-14	5.70	181.61	21	295	307	3.91
mc30-15	3.50	164.27	23	323	323	0.00
mc30-16	5.40	174.67	19	180	180	0.00
mc30-17	2.00	174.70	15	48	65	26.15
mc30-18	5.30	173.30	19	196	199	1.51
mc30-19	6.20	183.86	25	446	446	0.00
mc35-0	3.70	163.91	19	336	404	16.83
mc35-1	3.90	169.59	14	317	347	8.65
mc35-2	5.30	177.14	16	257	313	17.89
mc35-3	4.70	168.62	30	841	854	1.52
mc35-4	4.30	154.80	25	1245	1245	0.00
mc35-5	3.90	164.74	16	377	391	3.58
mc35-6	6.90	189.55	24	767	767	0.00
mc35-7	2.90	178.45	26	128	129	0.78
mc35-8	4.30	174.10	19	492	493	0.20
mc35-9	4.10	168.52	21	260	264	1.52
mc35-10	3.80	171.10	27	313	313	0.00
mc35-11	4.40	160.18	30	977	977	0.00
mc35-12	3.40	158.53	23	303	352	13.92
mc35-13	4.80	167.76	25	930	931	0.11
mc35-14	3.90	163.83	18	182	182	0.00
mc35-15	3.50	159.04	28	393	393	0.00
mc35-16	3.40	176.07	27	203	203	0.00
mc35-17	2.60	180.11	16	74	74	0.00
mc35-18	3.50	166.31	25	283	283	0.00
mc35-19	5.00	166.50	18	655	663	1.21
mc40-0	3.50	154.73	30	904	904	0.00
mc40-1	4.20	162.38	19	822	942	12.74
mc40-2	4.50	162.76	31	1916	1930	0.73
mc40-4	4.10	169.39	31	406	406	0.00
mc40-5	5.00	163.55	23	720	828	13.04
mc40-6	4.10	170.51	29	734	734	0.00
mc40-8	3.40	159.48	23	743	743	0.00
mc40-9	3.70	170.20	17	450	450	0.00
mc40-10	2.90	150.18	25	761	761	0.00
mc40-11	2.90	158.34	23	434	449	3.34
mc40-12	4.20	163.43	34	880	882	0.23
mc40-13	4.40	159.93	20	1173	1173	0.00
mc40-14	3.30	161.47	24	668	674	0.89

instance	root solution gap [%]	first solution gap [%]	best sol. found at depth	best sol. found at node	overall nodes	nodes after best solution [%]
mc40-15	4.30	158.12	17	2368	2368	0.00
mc40-16	4.30	157.14	19	1536	1690	9.11
mc40-17	3.10	155.55	34	684	684	0.00
mc40-19	4.00	159.39	28	822	973	15.52
mc45-0	3.20	153.29	23	793	932	14.91
mc45-1	2.80	160.56	22	427	427	0.00
mc45-6	4.40	161.15	26	1466	1466	0.00
mc45-7	4.00	164.15	27	1243	1243	0.00
mc45-8	3.90	156.50	24	1185	1185	0.00
mc45-9	2.80	155.65	38	402	412	2.43
mc45-11	2.80	159.48	21	569	569	0.00
mc45-13	4.10	159.40	29	1459	1459	0.00
mc45-17	4.00	165.67	34	594	594	0.00
mc45-18	3.80	163.88	40	752	752	0.00
mc45-19	3.00	157.65	18	623	623	0.00
mc50-3	3.50	161.79	32	585	601	2.66
mc50-6	2.10	152.70	25	460	460	0.00
mc50-16	2.80	160.37	31	803	803	0.00

Table A.24. – Information about the quality of the first solution and the root relaxation of the maximum cut instances, when solved using the MISDP branch-and-bound approach.

A.2.4. A note on some statistics

We provide four different tables two for the MIP solving runs and two the MISDP solving runs. For each model one of the tables presents the instances that could be solved within the time limit, the other table shows the instances that reached the time limit without being solved to optimality. All runs presented within this section had a time limit of two hours, as the effects are already visible for these test runs. Furthermore, we only present statistics for the pure MISDP branch-and-bound algorithm because this was the most successful solving approach, as the results of the previous sections show.

The first two tables (A.25 and A.26) show the statistics for the MIP solving runs. For all instances that could be solved by the MIP, we present the minimum solving time of the five runs, the maximum solving time and the average that is the arithmetic mean of all the solving times. Additionally, column four of Table A.25 shows the range of the solving times, by range we mean the difference of maximum and minimum solving time. In the last column we present the percentage of the range with respect to the maximum solving time. The same columns are presented for gap and nodes in Table A.26 for the instances that reached the time limit.

