# The Chvátal-Gomory Procedure for Integer SDPs with Applications in Combinatorial Optimization

Frank de Meijer *  Renata Sotirov †

**Abstract**

In this paper we study the well-known Chvátal-Gomory (CG) procedure for the class of integer semidefinite programs (ISDPs). We prove several results regarding the hierarchy of relaxations obtained by iterating this procedure. We also study different formulations of the elementary closure of spectrahedra. A polyhedral description of the elementary closure for a specific type of spectrahedra is derived by exploiting total dual integrality for SDPs. Moreover, we show how to exploit (strengthened) CG cuts in a branch-and-cut framework for ISDPs. Different from existing algorithms in the literature, the separation routine in our approach exploits both the semidefinite and the integrality constraints. We provide separation routines for several common classes of binary SDPs resulting from combinatorial optimization problems. In the second part of the paper we present a comprehensive application of our approach to the quadratic traveling salesman problem (QTSP). Based on the algebraic connectivity of the directed Hamiltonian cycle, two ISDPs that model the QTSP are introduced. We show that the CG cuts resulting from these formulations contain several well-known families of cutting planes. Numerical results illustrate the practical strength of the CG cuts in our branch-and-cut algorithm, which outperforms alternative ISDP solvers and is able to solve large QTSP instances to optimality.

**Keywords** integer semidefinite programming, Chvátal-Gomory procedure, total dual integrality, branch-and-cut, quadratic traveling salesman problem

## 1 Introduction

Convex integer nonlinear programs (CINLPs) are optimization problems in which the objective function is convex and the continuous relaxation of the feasible region is a convex set. Nonlinearities in CINLPs can appear in both the objective function and/or the constraints. Motivated by their numerous applications and their ability to generalize several well-known problem classes, CINLPs have been studied for decades. In this paper we focus on a specific class of CINLPs: the integer semidefinite programs (ISDPs). These problems can be formulated as:

$$\begin{aligned} \sup \quad & \mathbf{b}^\top \mathbf{x} \\ \text{s.t.} \quad & \mathbf{C} - \sum_{i=1}^{m} \mathbf{A}_i x_i \succeq \mathbf{0}, \quad \mathbf{x} \in \mathbb{Z}^m, \end{aligned} \tag{1}$$

with $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{C}, \mathbf{A}_i \in \mathcal{S}^n$, where $\mathcal{S}^n$ denotes the cone of symmetric matrices of order $n$. Since integer linear programs belong to the family of ISDPs, problems of the form (1) are generally $\mathcal{NP}$-hard to solve.

Although CINLPs have been studied extensively, see e.g., the survey of Bonami et al. [9], the special case of ISDPs has received attention only very recently. This is remarkable, as the mixture of positive semidefiniteness and integrality leads naturally to a broad range of applications, e.g., in

*CentER, Department of Econometrics and OR, Tilburg University, The Netherlands, `f.j.j.demeijer@uvt.nl`

†CentER, Department of Econometrics and OR, Tilburg University, The Netherlands, `r.sotirov@uvt.nl`

architecture [14, 67], signal processing [37, 53] and combinatorial optimization [38, 56]. For a more detailed overview of applications of ISDPs, we refer the reader to [38, 44].

Only a few solution approaches for solving SDPs with integrality constraints have been considered. Gally et al. [38] propose a general framework called SCIP-SDP for solving mixed integer semidefinite programs (MISDPs) using a branch-and-bound (B&B) procedure with continuous SDPs as subproblems. They show that strict duality of the relaxations is maintained in the B&B tree and study several solver components. Alternatively, Kobayashi and Takano [44] propose a cutting-plane algorithm that initially relaxes the positive semidefinite (PSD) constraint and solves a mixed integer linear programming problem, where the PSD constraint is imposed dynamically via cutting planes. This leads to a general branch-and-cut (B&C) algorithm for solving MISDPs. A third project that encounters general ISDPs is YALMIP [47]. However, it is noted by the authors of [38] and [44] that the branch-and-bound ISDP solver in YALMIP is not yet competitive to the performance of the other two methods. Recently, Matter and Pfetsch [48] study different presolving strategies for MISDPs for both the B&B and B&C approach.

Apart from solution methods for solving general ISDPs or MISDPs, there are several other approaches in the literature that aim to solve integer problems by utilizing SDP relaxations in a B&B framework. Although these approaches are very related to problems of the form (1) in the sense that they also combine semidefinite programs with a branching strategy, they differ in the sense that the problem at hand is not necessarily formulated as a MISDP. Examples are the BiqCrunch solver for constrained binary quadratic problems [45] and the Biq Mac solver for unconstrained binary quadratic problems [56].

In the light of improving the performance of the B&C algorithm of [44], we consider the exploitation of cutting planes for ISDPs. Practical algorithms for CINLPs have benefited a lot from the addition of strong cutting planes, see e.g., [4, 5, 7, 62], where many of these cutting plane frameworks are based on generalizations from integer linear programming. Among the most well-known cutting planes for integer linear programs (ILPs) are the Chvátal-Gomory (CG) cuts [15, 41]. Gomory [41] introduced these cuts to design the first finite cutting plane algorithm for ILPs. Chvátal [15] later generalized this notion and introduced the closure of all such cuts that leads to a hierarchy of relaxations of the ILP with increasing strength. Chvátal [15] and Schrijver [58] prove that this hierarchy is finite for bounded real polyhedra and rational polyhedra, respectively. Later on, the CG procedure is introduced for more general convex sets, see e.g., [22, 18, 24, 10, 19]. In particular, Çezik and Iyengar [13] show how to generate CG cuts for CINLPs where the continuous relaxation of the feasible region is conic representable.

A leading application in this work is a combinatorial optimization problem that can be modelled as an ISDP: the quadratic traveling salesman problem (QTSP). Jäger and Molitor [43] introduce the QTSP as the problem of finding a Hamiltonian cycle in a graph that minimizes the total interaction costs among consecutive arcs. The problem is motivated by an important application in bioinformatics [43, 33], but has also applications in telecommunication, precision farming and robotics, see e.g., [64, 28, 1]. The QTSP is $\mathcal{NP}$-hard in the strong sense and is currently considered as one of the hardest combinatorial optimization problems to solve in practice.

Several papers have studied the QTSP. In [31, 32, 35] the polyhedral structure of the asymmetric and symmetric QTSP–polytope is discussed. Rostami et al. [57] provide several lower bounding procedures for the QTSP, including a column generation approach. Woods and Punnen [65] provide different classes of neighbourhoods for the QTSP, while Staněk et al. [59] discuss several heuristics for the quadratic traveling salesman problem in the plane. The linearization problem for the QTSP is studied in [54]. Fischer et al. [33, 34] introduce several exact algorithms and heuristics for the asymmetric QTSP, while Aichholzer et al. [2] consider exact solution methods for the minimization and maximization version of the symmetric QTSP.

## 1.1 Main results and outline

In this paper we consider the Chvátal-Gomory procedure for ISDPs from a theoretical as well as a practical point of view. On the theoretical side, we derive several results on the elementary closure of all CG cuts for spectrahedra. On the practical side, we show how to apply these cuts in a generic branch-and-cut algorithm for ISDPs that exploits both the positive semidefiniteness and the integrality of the problem. We extensively study the application of this new approach to the QTSP, which confirms the practical strength of the proposed method.

We start by reformulating a CG cut for a spectrahedron in terms of its data matrices in combination with the elements from the dual cone. This leads to a constructive description of the elementary closure of spectrahedra rather than the implicit description that is known for general convex sets. Equivalent to the case of polyhedra, the elementary closure operation can be repeated, leading to a hierarchy of stronger approximations of the integer hull of the spectrahedron. We give a short proof for the finiteness of this hierarchy for bounded spectrahedra. For this class of spectrahedra we also provide a compact proof of a homogeneity property for the elementary closure operation that is based on a theorem of alternatives and Dirichlet's approximation theorem. We prove this property for halfspaces that are sufficient to describe any compact convex set. Homogeneity is the cornerstone in showing that the elementary closure of a bounded spectrahedron is polyhedral. Although the latter result is known in the literature, our proof significantly simplifies compared to the general proofs given in [19, 10]. Finally, we exploit the recently introduced notion of total dual integrality for SDPs [11] to derive the explicit polyhedron describing the elementary closure for a certain class of spectrahedra.

It is known that the practical strength of CG cuts in integer linear programming is mainly due to their application in branch-and-bound methods. In this vein, we propose a generic branch-and-cut (B&C) framework for ISDPs. Our algorithm initially relaxes the PSD constraint and solves a mixed integer linear program (MILP), where the PSD constraint is imposed iteratively via CG and/or strengthened CG cuts. To derive strengthened CG cuts, we use a similar approach to the one for rational polyhedra by Dash et al. [21]. Our B&C algorithm is an extension of the algorithm of [44], in which separation is only based on positive semidefiniteness without taking into account the integrality of the variables. Our approach also builds up on the work by Çezik and Iyengar [13], in which the authors leave the separation of CG cuts for conic problems as an open problem and do not include these cuts in their computational study. We provide examples of our approach for two common classes of binary SDPs that frequently appear in combinatorial optimization.

In the third part of this paper we apply our results to a difficult-to-solve combinatorial optimization problem: the quadratic traveling salesman problem. We derive two ISDP formulations of this problem based on the notion of algebaic connectivity. To solve these models using our B&C algorithm, we propose several CG separation routines and show that various of these routines lead to well-known cuts for the QTSP. Computational results on a large set of benchmark QTSP instances show that the practical potential of our new method is twofold. The method significantly outperforms the ISDP solvers from the literature, whereas it also provides competitive results to the state-of-the-art QTSP solution method of [33].

The paper is organized as follows. In Section 2 we study the Chvátal-Gomory procedure for spectrahedra. Section 3 provides a CG-based B&C framework for general ISDPs and provides specific CG separation routines for two classes of binary SDPs. In Section 4 we formally define the QTSP and present two ISDP formulations of this problem. Numerical results are given in Section 5.

## 1.2 Notation

A directed graph is given by $G = (N, A)$, where $N$ is a set of nodes and $A \subseteq N \times N$ is a set of arcs. We use $K_n$ to denote the complete directed graph on $n$ nodes, i.e., a directed graph in which every pair of nodes is connected by a bidirectional edge.

We denote by $\mathbf{0}_n \in \mathbb{R}^n$ the vector of all zeros, and by $\mathbf{1}_n \in \mathbb{R}^n$ the vector of all ones. The identity matrix and the matrix of ones of order $n$ are denoted by $\mathbf{I_n}$ and $\mathbf{J_n}$, respectively. We omit the subscripts of these matrices when there is no confusion about the order. The $i$-th elementary vector is denoted by $\mathbf{e}_i$ and we define $\mathbf{E_{ij}} := \mathbf{e}_i \mathbf{e}_j^\top$.

The set of integer numbers and nonnegative integer numbers is denoted by $\mathbb{Z}$ and $\mathbb{Z}_+$, respectively. For any integer vector $c \in \mathbb{Z}^m$, we let $\gcd(c)$ denote the greatest common divisor of the entries in $c$. We define the floor (resp. ceil) operator $\lfloor \cdot \rfloor$ (resp. $\lceil \cdot \rceil$) as the largest (resp. smallest) integer smaller (resp. larger) than or equal to the input number. For $n \in \mathbb{Z}_+$, we define the set $[n] := \{1, \dots, n\}$, whereas the power set $\mathcal{P}([n])$ denotes the set of all subsets of $[n]$. Also, for any $S \subseteq [n]$, we let $\mathbb{1}_\mathbf{S}$ be the binary indicator vector of $S$.

We let $\mathcal{S}^n$ be the set of all $n \times n$ real symmetric matrices and denote by $\mathbf{X} \succeq \mathbf{0}$ that a symmetric matrix $\mathbf{X}$ is positive semidefinite. We use $\mathbf{X} \succsim\neq \mathbf{0}$ to denote that $\mathbf{X}$ is positive semidefinite but not equal to the zero matrix. The cone of symmetric positive semidefinite matrices is defined as $\mathcal{S}^n_+ := \{\mathbf{X} \in \mathcal{S}^n \,:\, \mathbf{X} \succeq \mathbf{0}\}$. The trace of a square matrix $\mathbf{X} = (x_{ij})$ is given by $\operatorname{tr}(\mathbf{X}) = \sum_i x_{ii}$. For any $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n\times n}$ the trace inner product is defined as $\langle \mathbf{X}, \mathbf{Y}\rangle := \operatorname{tr}(\mathbf{X}^\top \mathbf{Y}) = \sum_{i=1}^n \sum_{j=1}^n x_{ij}y_{ij}$.

The operator $\operatorname{diag} : \mathbb{R}^{n\times n} \to \mathbb{R}^n$ maps a square matrix to a vector consisting of its diagonal elements. We denote by $\operatorname{Diag} : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ its adjoint operator.

# 2 The Chvátal-Gomory procedure for ISDPs

In this section we study the extension of the cutting-plane procedure by Chvátal [15] and Gomory [41] for integer linear programs to the class of integer semidefinite programs. We show that several concepts, such as the Chvátal-Gomory closure and the Chvátal rank, can be generalized to ISDPs. We start by recollecting the procedure for general convex sets.

## 2.1 The Chvátal-Gomory procedure

Let $C \subseteq \mathbb{R}^m$ be a non-empty closed convex set and let $C_I$ be its integer hull, i.e., $C_I := \operatorname{Conv}(C \cap \mathbb{Z}^m)$. The Chvátal-Gomory cutting-plane procedure is introduced by Chvátal [15] and Gomory [41] and is regarded to be among the most celebrated results in integer programming. The CG procedure aims at systematically identifying valid inequalities for $C$ that cut off non-integer solutions. By adding these new cuts to the relaxation and repeating this process, one obtains a hierarchy of stronger relaxations that converges to $C_I$.

The CG procedure relies on the notion of rational halfspaces. A rational halfspace is of the form $H = \{\mathbf{x} \in \mathbb{R}^m \,:\, \mathbf{c}^\top \mathbf{x} \le d\}$ for some $\mathbf{c} \in \mathbb{Q}^m, d \in \mathbb{Q}$. It is known that all such halfspaces can be represented by $\mathbf{c} \in \mathbb{Z}^m$ such that the entries of $\mathbf{c}$ are relatively prime. If $H = \{\mathbf{x} \in \mathbb{R}^m \,:\, \mathbf{c}^\top \mathbf{x} \le d\}$ with $\mathbf{c} \in \mathbb{Z}^m$, $\gcd(\mathbf{c}) = 1$, then $H_I = \{\mathbf{x} \in \mathbb{R}^m \,:\, \mathbf{c}^\top \mathbf{x} \le \lfloor d \rfloor\}$.

**Definition 1.** The elementary closure of a closed convex set $C$ is the set

$$c(C) := \bigcap_{\substack{(\mathbf{c},d)\in\mathbb{Q}^m\times\mathbb{Q} \\ C\subseteq H=\{\mathbf{x}\,:\,\mathbf{c}^\top\mathbf{x}\le d\}}} H_I. \tag{2}$$

Equivalently, the elementary closure of $C$ can be written as:

$$c(C) = \bigcap_{\substack{(\mathbf{c},d)\in\mathbb{Z}^m\times\mathbb{R} \\ C\subseteq\{\mathbf{x}\,:\,\mathbf{c}^\top\mathbf{x}\le d\}}} \left\{\mathbf{x} \in \mathbb{R}^m \,:\, \mathbf{c}^\top \mathbf{x} \le \lfloor d \rfloor\right\}, \tag{3}$$

and we will primarily use this form in this work. The inequalities that define $c(C)$ in (3) are known as CG cuts [41]. One can verify that $C_I \subseteq c(C)$. When $C$ is compact, we can exploit the following proposition due to Dadush et al. [18] and De Carli Silva and Tunçel [11].

**Proposition 1.** *If $C \subseteq \mathbb{R}^m$ is a compact convex set, then*

$$C = \bigcap_{\substack{(\mathbf{c},d)\in\mathbb{Z}^m\times\mathbb{R} \\ C\subseteq\{\mathbf{x}\,:\,\mathbf{c}^\top\mathbf{x}\le d\}}} \left\{\mathbf{x} \in \mathbb{R}^m \,:\, \mathbf{c}^\top \mathbf{x} \le d\right\}.$$

It follows from Proposition 1 that for compact convex sets $C$ we have $c(C) \subseteq C$. We can now repeat the procedure by defining $C^{(0)} := C$ and $C^{(k+1)} := c(C^{(k)})$ for all integer $k \ge 0$, where $C^{(k)}$ is referred to as the $k$th CG closure of $C$. For any compact convex set $C$ this leads to the hierarchy $C_I \subseteq \ldots \subseteq C^{(k+1)} \subseteq C^{(k)} \subseteq \ldots \subseteq C^{(0)} = C$. The smallest $k$ for which $C_I = C^{(k)}$ is known as the

*Chvátal rank* of $C$. In the same vein, the Chvátal rank of an inequality $\mathbf{c}^\top \mathbf{x} \leq d$ valid for $C_I$ is defined as the smallest $k$ such that $C^{(k)} \subseteq \{\mathbf{x} \in \mathbb{R}^m \, : \, \mathbf{c}^\top \mathbf{x} \leq d\}$.

The finiteness of the Chvátal rank is proven in the literature for bounded real polyhedra [15], unbounded rational polyhedra [58] and conic representable sets in the 0/1-cube [13]. However, the Chvátal rank for unbounded real polyhedra can be infinite as shown by Schrijver [58]. Schrijver also shows that the elementary closure of a rational polyhedron is a rational polyhedron. This result is later generalized to irrational polytopes [24], bounded rational ellipsoids [22], strictly convex bodies [18] and general compact convex sets [19, 10]. As a consequence, the Chvátal rank of these sets is also known to be finite.

## 2.2 The elementary closure of spectrahedra

We now apply the notions from Section 2.1 to integer semidefinite programming problems in standard primal and dual form. On top of the general definition given in the previous section, we derive alternative formulations of the elementary closure of spectrahedra.

Let $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{C} \in \mathcal{S}^n$ and $\mathbf{A_i} \in \mathcal{S}^n$ for all $i \in [m]$. An ISDP in standard primal form is given by:

$$(P_{ISDP}) \begin{cases} \inf & \langle \mathbf{C}, \mathbf{X} \rangle \\ \text{s.t.} & \langle \mathbf{A_i}, \mathbf{X} \rangle = b_i \qquad \text{for all } i \in [m], \\ & \mathbf{X} \succeq \mathbf{0}, \ \mathbf{X} \in \mathbb{Z}^{n \times n}, \end{cases} \tag{4}$$

while an ISDP in standard dual form is given by:

$$(D_{ISDP}) \begin{cases} \sup & \mathbf{b}^\top \mathbf{x} \\ \text{s.t.} & \mathbf{C} - \sum_{i=1}^m \mathbf{A_i} x_i \succeq \mathbf{0} \\ & \mathbf{x} \in \mathbb{Z}^m. \end{cases} \tag{5}$$

Using standard techniques, one can syntactically rewrite an integer SDP from primal form to dual form and vice versa. Consistent with most of the literature, we mainly consider, but not restrict ourselves to, ISDPs in dual form.

The continuous relaxation of the feasible set of (5) is defined as follows:

$$P := \left\{ \mathbf{x} \in \mathbb{R}^m \ : \ \mathbf{C} - \sum_{i=1}^m \mathbf{A_i} x_i \succeq \mathbf{0} \right\}. \tag{6}$$

The set $P$ is a spectrahedron that is a closed, semialgebraic and convex set. We define the integer hull of $P$ to be $P_I := \mathrm{Conv}(P \cap \mathbb{Z}^m)$, i.e., the convex hull of the integral points in $P$. In the remaining part of this section we study the elementary closure, see Definition 1, of spectrahedra in primal and dual standard form.

Using the fact that a matrix $\mathbf{C} - \sum_{i=1}^m \mathbf{A_i} x_i$ is positive semidefinite if and only if $\langle \mathbf{C} - \sum_{i=1}^m \mathbf{A_i} x_i, \mathbf{U} \rangle \geq 0$ for all $\mathbf{U} \in \mathcal{S}^n_+$, we can rewrite $P$ as follows:

$$\begin{aligned} P &= \left\{ \mathbf{x} \in \mathbb{R}^m \ : \ \langle \mathbf{C} - \sum_{i=1}^m \mathbf{A_i} x_i, \mathbf{U} \rangle \geq 0, \, \mathbf{U} \in \mathcal{S}^n_+ \right\} \\ &= \bigcap_{\mathbf{U} \in \mathcal{S}^n_+} \left\{ \mathbf{x} \in \mathbb{R}^m \ : \ \sum_{i=1}^m x_i \langle \mathbf{A_i}, \mathbf{U} \rangle \leq \langle \mathbf{C}, \mathbf{U} \rangle \right\}. \end{aligned} \tag{7}$$

Moreover, since $P$ is a closed convex set, we can write $P$ as the intersection of the halfspaces that contain it:

$$P = \bigcap_{\substack{(\mathbf{c}, d) \in \mathbb{R}^{m+1} \\ P \subseteq \{\mathbf{x} \, : \, \mathbf{c}^\top \mathbf{x} \leq d\}}} \left\{ \mathbf{x} \in \mathbb{R}^m \ : \ \mathbf{c}^\top \mathbf{x} \leq d \right\}. \tag{8}$$

It is clear that all halfspaces in the intersection of (7) are contained in the intersection (8). The following theorem shows the reverse statement.

**Theorem 1.** *Let* $P = \{\mathbf{x} \in \mathbb{R}^m \ : \ \mathbf{C} - \sum_{i=1}^m \mathbf{A_i} x_i \succeq \mathbf{0}\}$ *be a non-empty spectrahedron. Let* $(\mathbf{c}, d) \in \mathbb{R}^{m+1}$ *be such that* $P \subseteq \{\mathbf{x} \in \mathbb{R}^m \ : \ \mathbf{c}^\top \mathbf{x} \leq d\}$*. Then there exists a matrix* $\mathbf{U} \in \mathcal{S}_+^n$ *such that* $\langle \mathbf{A_i}, \mathbf{U} \rangle = c_i$ *for all* $i \in [m]$ *and* $\langle \mathbf{C}, \mathbf{U} \rangle \leq d$*.*

*Proof.* Define $S := \{\mathbf{Z} \in \mathcal{S}_+^n \ : \ \mathbf{Z} = \mathbf{C} - \sum_{i=1}^m \mathbf{A_i} x_i \succeq \mathbf{0}, \mathbf{x} \in P\}$ and let $F_S$ denote the minimal face of $\mathcal{S}_+^n$ that contains $S$ (where $F_S$ can also be the entire cone $\mathcal{S}_n^+$). From the theory on the facial geometry of the SDP cone, see e.g., [23], it follows that there exists some positive integer $k \leq n$ and an orthogonal matrix $\mathbf{W} \in \mathbb{R}^{n \times k}$ such that

$$F_S = \mathbf{W} \mathcal{S}_+^k \mathbf{W}^\top.$$

Now, $P$ can be regularized in the following way:

$$\begin{aligned} P &= \left\{ \mathbf{x} \in \mathbb{R}^m \ : \ \mathbf{Z} = \mathbf{C} - \sum_{i=1}^m \mathbf{A_i} x_i, \, \mathbf{Z} \in F_S \right\} \\ &= \left\{ \mathbf{x} \in \mathbb{R}^m \ : \ \mathbf{W} \mathbf{R} \mathbf{W}^\top = \mathbf{C} - \sum_{i=1}^m \mathbf{A_i} x_i, \, \mathbf{R} \succeq \mathbf{0} \right\} \\ &= \left\{ \mathbf{x} \in \mathbb{R}^m \ : \ \mathbf{R} = \mathbf{W}^\top \mathbf{C} \mathbf{W} - \sum_{i=1}^m \mathbf{W}^\top \mathbf{A_i} \mathbf{W} x_i, \, \mathbf{R} \succeq \mathbf{0} \right\}, \end{aligned}$$

where we used the fact that $\mathbf{W}^\top \mathbf{W} = \mathbf{I}$. Using this regularization, we know that there exists some $\hat{\mathbf{x}} \in \mathbb{R}^m$ such that $\hat{\mathbf{R}} = \mathbf{W}^\top \mathbf{C} \mathbf{W} - \sum_{i=1}^m \mathbf{W}^\top \mathbf{A_i} \mathbf{W} \hat{x}_i \succ 0$. Namely, if such $\hat{\mathbf{R}}$ would not exist, then $S$ would be contained in a proper face of $F_S$, which contradicts the minimality of $F_S$.