In Tables A.27 and A.28 we show the same values for the MISDP solving runs. These tables show that there are huge differences in the solving runs.

instance	average time	min. time	max. time	range	[%]
mc30-0	157.37	141.27	176.94	35.67	20.16
mc30-1	19.14	18.20	20.19	1.99	9.86
mc30-2	93.66	88.68	97.34	8.66	8.90
mc30-3	55.56	52.41	61.06	8.65	14.17
mc30-4	126.77	115.20	139.59	24.39	17.47
mc30-5	77.37	71.62	82.82	11.20	13.52
mc30-6	78.14	74.15	82.78	8.63	10.43
mc30-7	170.35	159.62	182.77	23.15	12.67
mc30-8	45.44	42.01	49.08	7.07	14.41
mc30-9	51.31	45.80	54.79	8.99	16.41
mc30-10	74.81	70.69	82.13	11.44	13.93
mc30-11	60.51	57.57	67.23	9.66	14.37
mc30-12	84.85	78.63	90.97	12.34	13.56
mc30-13	66.07	64.07	68.20	4.13	6.06
mc30-14	45.30	39.48	48.62	9.14	18.80
mc30-15	157.45	150.11	165.05	14.94	9.05
mc30-16	69.60	66.79	72.40	5.61	7.75
mc30-17	70.86	68.16	73.67	5.51	7.48
mc30-18	75.60	73.06	79.39	6.33	7.97
mc30-19	38.86	36.58	41.75	5.17	12.38
mc35-0	775.00	614.60	921.91	307.31	33.33
mc35-1	524.12	397.18	769.16	371.98	48.36
mc35-2	426.94	397.15	509.51	112.36	22.05
mc35-3	404.60	286.02	513.46	227.44	44.30
mc35-4	1902.30	1240.41	2326.59	1086.18	46.69
mc35-5	684.50	503.71	901.44	397.73	44.12
mc35-6	163.82	114.37	207.95	93.58	45.00
mc35-7	295.03	205.90	377.86	171.96	45.51
mc35-8	564.51	384.15	697.82	313.67	44.95
mc35-9	453.89	344.09	504.98	160.89	31.86
mc35-10	317.38	270.52	361.59	91.07	25.19
mc35-11	988.71	768.84	1225.34	456.50	37.25
mc35-12	1303.65	1184.15	1426.36	242.21	16.98
mc35-13	637.88	535.99	688.83	152.84	22.19
mc35-14	869.69	795.54	1034.92	239.38	23.13
mc35-15	1596.41	1100.43	2294.73	1194.30	52.05
mc35-16	322.35	240.11	472.91	232.80	49.23
mc35-17	343.09	264.87	413.07	148.20	35.88
mc35-18	623.96	480.70	774.82	294.12	37.96
mc35-19	801.60	568.68	1183.01	614.33	51.93
mc40-2	3960.25	3025.91	4867.02	1841.11	37.83

instance	average time	min. time	max. time	range	[%]
mc40-3	5879.97	4750.20	7199.71	2449.51	34.02
mc40-4	2142.15	1723.55	2933.94	1210.39	41.25
mc40-5	5083.34	4521.76	5640.55	1118.79	19.83
mc40-6	1787.34	1390.39	2181.30	790.91	36.26
mc40-8	4986.76	3523.31	6287.26	2763.95	43.96
mc40-9	1575.11	1258.57	1933.66	675.09	34.91
mc40-11	6846.12	5968.20	7199.75	1231.55	17.11
mc40-12	3821.84	3218.13	4521.11	1302.98	28.82
mc40-13	6971.72	6082.83	7199.75	1116.92	15.51
mc40-14	4416.41	3467.77	5188.01	1720.24	33.16
mc40-15	6527.66	5158.16	7199.75	2041.59	28.36
mc40-17	6541.75	5775.02	6933.51	1158.49	16.71
mc40-19	6985.74	6486.24	7199.78	713.54	9.91

Table A.25. – Statistics about the solving processes for the MIP for the instances that were solved to optimality.

	nodes					gap [%]				
instance	average	min.	max.	range	[%]	aver.	min.	max.	range	[%]
mc40-0	2292.80	1955	2497	542	21.71	2.72	2.72	2.72	0.00	0.00
mc40-1	1858.40	1532	2126	594	27.94	2.57	2.38	3.27	0.89	27.22
mc40-7	2147.20	1836	2636	800	30.35	2.04	2.04	2.04	0.00	0.00
mc40-10	1751.20	1438	1948	510	26.18	5.54	5.54	5.54	0.00	0.00
mc40-16	1698.20	1627	1826	199	10.90	2.37	2.21	2.47	0.26	10.53
mc40-18	2625.00	2242	2936	694	23.64	2.86	2.86	2.86	0.00	0.00
mc45-0	647.80	544	855	311	36.37	10.76	10.39	10.95	0.56	5.11
mc45-1	730.20	593	844	251	29.74	5.23	4.65	6.05	1.40	23.14
mc45-2	625.60	554	707	153	21.64	8.00	7.89	8.13	0.24	2.95
mc45-3	667.40	633	722	89	12.33	8.40	8.24	8.47	0.23	2.72
mc45-4	821.40	713	907	194	21.39	7.93	7.75	8.16	0.41	5.02
mc45-5	712.00	634	818	184	22.49	5.81	5.27	6.66	1.39	20.87
mc45-6	391.80	338	476	138	28.99	6.19	4.64	6.70	2.06	30.75
mc45-7	445.80	388	602	214	35.55	4.36	3.84	4.61	0.77	16.70
mc45-8	654.60	536	768	232	30.21	5.85	5.51	6.22	0.71	11.41
mc45-9	646.00	503	745	242	32.48	7.72	7.41	8.16	0.75	9.19
mc45-10	408.60	250	586	336	57.34	5.35	3.83	7.14	3.31	46.36
mc45-11	505.60	457	550	93	16.91	5.48	5.31	5.69	0.38	6.68
mc45-12	491.80	301	606	305	50.33	3.29	2.82	4.02	1.20	29.85
mc45-13	695.00	621	739	118	15.97	4.72	4.56	5.00	0.44	8.80
mc45-14	788.40	694	918	224	24.40	7.69	7.19	8.91	1.72	19.30
mc45-15	505.20	424	611	187	30.61	7.89	6.82	9.23	2.41	26.11
mc45-16	595.80	522	786	264	33.59	5.82	5.44	5.97	0.53	8.88
mc45-17	567.00	459	711	252	35.44	4.73	3.77	5.27	1.50	28.46

	nodes					gap [%]				
instance	average	min.	max.	range	[%]	aver.	min.	max.	range	[%]
mc45-18	617.00	537	688	151	21.95	4.34	4.15	4.61	0.46	9.98
mc45-19	611.20	595	658	63	9.57	6.16	6.02	6.20	0.18	2.90
mc50-0	295.40	272	327	55	16.82	9.09	8.63	9.74	1.11	11.40
mc50-1	254.40	182	307	125	40.72	10.97	10.65	11.70	1.05	8.97
mc50-2	383.20	277	518	241	46.53	12.66	10.71	13.96	3.25	23.28
mc50-3	186.40	149	243	94	38.68	10.70	10.19	10.99	0.80	7.28
mc50-4	112.40	100	129	29	22.48	15.73	14.91	16.27	1.36	8.36
mc50-5	115.40	102	128	26	20.31	15.10	15.10	15.11	0.01	0.07
mc50-6	175.20	125	238	113	47.48	13.80	13.09	14.14	1.05	7.43
mc50-7	71.40	56	107	51	47.66	16.76	16.65	16.84	0.19	1.13
mc50-8	90.00	80	100	20	20.00	14.46	14.46	14.46	0.00	0.00
mc50-9	208.40	198	227	29	12.78	14.26	14.25	14.27	0.02	0.14
mc50-10	100.60	95	110	15	13.64	15.56	15.55	15.56	0.01	0.06
mc50-11	196.60	143	240	97	40.42	12.16	11.65	12.85	1.20	9.34
mc50-12	174.20	132	222	90	40.54	13.76	12.50	14.40	1.90	13.19
mc50-13	264.40	198	303	105	34.65	9.80	9.62	10.17	0.55	5.41
mc50-14	325.80	290	372	82	22.04	12.51	12.28	12.68	0.40	3.15
mc50-15	268.00	251	288	37	12.85	11.15	11.13	11.20	0.07	0.62
mc50-16	125.00	88	160	72	45.00	12.16	8.76	15.34	6.58	42.89
mc50-17	89.40	71	112	41	36.61	13.83	13.10	14.32	1.22	8.52
mc50-18	202.20	35	286	251	87.76	10.31	8.92	15.22	6.30	41.39
mc50-19	292.00	193	355	162	45.63	12.36	12.06	13.23	1.17	8.84

Table A.26. – Some statistics about the solving processes for the MIP for the instances that reached the time limit.