Since $P \subseteq \{\mathbf{x} \in \mathbb{R}^m \ : \ \mathbf{c}^\top \mathbf{x} \leq d\}$, we have

$$\begin{aligned} d &\geq \sup_{\mathbf{x}} \left\{ \mathbf{c}^\top \mathbf{x} \ : \ \mathbf{W}^\top \mathbf{C} \mathbf{W} - \sum_{i=1}^m \mathbf{W}^\top \mathbf{A_i} \mathbf{W} x_i \succeq \mathbf{0} \right\} \\ &= \inf_{\overline{\mathbf{U}}} \left\{ \langle \mathbf{W}^\top \mathbf{C} \mathbf{W}, \overline{\mathbf{U}} \rangle \ : \ \langle \mathbf{W}^\top \mathbf{A_i} \mathbf{W}, \overline{\mathbf{U}} \rangle = c_i, \, i \in [m], \, \overline{\mathbf{U}} \succeq \mathbf{0} \right\}, \end{aligned}$$

where strong duality between the primal and dual SDP holds because the former problem contains a Slater feasible point $\hat{\mathbf{x}}$. By the same argument, we know that the infimum of the latter problem is attained. Hence, there exists some $\overline{\mathbf{U}} \succeq 0$ such that:

$$\langle \mathbf{W}^\top \mathbf{A_i} \mathbf{W}, \overline{\mathbf{U}} \rangle = c_i \quad \text{for all } i \in [m], \quad \text{and} \quad \langle \mathbf{W}^\top \mathbf{C} \mathbf{W}, \overline{\mathbf{U}} \rangle \leq d.$$

Using the properties of the trace inner product, it follows that

$$\langle \mathbf{A_i}, \mathbf{W} \overline{\mathbf{U}} \mathbf{W}^\top \rangle = c_i \quad \text{for all } i \in [m], \quad \text{and} \quad \langle \mathbf{C}, \mathbf{W} \overline{\mathbf{U}} \mathbf{W}^\top \rangle \leq d.$$

Taking $\mathbf{U} = \mathbf{W} \overline{\mathbf{U}} \mathbf{W}^\top \succeq \mathbf{0}$ provides the desired result. □

Using the representation of $P$ given by (7) and the result of Theorem 1, we now provide an alternative formulation of the elementary closure for spectrahedra of the form $P$. We have,

$$c(P) = \bigcap_{\substack{\mathbf{U} \in \mathcal{S}_+^n \text{ s.t.} \\ \langle \mathbf{A_i}, \mathbf{U} \rangle \in \mathbb{Z}, \ i \in [m]}} \left\{ x \in \mathbb{R}^m \ : \ \sum_{i=1}^m x_i \langle \mathbf{A_i}, \mathbf{U} \rangle \leq \lfloor \langle \mathbf{C}, \mathbf{U} \rangle \rfloor \right\}. \tag{9}$$

Hence, any possible CG cut for a spectrahedron is constructed by a matrix $\mathbf{U} \in \mathcal{S}_+^n$ such that $\langle \mathbf{A_i}, \mathbf{U} \rangle \in \mathbb{Z}$ for $i \in [m]$.

A similar alternative definition of the elementary closure of spectrahedra in standard primal form can be obtained. Let $Q \subseteq \mathcal{S}^n$ denote the continuous relaxation of the feasible set of (4), i.e.,

$$\begin{aligned} Q &= \{\mathbf{X} \in \mathcal{S}^n \,:\, \langle \mathbf{A_i}, \mathbf{X} \rangle = b_i, i \in [m],\, \mathbf{X} \succeq \mathbf{0}\} \\ &= \left\{\mathbf{X} \in \mathcal{S}^n \,:\, \langle \mathbf{A_i}, \mathbf{X} \rangle = b_i, i \in [m],\, \langle \mathbf{X}, \mathbf{U} \rangle \geq 0,\, \mathbf{U} \in \mathcal{S}^n_+\right\} \\ &= \left\{\mathbf{X} \in \mathcal{S}^n \,:\, \left\langle \mathbf{X}, \mathbf{U} + \sum_{i=1}^m \mathbf{A_i}\lambda_i \right\rangle \geq \sum_{i=1}^m b_i\lambda_i,\, \mathbf{U} \in \mathcal{S}^n_+,\, \boldsymbol{\lambda} \in \mathbb{R}^m\right\}, \end{aligned}$$

where the last equality follows from the fact that the choices $(\mathbf{U}, \boldsymbol{\lambda}) = (\mathbf{0}, \mathbf{e}_i)$ and $(\mathbf{U}, \boldsymbol{\lambda}) = (\mathbf{0}, -\mathbf{e}_i)$ lead to the cuts $\langle \mathbf{A_i}, \mathbf{X} \rangle \geq b_i$ and $\langle \mathbf{A_i}, \mathbf{X} \rangle \leq b_i$, respectively. Now, the elementary closure of $Q$ can be described by the following intersection of CG cuts:

$$c(Q) = \bigcap_{\substack{(\mathbf{U},\boldsymbol{\lambda}) \in \mathcal{S}^n_+ \times \mathbb{R}^m \text{ s.t.} \\ \mathbf{U} + \sum_{i=1}^m \mathbf{A_i}\lambda_i \in \mathbb{Z}^{n\times n}}} \left\{\mathbf{X} \in \mathcal{S}^n \,:\, \left\langle \mathbf{X}, \mathbf{U} + \sum_{i=1}^m \mathbf{A_i}\lambda_i \right\rangle \geq \left\lceil \sum_{i=1}^m b_i\lambda_i \right\rceil \right\}. \tag{10}$$

For many SDPs resulting from applications the spectrahedra that define the feasible sets are contained in the cone of nonnegative vectors or matrices. When $P \subseteq \mathbb{R}^m_+$ or $Q \subseteq \{\mathbf{X} \in \mathbb{R}^{n\times n} \,:\, \mathbf{X} \geq \mathbf{0}\}$, alternative equivalent formulations of the elementary closure can be given, see also [13].

**Theorem 2.** *Let* $P = \left\{\mathbf{x} \in \mathbb{R}^m_+ \,:\, \mathbf{C} - \sum_{i=1}^m \mathbf{A_i}x_i \succeq \mathbf{0}\right\}$. *Then* $c(P)$ *can equivalently be written as*

$$c(P) = \bigcap_{\mathbf{U} \in \mathcal{S}^n_+} \left\{\mathbf{x} \in \mathbb{R}^m \,:\, \sum_{i=1}^m x_i \lfloor \langle \mathbf{A_i}, \mathbf{U} \rangle \rfloor \leq \lfloor \langle \mathbf{C}, \mathbf{U} \rangle \rfloor \right\}. \tag{11}$$

*Similarly, let* $Q = \{\mathbf{X} \in \mathcal{S}^n \,:\, \langle \mathbf{A_i}, \mathbf{X} \rangle = b_i, i \in [m], \mathbf{X} \succeq \mathbf{0}, \mathbf{X} \geq \mathbf{0}\}$. *Then* $c(Q)$ *can equivalently be written as*

$$c(Q) = \bigcap_{(\mathbf{U},\boldsymbol{\lambda}) \in \mathcal{S}^n_+ \times \mathbb{R}^m} \left\{\mathbf{X} \in \mathcal{S}^n \,:\, \left\langle \mathbf{X}, \left\lceil \mathbf{U} + \sum_{i=1}^m \mathbf{A_i}\lambda_i \right\rceil \right\rangle \geq \left\lceil \sum_{i=1}^m b_i\lambda_i \right\rceil \right\}. \tag{12}$$

*Proof.* We prove the statement for the dual form (11). The proof for the primal form is similar.

Let $\overline{c(P)} := \bigcap_{\mathbf{U} \in \mathcal{S}^n_+} \{\mathbf{x} \in \mathbb{R}^m \,:\, \sum_{i=1}^m x_i \lfloor \langle \mathbf{A_i}, \mathbf{U} \rangle \rfloor \leq \lfloor \langle \mathbf{C}, \mathbf{U} \rangle \rfloor\}$ and let $c(P)$ be as given in (9). The inclusion $\overline{c(P)} \subseteq c(P)$ is obvious, as any halfspace in the intersection defining $c(P)$ is also in the intersection defining $\overline{c(P)}$. Now, consider a halfspace $\bar{H} = \{\mathbf{x} \in \mathbb{R}^m \,:\, \sum_{i=1}^m x_i \lfloor \langle \mathbf{A_i}, \mathbf{U} \rangle \rfloor \leq \lfloor \langle \mathbf{C}, \mathbf{U} \rangle \rfloor\}$ for some $\mathbf{U} \in \mathcal{S}^n_+$, that is included in the intersection defining $\overline{c(P)}$. Since $P \subseteq \mathbb{R}^n_+$, we know

$$\begin{aligned} P \subseteq \left\{\mathbf{x} \in \mathbb{R}^m_+ \,:\, \sum_{i=1}^m x_i \langle \mathbf{A_i}, \mathbf{U} \rangle \leq \langle \mathbf{C}, \mathbf{U} \rangle \right\} &\subseteq \left\{\mathbf{x} \in \mathbb{R}^m_+ \,:\, \sum_{i=1}^m x_i \lfloor \langle \mathbf{A_i}, \mathbf{U} \rangle \rfloor \leq \langle \mathbf{C}, \mathbf{U} \rangle \right\} \\ &\subseteq \left\{\mathbf{x} \in \mathbb{R}^m \,:\, \sum_{i=1}^m x_i \lfloor \langle \mathbf{A_i}, \mathbf{U} \rangle \rfloor \leq \langle \mathbf{C}, \mathbf{U} \rangle \right\}. \end{aligned}$$

Now we apply Theorem 1 to the latter halfspace. It follows that there exists a matrix $\mathbf{V} \in \mathcal{S}^n_+$ such that

$$\langle \mathbf{A_i}, \mathbf{V} \rangle = \lfloor \langle \mathbf{A_i}, \mathbf{U} \rangle \rfloor \quad \text{for all } i \in [m], \quad \text{and} \quad \langle \mathbf{C}, \mathbf{V} \rangle \leq \langle \mathbf{C}, \mathbf{U} \rangle.$$

We define the halfspace $H := \{\mathbf{x} \in \mathbb{R}^m \,:\, \sum_{i=1}^m x_i \langle \mathbf{A_i}, \mathbf{V} \rangle \leq \lfloor \langle \mathbf{C}, \mathbf{V} \rangle \rfloor\}$. Since $\lfloor \langle \mathbf{C}, \mathbf{V} \rangle \rfloor \leq \lfloor \langle \mathbf{C}, \mathbf{U} \rangle \rfloor$, it follows that the halfspace $\bar{H}$ contains the halfspace $H$, while $H$ is contained in the intersection of $c(P)$ given in (9). Since this construction can be repeated for all halfspaces in the intersection (11) defining $\overline{c(P)}$, it follows that $c(P) \subseteq \overline{c(P)}$. □

In Section 2.4 we provide a polyhedral description of the elementary closure of spectrahedra that satisfy the notion of total dual integrality.

## 2.3 The Chvátal rank of bounded spectrahedra

In this section we derive several results on the sequence of relaxations resulting from the Chvátal-Gomory procedure. Although some of these results are already known for general compact convex sets, we provide simplified proofs for the case of bounded spectrahedra.

Throughout this section we assume $P$ to be a spectrahedron of the form (6) that is bounded. For unbounded sets it is in general not even clear whether $C^{(k+1)} \subseteq C^{(k)}$.

The following result shows the finiteness of the CG procedure for bounded spectrahedra. Different from other proofs in the literature, our proof does not rely on the first Chvátal-Gomory closure being a rational polytope and can be extended to general bounded conic sets.

**Theorem 3.** *Let $P = \{\mathbf{x} \in \mathbb{R}^m \, : \, \mathbf{C} - \sum_{i=1}^m \mathbf{A_i} x_i \succeq \mathbf{0}\}$ be bounded. Then, $P^{(k)} = P_I$ for some finite $k$.*

*Proof.* Since $P$ is bounded, there exists an $M > 0$ such that $|x_i| < M$ for all $i \in [m]$. Now, let $Q$ be the following polytope:

$$Q := \left\{ \mathbf{x} \in \mathbb{R}^m \, : \, \begin{array}{r} \begin{bmatrix} \operatorname{diag}(\mathbf{A_1}) & \dots & \operatorname{diag}(\mathbf{A_m}) \end{bmatrix} \mathbf{x} \leq \operatorname{diag}(\mathbf{C}) \\ -M \leq x_i \leq M \quad \text{for all } i \in [m] \end{array} \right\}.$$

Clearly, $P \subseteq Q$, from which it follows that $P_I \subseteq Q_I$. For each $\mathbf{x} \in (Q_I \setminus P_I) \cap \mathbb{Z}^m$, we can find a halfspace containing $P$ that does not contain $\mathbf{x}$. As there are finitely many such points, we define $\bar{Q}$ as the intersection of $Q$ and all such halfspaces. The polytope $\bar{Q}$ contains $P$ and is such that $\bar{Q}_I = P_I$. It follows that $P^{(k)} \subseteq \bar{Q}^{(k)}$ for all $k \geq 0$. Using the result of Chvátal [15], we know that there exists a finite $k^*$ such that $\bar{Q}^{(k^*)} = \bar{Q}_I$. Consequently, we have

$$\bar{Q}_I = P_I \subseteq P^{(k^*)} \subseteq \bar{Q}^{(k^*)} = \bar{Q}_I,$$

from which it follows that $P^{(k^*)} = P_I$. □

Next, we aim to prove a homogeneity property of the CG procedure for bounded spectrahedra, which states that the elementary closure operation commutes with taking the intersection with supporting hyperplanes. This property plays a key role in showing that the elementary closure of $P$ is a rational polytope, following the proof of Braun and Pokutta [10]. We provide a simplified proof of this property for bounded spectrahedra, which can be seen as the conic analogue to a polyhedral result of Schrijver [58]. In the proof we restrict ourselves to halfspaces of the form $\{\mathbf{x} \in \mathbb{R}^m \, : \, \mathbf{w}^\top \mathbf{x} \leq d\}$ where $\mathbf{w} \in \mathbb{Z}^m$ and $d \in \mathbb{R}$. It follows from Proposition 1 that these halfspaces are sufficient to describe a compact convex set.

Before we show the main theorem, we need a chain of intermediate results, starting with a semidefinite version of the theorem of alternatives, see e.g., Balakrishnan and Vandenberghe [6].

**Proposition 2** (Theorem of the alternatives for SDP [6])**.** *Let $\mathbf{C}, \mathbf{A_1}, \dots, \mathbf{A_m} \in \mathcal{S}^n$. Then at most one of the following is true:*

1. *There exists an $\mathbf{X} \succ \mathbf{0}$, $\langle \mathbf{A_i}, \mathbf{X} \rangle = 0$ for all $i \in [m]$ and $\langle \mathbf{C}, \mathbf{X} \rangle \leq 0$;*
2. *There exists an $\mathbf{x} \in \mathbb{R}^m$ such that $\mathbf{C} - \sum_{i=1}^m \mathbf{A_i} x_i \succneq \mathbf{0}$.*

*Moreover, if there exists no $\mathbf{x} \in \mathbb{R}^m$ such that $\sum_{i=1}^m \mathbf{A_i} x_i \succneq \mathbf{0}$, then exactly one of the statements above is true.*

The following proposition shows that the assumption of boundedness is needed in order to state that exactly one of the statements in Proposition 2 is true.

**Proposition 3.** *Let $P = \{\mathbf{x} \in \mathbb{R}^m \, : \, \mathbf{C} - \sum_{i=1}^m \mathbf{A_i} x_i \succeq \mathbf{0}\}$ be a non-empty and bounded spectrahedron. Then there does not exist an $\mathbf{x} \in \mathbb{R}^m$ such that $\sum_{i=1}^m \mathbf{A_i} x_i \succneq \mathbf{0}$.*

*Proof.* Since $P$ is non-empty, there exists a point $\mathbf{x}^* \in P$, i.e., $\mathbf{C} - \sum_{i=1}^m \mathbf{A_i} x_i^* \succeq \mathbf{0}$. Now suppose there exists a point $\hat{\mathbf{x}}$ such that $\sum_{i=1}^m \mathbf{A_i} \hat{x}_i \succneqq \mathbf{0}$. Then clearly $\hat{\mathbf{x}} \neq \mathbf{0}_m$ and for all $t \geq 0$ we have

$$\mathbf{C} - \sum_{i=1}^m \mathbf{A_i} x_i^* + t \sum_{i=1}^m \mathbf{A_i} \hat{x}_i = \mathbf{C} - \sum_{i=1}^m \mathbf{A_i}(x_i^* - t\hat{x}_i) \succeq \mathbf{0},$$

i.e., $\mathbf{x}^* - t\hat{\mathbf{x}} \in P$ for all $t \geq 0$. Thus, $P$ is unbounded, so such $\hat{\mathbf{x}}$ cannot exist. □

Finally, we also need a variant of Dirichlet's approximation theorem, see e.g., [61].

**Proposition 4** (Dirichlet's Approximation Theorem, e.g., [61])**.** *Let $d \in \mathbb{R}$ and $N \geq 2$ be a positive integer. Then there exist integers $p$ and $q$ with $p > 0$ such that $|pd - q| \leq \frac{1}{N}$.*

We now derive its one-sided variant below.

**Corollary 1** (One-sided Approximation Theorem)**.** *Let $d \in \mathbb{R}$ and $N \geq 2$ be a positive integer number. Then there exists an integer $p \in \mathbb{Z}_+$ such that*

$$pd - \lfloor pd \rfloor \leq \frac{1}{N}.$$

*Proof.* By Dirichlet's Theorem, we know that for the given $d$ and $N$, there exist integers $q_1$ and $q_2$ with $q_1 > 0$ such that $|q_1 d - q_2| \leq \frac{1}{N}$. If $q_1 d \geq q_2$, then we have

$$q_1 d - \lfloor q_1 d \rfloor \leq q_1 d - q_2 = |q_1 d - q_2| \leq \frac{1}{N},$$

so the choice $p = q_1$ leads to the desired result. Next, we consider the case $q_1 d < q_2$, for which we have $-\frac{1}{N} \leq q_1 d - q_2 < 0$. Let $M \geq 1$ be the smallest integer such that $M(q_1 d - q_2) \leq -\frac{N-1}{N}$, which exists because $q_1 d - q_2 < 0$. For this $M$ we must have $-1 \leq M(q_1 d - q_2)$. Namely, if $M(q_1 d - q_2) < -1$, then $(M-1)(q_1 d - q_2) \leq -\frac{N-1}{N}$, contradicting the minimality of $M$. Thus,

$$-1 \leq M(q_1 d - q_2) \leq -\frac{N-1}{N} \quad \iff \quad 0 \leq Mq_1 d - (Mq_2 - 1) \leq \frac{1}{N}.$$

Since $Mq_2 - 1$ is integer, it follows that

$$Mq_1 d - \lfloor Mq_1 d \rfloor \leq Mq_1 d - (Mq_2 - 1) \leq \frac{1}{N},$$

so taking $p = Mq_1$ gives the desired result. □

We are now ready to present a simplified proof for the homogeneity property of the elementary closure of bounded spectrahedra.

**Theorem 4** (Homogeneity property of elementary closure)**.** *Let $P = \{\mathbf{x} \in \mathbb{R}^m : \mathbf{C} - \sum_{i=1}^m \mathbf{A_i} x_i \succeq \mathbf{0}\}$ be a bounded spectrahedron that is contained in a halfspace $\{\mathbf{x} \in \mathbb{R}^m : \mathbf{w}^\top \mathbf{x} \leq d\}$ with $\mathbf{w} \in \mathbb{Z}^m$ and $d \in \mathbb{R}$. Let $K := \{\mathbf{x} \in \mathbb{R}^m : \mathbf{w}^\top \mathbf{x} = d\}$. Then $c(P) \cap K = c(P \cap K)$.*

*Proof.* If $P$ is empty the claim is obvious, hence we assume that $P$ is non-empty.

The inclusion $c(P \cap K) \subseteq c(P) \cap K$ is trivial. In order to prove the reverse statement, we assume that $H$ is a rational halfspace containing $P \cap K$, i.e., $H = \{\mathbf{x} \in \mathbb{R}^m : \mathbf{v}^\top \mathbf{x} \leq \alpha\}$ where $\mathbf{v}$ is a vector of relative prime integers. It suffices to show that there exists a halfspace $\hat{H}$ containing $P$ such that $\hat{H}_I \cap K \subseteq H_I$. As $c(P \cap K)$ is the intersection of all such halfspaces $H$, we establish $c(P) \cap K \subseteq c(P \cap K)$.

For each $i \in [m]$ we define the following extended matrix $\tilde{\mathbf{A}}_{\mathbf{i}} \in \mathcal{S}^{n+2}$:

$$\tilde{\mathbf{A}}_{\mathbf{i}} := \begin{pmatrix} \mathbf{A_i} & \mathbf{0} & \mathbf{0} \\ \mathbf{0}^\top & -w_i & 0 \\ \mathbf{0}^\top & 0 & -v_i \end{pmatrix}.$$

We first show that there does not exist an $\mathbf{x} \in \mathbb{R}^m$ such that $\sum_{i=1}^m \tilde{\mathbf{A}}_\mathbf{i} x_i \succeq_{\neq} \mathbf{0}$. For the sake of contradiction, suppose such a vector exists, i.e., we have $\sum_{i=1}^m \mathbf{A}_\mathbf{i} \tilde{x}_i \succeq \mathbf{0}$, $\mathbf{w}^\top \tilde{\mathbf{x}} \leq 0$ and $\mathbf{v}^\top \tilde{\mathbf{x}} \leq 0$ for some $\tilde{\mathbf{x}}$, but not all of them are satisfied with equality. Since $P$ is non-empty and bounded, it follows from Proposition 3 that there does not exist an $\mathbf{x} \in \mathbb{R}^m$ such that $\sum_{i=1}^m \mathbf{A}_\mathbf{i} x_i \succeq_{\neq} \mathbf{0}$. Hence, we must have $\sum_{i=1}^m \mathbf{A}_\mathbf{i} \tilde{x}_i = \mathbf{0}$. This implies that either $\mathbf{w}^\top \tilde{\mathbf{x}} < 0$ or $\mathbf{v}^\top \tilde{\mathbf{x}} < 0$, or both.

Since $P$ is contained in $\{\mathbf{x} \in \mathbb{R}^m \ : \ \mathbf{w}^\top \mathbf{x} \leq d\}$, it follows from Theorem 1 that there exists $T \succeq \mathbf{0}$ such that $\langle \mathbf{A}_\mathbf{i}, \mathbf{T} \rangle = w_i$ for all $i \in [m]$. Since $\sum_{i=1}^m \mathbf{A}_\mathbf{i} \tilde{x}_i = \mathbf{0}$, we have

$$\left\langle \sum_{i=1}^m \mathbf{A}_\mathbf{i} \tilde{x}_i, \mathbf{T} \right\rangle = \sum_{i=1}^m \tilde{x}_i \langle \mathbf{A}_\mathbf{i}, \mathbf{T} \rangle = \mathbf{w}^\top \tilde{\mathbf{x}} = 0.$$

Since $P \cap K$ is contained in $H = \{\mathbf{x} \in \mathbb{R}^m \ : \ \mathbf{v}^\top \mathbf{x} \leq \alpha\}$, we can in a similar fashion show that $v_i = \langle \mathbf{A}_\mathbf{i}, \mathbf{S} \rangle + \beta w_i$ for some $\mathbf{S} \succeq \mathbf{0}$ and $\beta \in \mathbb{R}$. From this it follows that $\mathbf{v}^\top \tilde{\mathbf{x}} = 0$. We conclude that there exists no $\mathbf{x} \in \mathbb{R}^m$ such that $\sum_{i=1}^m \tilde{\mathbf{A}}_\mathbf{i} x_i \succeq_{\neq} \mathbf{0}$.

Next, we define the following extended matrix $\tilde{\mathbf{C}} \in \mathcal{S}^{n+2}$ and parameter $\epsilon > 0$:

$$\tilde{\mathbf{C}} := \begin{pmatrix} \mathbf{C} & \mathbf{0} & \mathbf{0} \\ \mathbf{0}^\top & -d & 0 \\ \mathbf{0}^\top & 0 & -(\alpha + \epsilon) \end{pmatrix} \quad \text{and} \quad \epsilon := \begin{cases} \frac{1}{2} \left( \lceil \alpha \rceil - \alpha \right) & \text{if } \alpha \text{ is not integer,} \\ \frac{1}{2} & \text{otherwise.} \end{cases}$$

Since $P \cap K$ is contained in $H$, it follows that $(P \cap K) \cap \{\mathbf{x} \in \mathbb{R}^m \ : \ \mathbf{v}^\top \mathbf{x} \geq \alpha + \epsilon\} = \emptyset$. Equivalently, we know that there does not exist an $\mathbf{x} \in \mathbb{R}^m$ such that $\tilde{\mathbf{C}} - \sum_{i=1}^m \tilde{\mathbf{A}}_\mathbf{i} x_i \succeq_{\neq} \mathbf{0}$. We can now apply Proposition 2 to this system, from where it follows that the first of the two alternative statements should be satisfied. Hence, there exist $\hat{\mathbf{U}} \succ \mathbf{0}$, $\lambda > 0$ and $\mu > 0$ such that $\langle \mathbf{A}_\mathbf{i}, \hat{\mathbf{U}} \rangle - w_i \lambda - v_i \mu = 0$ for all $i \in [m]$ and $\langle \mathbf{C}, \hat{\mathbf{U}} \rangle - d\lambda - (\alpha + \epsilon)\mu \leq 0$. Without loss of generality, we may assume that $\mu = 1$ and we define

$$\hat{\alpha} := \langle \mathbf{C}, \hat{\mathbf{U}} \rangle \quad \text{and} \quad \hat{v}_i := \langle \mathbf{A}_\mathbf{i}, \hat{\mathbf{U}} \rangle \text{ for all } i \in [m].$$

It follows from above that this particular $\hat{\alpha}$ and $\hat{\mathbf{v}}$ satisfy

$$\hat{\alpha} \leq \alpha + \epsilon + d\lambda \quad \text{and} \quad \hat{v}_i = v_i + w_i \lambda \text{ for all } i \in [m]. \tag{13}$$

Also, since $\hat{\mathbf{U}} \succ \mathbf{0}$, we know that for all $\mathbf{x} \in P$ we have

$$\hat{\mathbf{v}}^\top \mathbf{x} = \sum_{i=1}^m \langle \mathbf{A}_\mathbf{i}, \hat{\mathbf{U}} \rangle x_i = \left\langle \sum_{i=1}^m \mathbf{A}_\mathbf{i} x_i, \hat{\mathbf{U}} \right\rangle \leq \langle \mathbf{C}, \hat{\mathbf{U}} \rangle = \hat{\alpha}, \tag{14}$$

where we use the fact that $\langle \mathbf{C} - \sum_{i=1}^m \mathbf{A}_\mathbf{i} x_i, \hat{\mathbf{U}} \rangle \geq 0$. Observe that the tuple $(\lambda, \hat{\mathbf{v}}, \hat{\alpha})$ can be replaced by $(\lambda + \lambda_0, \hat{\mathbf{v}} + \lambda_0 \mathbf{w}, \hat{\alpha} + \lambda_0 d)$ for all $\lambda_0 \geq 0$ without affecting (13) and (14), where for the maintenance of (14) we use the fact that $P \subseteq \{\mathbf{x} \in \mathbb{R}^m \ : \ \mathbf{w}^\top \mathbf{x} \leq d\}$. Now we choose $\lambda_0$ such that $\lambda + \lambda_0 \in \mathbb{Z}_+$ and $d(\lambda + \lambda_0) - \lfloor d(\lambda + \lambda_0) \rfloor < \epsilon$, which can be done by Corollary 1. Moreover, we define $d_f := d(\lambda + \lambda_0) - \lfloor d(\lambda + \lambda_0) \rfloor$.