instance	average time	min. time	max. time	range	[%]
mc30-0	911.55	813.05	1144.55	331.50	28.96
mc30-1	94.86	83.47	112.44	28.97	25.76
mc30-2	95.61	84.68	118.07	33.39	28.28
mc30-3	191.90	171.98	232.34	60.36	25.98
mc30-4	248.53	223.61	300.82	77.21	25.67
mc30-5	137.20	117.00	159.51	42.51	26.65
mc30-6	249.41	229.79	282.75	52.96	18.73
mc30-7	582.86	514.87	709.99	195.12	27.48
mc30-8	101.63	84.05	122.02	37.97	31.12
mc30-9	204.62	192.06	249.21	57.15	22.93
mc30-10	162.12	152.72	175.76	23.04	13.11
mc30-11	180.88	174.71	183.87	9.16	4.98
mc30-12	549.00	541.30	560.87	19.57	3.49
mc30-13	82.90	78.41	87.74	9.33	10.63
mc30-14	383.51	363.83	403.42	39.59	9.81

instance	average time	min. time	max. time	range	[%]
mc30-15	391.07	382.90	398.91	16.01	4.01
mc30-16	218.20	213.86	224.39	10.53	4.69
mc30-17	84.71	81.72	87.98	6.26	7.12
mc30-18	287.26	268.83	302.98	34.15	11.27
mc30-19	600.39	577.69	635.73	58.04	9.13
mc35-0	1090.37	957.53	1161.05	203.52	17.53
mc35-1	855.90	799.93	925.01	125.08	13.52
mc35-2	899.95	786.77	1005.79	219.02	21.78
mc35-3	2070.39	1939.48	2265.44	325.96	14.39
mc35-4	3215.60	2947.16	3472.14	524.98	15.12
mc35-5	1047.74	967.29	1236.92	269.63	21.80
mc35-6	2083.98	1906.08	2247.52	341.44	15.19
mc35-7	389.11	363.21	441.40	78.19	17.71
mc35-8	1217.64	1080.74	1477.63	396.89	26.86
mc35-9	752.56	664.49	861.49	197.00	22.87
mc35-10	884.18	805.23	969.23	164.00	16.92
mc35-11	2501.49	2397.84	2684.45	286.61	10.68
mc35-12	953.90	859.48	1067.90	208.42	19.52
mc35-13	2281.37	2134.61	2468.55	333.94	13.53
mc35-14	468.62	432.83	534.19	101.36	18.97
mc35-15	1158.87	1065.99	1365.55	299.56	21.94
mc35-16	573.46	541.15	617.30	76.15	12.34
mc35-17	210.48	189.14	232.31	43.17	18.58
mc35-18	805.69	750.20	877.94	127.74	14.55
mc35-19	1698.72	1579.38	1862.26	282.88	15.19
mc40-0	4675.02	3907.28	6723.58	2816.30	41.89
mc40-1	4880.23	4145.79	6174.26	2028.47	32.85
mc40-4	2232.79	1916.44	3088.61	1172.17	37.95
mc40-5	4348.02	3786.61	5786.56	1999.95	34.56
mc40-6	3916.87	3548.84	4664.70	1115.86	23.92
mc40-8	3523.96	3161.45	4309.64	1148.19	26.64
mc40-9	2265.56	2045.18	2920.90	875.72	29.98
mc40-10	3984.38	3303.83	5006.41	1702.58	34.01
mc40-11	2465.58	2302.51	2976.94	674.43	22.66
mc40-12	4986.98	4407.79	6420.20	2012.41	31.34
mc40-13	6649.80	5991.55	7204.60	1213.05	16.84
mc40-14	3454.60	2917.13	4071.54	1154.41	28.35
mc40-17	3552.70	3032.62	4113.45	1080.83	26.28
mc40-19	5369.73	4955.05	6179.31	1224.26	19.81
mc45-1	4473.02	4038.99	4852.58	813.59	16.77
mc45-9	4751.70	4156.85	5276.34	1119.49	21.22
mc45-11	5872.94	5018.46	7215.54	2197.08	30.45
mc45-17	6702.10	6362.47	6960.83	598.36	8.60
mc45-18	6965.07	6008.96	7207.28	1198.32	16.63

instance	average time	min. time	max. time	range	[%]
mc45-19	6403.76	5617.86	6667.94	1050.08	15.75

Table A.27. – Some statistics about the solving processes for the MISDP for the instances that were solved to optimality.

	nodes					gap [%]				
instance	average	min.	max.	range	[%]	aver.	min.	max.	range	[%]
mc40-2	1426.00	1044	1586	542	34.17	163.42	163.17	164.05	0.88	0.54
mc40-3	1549.00	1115	1735	620	35.73	158.13	157.91	158.67	0.76	0.48
mc40-7	1350.00	1125	1467	342	23.31	159.50	159.28	159.92	0.64	0.40
mc40-15	1440.60	1131	1647	516	31.33	159.14	158.87	159.57	0.70	0.44
mc40-16	1346.20	1151	1490	339	22.75	157.74	157.54	158.04	0.50	0.32
mc40-18	1459.60	1313	1600	287	17.94	157.86	157.70	158.03	0.33	0.21
mc45-0	680.80	554	781	227	29.07	153.85	153.64	154.12	0.48	0.31
mc45-2	734.80	606	854	248	29.04	157.10	156.89	157.35	0.46	0.29
mc45-3	825.40	679	903	224	24.81	158.76	158.61	159.03	0.42	0.26
mc45-4	696.40	603	760	157	20.66	153.59	153.44	153.83	0.39	0.25
mc45-5	672.80	610	770	160	20.78	156.48	156.25	156.62	0.37	0.24
mc45-6	757.40	701	784	83	10.59	162.42	162.36	162.57	0.21	0.13
mc45-7	687.80	619	790	171	21.65	165.50	165.23	165.70	0.47	0.28
mc45-8	735.80	688	782	94	12.02	157.33	157.24	157.47	0.23	0.15
mc45-10	756.40	601	885	284	32.09	160.85	160.55	161.25	0.70	0.43
mc45-12	745.80	602	846	244	28.84	170.82	170.52	171.23	0.71	0.41
mc45-13	749.60	580	846	266	31.44	160.75	160.48	161.23	0.75	0.47
mc45-14	758.00	675	859	184	21.42	159.04	158.86	159.21	0.35	0.22
mc45-15	843.00	751	971	220	22.66	158.47	158.25	158.65	0.40	0.25
mc45-16	754.20	703	836	133	15.91	161.16	160.99	161.26	0.27	0.17
mc50-0	418.20	313	481	168	34.93	162.81	162.59	163.22	0.63	0.39
mc50-1	394.00	287	459	172	37.47	156.86	156.59	157.32	0.73	0.46
mc50-2	428.00	256	491	235	47.86	157.59	157.43	158.08	0.65	0.41
mc50-3	390.00	273	436	163	37.39	162.61	162.40	163.19	0.79	0.48
mc50-4	472.60	353	528	175	33.14	149.06	148.93	149.41	0.48	0.32
mc50-5	468.20	376	502	126	25.10	154.63	154.54	154.89	0.35	0.23
mc50-6	415.20	383	438	55	12.56	152.80	152.74	152.85	0.11	0.07
mc50-7	496.20	438	555	117	21.08	155.59	155.46	155.75	0.29	0.19
mc50-8	478.40	402	563	161	28.60	153.03	152.82	153.21	0.39	0.25
mc50-9	470.00	423	500	77	15.40	152.71	152.60	152.87	0.27	0.18
mc50-10	461.80	427	528	101	19.13	154.77	154.56	154.88	0.32	0.21
mc50-11	463.00	444	500	56	11.20	154.26	154.17	154.31	0.14	0.09
mc50-12	424.00	403	459	56	12.20	158.76	158.70	158.80	0.10	0.06
mc50-13	429.80	401	478	77	16.11	159.98	159.74	160.11	0.37	0.23
mc50-14	442.80	404	481	77	16.01	151.73	151.61	151.87	0.26	0.17
mc50-15	444.40	412	474	62	13.08	152.60	152.54	152.67	0.13	0.09

	nodes					gap [%]				
instance	average	min.	max.	range	[%]	aver.	min.	max.	range	[%]
mc50-16	415.00	326	486	160	32.92	161.38	161.20	161.66	0.46	0.28
mc50-17	356.60	313	406	93	22.91	161.50	161.29	161.69	0.40	0.25
mc50-18	459.40	402	541	139	25.69	160.39	160.14	160.55	0.41	0.26
mc50-19	438.60	390	515	125	24.27	149.16	149.00	149.27	0.27	0.18

Table A.28. – Some statistics about the solving processes for the MISDP for the instances that reached the time limit.

Bibliography

[Ach93] Wolfgang Achtziger. *Optimierung von einfach und mehrfach belasteten Stabwerken*, vol. 46. Bayreuther Mathematische Schriften, 1993.

[Ach96] Wolfgang Achtziger. *Truss topology optimization including bar properties different for tension and compression.* Structural and Multidisciplinary Optimization, 12:63 – 74, 1996.

[Ach99] Wolfgang Achtziger. *Local stability of trusses in the context of topology optimization – part I: Exact modelling.* Structural and Multidisciplinary Optimization, 17:235 – 246, 1999.