Define $\hat{H} := \{\mathbf{x} \in \mathbb{R}^m \ : \ (\hat{\mathbf{v}} + \lambda_0 \mathbf{w})^\top \mathbf{x} \leq \hat{\alpha} + \lambda_0 d\}$. It follows from (14) that $P \subseteq \hat{H}$. Moreover, we have

$$\begin{aligned} \hat{H}_I \cap K &\subseteq \{\mathbf{x} \in \mathbb{R}^m \ : \ (\hat{\mathbf{v}} + \lambda_0 \mathbf{w})^\top \mathbf{x} \leq \lfloor \hat{\alpha} + \lambda_0 d \rfloor\} \cap \{\mathbf{x} \in \mathbb{R}^m \ : \ \mathbf{w}^\top \mathbf{x} = d\} \\ &\subseteq \{\mathbf{x} \in \mathbb{R}^m \ : \ \mathbf{v}^\top \mathbf{x} + \mathbf{w}^\top \mathbf{x}(\lambda + \lambda_0) \leq \lfloor \alpha + \epsilon + d(\lambda + \lambda_0) \rfloor, \mathbf{w}^\top \mathbf{x} = d\} \\ &\subseteq \{\mathbf{x} \in \mathbb{R}^m \ : \ \mathbf{v}^\top \mathbf{x} + d_f \leq \lfloor \alpha + \epsilon + d_f \rfloor\} \\ &\subseteq \{\mathbf{x} \in \mathbb{R}^m \ : \ \mathbf{v}^\top \mathbf{x} \leq \lfloor \alpha \rfloor\} = H_I, \end{aligned}$$

where the last inclusion follows from the fact that $d_f \geq 0$ and $\epsilon + d_f < 1$ if $\alpha$ is integer and $\epsilon + d_f < \lceil \alpha \rceil - \alpha$ otherwise. □

The result of Theorem 4 holds for any halfspace $\{\mathbf{x} \in \mathbb{R}^m : \mathbf{w}^\top \mathbf{x} \leq d\}$ with $\mathbf{w} \in \mathbb{Z}^m$ containing $P$. In particular, it holds for all such halfspaces that support $P$, meaning that $P \cap K \neq \emptyset$, where $K$ is the corresponding hyperplane. In such case, the set $P \cap K$ defines a face of the spectrahedron. It is known that all proper faces of spectrahedra are exposed, meaning that they can be obtained as the intersection of $P$ with a supporting hyperplane. Note, however, that for the faces of bounded spectrahedra these hyperplanes are not necessarily such that the entries in $\mathbf{w}$ are integral, even if the data matrices describing the spectrahedron are rational (as is the case for polyhedra).

Homogeneity plays a key role in Braun and Pokutta's [10] proof for the polyhedrality of the elementary closure of compact convex sets. For the sake of completeness, we include this result here for the case of bounded spectrahedra.

**Theorem 5** (Dadush et al., [19], Braun and Pokutta [10])**.** *The elementary closure $c(P)$ of a bounded spectrahedron $P$ is a rational polytope.*

From Theorem 5 and the fact that the elementary closure of a rational polytope is again a rational polytope [58], it follows that the finite sequence

$$P = P^{(0)} \supseteq P^{(1)} \supseteq \ldots \supseteq P^{(k)} \supseteq P^{(k+1)} \supseteq \ldots \supseteq P_I,$$

consists of rational polyhedra from the first closure onwards.

## 2.4 A closed-form expression for the elementary closure of spectrahedra

In this section we derive a class of spectrahedra for which we can find an explicit expression for the elementary closure. For rational polyhedra such an expression can be derived from a total dual integral representation of the linear system [58]. It is therefore not surprising that a similar construction can be applied for bounded spectrahedra, albeit with a bit more technicalities.

Recently, De Carli Silva and Tunçel [11] introduced a notion of total dual integrality for SDPs. The authors of [11] argue that the term integrality in SDPs should be defined with care. For instance, the rank-one property that is sometimes used in the literature as the notion of SDP integrality is proven to be primal-dual asymmetric and therefore not the favoured choice. Instead, the following property is used to define integrality in SDPs:

**Definition 2** (Property (P$\mathbb{Z}$))**.** A matrix $\mathbf{X} \in \mathcal{S}_n^+$ satisfies integrality property (P$\mathbb{Z}$) if

$$\mathbf{X} = \sum_{S \subseteq [n]} \mathbf{y_S} \mathbb{1}_\mathbf{S} \mathbb{1}_\mathbf{S}^\top \quad \text{for some } \mathbf{y} : \mathcal{P}([n]) \to \mathbb{Z}_+. \tag{P$\mathbb{Z}$}$$

Note that the matrices $\mathbf{X}$ that satisfy property (P$\mathbb{Z}$) are also integral in the sense that $\mathbf{X} \in \mathbb{Z}^{n \times n}$. To overcome confusion between these definitions, we will always explicitly refer to property (P$\mathbb{Z}$) if that notion is meant.

Now we present the definition of total dual integrality for SDPs as introduced by [11].

**Definition 3** (Total dual integrality)**.** A linear matrix inequality $\mathbf{C} - \sum_{i=1}^m \mathbf{A_i} x_i \succeq \mathbf{0}$ is called totally dual integral if, for every integral $\mathbf{b} \in \mathbb{Z}^m$, the SDP dual to $\sup \left\{ \mathbf{b}^\top \mathbf{x} : \mathbf{C} - \sum_{i=1}^m \mathbf{A_i} x_i \succeq 0 \right\}$ has an optimal solution satisfying property (P$\mathbb{Z}$) whenever it has an optimal solution.

The following theorem shows a class of spectrahedra for which the rational polytope describing the elementary closure can be explicitly found.

**Theorem 6.** *Let $P = \{\mathbf{x} \in \mathbb{R}^m : \mathbf{C} - \sum_{i=1}^m \mathbf{A_i} x_i \succeq \mathbf{0}\}$ with $\mathbf{A_i} \in \mathbb{Z}^{n \times n} \cap \mathcal{S}^n$ for all $i \in [m]$ be such that the system $\mathbf{C} - \sum_{i=1}^m \mathbf{A_i} x_i \succeq \mathbf{0}$ is totally dual integral and satisfies Slater's condition. Define $\mathbf{B} \in \mathbb{Z}^{\mathcal{P}([n]) \times m}$ and $\mathbf{d} \in \mathbb{Z}^{\mathcal{P}([n])}$ such that:*

$$B_{S,i} := \left\langle \mathbf{A_i}, \mathbb{1}_\mathbf{S} \mathbb{1}_\mathbf{S}^\top \right\rangle \quad \text{and} \quad d_S := \left\lfloor \left\langle \mathbf{C}, \mathbb{1}_\mathbf{S} \mathbb{1}_\mathbf{S}^\top \right\rangle \right\rfloor,$$

*for all $S \subseteq [n]$ and $i \in [m]$. Then,*

$$c(P) = Q := \{\mathbf{x} \in \mathbb{R}^m : \mathbf{B}\mathbf{x} \leq \mathbf{d}\}.$$

*Proof.* To prove that $c(P) \subseteq Q$, let $S \subseteq [n]$ and define $\mathbf{U_S} := \mathbb{1}_\mathbf{S}\mathbb{1}_\mathbf{S}^\top$. Then, $\mathbf{U_S} \in \mathcal{S}^n_+$ and $\langle \mathbf{A_i}, \mathbf{U_S}\rangle \in \mathbb{Z}$ for all $i \in [m]$ by the integrality of the matrices $\mathbf{A_i}$. Consequently, we know that $P \subseteq \{\mathbf{x} \in \mathbb{R}^m : \sum_{i=1}^m x_i \langle \mathbf{A_i}, \mathbf{U_S}\rangle \leq \langle \mathbf{C}, \mathbf{U_S}\rangle\}$. It follows from (9) that $c(P) \subseteq \{\mathbf{x} \in \mathbb{R}^m : \sum_{i=1}^m x_i \langle \mathbf{A_i}, \mathbf{U_S}\rangle \leq \lfloor \langle \mathbf{C}, \mathbf{U_S}\rangle \rfloor\}$. Since all inequalities in $\mathbf{Bx} \leq \mathbf{d}$ are of this form, it follows that $c(P) \subseteq Q$.

To prove the reverse direction, let $H := \left\{\mathbf{x} \in \mathbb{R}^m : \mathbf{w}^\top\mathbf{x} \leq q\right\}$ be a rational halfspace containing $P$. Without loss of generality, we may assume that $\mathbf{w} \in \mathbb{Z}^m$. Since $P \subseteq H$, we have

$$q \geq \sup_{\mathbf{x}} \left\{ \mathbf{w}^\top\mathbf{x} : \mathbf{C} - \sum_{i=1}^m \mathbf{A_i} x_i \succeq \mathbf{0} \right\} \tag{15}$$

$$= \inf_{\mathbf{X}} \left\{ \langle \mathbf{C}, \mathbf{X}\rangle : \langle \mathbf{A_i}, \mathbf{X}\rangle = w_i,\, i \in [m],\, \mathbf{X} \succeq \mathbf{0} \right\}, \tag{16}$$

where strong duality among (15) and (16) holds since the former problem has a Slater feasible point. By the same argument, we know that the infimum in (16) is attained. Since $\mathbf{C} - \sum_{i=1}^m \mathbf{A_i} x_i \succeq \mathbf{0}$ is totally dual integral, it follows that there exists an optimal solution $\hat{\mathbf{X}}$ to (16) satisfying property (P$\mathbb{Z}$). In other words, there exists an $\hat{\mathbf{y}} \in \mathbb{Z}^{\mathcal{P}([n])}$ such that

$$\hat{\mathbf{X}} = \sum_{S \subseteq [n]} \hat{y}_S \mathbb{1}_\mathbf{S}\mathbb{1}_\mathbf{S}^\top, \quad \langle \mathbf{A_i}, \hat{\mathbf{X}}\rangle = w_i \text{ for all } i \in [m], \quad \hat{\mathbf{X}} \succeq 0.$$

Consequently, we have

$$\lfloor q \rfloor \geq \lfloor \langle \mathbf{C}, \hat{\mathbf{X}}\rangle \rfloor = \left\lfloor \sum_{S \subseteq [n]} \hat{y}_S \left\langle \mathbf{C}, \mathbb{1}_\mathbf{S}\mathbb{1}_\mathbf{S}^\top \right\rangle \right\rfloor \geq \sum_{S \subseteq [n]} \hat{y}_S \left\lfloor \left\langle \mathbf{C}, \mathbb{1}_\mathbf{S}\mathbb{1}_\mathbf{S}^\top \right\rangle \right\rfloor = \mathbf{d}^\top \hat{\mathbf{y}}.$$

Now, consider the following linear optimization problem and its corresponding dual:

$$\max\{\mathbf{w}^\top\mathbf{x} : \mathbf{Bx} \leq \mathbf{d}\} = \min\{\mathbf{d}^\top\mathbf{y} : \mathbf{y} \geq \mathbf{0}, \mathbf{y}^\top\mathbf{B} = \mathbf{w}^\top\}.$$

Since $\hat{\mathbf{y}} \geq \mathbf{0}$ and $(\hat{\mathbf{y}}^\top\mathbf{B})_i = \sum_{S \subseteq [n]} \hat{y}_S \langle \mathbf{A_i}, \mathbb{1}_\mathbf{S}\mathbb{1}_\mathbf{S}^\top\rangle = \langle \mathbf{A_i}, \hat{\mathbf{X}}\rangle = w_i$, the solution $\hat{\mathbf{y}}$ is feasible for the minimization problem above. This yields

$$\max\{\mathbf{w}^\top\mathbf{x} : \mathbf{Bx} \leq \mathbf{d}\} \leq \mathbf{d}^\top\hat{\mathbf{y}} \leq \lfloor q \rfloor.$$

Hence, $Q \subseteq \left\{\mathbf{x} \in \mathbb{R}^m : \mathbf{w}^\top\mathbf{x} \leq \lfloor q \rfloor\right\}$. Since this holds for all rational halfspaces $H$, it follows that $Q \subseteq c(P)$. ☐

The application of Theorem 6 is two-fold. At first, it can be used to characterize bounded spectrahedra for which $P = P_I$. Namely, if the matrix $\mathbf{C}$ is such that $\langle \mathbf{C}, \mathbb{1}_\mathbf{S}\mathbb{1}_\mathbf{S}^\top\rangle \in \mathbb{Z}$ for all $S \subseteq [n]$, then $P \subseteq Q$. This implies that the chain $Q = c(P) \subseteq P \subseteq Q$ holds with equality, hence $c(P) = P$. As $P^{(k)} = P_I$ for some finite $k$ for all bounded spectrahedra, we must have $P = P_I$. De Carli Silva and Tunçel [11] show that this, for example, happens for the SDP formulation of the Lovász theta function when the underlying graph is perfect.

On the other hand, when $\mathbf{C}$ is such that $\langle \mathbf{C}, \mathbb{1}_\mathbf{S}\mathbb{1}_\mathbf{S}^\top\rangle \notin \mathbb{Z}$ for some $S$, then it is possible that $c(P) \neq P$. In that case, the polyhedron $Q$ can be used as a closed-form expression for $c(P)$. A classical result states that for any rational polyhedron $P$ there exists a totally dual integral system that describes $P$ [40]. If a spectrahedral equivalent of this result exists, then finding the elementary closure would boil down to finding a TDI system representing the spectrahedron. This is a direction of future research.

## 2.5 Strengthened Chvátal-Gomory cuts

Dash et al. [21] consider a strengthening of the CG cut for rational polyhedra. We briefly present here their approach that can be applied to general convex sets.

For all $\mathbf{c} \in \mathbb{Z}^m$ such that $P \subseteq \{\mathbf{x} \in \mathbb{R}^m \,:\, \mathbf{c}^\top \mathbf{x} \leq d\}$, the corresponding CG cut is $\mathbf{c}^\top \mathbf{x} \leq \lfloor d \rfloor$. The validity of this cut follows from the inequality

$$\lfloor d \rfloor \geq \max \left\{ \mathbf{c}^\top \mathbf{x} \,:\, \mathbf{c}^\top \mathbf{x} \leq d,\, \mathbf{x} \in \mathbb{Z}^m \right\},$$

where equality holds if the entries in $\mathbf{c}$ are relatively prime. However, the gap between $\lfloor d \rfloor$ and $\max\{\mathbf{c}^\top \mathbf{x} \,:\, \mathbf{x} \in P \cap \mathbb{Z}^m\}$ can generally be very large. In order to reduce this gap, suppose that we know that $P \cap \mathbb{Z}^m$ is contained in some set $S \subseteq \mathbb{Z}^m$. Given a valid inequality $\mathbf{c}^\top \mathbf{x} \leq d$ for $P$, we define

$$\lfloor d \rfloor_{S,c} := \max \left\{ \mathbf{c}^\top \mathbf{x} \,:\, \mathbf{c}^\top \mathbf{x} \leq d,\, \mathbf{x} \in S \right\}. \tag{17}$$

By construction, $\mathbf{c}^\top \mathbf{x} \leq \lfloor d \rfloor_{S,c}$ is valid for $P \cap \mathbb{Z}^m$. We refer to these type of cuts as $S$-Chvátal-Gomory ($S$-CG) cuts. These cuts are at least as strong as standard CG cuts, since taking $S = \mathbb{Z}^m$ provides the standard CG cut. The geometric interpretation of an $S$-CG cut is that we shift the hyperplane $\{\mathbf{x} \in \mathbb{R}^m \,:\, \mathbf{c}^\top \mathbf{x} = d\}$ in the direction of $P \cap \mathbb{Z}^m$ until it hits a point in $S$. An example for $S$ is the set $\{0,1\}^m$ in the case of binary optimization problems.

# 3 A CG-based branch-and-cut algorithm for ISDPs

Solving ISDPs is a relatively new field of research for which only a few general-purpose solution approaches have been proposed. Gally et al. [38] present a B&B algorithm called SCIP-SDP for solving (M)ISDPs with continuous SDPs as subproblems. Alternatively, Kobayashi and Takano [44] propose a B&C algorithm that initially relaxes the PSD constraint and solves a mixed integer linear program (MILP), where the PSD constraint is imposed dynamically via cutting planes. Numerical results in [44] show that the B&C algorithm of [44] outperforms the B&B algorithm of [38]. The difference can be explained by the high performance of the current MILP solvers compared to the much less robust conic interior point methods that are used in [38]. It has to be noted, however, that an older version of SCIP-SDP with DSDP [8] as SDP solver was used in the computational results of [44], see also [48]. Another project that encounters MISDPs is YALMIP [47], although its performance is significantly inferior compared to the other two methods [38, 44].

In this section we present a generic B&C algorithm for solving ISDPs that exploits CG cuts of the underlying spectrahedron. This algorithm can be seen as an extension of the works of [44, 13]. In Section 3.1 we provide a general B&C framework for ISDPs which uses a cut generation routine based on $S$-CG cuts. Section 3.2 presents a separation routine for the special class of binary SDPs. This routine can efficiently generate deep cutting planes by exploiting both the positive semidefinite and the integrality constraints.

## 3.1 Generic Branch-and-Cut framework

We start this section by presenting the B&C framework proposed by Kobayashi and Takano [44] for ISDPs in standard dual form, see (5). However, the approach can be extended to problems in primal form in a straightforward way. We define

$$\mathcal{F} := \left\{ \mathbf{x} \in \mathbb{R}^m \,:\, \operatorname{diag}\left( \mathbf{C} - \sum_{i=1}^m \mathbf{A_i} x_i \right) \geq 0 \right\}, \tag{18}$$

which can be seen as the polyhedral part of the spectrahedron $P$, see (6). We assume that the problem of maximizing $\mathbf{b}^\top \mathbf{x}$ over $\mathcal{F}$ is bounded, which is a non-restrictive assumption whenever the original ISDP is bounded.

The B&C algorithm of [44] is based on a dynamic constraint generation known as a lazy constraint callback. The algorithm starts with optimizing over the set $\mathcal{F} \cap \mathbb{Z}^m$, i.e.,

$$\max \left\{ \mathbf{b}^\top \mathbf{x} \,:\, \mathbf{x} \in \mathcal{F} \cap \mathbb{Z}^m \right\}, \tag{19}$$

which can be solved using a B&B algorithm. Whenever an integer point $\hat{\mathbf{x}}$ is found in the branching tree, it is verified whether $\mathbf{C} - \sum_{i=1}^m \mathbf{A_i}\hat{x}_i \succeq \mathbf{0}$ is satisfied. If so, the solution is feasible for $(D_{ISDP})$ and provides a possibly better lower bound to prune other nodes in the tree. If not, then $\langle \mathbf{C} - \sum_{i=1}^m \mathbf{A_i}\hat{x}_i, \mathbf{dd}^\top \rangle < 0$ where $\mathbf{d}$ is a normalized eigenvector corresponding to the smallest eigenvalue of $\mathbf{C} - \sum_{i=1}^m \mathbf{A_i}\hat{x}_i$. This leads to the following valid constraint for $(D_{ISDP})$:

$$\left\langle \mathbf{C} - \sum_{i=1}^m \mathbf{A_i} x_i, \mathbf{dd}^\top \right\rangle \geq 0 \quad \Longleftrightarrow \quad \sum_{i=1}^m \langle \mathbf{A_i}, \mathbf{dd}^\top \rangle x_i \leq \langle \mathbf{C}, \mathbf{dd}^\top \rangle, \tag{20}$$

which separates $\hat{\mathbf{x}}$ from $P$. Now the algorithm adds to $\mathcal{F}$ a cut of type (20) to cut off the current point and continues the branching scheme using this additional constraint. This process is iterated until the optimality of a solution for $(D_{ISDP})$ is guaranteed by the B&B procedure.

Among all dual matrices $\mathbf{U} \in \mathcal{S}_+^n$, it follows from the Rayleigh principle that $\langle \mathbf{C} - \sum_{i=1}^m \mathbf{A_i}\hat{x}_i, \mathbf{U} \rangle$ is minimized by taking $\mathbf{U} = \mathbf{dd}^\top$ with $\mathbf{d}$ as defined above. In that sense, the cut (20) is the strongest cut with respect to violation in the PSD constraint. However, this type of separator ignores the fact that an optimal solution is also integer. In fact, the current point $\hat{\mathbf{x}}$ is separated from $P$, whereas ideally we want to separate it from $P_I$. We now propose an alternative stronger separator based on the CG procedure that exploits both the PSD and the integrality constraint.

Let $S \subseteq \mathbb{Z}^m$ be a set containing the feasible set of $(D_{ISDP})$. In case of no prior knowledge about $(D_{ISDP})$ we take $S = \mathbb{Z}^m$. Since $P$ and $c(P)$ share the same integer hull, it follows that if $\hat{\mathbf{x}} \notin P$, we also have $\hat{\mathbf{x}} \notin c(P)$. Hence, using (9), there exists a dual multiplier $\mathbf{U} \in \mathcal{S}_+^n$ with $\langle \mathbf{A_i}, \mathbf{U} \rangle \in \mathbb{Z}$ for all $i \in [m]$, such that $\sum_{i=1}^m \langle \mathbf{A_i}, \mathbf{U} \rangle \hat{x}_i > \lfloor \langle \mathbf{C}, \mathbf{U} \rangle \rfloor$. Taking such $\mathbf{U}$ and defining $\mathbf{v}(\mathbf{U}) := (\langle \mathbf{A_1}, \mathbf{U} \rangle, \ldots, \langle \mathbf{A_m}, \mathbf{U} \rangle)^\top$, we obtain the following $S$-CG cut:

$$\sum_{i=1}^m \langle \mathbf{A_i}, \mathbf{U} \rangle x_i \leq \lfloor \langle \mathbf{C}, \mathbf{U} \rangle \rfloor_{S, \mathbf{v}(\mathbf{U})}, \tag{21}$$

see (17). The cut (21) exploits both the PSD and the integrality constraints in $(D_{ISDP})$ by separating $\hat{\mathbf{x}}$ from $c(P)$ instead of only from $P$. As $c(P) \subseteq P$ for bounded spectrahedra, this type of cut is possibly stronger than the eigenvalue cut (20) for all $S$ containing $P \cap \mathbb{Z}^m$. Figure 1 depicts a simplified example indicating the geometric difference between the cuts (20) and (21).

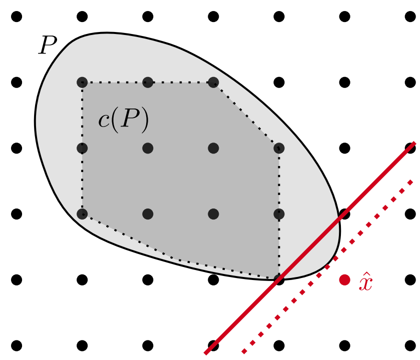



Figure 1: Simplified example of strengthened separation routine on spectrahedron $P$. The darkest area represents $c(P)$. The dotted line shows an eigenvalue cut (20) separating $\hat{\mathbf{x}}$ from $P$, the solid line shows a CG cut (21) separating $\hat{\mathbf{x}}$ from $c(P)$, where $S = \mathbb{Z}^m$.

Although there always exists a cut of the form (21) to separate $\hat{\mathbf{x}}$ from $c(P)$, it is not clear in general how to find an appropriate $\mathbf{U}$. Indeed, this is closely related to the CG separation problem, which was proven to be $\mathcal{NP}$-hard for the polyhedral case by Eisenbrand [26]. Fischetti and Lodi [36] show how to solve the separation problem for polyhedra using a mixed integer programming problem. Although we can extend this procedure to the class of spectrahedra, this implies solving a MISDP. Instead, we can adopt problem-specific separation routines that are efficient and provide strong cuts. For instance, in the next subsection we present a separation routine for two general classes of binary SDPs that have several applications in combinatorial optimization. Moreover, we later provide various separation routines for cuts of the form (21) for the quadratic traveling salesman problem.

Alongside extending the approach of Kobayashi and Takano [44], our framework also continues on the work of Çezik and Iyengar [13]. In [13] CG cuts for binary conic programs are introduced. It is noted that there is no method known for separating CG cuts from fractional points, and consequently the CG cuts are not included in the numerical experiments of [13]. Since our approach separates on integer points only, we partly resolve this issue for certain classes of problems by exploiting the underlying structure of the programs. As a result, we present the first practical algorithm that utilizes CG cuts in conic problems.

We end this section by providing a pseudocode of the generic B&C framework, see Algorithm 1. Suppose SeparationRoutine($\mathbf{C}, \mathbf{A_1}, \dots, \mathbf{A_m}, S, \hat{\mathbf{x}}$) is a problem-specific separation routine for constructing CG cuts of the form (21), where we assume this routine can generate multiple dual matrices at a time.