[Ach07] Tobias Achterberg. *Constraint integer programming.* PhD thesis, TU Berlin, 2007.

[Ach09] Tobias Achterberg. *SCIP: Solving constraint integer programs.* Mathematical Programming Computation, 1(1):1 – 41, 2009.

[AK06] Wolfgang Achtziger and Michal Kočvara. *Structural topology optimization with eigenvalues.* Technical report, University of Dortmund, 2006.

[Ali95] Farid Alizadeh. *Interior point methods in semidefinite programming with applications to combinatorial optimization.* SIAM Journal on Optimization, 5:13 – 51, 1995.

[AS07] Wolfgang Achtziger and Mathias Stolpe. *Truss topology optimization with discrete design variables – Guaranteed global optimality and benchmark examples.* Structural and Multidisciplinary Optimization, 34:1 – 20, 2007.

[AS08] Wolfgang Achtziger and Mathias Stolpe. *Global optimization of truss topology with discrete bar areas - part I: Theory of relaxed problems.* Computational Optimization and Applications, 40(2):247 – 280, 2008.

[AS09] Wolfgang Achtziger and Mathias Stolpe. *Global optimization of truss topology with discrete bar areas – part II: Implementation and numerical results.* Computational Optimization and Applications, 44:315 – 341, 2009.

[Bar02] Alexander Barvinok. *A course in convexity.* American Mathematical Society, 2002.

[BBTZ94] Martin P. Bendsøe, Aharon Ben-Tal, and Jochem Zowe. *Optimization methods for truss geometry and topology design.* Structural and Multidisciplinary Optimization, 7:141 – 159, 1994.

[BGH01] Srinivas Bollapragada, Omar Ghattas, and John N. Hooker. *Optimal design of truss structures by logic-based branch and cut.* Operations Research, 49(1):42 – 51, 2001.

[BS04] Martin P. Bendsøe and Ole Sigmund. *Topology Optimization – Theory, Methods and Applications.* Springer, 2004.

[BTB93] Aharon Ben-Tal and Martin P. Bendsøe. *A new method for optimal truss topology design.* SIAM Journal on Optimization, 3:322 – 358, 1993.

[BTJK^{+}00] Aharon Ben-Tal, Florian Jarre, Michael Kočvara, Arkardi Nemirovski, and Jochem Zowe. *Optimal design of trusses under a nonconvex global buckling constraint.* Optimization and Engineering, 1(2):189 – 219, 2000.

[BTN97] Aharon Ben-Tal and Arkardi Nemirovski. *Robust truss topology design via semidefinite programming.* SIAM Journal on Optimization, 7(4):991 – 1016, 1997.

[BTN01] Aharon Ben-Tal and Arkadi Nemirovski. *Lectures on Modern Convex Optimization.* Society for Industrial and Applied Mathematics, Philadephia, PA, 2001.

[BTZ95] Aharon Ben-Tal and Michael Zibulevsky. *Penalty/barrier multiplier methods for convex programming problems.* SIAM Journal on Optimization, 7:347 – 366, 1995.

[BY05] Steven J. Benson and Yinyu Ye. *DSDP5 user guide – software for semidefinite programming.* Technical Report ANL/MCS-TM-277, Argonne National Laboratory, 2005.

[BY08] Steven J. Benson and Yinyu Ye. *Algorithm 875: DSDP5 – software for semidefinite programming.* ACM Transactions on Mathematical Software, 34(3), 2008.

[CLRS01] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms.* MIT Press and McGraw-Hill, 2001.

[CPL12] Optimization Software CPLEX. *IBM ILOG CPLEX Optimization Studio Version 12.4*, 2012. `www.ibm.com/software/integration/optimization/cplex-optimization-studio`.

[CRC16] CRC 805: Collaborative Research Center 805. *Control of uncertainty in load-carrying structures in mechanical engineering*, 2009 – 2016. TU Darmstadt, `www.sfb805.tu-darmstadt.de`.

[Dhi97] Inderjit S. Dhillon. *A new $o(n^2)$ algorithm for the symmetric tridiagonal eigenvalue/eigenvector problem.* Technical Report UCB/CSD-97-971, Computer Science Division, UC Berkeley, 1997.

[dKRT97] Etienne de Klerk, Cornelis Roos, and Tamás Terlaky. *Initialization in semidefinite programming via a self-dual skew-symmetric embedding.* Operations Research Letters, 20(5):213 – 221, 1997.

[DMSR04] Inderjit Dhillon, Osni Marques, Ken Stanley, and Jason Riedy. *DSYEVR*, 2004. `www.netlib.org/lapack/double/dsyevr.f`.

[DP04a] Inderjit S. Dhillon and Beresford N. Parlett. *Multiple representations to compute orthogonal eigenvectors of symmetric tridiagonal matrices.* Linear Algebra and its Applications, 387:1 – 28, 2004.

[DP04b] Inderjit S. Dhillon and Beresford N. Parlett. *Orthogonal eigenvectors and relative gaps.* SIAM Journal on Matrix Analysis and Applications, 25, 2004.

[EPH01] Jonathan Eckstein, Cynthia A. Phillips, and William E. Hart. *PICO: An object-oriented framework for parallel branch-and-bound.* In *Inherently Parallel Algorithms in Feasibility and Optimization and their Applications*, Studies in Computational Mathematics 8, pp. 219 – 265. Elsevier Science, 2001.

[FFK$^+$08] Katsuki Fujisawa, Mituhiro Fukuda, Kazuhiro Kobayashi, Masakazu Kojimaand, Kazuhide Nakata, Maho Nakata, and Makoto Yamashita. *SDPA (SemiDefinite Programming Algorithm) and SDPA-GMP user's manual – Version 7.1.1.* Technical report, Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, 2008. `www.sdpa.sourceforge.net`.

[FKMN01] Mituhiro Fukuda, Masakazu Kojima, Kazuo Murota, and Kazuhide Nakata. *Exploiting sparsity in semidefinite programming via matrix completion I. General framework.* SIAM Journal on Optimization, 11(3):647 – 674, 2000/01.

[GUR11] Optimization Software GUROBI. *GUROBI Optimizer Version 4.6*, 2011. `www.gurobi.com`.

[Hab10] Kai Habermehl. *Personal communication*, 2010. TU Darmstadt.

[Hel00] Christoph Helmberg. *Semidefinite Programming for Combinatorial Optimization.* ZIB-Report, Habilitationsschrift, 2000.

[HJ90] Roger A. Horn and Charles R. Johnson. *Matrix Analysis.* Cambridge University Press, 1990.

[HUL04] Jean-Baptiste Hiriart-Urruty and Claude Lemarechal. *Fundamentals of convex analysis.* Springer, 2004.

[JKZ95] Florian Jarre, Michael Kočvara, and Jochem Zowe. *Optimal truss design by interior point methods.* SIAM Journal on Optimization, 8(4):1084 – 1107, 1995.

[Kar84] Narendra Karmarkar. *A new polynomial-time algorithm for linear programming.* Combinatorica, 4:373 – 395, 1984.

[KG10] Yoshihiro Kanno and Xu Guo. *A mixed integer programming for robust truss topology optimization with stress constraints.* International Journal for Numerical Methods in Engineering, 83(13):1675 – 1699, 2010.

[Kiw90] Krzysztof C. Kiwiel. *Proximity control in bundle methods for convex nondifferentiable minimization.* Mathematical Programming, 46:105 – 122, 1990.

[KO06] Michael Kočvara and Jiří Outrata. *Effective reformulations of the truss topology design problem.* Optimization and Engineering, 7(2):201 – 219, 2006.

[KS03] Michael Kočvara and Michael Stingl. *PENNON – A generalized augmented lagrangian method for semidefinite programming.* In Gianni Di Pillo and Almerico Murli, editors, *High Performance Algorithms and Software for Nonlinear Optimization*, vol. 82 of *Applied Optimization*, pp. 297 – 315. Kluwer Academic Publishers, Dordrecht, 2003.

[KT04] Yoshihiro Kanno and Izuru Takewaki. *Direct evaluation of robustness functions of trusses associated with stress constraints.* Technical report, Kyoto University, 2004.

[KT05] Yoshihiro Kanno and Izuru Takewaki. *Sequential semidefinite program for robust truss optimization based on robustness functions associated with stress constraints.* Technical report, Kyoto University, 2005.

[KZN00] Michael Kočvara, Jochem Zowe, and Arkardi Nemirovski. *Cascading – an approach to robust material optimization.* Computers and Structures, 75:431 – 442, 2000.

[LA11] Jean Lasserre and Miguel Anjos. *Handbook on Semidefinite, Conic and Polynomial Optimization.* Springer, 2011.

[LD60] Ailsa H. Land and Alison G. Doig. *An automatic method of solving discrete programming problems.* Econometrica, 28:497 – 520, 1960.

[LR05] Monique Laurent and Franz Rendl. *Semidefinite programming and integer programming.* In Karen Aardal, George Nemhauser, and Robert Weismantel, editors, *Handbook on Discrete Optimization*, vol. 12 of *Handbooks in Operations Research and Management Science*, pp. 393 – 514. Elsevier B.V., 2005.