**Algorithm 1:** CG-based B&C algorithm for solving ($D_{ISDP}$)

**Input:** $\mathbf{C}, \mathbf{A_i}$, $i \in [m]$ , $S$, $\epsilon > 0$

Initialize $\mathcal{F}$ as defined in (18).

**B&B procedure:** Start or continue the branch-and-bound algorithm for solving the MILP $\max \left\{ \mathbf{b}^\top \mathbf{x} \,:\, \mathbf{x} \in \mathcal{F} \cap \mathbb{Z}^m \right\}$ incorporating the callback function below at each node in the branching tree.

**Callback procedure:** **if** *an integer point* $\hat{\mathbf{x}} \in \mathcal{F}$ *is found* **then**

  **if** $\lambda_{\min} \left( \mathbf{C} - \sum_{i=1}^m \mathbf{A_i} \hat{x}_i \right) < -\epsilon$ **then**

    Call SeparationRoutine($\mathbf{C}, \mathbf{A_1}, \dots, \mathbf{A_m}, S, \hat{\mathbf{x}}$) which provides matrices $\mathbf{U_j}$, $j \in [K]$. Add the cuts

$$\sum_{i=1}^m \langle \mathbf{A_i}, \mathbf{U_j} \rangle x_i \leq \lfloor \langle \mathbf{C}, \mathbf{U_j} \rangle \rfloor_{S, \mathbf{v}(\mathbf{U_j})} \quad \text{for } j \in [K] \text{ to } \mathcal{F}.$$

  **else**

    Use $\hat{\mathbf{x}}$ to cut off other nodes in the branching tree.

  **end**

  Return to Step 2

**end**

**Output:** $\hat{\mathbf{x}}, OPT := \mathbf{b}^\top \mathbf{x}$

## 3.2 A separation routine for binary SDPs

We now focus on binary semidefinite programming problems in primal form, i.e.,

$$\begin{cases} \inf & \langle \mathbf{C}, \mathbf{X} \rangle \\ \text{s.t.} & \langle \mathbf{A_i}, \mathbf{X} \rangle = b_i \qquad \text{for all } i \in [m] \\ & \mathbf{X} \succeq \mathbf{0}, \; \mathbf{X} \in \{0,1\}^{n \times n}. \end{cases} \tag{$P_{BSDP}$}$$

In this section we present a separation routine for generating CG cuts for problems of the form ($P_{BSDP}$) and provide two illustrative examples. To do so, we use the following characterization of binary PSD matrices.

**Proposition 5** (Letchford and Sørensen [46])**.** *Let* $\mathbf{X} \in \{0,1\}^{n \times n}$ *be a symmetric matrix. Then* $\mathbf{X} \succeq \mathbf{0}$ *if and only if* $\mathbf{X} = \sum_{i=1}^k \mathbf{x_i} \mathbf{x_i}^\top$ *for some* $\mathbf{x_i} \in \{0,1\}^n$*,* $i \in [k]$*.*

The result of Proposition 5 has an interpretation in terms of the complete graph $K_n$. Namely, each $\mathbf{x_i}$ can be seen as the characteristic vector of a clique in $K_n$. Since the diagonal entries of $\mathbf{X}$ may not exceed one, the cliques induced by $\mathbf{x_i}, i \in [k]$, are non-overlapping. Hence, $\mathbf{X}$ may be thought of as being the characteristic matrix of a set of non-overlapping cliques in $K_n$.

Suppose we solve ($P_{BSDP}$) using the B&C algorithm presented in Section 3.1. In a certain node in the branching tree we have obtained a symmetric matrix $\hat{\mathbf{X}} \in \{0,1\}^{n \times n}$ that satisfies $\langle \mathbf{A_i}, \hat{\mathbf{X}} \rangle = b_i$ for all $i \in [m]$. The separation oracle that we present below distinguishes two types of certificates for $\hat{\mathbf{X}}$ not being positive semidefinite. The first one is obtained by a so-called dominated diagonal, i.e., $\hat{X}_{ii} = 0$, while $\hat{X}_{ij} = 1$ for some $j$, which clearly implies that $\hat{\mathbf{X}} \not\succeq \mathbf{0}$. The second certificate is the presence of a so-called conflicting vertex. That is, there exists a vertex in $K_n$ that is contained in two separate cliques implied by $\hat{\mathbf{X}}$. By Proposition 5, it follows that $\hat{\mathbf{X}} \not\succeq \mathbf{0}$. In particular, the two certificates correspond to the existence of the following induced submatrices in $\hat{\mathbf{X}}$ (up to a permutation of the rows and columns):

$$\begin{array}{c c} & \begin{array}{cc} i & j \end{array} \\ \begin{array}{c} i \\ j \end{array} & \begin{pmatrix} 0 & 1 \\ 1 & \star \end{pmatrix} \end{array} \quad \text{and} \quad \begin{array}{c c} & \begin{array}{ccc} i & j & k \end{array} \\ \begin{array}{c} i \\ j \\ k \end{array} & \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \end{array},$$

where $\star$ indicates a position that can be either 0 or 1. The following result shows that these certificates are necessary and sufficient to characterize positive semidefiniteness.

**Proposition 6.** *Let $\hat{\mathbf{X}} = (\hat{x}_{ij})$ be binary and symmetric. Then, $\hat{\mathbf{X}}$ is positive semidefinite if and only if $\hat{\mathbf{X}}$ contains no dominated diagonal or conflicting vertex.*

*Proof.* Necessity follows from the discussion above. To prove necessity, let $D(i) := \{j \in [n] \,:\, \hat{x}_{ij} = 1\}$ for all $i \in [n]$ with $\hat{x}_{ii} = 1$. If $\hat{x}_{ij} = 1$ and $\hat{x}_{ik} = 1$, it must follow that $\hat{x}_{jk} = 1$, otherwise $i$ would be conflicting. Hence, the sets $D(i)$ for all $i$ with $\hat{x}_{ii} = 1$ are cliques. Since $i \in D(j)$ if and only if $j \in D(i)$, it follows that $D(i)$ and $D(j)$ are either the same or disjoint for all $i, j$. Let $\mathcal{D}$ be the entire collection of pairwise disjoint cliques after removing duplicates. Then, $\hat{\mathbf{X}} = \sum_{D \in \mathcal{D}} \mathbb{1}_{\mathbf{D}} \mathbb{1}_{\mathbf{D}}^\top$, hence by Proposition 5 it follows that $\hat{\mathbf{X}}$ is positive semidefinite. □

Algorithm 2 shows a pseudocode for checking positive semidefiniteness for binary matrices.

**Algorithm 2:** Separation routine for binary SDPs

**Input:** $\hat{\mathbf{X}} \in \{0,1\}^{n \times n} \cap \mathcal{S}^n$

Initialize $\mathbf{U} = \mathbf{0}$
**for** $i \in \{1, \ldots, n\}$ **do**
    **if** $\hat{x}_{ii} = 0$ **then**
        **if** $\exists j$ *s.t.* $\hat{x}_{ij} = 1$ **then**
            Let $\mathbf{U} \leftarrow (\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\top$ and RETURN    ▷ Dominated diagonal certificate
        **end**
    **else**
        Let $D \leftarrow \{j \in [n] \,:\, \hat{x}_{ij} = 1\}$
        **if** $\exists j, k \in D$ *s.t.* $j \neq k, i \neq j, i \neq k$ *and* $\hat{x}_{jk} = 0$ **then**
            Let $\mathbf{U} \leftarrow (\mathbf{e}_j + \mathbf{e}_k - \mathbf{e}_i)(\mathbf{e}_j + \mathbf{e}_k - \mathbf{e}_i)^\top$ and RETURN    ▷ Conflicting vertex certificate
        **end**
    **end**
**end**
If $\mathbf{U} = \mathbf{0}$, then $\hat{\mathbf{X}}$ is positive semidefinite

**Output:** $\mathbf{U}$

Algorithm 2 provides an integer dual matrix $\mathbf{U}$ inducing a cut $\langle \mathbf{U}, \mathbf{X} \rangle \geq 0$ separating $\hat{\mathbf{X}}$ from the feasible set of $(P_{BSDP})$. This cut can be further strengthened by exploiting the affine constraints in a CG rounding step. We show how this can be done for two classes of binary semidefinite programming problems. Both types play an important role in combinatorial optimization.

**Example 1** (Binary SDPs over the elliptope)**.** Suppose we have the following binary SDP:

$$\begin{cases} \inf & \langle \mathbf{C}, \mathbf{X} \rangle \\ \text{s.t.} & \langle \mathbf{A_i}, \mathbf{X} \rangle = b_i \qquad \text{for all } i \in [m] \\ & \operatorname{diag}(\mathbf{X}) = \mathbf{1} \\ & \mathbf{X} \succeq \mathbf{0},\ \mathbf{X} \in \{0,1\}^{n \times n}. \end{cases} \tag{$P_1$}$$

The problem $(P_1)$ can be solved using the B&C algorithm of Section 3.1 by initially setting $\mathcal{F} := \{\mathbf{X} \in \mathcal{S}^n \,:\, \langle \mathbf{A_i}, \mathbf{X} \rangle = b_i, i \in [m], \operatorname{diag}(\mathbf{X}) = \mathbf{1}, \mathbf{0} \leq \mathbf{X} \leq \mathbf{J}\}$. At a certain node in the branching tree a matrix $\hat{\mathbf{X}} \in \mathcal{F} \cap \{0,1\}^{n \times n}$ is considered. If $\hat{\mathbf{X}}$ is not positive semidefinite, then Algorithm 2 provides an integer dual matrix $\mathbf{U}$. Since $\operatorname{diag}(\mathbf{X}) = \mathbf{1}$ is included in $\mathcal{F}$, this dual matrix results from a conflicting vertex certificate. Suppose the output of Algorithm 2 is $\mathbf{U} = (\mathbf{e}_j + \mathbf{e}_k - \mathbf{e}_i)(\mathbf{e}_j + \mathbf{e}_k - \mathbf{e}_i)^\top$ for some distinct $i, j, k$. We can now further strengthen the cut $\langle \mathbf{U}, \mathbf{X} \rangle \geq 0$ using the constraint diag($\mathbf{X}$) = $\mathbf{1}$ and the fact that $\mathbf{U}$ is integer. Namely, taking the linear combination of $\langle \mathbf{U}, \mathbf{X}\rangle \geq 0$, $x_{ii} = 1, x_{jj} = 1$ and $x_{kk} = 1$, each with weight $\frac{1}{2}$, yields:

$$\frac{1}{2}\left(\langle \mathbf{U}, \mathbf{X}\rangle + x_{ii} + x_{jj} + x_{kk}\right) \geq \frac{1}{2}(0+1+1+1) \iff \left\langle \frac{1}{2}\mathbf{U} + \frac{1}{2}\left(\mathbf{E}_{ii} + \mathbf{E}_{jj} + \mathbf{E}_{kk}\right), \mathbf{X} \right\rangle \geq 1\frac{1}{2}.$$

Since $\mathbf{X} \in \mathcal{S}_n$ and all coefficients on the left hand side are integral, we can strengthen the cut using a CG rounding step:

$$\left\langle \frac{1}{2}\mathbf{U} + \frac{1}{2}\left(\mathbf{E}_{ii} + \mathbf{E}_{jj} + \mathbf{E}_{kk}\right), \mathbf{X} \right\rangle \geq \left\lceil 1\frac{1}{2} \right\rceil = 2.$$

This cut is equivalent to $x_{jk} + 1 \geq x_{ik} + x_{ij}$, which is one of the triangle inequalities resulting from the Boolean Quadric Polytope [52]. These cuts are facet defining for the binary PSD polytope [46].

**Example 2** (Binary SDPs over the simplex)**.** Many combinatorial optimization problems have formulations including a constraint on the trace of the matrix variable, i.e.,

$$\begin{cases} \inf & \langle \mathbf{C}, \mathbf{X}\rangle \\ \text{s.t.} & \langle \mathbf{A_i}, \mathbf{X}\rangle = b_i \qquad \text{for all } i \in [m] \\ & \text{tr}(\mathbf{X}) = K \\ & \mathbf{X} \succeq \mathbf{0},\ \mathbf{X} \in \{0,1\}^{n\times n}, \end{cases} \tag{$P_2$}$$

for some $K \in \mathbb{N}$. One can solve $(P_2)$ using Algorithm 1 with $\mathcal{F} := \{\mathbf{X} \in \mathcal{S}^n : \langle \mathbf{A_i}, \mathbf{X}\rangle = b_i, i \in [m],\ \text{tr}(\mathbf{X}) = K,\ \mathbf{0} \leq \mathbf{X} \leq \mathbf{J}\}$. Assume that the separation routine provides a dual matrix $\mathbf{U} = (\mathbf{e}_j + \mathbf{e}_k - \mathbf{e}_i)(\mathbf{e}_j + \mathbf{e}_k - \mathbf{e}_i)^\top$ for some distinct $i, j, k$. Now taking the linear combination of $\langle \mathbf{U}, \mathbf{X}\rangle \geq 0$, $\text{tr}(\mathbf{X}) = K$ and $x_{ll} \geq 0$ for all $l \notin \{i, j, k\}$, each with weight $\frac{1}{2}$, yields:

$$\left\langle \frac{1}{2}\mathbf{U} + \frac{1}{2}\mathbf{I} + \frac{1}{2}\sum_{l \notin \{i,j,k\}} \mathbf{E_{ll}},\ \mathbf{X} \right\rangle \geq \frac{1}{2}K.$$

For $K$ odd, we can strengthen the cut by replacing the right-hand side by $\lceil \frac{1}{2}K \rceil$. This procedure can be repeated for dual matrices resulting from a dominated diagonal certificate.

# 4 The Chvátal-Gomory procedure for ISDP formulations of the QTSP

In this section we provide an in-depth study on solving the Quadratic Traveling Salesman Problem using our B&C approach. We formally define the QTSP in Section 4.1. In Section 4.2 we derive two ISDP formulations of the QTSP. Our first ISDP model exploits the algebraic connectivity of a directed tour. Our second formulation exploits the algebraic connectivity of a directed tour and the distance two matrix that originates from the product of a tour matrix with itself. Finally, in Section 4.3 we derive CG cuts for the two ISDPs and show that we can obtain various classes of well-known cuts in this way.

## 4.1 The Quadratic Traveling Salesman Problem

Let $G = (N, A)$ be a directed simple graph on $n := |N|$ nodes and $m := |A|$ arcs. A directed cycle $\mathcal{C}$ in $G$ that visits all the nodes exactly once is called a directed Hamiltonian cycle or a directed tour in $G$. For the sake of simplicity, we often omit the adjective 'directed' in the sequel.

A tour in $G$ can be represented by a binary matrix $\mathbf{X} = (x_{ij}) \in \{0,1\}^{n\times n}$ such that $x_{ij} = 1$ if and only if arc $(i, j)$ is used in the tour. We refer to such a matrix as a *tour matrix*. The set of all tour matrices in $G$ is defined as follows:

$$\mathcal{T}_n(G) := \left\{ \mathbf{X}^{\mathcal{C}} \in \{0,1\}^{n\times n} : x^{\mathcal{C}}_{ij} = 1 \text{ if and only if } (i,j) \in \mathcal{C} \text{ for Hamiltonian cycle } \mathcal{C} \right\}. \tag{22}$$

It follows from (22) that for all $\mathbf{X} \in \mathcal{T}_n(G)$ we have $x_{ij} = 0$ if $(i, j) \notin A$. In particular, $\text{diag}(\mathbf{X}) = \mathbf{0}_n$. Given a distance matrix $\mathbf{D} = (d_{ij}) \in \mathbb{R}^{n\times n}$, the (linear) traveling salesman problem (TSP) is the problem of finding a Hamiltonian cycle $\mathcal{C}$ of $G$ that minimizes $\sum_{(i,j)\in\mathcal{C}} d_{ij}$. As $G$ is directed and $\mathbf{D}$ is not necessarily symmetric, this version of the problem is sometimes referred to as the asymmetric traveling salesman problem. Using the set defined in (22), we can state the TSP as follows:

$$TSP(\mathbf{D}, G) := \min \left\{ \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \ : \ \mathbf{X} \in \mathcal{T}_n(G) \right\}. \tag{23}$$

We now define the quadratic version of the TSP, where the total cost is given by the sum of interaction costs between arcs used in the tour. In accordance with most of the literature, we assume that a quadratic cost is incurred only if two arcs are placed in succession on the tour, see e.g., [31, 32, 33, 43, 57]. To model this problem, we define the set of the so-called 2-arcs of $G$, i.e.,

$$\mathcal{A} := \{(i, j, k) \ : \ (i, j), (j, k) \in A, |\{i, j, k\}| = 3\}, \tag{24}$$

which consists of all node triples of $G$ that can be placed in succession on a cycle. Now let $\mathbf{Q} = (q_{ijk}) \in \mathbb{R}^{n\times n\times n}$ be a cost matrix such that $q_{ijk} = 0$ if $(i, j, k) \notin \mathcal{A}$. Then the quadratic traveling salesman problem (QTSP) is formulated as:

$$QTSP(\mathbf{Q}, G) := \min \left\{ \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} q_{ijk} x_{ij} x_{jk} \ : \ \mathbf{X} \in \mathcal{T}_n(G) \right\}. \tag{25}$$

Since the in- and outdegree of each node on a Hamiltonian cycle is exactly one, we have $\mathbf{X1} = \mathbf{1}$ and $\mathbf{X}^\top \mathbf{1} = \mathbf{1}$ for all $\mathbf{X} \in \mathcal{T}_n(G)$. The set of square binary matrices that satisfy this property is known as the set of permutation matrices $\Pi_n$, i.e.,

$$\Pi_n := \left\{ \mathbf{X} \in \{0, 1\}^{n\times n} \ : \ \mathbf{X1} = \mathbf{1}, \, \mathbf{X}^\top \mathbf{1} = \mathbf{1} \right\}.$$

The permutation matrices that additionally satisfy $\text{diag}(\mathbf{X}) = \mathbf{0}_n$ induce a disjoint cycle cover in $K_n$.

Similar to the definition of $\mathcal{T}_n(G)$, we can also restrict $\Pi_n$ to the entries induced by $G$. That is, $\Pi_n(G)$ has a zero on position $(i, j)$ whenever $(i, j) \notin A$.

## 4.2 ISDP based on algebraic connectivity in directed graphs

Cvetković et al. [16] derive an ISDP formulation of the symmetric linear TSP based on algebraic connectivity. We now exploit the equivalent of this notion for directed graphs to derive two ISDP formulations of the QTSP. Different from our approach, there was no attempt in [16] to solve the ISDP itself, only its SDP relaxation.

Let $\mathbf{D_G}$ be an $n \times n$ diagonal matrix that contains the outdegrees of the nodes of $G$ on the diagonal. Moreover, let $\mathbf{A_G}$ denote the adjacency matrix of $G$. That is, $(A_G)_{ij} = 1$ if there exists an arc from $i$ to $j$ in $G$, and $(A_G)_{ij} = 0$ otherwise. We define the directed out-degree Laplacian matrix of $G$ as $\mathbf{L_G} := \mathbf{D_G} - \mathbf{A_G}$. The matrix $\mathbf{L_G}$ can be asymmetric and has a zero eigenvalue with corresponding eigenvector $\mathbf{1}_n$. Observe that there exist also other ways for defining the directed graph Laplacian of $G$, see e.g., [12]. Wu [66] generalized Fiedler's notion of algebraic connectivity of an undirected graph [30] to directed graphs, by exploiting the out-degree Laplacian matrix.

**Definition 4.** The algebraic connectivity of a directed graph $G$ is given by

$$a(G) := \min_{\mathbf{x}\in S} \mathbf{x}^\top \mathbf{L_G} \mathbf{x} = \min_{\substack{\mathbf{x}\in\mathbb{R}^n \\ \mathbf{x}\neq\mathbf{0}, \mathbf{x}\perp\mathbf{1}_n}} \frac{\mathbf{x}^\top \mathbf{L_G} \mathbf{x}}{\mathbf{x}^\top \mathbf{x}} = \lambda_{\min} \left( \frac{1}{2} \mathbf{W}^\top \left( \mathbf{L_G} + \mathbf{L_G}^\top \right) \mathbf{W} \right),$$

where $S := \{\mathbf{x} \in \mathbb{R}^n \ : \ \mathbf{x} \perp \mathbf{1}_n \, , \ \|\mathbf{x}\|_2 = 1\}$ and $\mathbf{W} \in \mathbb{R}^{n\times(n-1)}$ is a matrix whose columns form an orthonormal basis for $\mathbf{1}_n^\perp$.

The last equality in Definition 4 follows from the Courant-Fischer theorem. Observe that $a(G)$ is not necessarily equal to the second smallest eigenvalue of the directed Laplacian matrix, which is the definition of its undirected counterpart. The algebraic connectivity $a(G)$ as defined in Definition 4 is a real number that can be negative.

A directed graph is called *balanced* if for each node its indegree is equal to its outdegree. Let $\mathbf{B} \in \{-1, 0, 1\}^{n\times m}$ be the signed incidence matrix of $G$, i.e., $B_{i,e} = -1$ if arc leaves node $i$, $B_{i,e} = 1$ if $e$ enters node $i$ and $B_{i,e} = 0$ otherwise. One can verify that $G$ is balanced if and only if $\mathbf{L_G} + \mathbf{L_G}^\top = \mathbf{B}\mathbf{B}^\top$. This implies that for balanced graphs the matrix $\frac{1}{2}(\mathbf{L_G} + \mathbf{L_G}^\top)$ is positive semidefinite. Wu [66] observes that if $G$ is balanced, then

$$a(G) = \lambda_2 \left( \frac{1}{2} \left( \mathbf{L_G} + \mathbf{L_G}^\top \right) \right) \geq 0.$$

A directed graph is called *strongly connected* if for every pair of distinct nodes $u, v \in N$ there exists a directed path from $u$ to $v$ in $G$. The balanced graphs that are strongly connected are characterized by their algebraic connectivity, see Proposition 7 below. Connectedness of directed graphs is also studied in [12, 63].

**Proposition 7** (Wu [66])**.** *Let a directed graph $G$ be balanced. Then, $a(G) > 0$ if and only if $G$ is strongly connected.*

This characterization can be exploited to derive a certificate for a tour matrix via a linear matrix inequality. In order to do so, we consider the spectrum of a Hamiltonian cycle. Let $\mathcal{C}$ be a Hamiltonian cycle in $G$ corresponding to the tour matrix $\mathbf{X} \in \mathcal{T}_n(G)$, see (22). We then have $\frac{1}{2}\left(\mathbf{L}_\mathcal{C} + \mathbf{L}_\mathcal{C}^\top\right) = \mathbf{I_n} - \frac{1}{2}(\mathbf{X}+\mathbf{X}^\top)$. The matrix $\mathbf{X}+\mathbf{X}^\top$ with $\mathbf{X} \in \mathcal{T}_n(G)$ has the same spectrum as the adjacency matrix of the standard undirected $n$-cycle. As a result, the spectrum of $\frac{1}{2}(\mathbf{X}+\mathbf{X}^\top)$ is given by $\cos\left(\frac{2\pi j}{n}\right)$ for $j = 1, \ldots, n$, see e.g., [16]. From this, it follows that the spectrum of $\frac{1}{2}\left(\mathbf{L}_\mathcal{C} + \mathbf{L}_\mathcal{C}^\top\right)$ is given by

$$1 - \cos\left(\frac{2\pi j}{n}\right) \quad \text{for } j = 1, \ldots, n,$$

and the algebraic connectivity of a directed Hamiltonian cycle $\mathcal{C}$ is $a(\mathcal{C}) = 1 - \cos(2\pi/n)$. We define:

$$k_n := \cos\left(\frac{2\pi}{n}\right) \quad \text{and} \quad h_n := 1 - k_n. \tag{26}$$

Next, we extend a result by Cvetković et al. [16] from undirected to directed Hamiltonian cycles.

**Theorem 7.** *Let $H$ be a spanning subgraph of a directed graph $G$ where the in- and outdegree equals one for all nodes in $H$. Let $\mathbf{X}$ be its adjacency matrix and let $\alpha, \beta \in \mathbb{R}$ be such that $\alpha \geq h_n/n$ and $k_n \leq \beta < 1$, with $k_n, h_n$ as defined in (26). Then, $H$ is a directed Hamiltonian cycle if and only if*

$$\mathbf{Z} := \beta\mathbf{I_n} + \alpha\mathbf{J_n} - \frac{1}{2}\left(\mathbf{X} + \mathbf{X}^\top\right) \succeq \mathbf{0}.$$

*Proof.* Let $\mathbf{L_H}$ be the Laplacian matrix of $H$ and let $\mathbf{W}$ be as given in Definition 4. Then $a(H) = \lambda_{\min}\left(\frac{1}{2}\mathbf{W}^\top\left(\mathbf{L_H} + \mathbf{L_H}^\top\right)\mathbf{W}\right)$.

Let $\mathbf{Z} \succeq \mathbf{0}$. This implies that $\mathbf{W}^\top\mathbf{Z}\mathbf{W} \succeq \mathbf{0}$, i.e.,

$$\begin{aligned}
\mathbf{W}^\top\mathbf{Z}\mathbf{W} &= \mathbf{W}^\top\left(\beta\mathbf{I_n} + \alpha\mathbf{J_n} - \frac{1}{2}\left(\mathbf{X} + \mathbf{X}^\top\right)\right)\mathbf{W} \\
&= \beta\mathbf{W}^\top\mathbf{W} + \alpha\mathbf{W}^\top\mathbf{J_n}\mathbf{W} - \frac{1}{2}\mathbf{W}^\top\left(\mathbf{X} + \mathbf{X}^\top\right)\mathbf{W} \\
&= \beta\mathbf{I_{n-1}} - \frac{1}{2}\mathbf{W}^\top\left(\mathbf{X} + \mathbf{X}^\top\right)\mathbf{W}
\end{aligned}$$

$$= (\beta - 1)\mathbf{I_{n-1}} + \frac{1}{2}\mathbf{W}^\top \left(\mathbf{L_H} + \mathbf{L_H}^\top\right)\mathbf{W} \succeq \mathbf{0},$$

where we used the fact that $\mathbf{J_n W} = \mathbf{0}$ and $\frac{1}{2}(\mathbf{L_H} + \mathbf{L_H}^\top) = \mathbf{I_n} - \frac{1}{2}(\mathbf{X} + \mathbf{X}^\top)$. The linear matrix inequality above can be rewritten as

$$\frac{1}{2}\mathbf{W}^\top \left(\mathbf{L_H} + \mathbf{L_H}^\top\right)\mathbf{W} \succeq (1-\beta)\mathbf{I_{n-1}} \quad \Longrightarrow \quad a(H) = \lambda_{\min}\left(\frac{1}{2}\mathbf{W}^\top \left(\mathbf{L_H} + \mathbf{L_H}^\top\right)\mathbf{W}\right) \geq 1 - \beta.$$

Since $\beta < 1$, we have $\alpha(H) > 0$. Because $H$ is balanced, it follows from Proposition 7 that $H$ is strongly connected and, thus, $H$ is a directed Hamiltonian cycle.

Conversely, let $H$ be a directed Hamiltonian cycle. Then, $a(H) = \lambda_{\min}\left(\frac{1}{2}\mathbf{W}^\top \left(\mathbf{L_H} + \mathbf{L_H}^\top\right)\mathbf{W}\right) = 1 - k_n$. Since $\beta \geq k_n$, we have

$$\frac{1}{2}\mathbf{W}^\top \left(\mathbf{L_H} + \mathbf{L_H}^\top\right)\mathbf{W} - (1-\beta)\mathbf{I_{n-1}} \succeq \mathbf{0} \quad \Longleftrightarrow \quad \mathbf{W}^\top\mathbf{Z}\mathbf{W} \succeq \mathbf{0},$$

following the same derivation as above. Now, let $\mathbf{x} \in \mathbb{R}^n$. Since the columns of $\mathbf{W}$ form a basis for $\mathbf{1}_n^\perp$, $\mathbf{x}$ can be written as $\mathbf{x} = \mathbf{W}\mathbf{y} + \delta\mathbf{1}_n$ for some $\mathbf{y} \in \mathbb{R}^{n-1}$ and $\delta \in \mathbb{R}$. This yields:

$$\begin{aligned}\mathbf{x}^\top\mathbf{Z}\mathbf{x} &= \mathbf{y}^\top\mathbf{W}^\top\mathbf{Z}\mathbf{W}\mathbf{y} + 2\delta\mathbf{y}^\top\mathbf{W}^\top\mathbf{Z}\mathbf{1}_n + \delta^2\mathbf{1}_n^\top\mathbf{Z}\mathbf{1}_n \\ &= \underbrace{\mathbf{y}^\top\mathbf{W}^\top\mathbf{Z}\mathbf{W}\mathbf{y}}_{\geq 0} + \underbrace{2\delta\mathbf{y}^\top\mathbf{W}^\top\left((\beta-1)\mathbf{1}_n + \alpha n\mathbf{1}_n\right)}_{=0} + \underbrace{\delta^2 n\left((\beta-1) + \alpha n\right)}_{\geq 0},\end{aligned}$$

where we used the facts that $\mathbf{W}^\top\mathbf{Z}\mathbf{W} \succeq \mathbf{0}, \mathbf{W}^\top\mathbf{1}_n = \mathbf{0}$ and $\beta - 1 + \alpha n \geq k_n - 1 + n\frac{1-k_n}{n} = 0$. Thus, $\mathbf{Z} \succeq \mathbf{0}$. □

In order to present our first ISDP formulation of the QTSP, we derive an explicit expression for the set $\mathcal{T}_n(G)$ and linearize the objective function. The former can be done using Theorem 7. The set $\mathcal{T}_n(G)$ can be fully characterized by the permutation matrices that satisfy a linear matrix inequality. That is,

$$\mathcal{T}_n(G) = \Pi_n(G) \cap \left\{\mathbf{X} \in \mathcal{S}^n \;:\; \beta\mathbf{I_n} + \alpha\mathbf{J_n} - \frac{1}{2}(\mathbf{X} + \mathbf{X}^\top) \succeq \mathbf{0}\right\}, \tag{27}$$

for all $\alpha \geq h_n/n$ and $k_n \leq \beta < 1$. Recall that $\Pi_n(G)$ is the set of permutation matrices implied by $G$, see Section 4.1.

To linearize the objective function, we follow the same construction as proposed by Fischer et al. [33]. For all two-arcs $(i,j,k) \in \mathcal{A}$, see (24), we define a variable $y_{ijk} := x_{ij}x_{jk}$. This equality can be guaranteed by the introduction of the following set of linear coupling constraints:

$$x_{ij} = \sum_{\substack{k \in N: \\ (k,i,j) \in \mathcal{A}}} y_{kij} = \sum_{\substack{k \in N: \\ (i,j,k) \in \mathcal{A}}} y_{ijk} \text{ for all } (i,j) \in A \quad \text{and} \quad y_{ijk} \geq 0 \text{ for all } (i,j,k) \in \mathcal{A}.$$

We define the following set:

$$\mathcal{F}_1 := \left\{(\mathbf{y}, \mathbf{X}) \in \{0,1\}^{\mathcal{A}} \times \Pi_n(G) \;:\; x_{ij} = \sum_{\substack{k \in N: \\ (k,i,j) \in \mathcal{A}}} y_{kij} = \sum_{\substack{k \in N: \\ (i,j,k) \in \mathcal{A}}} y_{ijk} \quad \forall (i,j) \in A\right\}. \tag{28}$$

Now, our first ISDP formulation of the QTSP is as follows:

$$\begin{cases} \min & \displaystyle\sum_{(i,j,k) \in \mathcal{A}} q_{ijk}y_{ijk} \\ \text{s.t.} & \beta\mathbf{I_n} + \alpha\mathbf{J_n} - \dfrac{1}{2}\left(\mathbf{X} + \mathbf{X}^\top\right) \succeq \mathbf{0} \\ & (\mathbf{y}, \mathbf{X}) \in \mathcal{F}_1, \end{cases} \tag{$ISDP_1$}$$

where $\alpha \geq h_n/n$ and $k_n \leq \beta < 1$. One can verify that setting $\alpha = h_n/n$ and $\beta = k_n$ leads to the strongest linear matrix inequality among all possible values for $\alpha$ and $\beta$. Thus, we use these values in the computational results of Section 5.

**Remark 1.** In fact, we do not need to enforce integrality on $\mathbf{y}$ explicitly. Namely, if $\mathbf{X} \in \mathcal{T}_n(G)$, it follows from the integrality of $\mathbf{X}$ and the coupling constraints that $y_{ijk} = 1$ if $(i, j, k) \in \mathcal{A}$ is used in the tour and 0 otherwise. Hence, when optimizing over $\mathcal{F}_1$ using a B&B or B&C algorithm, we relax the integrality constraint on $\mathbf{y}$ and branch on $\mathbf{X}$ only.

In what follows, we further exploit properties of tour matrices to derive our second ISDP formulation of the QTSP. Let $\mathbf{X} \in \mathcal{T}_n(G)$ be a tour matrix and define $\mathbf{X^{(2)}} = (x^{(2)}_{ij}) := \mathbf{X} \cdot \mathbf{X}$. For $i, k \in N$ we have $x^{(2)}_{ik} = \sum_{j=1}^{n} x_{ij}x_{jk} = \sum_{j \in N:(i,j,k)\in\mathcal{A}} y_{ijk}$, where the last equality follows from the definition of $\mathbf{y}$. Thus, $\mathbf{X^{(2)}}$ is a binary matrix and $x^{(2)}_{ik} = 1$ if and only if the length of the shortest directed path from $i$ to $k$ in the subgraph induced by $\mathbf{X}$ is equal to two.

We can again characterize a tour matrix as in Theorem 7 by combining the variables $\mathbf{X}$ and $\mathbf{X^{(2)}}$. Observe that the directed graph induced by $\mathbf{X^{(2)}}$ is balanced with in- and outdegree one, and circulant (but not strongly connected for even $n$). Moreover, the circulant graph $\mathcal{C}_2$ corresponding to $\mathbf{X} + \mathbf{X^{(2)}}$ is strongly connected and balanced with in- and outdegree two. The spectrum of $\frac{1}{2}((\mathbf{X} + \mathbf{X^{(2)}}) + (\mathbf{X} + \mathbf{X^{(2)}})^\top)$ for any $\mathbf{X} \in \mathcal{T}_n(G)$ and $\mathbf{X^{(2)}} = \mathbf{X} \cdot \mathbf{X}$ is given by

$$\cos\left(\frac{2\pi j}{n}\right) + \cos\left(\frac{4\pi j}{n}\right) \quad \text{for } j = 1, \ldots, n, \tag{29}$$

which results in the algebraic connectivity of $\mathcal{C}_2$ being $a(\mathcal{C}_2) = 2 - (\cos(2\pi/n) + \cos(4\pi/n))$. We define

$$k^{(2)}_n := \cos\left(\frac{2\pi}{n}\right) + \cos\left(\frac{4\pi}{n}\right) \quad \text{and} \quad h^{(2)}_n := 2 - k^{(2)}_n. \tag{30}$$

Now, we are ready to state the following theorem.

**Theorem 8.** *Let $H$ be a spanning subgraph of a directed graph $G$ where the in- and outdegree equals one for all nodes in $H$. Let $\mathbf{X}$ be its adjacency matrix and let $\mathbf{X^{(2)}} := \mathbf{X} \cdot \mathbf{X}$ be the distance two adjacency matrix. Let $\alpha^{(2)}, \beta^{(2)} \in \mathbb{R}$ be such that $\alpha^{(2)} \geq h^{(2)}_n/n$ and $k^{(2)}_n \leq \beta^{(2)} < 2$, with $k^{(2)}_n$, $h^{(2)}_n$ as defined in* (30)*. Then $H$ is a directed Hamiltonian cycle if and only if*

$$\mathbf{Z} := \beta^{(2)}\mathbf{I_n} + \alpha^{(2)}\mathbf{J_n} - \frac{1}{2}\left((\mathbf{X} + \mathbf{X^{(2)}}) + (\mathbf{X} + \mathbf{X^{(2)}})^\top\right) \succeq \mathbf{0}.$$

*Proof.* Let $\tilde{H}$ be the subgraph of $G$ that has adjacency matrix $\mathbf{X} + \mathbf{X^{(2)}}$. Observe that $\tilde{H}$ is balanced, and thus, $\tilde{H}$ is strongly connected if and only if $a(\tilde{H}) > 0$.

Let $\mathbf{Z} \succeq \mathbf{0}$, which implies that $\mathbf{W}^\top\mathbf{Z}\mathbf{W} \succeq \mathbf{0}$. Now we can use a similar derivation as in the proof of Theorem 7, which results in the following:

$$\frac{1}{2}\mathbf{W}^\top\left(\mathbf{L_{\tilde{H}}} + \mathbf{L_{\tilde{H}}}^\top\right)\mathbf{W} \succeq \left(2 - \beta^{(2)}\right)\mathbf{I_{n-1}} \implies a(\tilde{H}) = \lambda_{\min}\left(\frac{1}{2}\mathbf{W}^\top\left(\mathbf{L_{\tilde{H}}} + \mathbf{L_{\tilde{H}}}^\top\right)\mathbf{W}\right) \geq 2 - \beta^{(2)}.$$

Since $\beta^{(2)} < 2$, we have $a(\tilde{H}) > 0$, and thus, $\tilde{H}$ is strongly connected. As $\tilde{H}$ is the union of a directed cycle cover and its implied distance two graph, $\tilde{H}$ can only be strongly connected if $H$ is strongly connected. We conclude that $H$ is a Hamiltonian cycle.

Conversely, let $H$ be a Hamiltonian cycle. In that case, the algebraic connectivity of $\tilde{H}$ is $a(\tilde{H}) = 2 - k^{(2)}_n$, i.e., $\lambda_{\min}\left(\frac{1}{2}\mathbf{W}^\top\left(\mathbf{L_{\tilde{H}}} + \mathbf{L_{\tilde{H}}}^\top\right)\mathbf{W}\right) = 2 - k^{(2)}_n$. Since $\beta^{(2)} \geq k^{(2)}_n$, this yields

$$\frac{1}{2}\mathbf{W}^\top\left(\mathbf{L_{\tilde{H}}} + \mathbf{L_{\tilde{H}}}^\top\right)\mathbf{W} - \left(2 - \beta^{(2)}\right)\mathbf{I_{n-1}} \succeq \mathbf{0} \iff \mathbf{W}^\top\mathbf{Z}\mathbf{W} \succeq \mathbf{0}.$$

Now we can use the same argument as in the proof of Theorem 7 to show that $\mathbf{Z} \succeq 0$ where $\beta, \alpha$ and $k_n$ are replaced by $\beta^{(2)}, \alpha^{(2)}$ and $k^{(2)}_n$, respectively. □

We define the set $\mathcal{F}_2$ as follows:

$$\mathcal{F}_2 := \left\{ \left(\mathbf{y}, \mathbf{X}, \mathbf{X^{(2)}}\right) \in \mathcal{F}_1 \times \Pi_n(G^2) \,:\, x_{ik}^{(2)} = \sum_{\substack{j \in N: \\ (i,j,k) \in \mathcal{A}}}^{n} y_{ijk} \;\forall (i,k) \in A^2 \right\}, \tag{31}$$

where

$$\Pi_n(G^2) := \left\{ \mathbf{X^{(2)}} \in \{0,1\}^{n \times n} \,:\, \mathbf{X^{(2)}}\mathbf{1} = \mathbf{1},\, (\mathbf{X^{(2)}})^\top \mathbf{1} = \mathbf{1},\, \mathrm{diag}(\mathbf{X^{(2)}}) = \mathbf{0},\; x_{ij}^{(2)} = 0 \;\forall (i,j) \notin A^2 \right\},$$

and $A^2$ is the set of node pairs $(i,j)$ for which there exists a directed path from $i$ to $j$ of length 2. The set $\mathcal{F}_2$ and the result of Theorem 8 lead to our second ISDP formulation of the QTSP:

$$\begin{cases} \min & \sum_{(i,j,k) \in \mathcal{A}} Q_{ijk} y_{ijk} \\ \text{s.t.} & \beta \mathbf{I_n} + \alpha \mathbf{J_n} - \frac{1}{2}\left(\mathbf{X} + \mathbf{X}^\top\right) \succeq \mathbf{0} \\ & \beta^{(2)} \mathbf{I_n} + \alpha^{(2)} \mathbf{J_n} - \frac{1}{2}\left((\mathbf{X} + \mathbf{X^{(2)}}) + (\mathbf{X} + \mathbf{X^{(2)}})^\top\right) \succeq \mathbf{0} \\ & (\mathbf{y}, \mathbf{X}, \mathbf{X^{(2)}}) \in \mathcal{F}_2, \end{cases} \tag{$ISDP_2$}$$

where $\alpha \geq h_n/n$, $k_n \leq \beta < 1$, $\alpha^{(2)} \geq h_n^{(2)}/n$ and $k_n^{(2)} \leq \beta^{(2)} < 2$. Again the choice of $\alpha, \beta, \alpha^{(2)}$ and $\beta^{(2)}$ equal to their lower bounds provides the strongest continuous relaxation.

It follows from Theorem 8 that one can remove the first linear matrix inequality in $(ISDP_2)$ and still obtain an exact formulation of the QTSP. However, the bound obtained from the SDP relaxation of $(ISDP_2)$ dominates the bound obtained from the SDP relaxation of $(ISDP_1)$. In that sense, the formulation $(ISDP_2)$ can be seen as a level two formulation of the QTSP, whose continuous relaxation is stronger than that of the first level formulation. An additional advantage of the level two formulation is that both linear matrix inequalities may be used to generate CG cuts, as we show in the following section.

In the same vein, one can construct level $k$ formulations of the QTSP for $k = 3, \dots, n$. This leads to a hierarchy of formulations, whose SDP relaxations are of increasing strength and complexity.

### 4.3 Chvátal-Gomory cuts for the ISDPs of the QTSP

In order to solve $(ISDP_1)$ and $(ISDP_2)$ using our B&C algorithm, we study various CG-based separation routines for the QTSP. We first derive a general CG cut generator for the formulations $(ISDP_1)$ and $(ISDP_2)$. Thereafter, we show how different types of well-known inequalities for the QTSP can be derived as CG cuts of the formulations $(ISDP_1)$ and $(ISDP_2)$.

Let us consider $(ISDP_1)$. The set $\mathcal{F}_1$, see (28), consists of all tuples $(\mathbf{y}, \mathbf{X})$ where $\mathbf{X}$ represents a node-disjoint cycle cover in $G$. Our B&C algorithm starts with optimizing over the set $\mathcal{F}_1$, where we are allowed to relax the integrality of $\mathbf{y}$ at no cost, see Remark 1. If an integer point $(\hat{\mathbf{y}}, \hat{\mathbf{X}})$ is found in the branching tree, it is verified whether $\lambda_{\min}\left(\beta \mathbf{I_n} + \alpha \mathbf{J_n} - \frac{1}{2}\left(\hat{\mathbf{X}} + \hat{\mathbf{X}}^\top\right)\right) \geq 0$. If so, then $\hat{\mathbf{X}} \in \mathcal{T}_n(G)$ and we have found a possibly new incumbent solution. If not, then $\hat{\mathbf{X}}$ is the adjacency matrix of a node-disjoint cycle cover that is not a Hamiltonian cycle. Therefore we have to generate dual matrices that cut off the current point.

The first separation routine that we present is based on finding a set of integer eigenvectors corresponding to a negative eigenvalue of $\beta \mathbf{I_n} + \alpha \mathbf{J_n} - \frac{1}{2}\left(\hat{\mathbf{X}} + \hat{\mathbf{X}}^\top\right)$.

**Proposition 8.** *Let $\mathbf{X} \in \Pi_n(G)$ be the adjacency matrix of a directed node-disjoint cycle cover consisting of $k \geq 2$ cycles. Let $\{S_1, \dots, S_k\}$ be the partition of the nodes implied by the cycle cover and define for each $l \in [k]$ the vector*

$$v_i^l := \begin{cases} n - |S_l| & \text{if } i \in S_l \\ -|S_l| & \text{if } i \notin S_l. \end{cases}$$

*Then* $\left\langle \mathbf{v^l}(\mathbf{v^l})^\top,\, \beta\mathbf{I_n} + \alpha\mathbf{J_n} - \frac{1}{2}(\mathbf{X}+\mathbf{X}^\top)\right\rangle < 0$ *for all* $l \in [k]$.

*Proof.* The vectors $\mathbf{v^l}$ are eigenvectors of $\mathbf{X}$ and $\mathbf{X}^\top$ corresponding to eigenvalue 1. Therefore we have:

$$\begin{aligned}\left(\beta\mathbf{I_n} + \alpha\mathbf{J_n} - \frac{1}{2}(\mathbf{X}+\mathbf{X}^\top)\right)\mathbf{v^l} &= \beta\mathbf{v^l} + \alpha\left((n-|S_l|)\cdot|S_l| + (n-|S_l|)\cdot(-|S_l|)\right)\mathbf{1} - \frac{1}{2}\mathbf{v^l} - \frac{1}{2}\mathbf{v^l} \\ &= (\beta-1)\mathbf{v^l},\end{aligned}$$

from where it follows that $\mathbf{v^l}$ is an eigenvector of $\beta\mathbf{I_n} + \alpha\mathbf{J_n} - \frac{1}{2}(\mathbf{X}+\mathbf{X}^\top)$ corresponding to eigenvalue $\beta - 1$. Since we assume $\beta < 1$, this eigenvalue is negative, from which the conclusion follows. □

The result of Proposition 8 can be used within our B&C algorithm in the following way. Let $\{S_1, \dots, S_k\}$ be the partition of the nodes implied by the current solution $\hat{\mathbf{X}}$ and let $\mathbf{U^l} := \mathbf{v^l}(\mathbf{v^l})^\top$ where $\mathbf{v^l}$ is as defined in Proposition 8. Then for each $l \in [k]$ we construct the following CG cuts:

$$\left\langle \mathbf{U^l}, \frac{1}{2}(\mathbf{X}+\mathbf{X}^\top)\right\rangle \leq \lfloor\langle \mathbf{U^l}, \beta\mathbf{I_n} + \alpha\mathbf{J_n}\rangle\rfloor \quad \Longleftrightarrow \quad \langle \mathbf{U^l}, \mathbf{X}\rangle \leq \lfloor\langle \mathbf{U^l}, \beta\mathbf{I_n} + \alpha\mathbf{J_n}\rangle\rfloor, \tag{32}$$

which cut off the current point. Observe that the choice $\alpha = h_n/n$ and $\beta = k_n$ leads to non-integer values for $\alpha$ and $\beta$, i.e., the CG rounding step provides a strengthened eigenvalue cut.

Since the result of Proposition 8 can be repeated for the extended linear matrix inequality in Theorem 8, we also obtain the following CG cuts with respect to $(ISDP_2)$:

$$\left\langle \mathbf{U^l}, \mathbf{X} + \mathbf{X^{(2)}}\right\rangle \leq \left\lfloor\langle \mathbf{U^l}, \beta^{(2)}\mathbf{I_n} + \alpha^{(2)}\mathbf{J_n}\rangle\right\rfloor \qquad \forall l \in [k]. \tag{33}$$

Next, we consider the class of subtour elimination constraints. It has been shown by Çezik and Iyengar [13] that the ordinary subtour elimination constraints defined by Dantzig et al. [20] can be obtained as CG cuts for the symmetric TSP, provided that $\alpha$ and $\beta$ equal their lower bounds. We extend the result from [13] and present five types of subtour elimination constraints that are in fact (strengthened) CG cuts of $(ISDP_1)$ and/or $(ISDP_2)$, see Table 1. Many of these constraints do not follow directly from the linear matrix inequalities, but require the addition of a positive multiple of a subset of the affine constraints. It is shown by Fischer [32] that the inequalities IV and V of Table 1 define facets of the asymmetric quadratic traveling salesman polytope.

In Appendix A, we explicitly derive these inequalities as (strengthened) CG cuts.

# 5 Computational Results

In this section we test our ISDP formulations of the QTSP, see Section 4. We solve the ISDPs using various settings of our CG-based B&C framework, see Algorithm 1, where we include different sets of cuts from Section 4.3 in the separation routines. We compare the performance of our approach with the two other ISDP solvers from the literature.

## 5.1 Design of numerical experiments

In total we compare seven different approaches, among which two from the literature and five variants of our B&C approach. The former class consists of the following:

- ***KT*:** The B&C algorithm of Kobayashi and Takano [44], see Section 3.1.
- ***SCIP-SDP*:** The general ISDP solver of Gally et al. [38]. This approach is based on solving continuous SDPs in a B&B framework.

A third project that is known for its ability to solve ISDPs is YALMIP [47]. Preliminary experiments show, however, that the solver of [47] is significantly outperformed by the solvers from [38] and [44]. Therefore, we do not take the solver of YALMIP into account.