[Mic04] A. Michell. *The limits of economy of material in frame-structures.* Philosophical Magazine Series 6, 8(47):589 – 597, 1904.

[Mit03] Hans D. Mittelmann. *An independent benchmarking of SDP and SOCP solvers.* Mathematical Programming B, 95:407 – 430, 2003.

[MS12] Sonja Mars and Lars Schewe. *An SDP-package for SCIP.* Technical report, TU Darmstadt, 2012.

[Nem04] Arkardi Nemirovski. *Interior point polynomial time methods in convex programming, lecture notes*, 2004.

[NFF$^+$03] Kazuhide Nakata, Katsuki Fujisawa, Mituhiro Fukuda, Masakazu Kojima, and Kazuo Murota. *Exploiting sparsity in semidefinite programming via matrix completion II. Implementation and numerical results.* Mathematical Programming B, 95(2):303 – 327, 2003.

[NN94] Yuri Nesterov and Arkadi Nemirovski. *Interior-Point Polynomial Algorithms in Convex Programming*, vol. 13 of *Studies in Applied Mathematics.* Society for Industrial and Applied Mathematics, 1994.

[NT97] Yuri E. Nesterov and Micheal J. Todd. *Self-scaled barriers and interior-point methods for convex programming.* Mathematics of Operations Research, 22(1):1 – 42, 1997.

[QBM12] Andrea Qualizza, Pietro Belotti, and François Margot. *Linear programming relaxations of quadratically constrained quadratic programs.* In Jon Lee and Sven Leyffer, editors, *Mixed Integer Nonlinear Programming*, vol. 154 of *The IMA Volumes in Mathematics and its Applications*, pp. 407 – 426. Springer New York, 2012.

[Rin86] Ulf Torbjörn Ringertz. *A branch and bound algorithm for topology optimization of truss structures.* Engineering Optimization, 10(2):111 – 124, 1986.

[Roc70] R. Tyrrell Rockafellar. *Convex Analysis.* Princeton Mathematical Series. Princeton University Press, 1970.

[RRW10] Franz Rendl, Giovanni Rinaldi, and Angelika Wiegele. *Solving Max-Cut to optimality by intersecting semidefinite and polyhedral relaxations.* Mathematical Programming, 121(2):307, 2010.

[Sch11] Jakob Schelbert. *Personal communication*, 2011. FAU Erlangen-Nürnberg.

[Sch12] Lars Schewe. *Personal communication*, 2009 – 2012. FAU Erlangen-Nürnberg.

[SCI12] Optimization Software SCIP 3.0. *Solving constraint integer programmes version 3.0*, 2012. `http://scip.zib.de`.

[SS03] Mathias Stolpe and Krister Svanberg. *Modelling topology optimization problems as linear mixed 0–1 programs.* International Journal for Numerical Methods in Engineering, 57:723 – 739, 2003.

[Ste12] Peter Steinke. *Eigenfrequenzen und Schwingungsformen von Stäben und Balken.* In *Finite-Elemente-Methode*, chapter 11, pp. 343 – 359. Springer Berlin Heidelberg, 2012.

[Sti06] Michael Stingl. *On the solution of nonlinear semidefinite programs by augmented lagrangian methods.* PhD thesis, FAU Erlangen-Nürnberg, 2006.

[Sto07] Mathias Stolpe. *On the reformulation of topology optimization problems as linear or convex quadratic mixed 0–1 programs.* Optimization and Engineering, 8:163 – 192, 2007.

[Str97] Bjarne Stroustrup. *The C++ Programming Language.* Addison Wesley Longman, Reading, MA., 1997.

[Swa73] Kandadai. N. Swamy. *On Sylvester's criterion for positive-semidefinite matrices.* IEEE Transactions on automatic control, 18:306, 1973.

[VB96] Lieven Vandenberghe and Stephen Boyd. *Semidefinite programming.* SIAM Review, 28(1):49 – 95, 1996.

[Wri97] Stephen J. Wright. *Primal-Dual Interior-Point Methods.* Other Titles in Applied Mathematics. SIAM, 1997.

[WSV00] Henry Wolkowicz, Romesh Saigal, and Lieven Vandenberghe. *Handbook of Semidefinite Programming – Theory, Algorithms, and Applications*, vol. 27 of *International Series in Operations Research and Management Science.* Springer, 2000.

[YK10] Kazuo Yonekura and Yoshihiro Kanno. *Global optimization of robust truss topology via mixed integer semidefinite programming.* Optimization Engineering, 11(3):355 – 379, 2010.