On top of the approaches from the literature, we consider five variants of our B&C procedure that differ in the initial feasible set and the type of cuts that we add in the separation routine:

| | Inequality | Description |
|---|---|---|
| I | $\sum_{\substack{i \in S \\ j \in S}} x_{ij} \leq \lvert S \rvert - 1, \quad \forall S \subset N, 2 \leq \lvert S \rvert < n$ | CG cut of $\beta \mathbf{I_n} + \alpha \mathbf{J_n} - \frac{1}{2}\left(\mathbf{X} + \mathbf{X}^\top\right) \succeq \mathbf{0}$ with dual multiplier $\mathbf{U} = \mathbb{1}_\mathbf{S} \mathbb{1}_\mathbf{S}^\top$. |
| II | $\sum_{\substack{i \in S \\ j \notin S}} x_{ij} \geq 1, \quad \forall S \subset N, 2 \leq \lvert S \rvert < n$ | CG cut of $\beta \mathbf{I_n} + \alpha \mathbf{J_n} - \frac{1}{2}\left(\mathbf{X} + \mathbf{X}^\top\right) \succeq \mathbf{0}$ with dual multiplier $\mathbf{U} = \mathbb{1}_\mathbf{S} \mathbb{1}_\mathbf{S}^\top$ and $-\mathbf{X1} = -\mathbf{1}$ with dual multiplier $\mathbb{1}_\mathbf{S}$. |
| III | $\sum_{l=1}^{k} \sum_{\substack{i \in S_l \\ j \in S_l}} x_{ij} - \sum_{l \neq p} \sum_{\substack{i \in S_l \\ j \in S_p}} x_{ij} \leq n - 2k$ <br> $\forall (S_1, \dots, S_k), \cup_{l=1}^k S_l = N, S_l \cap S_p = \emptyset \ \forall l \neq p$ | CG cut of $\beta \mathbf{I_n} + \alpha \mathbf{J_n} - \frac{1}{2}\left(\mathbf{X} + \mathbf{X}^\top\right) \succeq \mathbf{0}$ with dual multiplier $\mathbf{U} = 2\sum_{l=1}^k \mathbb{1}_{\mathbf{S_l}} \mathbb{1}_{\mathbf{S_l}}^\top$ and $-\mathbf{X1} = -\mathbf{1}$ with dual multiplier $\mathbf{1}$. |
| IV | $x_{ij} + x_{ji} + \sum_{\substack{k \in N: \\ (i,k,j) \in \mathcal{A}}} y_{ikj} + \sum_{\substack{k \in N: \\ (j,k,i) \in \mathcal{A}}} y_{jki} \leq 1$ <br> $\forall i, j \in N, i \neq j, n \geq 5$ | $S$-CG cut of $\beta^{(2)} \mathbf{I_n} + \alpha^{(2)} \mathbf{J_n} - \frac{1}{2}\big((\mathbf{X} + \mathbf{X^{(2)}}) + (\mathbf{X} + \mathbf{X^{(2)}})^\top\big) \succeq \mathbf{0}$ with dual multiplier $\mathbf{U} = \mathbb{1}_{\{\mathbf{i},\mathbf{j}\}} \mathbb{1}_{\{\mathbf{i},\mathbf{j}\}}^\top$ and $\sum_{\substack{k \in N: \\ (i,k,j) \in \mathcal{A}}} y_{ikj} - x_{ij}^{(2)} = 0$, $\sum_{\substack{k \in N: \\ (j,k,i) \in \mathcal{A}}} y_{jki} - x_{ji}^{(2)} = 0$, $-x_{ii} = 0$, $-x_{jj} = 0$, $-x_{ii}^{(2)} = 0$ and $-x_{jj}^{(2)} = 0$, each with dual multiplier 1. |
| V | $\sum_{\substack{i \in S \\ j \in S}} x_{ij} + \sum_{\substack{i \in S \\ j \in S}} \sum_{\substack{k \in N \setminus S: \\ (i,k,j) \in \mathcal{A}}} y_{ikj} \leq \lvert S \rvert - 1$ <br> $\forall S \subset N, 2 \leq \lvert S \rvert < \frac{1}{2} n$ | $S$-CG cut of $\beta^{(2)} \mathbf{I_n} + \alpha^{(2)} \mathbf{J_n} - \frac{1}{2}\big((\mathbf{X} + \mathbf{X^{(2)}}) + (\mathbf{X} + \mathbf{X^{(2)}})^\top\big) \succeq \mathbf{0}$ with dual multiplier $\mathbf{U} = \mathbb{1}_\mathbf{S} \mathbb{1}_\mathbf{S}^\top$ and $\sum_{k \in N:(i,k,j) \in \mathcal{A}} y_{ikj} - x_{ij}^{(2)} = 0$, for all $i, j \in S$, each with dual multiplier 1, and $-y_{ikj} \leq 0$ for all $(i,k,j) \in \mathcal{A}$ with $i, k, j \in S$, each with dual multiplier 1. |

Table 1: Five types of subtour elimination constraints for the QTSP that can be obtained as (strengthened) CG cuts of $(ISDP_1)$ and/or $(ISDP_2)$. The third column describes which (in)equalities and dual multipliers are used to construct the inequality.

- ***CG1*:** In this setting we solve $(ISDP_1)$ where we initially optimize over $\mathcal{F}_1$, see (28). In the separation routine we add the CG cut of the form (32) for each subtour present in the current candidate solution.
- ***CG2*:** In this setting we solve the second QTSP formulation $(ISDP_2)$. We initially optimize over $\mathcal{F}_2$, see (31), and in each callback iteration we add the CG cuts of the form (32) and (33) for each subtour in the current candidate solution.
- ***SEC-simple*:** In this setting we solve $(ISDP_1)$ by starting from optimizing over $\mathcal{F}_1$, see (28). In the callback procedure, we add the ordinary subtour elimination constraints, see Type I in Table 1, for all subtours in the current candidate solution.
- ***SEC*:** This setting solves $(ISDP_2)$ with subtour elimination constraints of Type I, IV and V from Table 1. The latter type of constraint is added only for the subtours of size less than $\frac{1}{2}n$. Since the order two variables $\mathbf{X^{(2)}}$ in this setting do not appear directly in the cutting planes, we eliminate them also from the initial MILP based on preliminary tests. That is, we start optimizing over $\mathcal{F}_1$, see (28). Moreover, based on a result by Fischer et al. [33] we also add additional cuts to forbid subtours of three nodes. For a triple $i, j, k$ of distinct nodes, the following cut is valid for any tour:

  $$y_{ijk} + y_{kij} \leq x_{ij}.$$

  We add this cut for all distinct $i, j, k \in S$ in the separation routine whenever a subtour on $S$ with $\lvert S \rvert = 3$ is present in the current candidate solution. Observe that there are six of them for each triple of nodes.

- ***SEC-CG*:** This setting solves $(ISDP_2)$, starting from $\mathcal{F}_2$, see (31). In the separation routines, we add all the cuts that are included in the previous setting SEC. Moreover, on top of that we also add the CG cuts (32) and (33) in the callback procedure.

Recall that the separation routines are only called at integer points, which represent cycle covers of $G$. Therefore, the separation of all mentioned cuts boils down to identifying the subtours in the cycle cover. Also, recall that the integrality of $\mathbf{y}$ is relaxed in all settings, see Remark 1.

The setting SEC looks similar to the best exact QTSP solving strategy of Fischer et al. [32]. However, there are two main differences between the methods. First, our separation routine is only called on integer points, while the algorithm of [33] separates on fractional points. The separation on integer points is computationally very cheap compared to the fractional separation method applied by [33]. Consequently, the former separation can lead to superior behavior, as observed by Aichholzer et al. [2] for the symmetric QTSP. Second, our approach results from a more general B&C framework for solving integer SDPs, which is not limited to the QTSP.

Notice that the derived CG cuts of Type II and III from Table 1 are not added in the test settings. Preliminary experiments have shown that the cut-set subtour elimination constraints (Type II of Table 1) have similar practical behaviour compared to the ordinary subtour elimination constraints. Also, preliminary tests show that the addition of one merged Type III cut instead of all separate Type I cuts leads to worse behaviour in terms of overall computation time. We expect this difference to be caused by the sparsity of the Type I cuts, compared to the very dense Type III cuts.

For our tests, we consider three types of instances[1]:

- **Real instances from bioinformatics:** Jäger and Molitor [43], Fischer [32] and Fischer et al. [33, 34] consider an important application of the QTSP in computational biology. In order to recognise transcription factor binding sites or RNA splice sites in a given set of DNA sequences, Permuted Markov (PM) models [27] or Permuted Variable Length Markov (PVLM) models [68] can be used. Finding the optimal order two PM or PVLM model boils down to solving a QTSP instance. We consider three classes of bioinformatics instances used in [31, 32], which are denoted by 'bma', 'map' and 'ml'. Each class consists of 38 instances with $n \in \{3, \dots, 40\}$.

- **Reload instances:** The reload instances are the same as the ones used by Rostami et al. [57] and De Meijer and Sotirov [50]. The reload model [64] is inspired by logistics and energy distribution, where a certain cost is incurred whenever the underlying type of arc in a network changes, e.g., the means of transport. Let $G$ be a directed graph where each arc $(i, j)$ is present with probability $p$. Each arc in $G$ is randomly assigned a color from a color set $L$ with cardinality $c$. If two successive arcs $e$ and $f$ have colors $s$ and $t$, respectively, the quadratic cost among $e$ and $f$ equals $r(s,t)$, where $r : L \times L \to \mathbb{R}$ is a reload cost function such that $r(s,s) = 0$ for all $s \in L$. We consider two types of reload classes:

  - *Reload class 1*: For each pair of distinct colors $s, t \in L$ the reload cost equals $r(s,t) = 1$;
  - *Reload class 2*: For each pair of distinct colors $s, t \in L$, the reload cost $r(s,t)$ is chosen uniformly at random from $\{1, \dots, 10\}$.

  For each class, we consider 10 distinct instances for each possible combination of $n \in \{10, 15, 20\}$, $p \in \{0.5, 1\}$ and $c \in \{5, 10, 20\}$. Thus, in total we consider 360 reload instances.

- **Turn cost instances:** The special case of the QTSP where the nodes are points in Euclidean space and the angle cost of a tour is the sum of the direction changes at the points is called the Angular-Metric Traveling Salesman Problem (Angle-TSP) [1]. The Angle-TSP is motivated by robotics and VLSI design and proven to be $\mathcal{NP}$-hard [1]. The problem is in the literature also known as the Minimum Bends Traveling Salesman Problem [60]. We consider two classes of this type:

[1] Instances can be downloaded from `https://github.com/frankdemeijer/CGforISDP`.

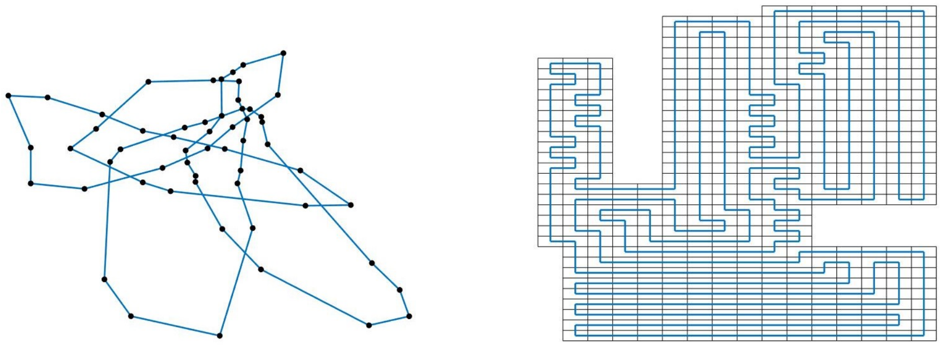
– *TSPLIB instances:* The TSP library (TSPLIB) [55] contains a broad set of TSP test instances, among which a large number of Euclidean instances. We construct a corresponding QTSP instance as follows: Given points $v_1, \ldots, v_n$ in $\mathbb{R}^2$, we let $G$ be the complete graph on $n$ vertices. For $i, j, k$, $i \neq j$, $j \neq k$, $i \neq k$, we define $q_{ijk}$ to be proportional to the angle between edges $\{i,j\}$ and $\{j,k\}$. More precisely,

$$q_{ijk} := \left\lceil 10 \cdot \left(1 - \frac{1}{\pi} \arccos\left(\frac{(v_i - v_j)^\top (v_k - v_j)}{\|v_i - v_j\| \cdot \|v_j - v_k\|}\right)\right)\right\rceil .$$

This cost structure is similar to the angle-distance costs considered in Fischer et al. [33] and De Meijer and Sotirov [49]. In total, we consider 9 TSPLIB instances with $n$ ranging from 15 to 70. Figure 2a depicts one of the TSPLIB instances including its optimal tour with respect to the defined quadratic cost structure.

– *Grid instances:* Fekete and Krupke [28, 29] consider problems of computing optimal covering tours and cycle covers under a turn cost model, see also Arkin et al. [3]. These problems have many practical applications, such as pest control and precision farming. Following this line, we consider the Angle-TSP on grid graphs. We construct a 2D connected grid graph using the Type II instance generator of [29]. Given the vertex coordinate vectors $v^1, \ldots, v^n \in \{0, \ldots, N_1\} \times \{0, \ldots, N_2\}$ for integers $N_1, N_2$, we include an edge between vertex $i$ and $j$ if and only if ($v_1^i = v_1^j$ and $|v_2^i - v_2^j| = 1$) or ($v_2^i = v_2^j$ and $|v_1^i - v_1^j| = 1$). If two edges $\{i,j\}$ and $\{j,k\}$ are present, the quadratic costs are computed similar as for the TSPLIB instances. In total we consider 9 grid instances with $N_1$ and $N_2$ running from 20 to 80, corresponding to $n$ ranging from 430 to 2646. An example of a grid instance including its minimum bend tour is given in Figure 2b.

Both types of turn cost instances are in fact instances of the symmetric QTSP, as they are defined on undirected graphs. To account for this, we use symmetrized versions of ($ISDP_1$) and ($ISDP_2$) instead. We refer to Appendix B for the construction of these formulations.



(a) Instance 'kn57' (b) Instance 'grid1'

Figure 2: Optimal tours of two turn instances: the TSPLIB instance 'kn57' ($n = 57$) and the grid instance 'grid1' ($n = 430$). Each square in Figure 2b represents a vertex in the grid graph.

All our algorithms, including the algorithm of Kobayashi and Takano [44], are implemented in Julia 1.5.3 using JuMP v0.21.10 [25] to model the mathematical optimization problems. In particular, we exploit the solver-independent lazy constraint callback option of JuMP to include the separation routines. Solving the underlying MILP in the subproblems is done using Gurobi v9.10 [42] in the default settings including built-in cuts. Experiments are carried out on a PC with an Intel(R) Core(TM) i7-8700 CPU, 3.20GHz, 8GB RAM. To run SCIP-SDP, we use SCIP-SDP version 3.2.0 on the NEOS Server [17], where the B&B framework of SCIP 7.0.0 [39] and the SDP solver Mosek 9.2 [51] are combined in the default configuration.

Observe that an older version of SCIP-SDP with DSDP [8] as SDP solver was used in the numerical experiments of [44], which partly explains the poor behaviour of SCIP-SDP compared to the B&C algorithm of [44]. However, our computational study that uses SCIP-SDP with the state-of-the-art SDP solver Mosek [51] also shows superior behaviour of the B&C algorithms.

We test all seven settings on the bioinformatics and reload instances. Since these instance classes give a clear and consistent overview of the superior approaches, we restrict ourselves to the best three settings for the turn cost instances. The maximum computation time for all our approaches is set to 8 hours, which is in correspondence with the maximum computation time on the NEOS Server [17].

## 5.2 Comparison of approaches

Table 2 and Figure 3 provide an overview of the performance on the instances from bioinformatics. For each setting, the average values in Table 2 are only computed over the instances that could be solved to optimality for that setting. An extended table on the results per instance can be found in Appendix C. It is clear that our B&C settings significantly outperform the other two ISDP solvers SCIP-SDP and KT, which can solve at most 60% of the instances to optimality. Since the separation routine of CG1 is based on the identification of an integer eigenvector corresponding to a negative eigenvalue, the settings KT and CG1 are almost identical apart from the CG rounding step. The large decrease in the number of branching nodes of CG1 compared to KT is remarkable. This indicates that the effect of deeper cuts as shown in Figure 1 is not solely theoretical, but also turns out substantial from a practical point of view.

When comparing the five different separation routines of our B&C approach, we also see a clear pattern. The settings SEC and SEC-CG turn out to be superior, being able to solve all instances within short computation times. Although SEC generally provides the fastest algorithm, it sometimes happens that SEC-CG solves the instance faster, see Figure 3, due to the smaller number of B&C nodes. This shows that the additional CG cuts can sometimes improve on the subtour elimination constraints. The two approaches are followed by SEC-simple, which is able to solve instances up to $n = 35$ to optimality. This difference is mainly due to the strengthened subtour elimination cuts (type IV and V in Table 1) that work well for the bioinformatics instances, as also noted by Fischer et al. [33]. Finally, the settings CG1 and CG2 are only able to solve instances up to $n = 32$ and $n = 27$, respectively. Although the distance two CG cuts (33) significantly reduce the number of needed branching steps, the overall computation time is larger due to the increase in the number of variables and constraints in CG2.

| Type | Statistic | SCIP-SDP | KT | CG1 | CG2 | SEC-simple | SEC | SEC-CG |
|---|---|---|---|---|---|---|---|---|
| **bma** | Instances solved (%) | 34.21 | 60.53 | 78.95 | 65.79 | 84.21 | **100** | **100** |
| | Average comp. time | 1519 | 1581 | 846.1 | 639.40 | 817.43 | **28.31** | 182.3 |
| | Average B&C nodes | 30308 | 854964 | 119144 | 42515 | 85984 | 200.4 | 142.5 |
| | Average time per node | 0.025 | 0.001 | 0.002 | 0.006 | 0.005 | 0.184 | 1.158 |
| **map** | Instances solved (%) | 34.21 | 57.89 | 78.95 | 65.79 | 81.58 | **100** | **100** |
| | Average comp. time | 2247 | 1721 | 1768 | 806.6 | 911.4 | **25.83** | 199.6 |
| | Average B&C nodes | 30385 | 896340 | 245732 | 56197 | 79869 | 244 | 173 |
| | Average time per node | 0.037 | 0.001 | 0.002 | 0.009 | 0.004 | 0.496 | 2.094 |
| **ml** | Instances solved (%) | 34.21 | 57.89 | 76.32 | 65.79 | 81.58 | **100** | **100** |
| | Average comp. time | 2891 | 1315 | 460.9 | 805.6 | 520.2 | **27.34** | 221.7 |
| | Average B&C nodes | 33185 | 658640 | 86743 | 44342 | 51495 | 252.0 | 186.3 |
| | Average time per node | 0.034 | 0.001 | 0.002 | 0.007 | 0.005 | 0.096 | 0.961 |

Table 2: Summary table of the performance on the bioinformatics instances per setting and per instance type. The best performing setting per row is given in bold.

Next, we discuss the results on the set of reload instances. For both class 1 and 2 and for each value of $n, p$ and $c$ we consider 10 randomly generated instances. The averaged results for each combination of parameters can be found in Appendix C, see Table 12 and 13. In general, we see that the computation times increase with the number of nodes $n$ and the graph density $p$. On the other hand, if the number of colors $c$ increases, the instances become easier to solve as the number of (optimal) solutions will decrease. For the same reason, the instances from reload class 1 seem more difficult to solve than the instances from reload class 2. Table 3 shows a summary of the results accumulated over the number of colors $c$. Accordingly, Figure 4 shows the spread of the computation times, where we also accumulate both reload classes.

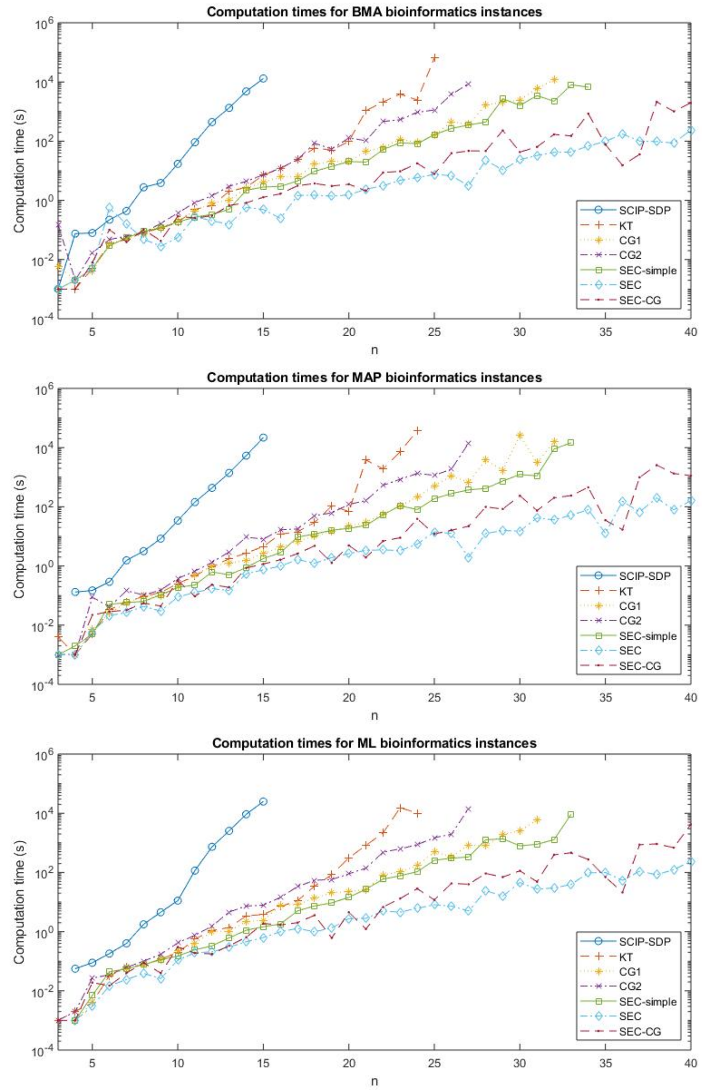



Figure 3: Computation times versus instance size for the bioinformatics classes 'bma' (top), 'map' (middle) and 'ml' (bottom). The computation times are given on a logarithmic scale.

When comparing the different settings, we draw similar conclusions as before. Note that SCIP-SDP performs very poorly on the reload instances. The difference between KT and CG1 is not as significant as before, although CG1 is still favourable above KT on almost all instance types. The settings that involve the variables $\mathbf{X^{(2)}}$ in the root node, i.e., CG2 and SEC-CG, are outperformed by SEC-simple and SEC. Apparently, the increase in the number of variables does not contribute much to the pruning of the branching tree. In fact, the results in Appendix C even suggest that the number of branching nodes sometimes becomes larger. However, the $S$-CG cuts resulting from $(ISDP_2)$ do contribute to the pruning of the tree, as is suggested by the strong performance of SEC. The settings SEC and SEC-simple overall perform best, where SEC is doing better on reload class 1, while SEC-simple is doing slightly better on reload class 2. The setting SEC can solve almost all reload instances within 100 seconds, while the majority of the instances are solved within 1 second.

| Instance | | | | Average computation times (s) | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Class** | $n$ | $p$ | **OPT** | **SDP-SCIP** | **KT** | **CG1** | **CG2** | **SEC-simple** | **SEC** | **SEC-CG** |
| **1** | 10 | 0.5 | 6.233 | 0.161 | 0.035 | 0.028 | 0.035 | 0.024 | **0.019** | 0.031 |
| | 10 | 1 | 3.3 | 1.627 | 0.133 | 0.127 | 0.165 | 0.128 | **0.113** | 0.171 |
| | 15 | 0.5 | 6.367 | 2.256 | 0.158 | 0.160 | 0.251 | 0.142 | **0.139** | 0.223 |
| | 15 | 1 | 2.8 | 244.0 | 1.426 | 1.124 | 4.503 | 1.095 | **1.040** | 2.825 |
| | 20 | 0.5 | 6.2 | 82.08 | 0.610 | 0.625 | 1.510 | 0.483 | **0.465** | 1.237 |
| | 20 | 1 | 2.314 | 3908 | 183.8 | 91.56 | 1910 | 162.5 | **43.21** | 3278 |
| **2** | 10 | 0.5 | 22.74 | 0.185 | 0.036 | 0.039 | 0.049 | **0.029** | **0.029** | 0.044 |
| | 10 | 1 | 8.2 | 0.962 | 0.164 | 0.148 | 0.152 | **0.116** | 0.139 | 0.157 |
| | 15 | 0.5 | 22.73 | 2.989 | 0.193 | 0.174 | 0.255 | 0.173 | **0.172** | 0.261 |
| | 15 | 1 | 6.767 | 277.5 | 1.363 | 1.768 | 4.643 | 1.293 | **1.190** | 3.290 |
| | 20 | 0.5 | 18.1 | 58.68 | 0.575 | 0.585 | 1.246 | **0.552** | 0.576 | 1.352 |
| | 20 | 1 | 4.745 | 2689 | 43.99 | 20.88 | 1187 | **11.89** | 16.88 | 850.2 |

Table 3: Overview of average computation times for the reload instances. Each row provides averages of 30 instances, namely 10 random instances for each value of $c = 5, 10, 20$.

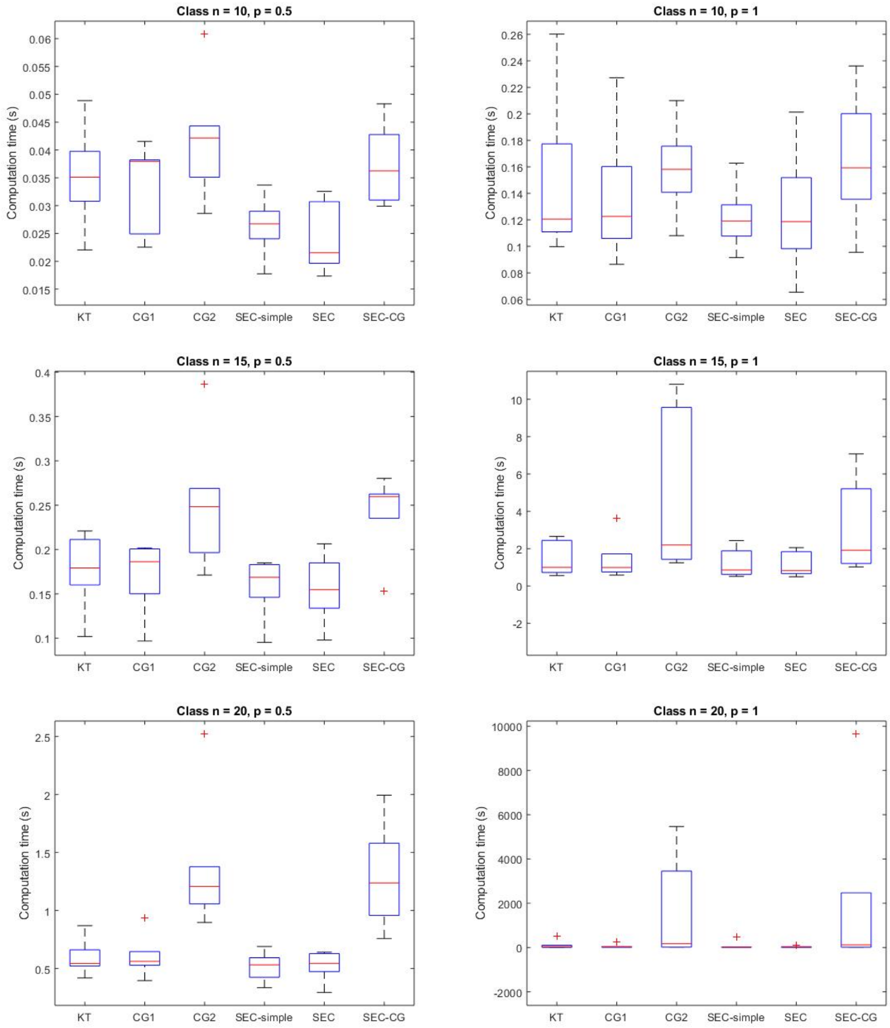



Figure 4: Boxplots showing the computation times for the reload instances for different values of $n$ and $p$, accumulated over the reload class and the number of colors $c$. We omit the results of SCIP-SDP, since these computation times are several magnitudes larger.

Finally, we consider the turn cost instances. From the class of bioinformatics and reload instances it is clear that the settings SEC-simple, SEC and SEC-CG generally perform best. Hence, we restrict the numerical results on the turn cost instances to these three settings. Table 4 and 5 show the computation times and number of branching nodes for the TSPLIB and grid instances, respectively.

The TSPLIB graphs are complete graphs, and hence we can only solve up to $n = 70$ for this instance type. We are able to solve all TSPLIB instances in a time span 900 seconds. Since the grid instances are more sparse, we can solve much larger instance sizes to optimality. For this type, instances up to 2646 nodes (!) can be solved to optimality within 15 seconds. These are currently the largest solved QTSP instances in the literature.

When comparing the three settings, we see that SEC-simple and SEC perform slightly better than SEC-CG on the turn cost instances. Since the different separation routines lead to different relaxations, the branching strategy between the methods can differ. Not surprisingly, the favourable setting is often the one with the smallest number of B&C nodes, regardless of the time per branching node. Taking both the TSPLIB and grid instances into account, this happens slightly more often for the setting SEC-simple.

| | | | | **SEC-simple** | | **SEC** | | **SEC-CG** | |
|---|---|---|---|---|---|---|---|---|---|
| **Instance** | $n$ | $m$ | **OPT** | **Comput. time (s)** | **Number of nodes** | **Comput. time (s)** | **Number of nodes** | **Comput. time (s)** | **Number of nodes** |
| **lau15** | 15 | 105 | 47 | 0.278 | 1 | 0.458 | 1 | **0.115** | 1 |
| **wg22** | 22 | 231 | 63 | 0.643 | 1 | **0.436** | 1 | 0.490 | 1 |
| **bays29** | 29 | 406 | 78 | 1.519 | 96 | **0.905** | 93 | 0.949 | 78 |
| **dantzig42** | 42 | 861 | 96 | **11.25** | 994 | 12.04 | 1059 | 21.20 | 1458 |
| **att48** | 48 | 1128 | 105 | 53.07 | 4104 | **47.64** | 3627 | 55.52 | 3375 |
| **berlin52** | 52 | 1326 | 118 | **702.9** | 36523 | 1115 | 49265 | 1070 | 41041 |
| **kn57** | 57 | 1596 | 120 | 153.1 | 2425 | **110.8** | 1539 | 138.6 | 1804 |
| **wg59** | 59 | 1711 | 121 | **391.2** | 10503 | 898.7 | 13627 | 650.1 | 10269 |
| **st70** | 70 | 2415 | 137 | 861.8 | 8596 | **838.1** | 4649 | 1862 | 12222 |

Table 4: Computation times and number of branching nodes for the TSPLIB instances.

| | | | | **SEC-simple** | | **SEC** | | **SEC-CG** | |
|---|---|---|---|---|---|---|---|---|---|
| **Instance** | $n$ | $m$ | **OPT** | **Comput. time (s)** | **Number of nodes** | **Comput. time (s)** | **Number of nodes** | **Comput. time (s)** | **Number of nodes** |
| **grid1** | 430 | 795 | 620 | 1.431 | 6 | 1.538 | 1 | **1.020** | 1 |
| **grid2** | 734 | 1393 | 460 | **14.86** | 2781 | 20.29 | 7942 | 19.33 | 2562 |
| **grid3** | 880 | 1672 | 590 | **3.303** | 30 | 5.019 | 78 | 5.945 | 207 |
| **grid4** | 960 | 1802 | 840 | 4.954 | 3 | **3.731** | 1 | 7.507 | 1 |
| **grid5** | 1038 | 1965 | 440 | 8.452 | 24 | **4.514** | 16 | 8.192 | 10 |
| **grid6** | 1214 | 2335 | 480 | 19.67 | 57 | **15.61** | 25 | 23.27 | 55 |
| **grid7** | 1302 | 2493 | 730 | **9.121** | 330 | 17.83 | 177 | 14.65 | 181 |
| **grid8** | 1788 | 3469 | 540 | 4.800 | 1 | 4.917 | 1 | **4.619** | 1 |
| **grid9** | 2646 | 5172 | 760 | 13.79 | 1 | 13.80 | 1 | **13.39** | 1 |

Table 5: Computation times and number of branching nodes for the grid instances.

# 6 Conclusions

In this work we study the Chvátal-Gomory cuts for spectrahedra and their strength in solving integer semidefinite programs resulting from combinatorial optimization problems. Accordingly, this paper increases the theoretical understanding of integer semidefinite programming, which in turn contributes to new solution techniques for this type of problems.

In Section 2 we study the elementary closure of spectrahedra and the hierarchy obtained by iterating this procedure. Using an alternative formulation of the elementary closure, see (9), we provide simple proofs of several properties, including a homogeneity property for bounded spectrahedra, see Theorem 4. We also provide a simple proof for the finiteness of the CG procedure for bounded spectrahedra, see Theorem 3. Although some of the here presented results are already known in the literature, the proofs we present are considerably simpler and are mainly based on concepts from mathematical optimization and number theory. We also present the polyhedral description of the elementary closure of spectrahedra whose defining linear matrix inequality is totally dual integral, see Theorem 6. To the best of our knowledge, this is the first such description for the elementary closure of a non-polyhedral set.

A generic B&C algorithm for ISDPs based on strengthened CG cuts is presented in Section 3, see Algorithm 1. Our algorithm is a refinement of the algorithm from [44], where the authors use eigenvector based inequalities to separate infeasible integer points. Moreover, our work can be seen as an extension of [13], in which the authors introduce CG cuts for conic programs, but leave the efficient separation of CG cuts as an open problem. Our numerical results indicate the effectiveness of the use of deeper CG cuts. We also provide a separation routine for binary SDPs originating from combinatorial optimization problems, see Algorithm 2.

In Section 4 we extensively study the application of our approach to the quadratic traveling salesman problem. Based on a generalization of the notion of algebraic connectivity to directed graphs, we present two exact ISDP formulations of the QTSP, see ($ISDP_1$) and ($ISDP_2$). We show that the simplest CG separation routine boils down to finding integer eigenvectors of the adjacency matrix of a node-disjoint cycle cover, see Proposition 8. However, more intricate dual multipliers lead to some well-known families of cuts, e.g., the ordinary and strengthened versions of the subtour elimination constraints, see Table 1. We test several variants of our B&C procedure that involve different separation routines.

Numerical results on the QTSP show that our B&C algorithm significantly outperforms the two alternative ISDP solvers of [38] and [44]. For the real instances from bioinformatics [33, 34], these solvers are able to solve instances up to only $n = 15$ and $n = 25$, respectively, whereas our method can solve all instances up to $n = 40$ in a short timespan. As one would expect, the extension to CG inequalities leads to deeper cuts, which successfully reduces the size of the branching tree compared to [44]. From all considered separation routines, it turns out that the setting SEC, see page 24, is overall most effective. This setting was able to solve all 492 tested QTSP instances to optimality within 5 minutes, where the largest instance contains $m = 5172$ arcs. This is currently the largest solved QTSP instance in the literature.

Our work inspires several future research directions. On the theoretical side, an interesting open problem is whether the classical result of [40] stating that any rational polyhedron can be described by a totally dual integral system can be extended to the case of bounded spectrahedra. If so, the result of Theorem 6 provides the polyhedral description of the elementary closure of any bounded spectrahedron. Another direction of study is the application of our B&C algorithm to other optimization problems that can be formulated as ISDPs. We expect the exploitation of CG cuts in the branching scheme to be effective for such ISDPs.

**Acknowledgements.** We thank Anja Fischer and Borzou Rostami for sharing with us the instances from bioinformatics and reload instances, respectively.

# References

[1] A. Aggarwal, D. Coppersmith, S. Khanna, R. Motwani, and B. Schieber. The angular-metric traveling salesman problem. *SIAM. J. Comput.*, 29:697–711, 1999.

[2] O. Aichholzer, A. Fischer, F. Fischer, J.F. Meier, U. Pferschy, A. Pilz, and R. Staněk. Minimization and maximization versions of the quadratic travelling salesman problem. *Optim.*, 66(4):521–546, 2017.

[3] E. M. Arkin, M. A. Bender, E. D. Demaine, S. P. Fekete, J. S. B. Mitchell, and S. Sethia. Optimal covering tours with turn costs. *SIAM. J. Comput.*, 35(3):531–566, 2005.

[4] A. Atamtürk and V. Narayanan. Cuts for conic mixed-integer programming. In Matteo Fischetti and David P. Williamson, editors, *Integer Programming and Combinatorial Optimization*, pages 16–29, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[5] A. Atamtürk and V. Narayanan. Conic mixed-integer rounding cuts. *Math. Program.*, 122(1):1–20, 2010.

[6] V. Balakrishnan and L. Vandenberghe. Semidefinite programming duality and linear time-invariant systems. *IEEE Trans. Autom. Control*, 48(1):30–41, 2003.

[7] P. Belotti, J.C. Góez, I. Pólik, T.K. Ralphs, and T. Terlaky. A complete characterization of disjunctive conic cuts for mixed integer second order cone optimization. *Discrete Optim.*, 24:3–31, 2017.

[8] S.J. Benson, Y. Ye, and X. Zhang. Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM J. Optim.*, 10(2):443–461, 2000.

[9] P. Bonami, M. Kilinç, and J. Linderoth. *Mixed Integer Nonlinear Programs*, volume 154 of *The IMA Volumes in Mathematics and its Applications*, chapter Algorithms and Software for Convex Mixed Integer Nonlinear Programs, pages 1–39. Springer, New York, NY, 2012.

[10] G. Braun and S. Pokutta. A short proof for the polyhedrality of the Chvátal-Gomory closure of a compact convex set. *Oper. Res. Lett.*, 42(5):307–310, 2014.

[11] M.K. De Carli Silva and L. Tunçel. A notion on total dual integrality for convex, semidefinite, and extended formulations. *SIAM J. Discrete Math.*, 34(1):470–496, 2018.

[12] J. S. Caughman and J. J. P. Veerman. Kernels of directed graph Laplacians. *Electron. J. Comb.*, 13:R39, 2006.

[13] M.T. Çezik and G. Iyengar. Cuts for mixed 0-1 conic programming. *Math. Program.*, 104:179–202, 2005.

[14] A. Cerveira, A. Agra, F. Bastos, and J. Gromicho. A new branch and bound method for a discrete truss topology design problem. *Comput. Optim. Appl.*, 54(1):163–187, 2013.

[15] V. Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Math.*, 4:305–337, 1973.

[16] D. Cvetković, M. Čangalović, and V. Kovačević-Vujčić. Semidefinite programming methods for the symmetric traveling salesman problem. In G. Cornuéjols, R.E. Burkard, and G.J. Woeginger, editors, *Integer programming and Combinatorial Optimization (IPCO 1999)*, volume 1610 of *Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, 1999.

[17] J. Czyzyk, M.P. Mesnier, and J.J. Moré. The NEOS server. *IEEE Comput. Sci. Eng.*, 5(3):68–75, 1998.

[18] D. Dadush, S.S. Dey, and J.P. Vielma. The Chvátal-Gomory closure of a strictly convex body. *Math. Oper. Res.*, 36(2):227–239, 2011.

[19] D. Dadush, S.S. Dey, and J.P. Vielma. On the Chvátal-Gomory closure of a compact convex set. *Math. Program.*, 145:327–348, 2014.

[20] G. Dantzig, R. Fulkerson, and S. Johnson. Solution of a large-scale traveling-salesman problem. *Oper. Res.*, 2:393–410, 1954.

[21] S. Dash, O. Günlük, and D. Lee. On a generalization of the Chvátal-Gomory closure. *Math. Program.*, 2021. `https://doi.org/10.1007/s10107-021-01697-0`.

[22] S.S. Dey and J.P. Vielma. The Chvátal-Gomory closure of an ellipsoid is a polyhedron. In F. Eisenbrand and F.B. Shepherd, editors, *14th International Conference on Integer Programing and Combinatorial Optimization (IPCO2010)*, volume 6080 of *Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, 2010.

[23] D. Drusvyatskiy and H. Wolkowicz. The many faces of degeneracy in conic optimization. *Found. Trends Optim.*, 3(2):77–170, 2017.

[24] J. Dunkel and A.S. Schulz. The Gomory-Chvátal closure of a nonrational polytope is a rational polytope. *Math. Oper. Res.*, 38(1):63–91, 2012.

[25] I. Dunning, J. Huchette, and M. Lubin. JuMP: A modeling language for mathematical optimization. *SIAM Rev.*, 59(2):295–320, 2017.

[26] F. Eisenbrand. On the membership problem for the elementary closure of a polyhedron. *Combinatorica*, 19:297–300, 1999.

[27] K. Ellrott, C. Yang, F.M. Sladek, and T. Jiang. Identifying transcription factor binding sites through Markov chain optimization. *Bioinformatics*, 18:100–109, 2002.

[28] S. P. Fekete and D. Krupke. Covering tours and cycle covers with turn costs: hardness and approximation. In P. Heggernes, editor, *International Conference on Algorithms and Complexity (CIAC2019)*, volume 11485 of *Lecture Notes in Computer Science*, Cham, 2019. Springer.

[29] S.P. Fekete and D. Krupke. Practical methods for computing large covering tours and cycle covers with turn cost. In S. Kobourov and H. Meyerhenke, editors, *Meeting on Algorithm Engineering and Experiments (ALENEX2019)*, 2019.

[30] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Math. J.*, 23(2):298–305, 1973.

[31] A. Fischer. *A polyhedral study of quadratic traveling salesman problems*. PhD thesis, Chemnitz University of Technology, 2013.

[32] A. Fischer. An analysis of the asymmetric quadratic traveling salesman polytope. *SIAM J. Discrete Math.*, 28(1):240–276, 2014.

[33] A. Fischer, F. Fischer, G. Jäger, J. Keilwagen, P. Molitor, and I. Grosse. Exact algorithms and heuristics for the quadratic traveling salesman problem with an application in bioinformatics. *Discret. Appl. Math.*, 166:87–114, 2014.

[34] A. Fischer, F. Fischer, G. Jäger, J. Keilwagen, P. Molitor, and I. Grosse. Computational recognition of RNA splice sites by exact algorithms for the quadratic traveling salesman problem. *Comput.*, 3(2):285–298, 2015.

[35] A. Fischer and C. Helmberg. The symmetric quadratic traveling salesman problem. *Math. Program.*, 142(1–2):205–254, 2013.

[36] M. Fischetti and A. Lodi. Optimizing over the first Chvátal closure. *Math. Program.*, 110:3–20, 2007.

[37] T. Gally and M.E. Pfetsch. Computing restricted isometry constants via mixed-integer semidefinite programming. Optimization Online: `http://www.optimization-online.org/DB_FILE/2016/04/5395.pdf`, 2016.

[38] T. Gally, M.E. Pfetsch, and S. Ulbrich. A framework for solving mixed-integer semidefinite programs. *Optim. Methods Softw.*, 33(3):594–632, 2018.

[39] G. Gamrath, D. Anderson, K. Bestuzheva, W. Chen, L. Eifler, M. Gasse, P. Gemander, A. Gleixner, L. Gottwald, K. Halbig, G. Hendel, C. Hojny, T. Koch, P. Le Bodic, S.J. Maher, F. Matter, M. Miltenberger, E. Mühmer, B. Müller, M.E. Pfetsch, F. Schlösser, F. Serrano, Y. Shinano, C. Tawfik, S. Vigerske, F. Wegscheider, D. Weninger, and J. Witzig. The SCIP Optimization Suite 7.0. Technical report, Optimization Online, March 2020.

[40] F.R. Giles and W.R. Pulleyblank. Total dual integrality and integer polyhedra. *Linear Algebra Appl.*, 25:191–196, 1979.

[41] R.E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bull. Am. Math. Soc.*, 64(5):275–278, 1958.

[42] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2021. `https://www.gurobi.com`.

[43] G. Jäger and P. Molitor. Algorithms and experimental study for the traveling salesman problem of second order. In B. Yang, D.Z. Du, and C.A. Wang, editors, *International Conference on Combinatorial Optimization and Applications (COCOA2008)*, volume 5165 of *Lecture Notes in Computer Science*, pages 211–224. Springer, Berlin, Heidelberg, 2008.

[44] K. Kobayashi and Y. Takano. A branch-and-cut algorithm for solving mixed-integer semidefinite optimization problems. *Comput. Optim. Appl.*, 75:493–513, 2020.

[45] N. Krislock, J. Malick, and F. Roupin. A semidefinite branch-and-bound method for solving binary quadratic problems. *ACM Trans. Math. Softw.*, 43(4), 2017.

[46] A.N. Letchford and M.M. Sørensen. Binary positive semidefinite matrices and associated integer polytopes. *Math. Program. Series A*, 131:253–271, 2012.

[47] J. Löfberg. YALMIP : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[48] F. Matter and M.E. Pfetsch. Presolving for mixed-integer semidefinite optimization. Optimization Online: `http://www.optimization-online.org/DB_HTML/2021/10/8614.html`, 2021.

[49] F. de Meijer and R. Sotirov. The quadratic cycle cover problem: special cases and efficient bounds. *J. Comb. Optim.*, 39:1096–1128, 2020.

[50] F. de Meijer and R. Sotirov. SDP-based bounds for the quadratic cycle cover problem via cutting plane augmented Lagrangian methods and reinforcement learning. *INFORMS J. Comput.*, 33(4):1262–1276, 2021.

[51] A. P. S. MOSEK. *The MOSEK optimization toolbox for MATLAB manual. Version 9.2.*, 2020.

[52] M. Padberg. The boolean quadric polytope: some characteristics, facets and relatives. *Math. Program.*, 45:139–172, 1989.

[53] A. Philipp, S. Ulbrich, Y. Cheng, and M. Pesavento. Multiuser downlink beamforming with interference cancellation using a sdp-based branch-and-bound algorithm. In *Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7724–7728. IEEE, 2014.

[54] A.P. Punnen, M. Walter, and B.D. Woods. A characterization of linearizable instances of the quadratic traveling salesman problem. arXiv:1708.07217, 2018.

[55] G. Reinelt. TSPLIB - a traveling salesman problem library. *INFORMS J. Comput.*, 3:376–384, 1991.

[56] F. Rendl, G. Rinaldi, and A. Wiegele. Solving Max-Cut to optimality by intersecting semidefinite and polyhedral relaxations. *Math. Program.*, 121(2):307, 2010.

[57] B. Rostami, F. Malucelli, P. Belotti, and S. Gualandi. Lower bounding procedure for the asymmetric quadratic traveling salesman problem. *Eur. J. Oper. Res.*, 253(3):584–592, 2016.

[58] A. Schrijver. On cutting planes. *Ann. Discrete Math.*, 9:291–296, 1980.

[59] R. Staněk, P. Greistorfer, K. Ladner, and U. Pferschy. Geometric and LP-based heuristics for angular travelling salesman problems in the plane. *Comput. Oper. Res.*, 108:97–111, 2019.

[60] C. Stein and D.P. Wagner. Approximation algorithms for the minimum bends traveling salesman problem. In K. Aardal and B. Gerards, editors, *Integer Programming and Combinatorial Optimization (IPCO 2001)*, volume 2081 of *Lecture Notes in Computer Science*, pages 406–425. Springer, Berlin, Heidelberg, 2001.

[61] J. Stillwell. *Elements of Number Theory.* Undergraduate Texts in Mathematics. Springer-Verlag, New York, NY, 1 edition, 2003.

[62] R.A. Stubbs and S. Mehrotra. A branch-and-cut method for 0-1 mixed convex programming. *Math. Program.*, 86:515–532, 1999.

[63] J. J. P. Veerman and R. Lyons. A primer on Laplacian dynamics in directed graphs. *Nonlinear Phenom. Complex Syst.*, 23(2):196–206, 2020.

[64] H. Wirth and J. Steffan. Reload cost problems: minimum diameter spanning tree. *Discrete Appl. Math.*, 113:73–85, 2001.

[65] B.D. Woods and A.P. Punnen. A class of exponential neighbourhoods for the quadratic travelling salesman problem. *J. Comb. Optim.*, 40(2):303–332, 2020.

[66] C.W. Wu. Algebraic connectivity of directed graphs. *Linear Multilinear Algebra*, 53(3):203–223, 2005.

[67] K. Yonekura and Y. Kanno. Global optimization of robust truss topology via mixed integer semidefinite programming. *Optim. Eng.*, 11(3):355–379, 2010.

[68] X. Zhao, H. Huang, and T.P. Speed. Finding short DNA motifs using permuted Markov models. *J. Comput. Biol.*, 12:894–906, 2005.

# A Derivation of subtour elimination constraints as CG cuts

In this appendix we elaborate on the construction of the five types of subtour elimination constraints given in Table 1 as ($S$-)CG cuts.

## A.1 Ordinary subtour elimination constraint

Let $S \subseteq N$ with $|S| < n$. The well-known subtour elimination constraint corresponding to $S$ can be obtained as a CG cut, see also [13]. Let $\mathbb{1}_\mathbf{S}$ be the indicator vector of the set of nodes $S$. Then

$$\left\langle \mathbb{1}_\mathbf{S}\mathbb{1}_\mathbf{S}^\top,\, \beta\mathbf{I_n} + \alpha\mathbf{J_n} - \frac{1}{2}\left(\mathbf{X} + \mathbf{X}^\top\right)\right\rangle \geq 0$$

is a valid cut. Applying the CG procedure to this cut, yields

$$\left\langle \mathbb{1}_\mathbf{S}\mathbb{1}_\mathbf{S}^\top, \frac{1}{2}(\mathbf{X} + \mathbf{X}^\top)\right\rangle \leq \left\lfloor \left\langle \mathbb{1}_\mathbf{S}\mathbb{1}_\mathbf{S}^\top, \beta\mathbf{I_n} + \alpha\mathbf{J_n}\right\rangle \right\rfloor \quad \Longleftrightarrow \quad \sum_{i\in S, j\in S} x_{ij} \leq \lfloor |S|\left(\beta + \alpha|S|\right)\rfloor .$$

If $\beta = k_n$ and $\alpha = h_n/n$, then for all $S$ with $|S| < n$ we have $\beta + \alpha|S| < 1$. Hence, the CG cut above implies

$$\sum_{i\in S, j\in S} x_{ij} \leq |S| - 1. \tag{34}$$

The cut (34) is the common subtour elimination constraint introduced by Dantzig et al. [20].

## A.2 Cut-set subtour elimination constraints

The cut-set subtour elimination constraints are known to be equivalent to the ordinary subtour elimination constraints of [20]. It is therefore no surprise that these cuts can be obtained similarly as the ordinary subtour elimination constraints.

Let $\mathbf{U} = \mathbb{1}_\mathbf{S}\mathbb{1}_\mathbf{S}^\top$ be the dual multiplier of the linear matrix inequality $\beta\mathbf{I_n} + \alpha\mathbf{J_n} - \frac{1}{2}\left(\mathbf{X} + \mathbf{X}^\top\right)$ and let $\mathbb{1}_\mathbf{S}$ be the dual multiplier of the constraints $-\mathbf{X1} = -\mathbf{1}$. The sum of these constraints yields

$$\left\langle \mathbb{1}_\mathbf{S}\mathbb{1}_\mathbf{S}^\top, \frac{1}{2}(\mathbf{X}+\mathbf{X}^\top)\right\rangle - \mathbb{1}_\mathbf{S}^\top\mathbf{X1} \leq \left\lfloor \left\langle \mathbb{1}_\mathbf{S}\mathbb{1}_\mathbf{S}^\top, \beta\mathbf{I_n} + \alpha\mathbf{J_n}\right\rangle - \mathbb{1}_\mathbf{S}^\top\mathbf{1}\right\rfloor$$
$$\iff \quad -\sum_{i\in S, j\notin S} x_{ij} \leq \lfloor |S|\left(\beta + \alpha|S|\right)\rfloor - |S|.$$

If $\beta = k_n$ and $\alpha = h_n/n$, then the right hand side becomes $|S| - 1 - |S| = -1$, which yields the desired cut.

## A.3 Merged subtour elimination constraint

Let $(S_1, \ldots, S_k)$ be a partition of the node set of $G$, i.e., $\bigcup_{l=1}^k S_l = N$ and $S_l \cap S_p = \emptyset$ for all $l \neq p$. We can obtain a merged subtour elimination constraint via the CG procedure in the following way.

Let $\mathbf{U} = 2\sum_{l=1}^k \mathbb{1}_{\mathbf{S_l}}\mathbb{1}_{\mathbf{S_l}}^\top$ be the dual multiplier for $\beta\mathbf{I_n} + \alpha\mathbf{J_n} - \frac{1}{2}\left(\mathbf{X} + \mathbf{X}^\top\right)$. Since each dual multiplier $\mathbb{1}_{\mathbf{S_l}}\mathbb{1}_{\mathbf{S_l}}^\top$ leads to a CG cut of Type I in Table 1, its weighted sum also belongs to the elementary closure and looks as follows:

$$2\sum_{l=1}^k \sum_{\substack{i\in S_l\\ j\in S_l}} x_{ij} \leq 2\sum_{l=1}^k (|S_l| - 1) = 2(n-k).$$

Now we add to this cut the equality $-\mathbf{X1} = -\mathbf{1}$ with dual multiplier $\mathbf{1}$, which yields the desired merged cut

$$2\sum_{l=1}^k \sum_{\substack{i\in S_l\\ j\in S_l}} x_{ij} - \mathbf{1}^\top\mathbf{X1} \leq 2(n-k) - \mathbf{1}^\top\mathbf{1} \quad \iff \quad \sum_{l=1}^k \sum_{\substack{i\in S_l\\ j\in S_l}} x_{ij} - \sum_{l\neq p}\sum_{\substack{i\in S_l\\ j\in S_p}} x_{ij} \leq n - 2k.$$

## A.4 Strengthened subtour elimination constraints of size two

Let $i \neq j$ and define $\mathbf{U} = \mathbb{1}_{\{\mathbf{i},\mathbf{j}\}}\mathbb{1}_{\{\mathbf{i},\mathbf{j}\}}^\top$. Taking $\mathbf{U}$ as the dual multiplier with respect to $\beta^{(2)}\mathbf{I_n} + \alpha^{(2)}\mathbf{J_n} - \frac{1}{2}\left((\mathbf{X} + \mathbf{X^{(2)}}) + (\mathbf{X} + \mathbf{X^{(2)}})^\top\right) \succeq \mathbf{0}$, provides the following valid cut:

$$\left\langle \mathbb{1}_{\{\mathbf{i},\mathbf{j}\}}\mathbb{1}_{\{\mathbf{i},\mathbf{j}\}}^\top, \beta^{(2)}\mathbf{I_n} + \alpha^{(2)}\mathbf{J_n} - \frac{1}{2}\left((\mathbf{X}+\mathbf{X^{(2)}}) + (\mathbf{X}+\mathbf{X^{(2)}})^\top\right)\right\rangle \geq 0.$$

Moreover, adding the coupling constraints $\sum_{k\in N:(i,k,j)\in\mathcal{A}} y_{ikj} - x_{ij}^{(2)} = 0$ and $\sum_{k\in N:(j,k,i)\in\mathcal{A}} y_{jki} - x_{ji}^{(2)} = 0$, each with dual multiplier 1, and the constraints $-x_{ii} = 0$, $-x_{jj} = 0$, $-x_{ii}^{(2)} = 0$ and $-x_{jj}^{(2)} = 0$, also each with dual mulitplier 1, gives

$$x_{ij} + x_{ji} + \sum_{\substack{k\in N:\\ (i,k,j)\in\mathcal{A}}} y_{ikj} + \sum_{\substack{k\in N:\\ (j,k,i)\in\mathcal{A}}} y_{jki} \leq 2\beta^{(2)} + 4\alpha^{(2)}.$$

We now take $\beta^{(2)} = k_n^{(2)}$ and $\alpha^{(2)} = h_n^{(2)}/n$. Applying the standard CG procedure to this inequality results in the cut

$$x_{ij} + x_{ji} + \sum_{\substack{k\in N:\\ (i,k,j)\in\mathcal{A}}} y_{ikj} + \sum_{\substack{k\in N:\\ (j,k,i)\in\mathcal{A}}} y_{jki} \leq \left\lfloor 2k_n^{(2)} + 4\frac{h_n^{(2)}}{n}\right\rfloor. \tag{35}$$

The right hand side of this cut equals one if $5 \le n \le 7$, two if $8 \le n \le 12$ and three if $n \ge 13$.

For $n \ge 5$, we can strengthen this cut by applying the $S$-CG procedure as explained in Section 2.5. Since the cut (35) only involves variables $\mathbf{y}$ and $\mathbf{X}$, we can restrict the set $S$ to the space corresponding to these variables. Let $S = \mathcal{F}_1 \cap \left(\{0,1\}^{\mathcal{A}} \times \mathcal{T}_n(G)\right)$ and let $c_1$ be the coefficient vector of the left hand side in (35). Then the strengthened rounding looks as follows:

$$\left\lfloor 2k_n^{(2)} + 4\frac{h_n^{(2)}}{n} \right\rfloor_{S,c_1} := \max\left\{ x_{ij} + x_{ji} + \sum_{\substack{k\in N:\\(i,k,j)\in\mathcal{A}}} y_{ikj} + \sum_{\substack{k\in N:\\(j,k,i)\in\mathcal{A}}} y_{jki} \;:\; (35),\, (\mathbf{y},\mathbf{X}) \in S \right\}.$$

One can verify that the value of this maximization is equal to 1 for $n \ge 5$. Namely, if it would be greater than 1, this implies a subtour of size two (if $x_{ij} = x_{ji} = 1$), size three (e.g., if $x_{ij} = 1$ and $y_{jki} = 1$ for some $k \in N \setminus \{i,j\}$) or size four (e.g., if $y_{ikj} = 1$ and $y_{jli} = 1$ for some distinct $k, l \in N \setminus \{i,j\}$), which contradicts the fact that $\mathbf{X} \in \mathcal{T}_n(G)$. We conclude that $\left\lfloor 2k_n^{(2)} + 4\frac{h_n^{(2)}}{n} \right\rfloor_{S,c_1} = 1$. Thus, we obtain the strengthened CG cut

$$x_{ij} + x_{ji} + \sum_{\substack{k\in N:\\(i,k,j)\in\mathcal{A}}} y_{ikj} + \sum_{\substack{k\in N:\\(j,k,i)\in\mathcal{A}}} y_{jki} \le 1.$$

## A.5 Strenghtened subtour elimination constraints

Let $S \subset N$ with $2 \le |S| < \frac{1}{2}n$ and define $\mathbf{U} = \mathbb{1}_\mathbf{S}\mathbb{1}_\mathbf{S}^\top$. Taking $\mathbf{U}$ as the dual multiplier with respect to $\beta^{(2)}\mathbf{I_n} + \alpha^{(2)}\mathbf{J_n} - \frac{1}{2}\left((\mathbf{X} + \mathbf{X^{(2)}}) + (\mathbf{X} + \mathbf{X^{(2)}})^\top\right) \succeq \mathbf{0}$ provides the inequality

$$\left\langle \mathbb{1}_\mathbf{S}\mathbb{1}_\mathbf{S}^\top, \beta^{(2)}\mathbf{I_n} + \alpha^{(2)}\mathbf{J_n} - \frac{1}{2}\left((\mathbf{X} + \mathbf{X^{(2)}}) + (\mathbf{X} + \mathbf{X^{(2)}})^\top\right)\right\rangle \ge 0.$$

For all $i,j \in S$ we now add the coupling constraints $\sum_{k\in N:(i,k,j)\in\mathcal{A}} y_{ikj} - x_{ij}^{(2)} = 0$ with dual multiplier 1. Moreover, for all $(i,k,j) \in \mathcal{A}$ with $i,k,j \in S$ we add the constraint $-y_{ikj} \le 0$ with multiplier 1. This yields the following valid cut

$$\sum_{\substack{i\in S\\j\in S}} x_{ij} + \sum_{\substack{i\in S\\j\in S}} \sum_{\substack{k\in N\setminus S:\\(i,k,j)\in\mathcal{A}}} y_{ikj} \le |S|\beta^{(2)} + |S|^2\alpha^{(2)}.$$

Again, we take $\beta^{(2)} = k_n^{(2)}$ and $\alpha^{(2)} = h_n^{(2)}/n$. The standard CG rounding step yields

$$\sum_{\substack{i\in S\\j\in S}} x_{ij} + \sum_{\substack{i\in S\\j\in S}} \sum_{\substack{k\in N\setminus S:\\(i,k,j)\in\mathcal{A}}} y_{ikj} \le \left\lfloor |S|\left(k_n^{(2)} + |S|\frac{h_n^{(2)}}{n}\right)\right\rfloor. \tag{36}$$

Since $|S| < \frac{1}{2}n$, we know

$$\left\lfloor |S|\left(k_n^{(2)} + |S|\frac{h_n^{(2)}}{n}\right)\right\rfloor \le \left\lfloor |S|\left(k_n^{(2)} + \frac{1}{2}n\frac{2-k_n^{(2)}}{n}\right)\right\rfloor = \left\lfloor |S|\left(1 + \frac{1}{2}k_n^{(2)}\right)\right\rfloor \le 2|S| - 1.$$

However, similar to the approach in Appendix A.4, we obtain a tighter bound if we apply the strengthened CG procedure. Let $T = \mathcal{F}_1 \cap \left(\{0,1\}^{\mathcal{A}} \times \mathcal{T}_n(G)\right)$ and let $c_2$ be the coefficient vector of the left hand side of (36). Then,

$$\left\lfloor |S|\left(k_n^{(2)} + |S|\frac{h_n^{(2)}}{n}\right)\right\rfloor_{T,c_2} := \max\left\{ \sum_{\substack{i\in S\\j\in S}} x_{ij} + \sum_{\substack{i\in S\\j\in S}} \sum_{\substack{k\in N\setminus S:\\(i,k,j)\in\mathcal{A}}} y_{ikj} \;:\; (36),\, (\mathbf{y},\mathbf{X}) \in T \right\}.$$

It can be verified that this maximum is equal to $|S|-1$ for all $S$ with $|S| < \frac{1}{2}n$. Namely, if $(\mathbf{y}, \mathbf{X}) \in T$, we cannot have both $x_{ij} = 1$ and $y_{ikj} = 1$ for some $k \in N$. Hence, $x_{ij} + \sum_{k \in N \setminus S:(i,k,j)\in\mathcal{A}} y_{ikj} \leq 1$ for all $i, j \in S$. If we now sum over all $i, j \in S$, the result must be at most $|S| - 1$, otherwise a subtour would exist. The strengthened CG cut corresponding to (36) becomes

$$\sum_{\substack{i\in S\\ j\in S}} x_{ij} + \sum_{\substack{i\in S\\ j\in S}} \sum_{\substack{k\in N\setminus S:\\ (i,k,j)\in\mathcal{A}}} y_{ikj} \leq |S| - 1.$$

# B The symmetric quadratic traveling salesman problem

In this appendix we briefly consider the symmetric quadratic traveling salesman problem (SQTSP). Although this problem is very related to the asymmetric version used in the rest of the paper (that we continue to denote by QTSP), the underlying model is different. We show how to construct this model and how all cuts for the QTSP can be extended to the symmetric case.

Let $G = (V, E)$ be an undirected graph, where $E$ consists of undirected pairs of nodes $\{i, j\}$ $(= \{j, i\})$, $i, j \in V$. We define $\mathcal{E} = \{\langle i, j, k\rangle = \langle k, j, i\rangle \ : \ i, j, k \in V, |\{i, j, k\}| = 3\}$ to be the set of two-edges in $G$, where a two-edge is a sequence of three distinct nodes where the reverse sequence is regarded as identical. Given is a quadratic cost matrix $\mathbf{Q} = (q_{ijk})$, where a cost is zero if $\langle i, j, k\rangle \notin \mathcal{E}$.

The goal of the SQTSP is to find an undirected Hamiltonian cycle in $G$ such that the total quadratic cost is minimized. To model this problem, let $\bar{\mathbf{x}} \in \{0, 1\}^E$ and $\bar{\mathbf{y}} \in \{0, 1\}^{\mathcal{E}}$ denote indicator vectors that are 1 if and edge, respectively two-edge, is present in the solution and 0 otherwise. We aim to find a tuple $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ with $\bar{y}_{ijk} = \bar{x}_{ij}\bar{x}_{jk}$, representing a Hamiltonian cycle such that $\sum_{\langle i,jk\rangle\in\mathcal{E}} q_{ijk}y_{ijk}$ is minimized.

The symmetric equivalent of the set $\mathcal{F}_1$, see (28), is now given by:

$$\mathcal{F}_1^s := \left\{ (\bar{\mathbf{y}}, \bar{\mathbf{x}}) \in \{0,1\}^{\mathcal{E}} \times \{0,1\}^E \ : \ \begin{array}{r} \displaystyle\sum_{e\in\delta(i)} \bar{x}_e = 2 \quad \forall i \in V \\ \displaystyle\bar{x}_{ij} = \sum_{\substack{k\in V\\ \langle i,j,k\rangle\in\mathcal{E}}} \bar{y}_{ijk} = \sum_{\substack{k\in V\\ \langle k,i,j\rangle\in\mathcal{E}}} \bar{y}_{kij} \ \forall \{i,j\} \in E \end{array} \right\},$$

where $\delta(i) \in V$ denotes the set of edges that are incident to $i$. The formulation used in $\mathcal{F}_1^s$ is introduced by Fischer and Helmberg [35] where it is shown that the equation $\bar{y}_{ijk} = \bar{x}_{ij}\bar{x}_{jk}$ is indeed established for all $\langle i, j, k\rangle \in \mathcal{E}$. Moreover, similar to the asymmetric case, we can relax the integrality of $\bar{\mathbf{y}}$, since it is enforced by the integrality of $\bar{\mathbf{x}}$ and the coupling constraints, see Remark 1. It follows that the tuples in $\mathcal{F}_1^s$ are characteristic vectors of node-disjoint cycle covers in $G$, where the smallest cycles have size 3 due to the definition of $\mathcal{E}$.

The B&C algorithm presented in Section 3 can now be applied to the SQTSP, starting from optimizing over $\mathcal{F}_1^s$. In order to cut off solutions that do not correspond to a Hamiltonian cycle in $G$, we need separation routines for the symmetric version. Instead of providing symmetric equivalents to all QTSP cutting planes derived in Section 4.3, we present a transformation that maps any valid cut for the asymmetric version to a cut for the SQTSP. To that end, we introduce a directed graph $H = (V, A)$ that is defined on the same set of nodes as the undirected graph $G$, where $A$ is such that it contains both pairs $(i, j)$ and $(j, i)$ whenever the corresponding edge $\{i, j\}$ is contained in $G$. Moreover, we define the cost of each two-arc $(i, j, k)$ in $H$ to be equal to $q_{\langle i,j,k\rangle}$ for the corresponding two-edge $\langle i, j, k\rangle$ in $G$. Let $\mathcal{I}_{\mathrm{S}}$ denote the original SQTSP instance and let $\mathcal{I}_{\mathrm{A}}$ denote the corresponding asymmetric instance.

The variables in the two programs can now be related as follows: Let $(\mathbf{y}, \mathbf{X})$ be variables in $\mathcal{I}_{\mathrm{A}}$ and define the tuple $(\bar{\mathbf{y}}, \bar{\mathbf{x}})$ by

$$\begin{aligned} \bar{x}_{ij} &= x_{ij} + x_{ji} \quad \text{for all} \quad \{i, j\} \in E \\ \bar{y}_{ijk} &= y_{ijk} + y_{kji} \quad \text{for all} \quad \langle i, j, k\rangle \in \mathcal{E}. \end{aligned}$$

From the constraints in $\mathcal{F}_1$ and $\mathcal{F}_1^s$, it follows that any solution $(\mathbf{y}, \mathbf{X})$ in $\mathcal{I}_\mathrm{A}$ leads to a solution $(\bar{\mathbf{y}}, \bar{\mathbf{x}})$ in $\mathcal{I}_\mathrm{S}$ with the same objective value. Reversely, any solution $(\bar{\mathbf{y}}, \bar{\mathbf{x}})$ in $\mathcal{I}_\mathrm{S}$ leads to a solution (or actually two solutions, one for each direction) $(\mathbf{y}, \mathbf{X})$ in $\mathcal{I}_\mathrm{A}$ with the same objective value. As a result, any valid cut for $\mathcal{I}_\mathrm{A}$ is also valid for $\mathcal{I}_\mathrm{S}$.

This implies that all cuts defined in Section 4.3 can be converted to cuts for the SQTSP. Namely, given a cut for $\mathcal{I}_\mathrm{A}$, we define the coefficient on $\bar{x}_{ij}$ to be the sum of the coefficients on $x_{ij}$ and $x_{ji}$ for all edges $\{i, j\} \in E$. Similarly, we define the coefficient on $\bar{y}_{ijk}$ to be the sum of the coefficients on $y_{ijk}$ and $y_{kji}$ for all two-edges $\langle i, j, k\rangle \in \mathcal{E}$. If no more violated cuts can be found in $\mathcal{I}_\mathrm{A}$, the corresponding solution in $\mathcal{I}_\mathrm{S}$ is also optimal. This proves the validity of the B&C algorithm for the symmetric version of the problem.

# C Extended computational results

In this appendix we present a complete overview of the computational results from which the summarized results in Section 5 follow. We start by considering the instances from bioinformatics, after which we present results for the reload instances. No additional results are presented for the turn instances, since for these instances the complete overview is already given in Section 5.

In all tables showing computation times, the setting that provides the shortest time is presented in bold for each instance. Moreover, a '-' indicates that a given algorithm could not solve the instance within 8 hours.

The computation times and the number of branching nodes for the class of 'bma' instances from bioinformatics are given in Table 6 and 7, respectively. Table 8 and 9 provide computation times and number of branching nodes for the 'map' instances, respectively. The computation times and the number of branching nodes for the 'ml' instances are presented in Table 10 and 11, respectively.

Finally, we present a more elaborate overview of the reload instances. For each of the two classes and for different values of $n, p$ and $c$, 10 randomly generated instances are considered. In order to save space, we only present the results that are averaged over these 10 similar instances. Table 12 and 13 present the computation times and number of branching nodes, respectively.

| Instance | SCIP-SDP | KT | CG1 | CG2 | SEC-simple | SEC | SEC-CG |
|---|---|---|---|---|---|---|---|
| bma2_3 | **0.001** | 0.001 | 0.006 | 0.153 | **0.001** | **0.001** | **0.001** |
| bma2_4 | 0.075 | **0.001** | 0.002 | 0.002 | 0.002 | 0.002 | **0.001** |
| bma2_5 | 0.079 | **0.005** | 0.004 | 0.017 | **0.005** | **0.005** | 0.008 |
| bma2_6 | 0.221 | 0.034 | 0.034 | 0.047 | **0.029** | 0.555 | 0.101 |
| bma2_7 | 0.436 | 0.048 | 0.051 | 0.062 | 0.054 | 0.157 | **0.039** |
| bma2_8 | 2.76 | 0.083 | 0.073 | 0.081 | 0.090 | **0.049** | 0.102 |
| bma2_9 | 3.85 | 0.114 | 0.121 | 0.165 | 0.120 | **0.028** | 0.043 |
| bma2_10 | 17.11 | 0.179 | 0.205 | 0.374 | 0.185 | **0.056** | 0.29 |
| bma2_11 | 91.38 | 0.496 | 0.442 | 0.841 | 0.272 | 0.281 | **0.251** |
| bma2_12 | 440.6 | 0.667 | 0.8 | 1.428 | 0.333 | **0.203** | 0.303 |
| bma2_13 | 1327 | 2.005 | 1.006 | 2.954 | 0.501 | **0.154** | 0.671 |
| bma2_14 | 4794 | 2.76 | 2.524 | 4.396 | 2.190 | **0.582** | 0.82 |
| bma2_15 | 13075 | 7.118 | 4.136 | 7.313 | 2.811 | **0.488** | 1.27 |
| bma2_16 | - | 11.83 | 6.129 | 12.24 | 2.962 | **0.247** | 1.66 |
| bma2_17 | - | 24.19 | 6.214 | 23.87 | 4.509 | **1.457** | 3.105 |
| bma2_18 | - | 58.43 | 16.42 | 84.56 | 9.579 | **1.489** | 3.699 |
| bma2_19 | - | 49.98 | 20.66 | 54.24 | 13.97 | **1.386** | 3.043 |
| bma2_20 | - | 98.18 | 20.37 | 127.9 | 20.40 | **1.516** | 3.496 |
| bma2_21 | - | 1068 | 45.41 | 103.2 | 19.64 | 2.387 | **2.086** |
| bma2_22 | - | 2120 | 59.95 | 464.3 | 52.09 | **3.093** | 8.663 |
| bma2_23 | - | 3855 | 117 | 527.0 | 87.85 | **4.76** | 9.586 |
| bma2_24 | - | 2461 | 90.07 | 945.9 | 81.87 | **5.866** | 17.73 |
| bma2_25 | - | 26594 | 160 | 1129 | 166.3 | **7.53** | 7.765 |
| bma2_26 | - | - | 451.6 | 3777 | 269.2 | **6.779** | 38.962 |
| bma2_27 | - | - | 372.6 | 8718 | 350.0 | **3.108** | 46.95 |
| bma2_28 | - | - | 1628 | - | 440.7 | **22.47** | 46.5 |
| bma2_29 | - | - | 2095 | - | 2700 | **10.65** | 229 |
| bma2_30 | - | - | 2453 | - | 1614 | **23.59** | 42.8 |
| bma2_31 | - | - | 5997 | - | 3434 | **32.72** | 63.66 |
| bma2_32 | - | - | 11835 | - | 2227 | **41.79** | 168.2 |
| bma2_33 | - | - | - | - | 7985 | **42.97** | 150 |
| bma2_34 | - | - | - | - | 6673 | **67.8** | 845.3 |
| bma2_35 | - | - | - | - | - | 98.17 | **77.03** |
| bma2_36 | - | - | - | - | - | 173.3 | **15.26** |
| bma2_37 | - | - | - | - | - | 101.1 | **35.64** |
| bma2_38 | - | - | - | - | - | **96.33** | 2127 |
| bma2_39 | - | - | - | - | - | **86.46** | 1012 |
| bma2_40 | - | - | - | - | - | **236.2** | 1964 |

Table 6: Computation times (s) for bioinformatics instances from the 'bma' class.

| Instance | SCIP-SDP | KT | CG1 | CG2 | SEC-simple | SEC | SEC-CG |
|---|---|---|---|---|---|---|---|
| bma2_3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| bma2_4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| bma2_5 | 15 | 1 | 1 | 1 | 1 | 1 | 1 |
| bma2_6 | 31 | 21 | 15 | 8 | 1 | 1 | 4 |
| bma2_7 | 91 | 44 | 44 | 56 | 20 | 1 | 1 |
| bma2_8 | 223 | 188 | 162 | 117 | 106 | 5 | 10 |
| bma2_9 | 488 | 439 | 468 | 391 | 101 | 1 | 1 |
| bma2_10 | 1477 | 777 | 1072 | 1194 | 289 | 1 | 238 |
| bma2_11 | 3907 | 3397 | 2639 | 2608 | 431 | 48 | 44 |
| bma2_12 | 13601 | 3428 | 3634 | 4060 | 670 | 54 | 50 |
| bma2_13 | 29649 | 8517 | 2240 | 3160 | 947 | 1 | 177 |
| bma2_14 | 94149 | 6175 | 11496 | 3545 | 4977 | 24 | 50 |
| bma2_15 | 250373 | 20247 | 10150 | 3991 | 3981 | 9 | 41 |
| bma2_16 | - | 27932 | 9935 | 6401 | 5960 | 1 | 106 |
| bma2_17 | - | 63530 | 8223 | 9184 | 7025 | 308 | 169 |
| bma2_18 | - | 123582 | 24108 | 36441 | 8876 | 91 | 216 |
| bma2_19 | - | 70434 | 22172 | 11652 | 10536 | 1 | 40 |
| bma2_20 | - | 160135 | 18042 | 28155 | 13877 | 5 | 13 |
| bma2_21 | - | 1022644 | 39130 | 14514 | 10066 | 57 | 1 |
| bma2_22 | - | 1700860 | 32304 | 72836 | 26889 | 292 | 113 |
| bma2_23 | - | 1832083 | 75655 | 77233 | 42630 | 128 | 85 |
| bma2_24 | - | 726287 | 37575 | 104902 | 29058 | 172 | 91 |
| bma2_25 | - | 13893446 | 74107 | 77935 | 52053 | 155 | 28 |
| bma2_26 | - | - | 196693 | 280367 | 34152 | 89 | 228 |
| bma2_27 | - | - | 103566 | 324133 | 72630 | 3 | 133 |
| bma2_28 | - | - | 299599 | - | 90531 | 549 | 130 |
| bma2_29 | - | - | 491304 | - | 514762 | 264 | 714 |
| bma2_30 | - | - | 406116 | - | 204763 | 224 | 254 |
| bma2_31 | - | - | 755325 | - | 481742 | 493 | 53 |
| bma2_32 | - | - | 948536 | - | 221932 | 637 | 245 |
| bma2_33 | - | - | - | - | 488976 | 483 | 122 |
| bma2_34 | - | - | - | - | 423497 | 575 | 583 |
| bma2_35 | - | - | - | - | - | 723 | 49 |
| bma2_36 | - | - | - | - | - | 690 | 3 |
| bma2_37 | - | - | - | - | - | 416 | 3 |
| bma2_38 | - | - | - | - | - | 240 | 903 |
| bma2_39 | - | - | - | - | - | 205 | 259 |
| bma2_40 | - | - | - | - | - | 668 | 258 |

Table 7: Number of branching nodes for bioinformatics instances from the 'bma' class.

| Instance | SCIP-SDP | KT | CG1 | CG2 | SEC-simple | SEC | SEC-CG |
|---|---|---|---|---|---|---|---|
| map2_3 | **0** | 0.004 | **0** | 0.001 | 0.001 | 0.001 | **0** |
| map2_4 | 0.132 | **0.001** | **0.001** | **0.001** | 0.002 | **0.001** | **0.001** |
| map2_5 | 0.149 | **0.005** | 0.007 | 0.088 | **0.005** | 0.006 | 0.022 |
| map2_6 | 0.291 | 0.033 | 0.026 | 0.042 | 0.051 | **0.021** | 0.029 |
| map2_7 | 1.551 | 0.06 | 0.057 | 0.147 | 0.058 | **0.027** | 0.032 |
| map2_8 | 3.175 | 0.094 | 0.095 | 0.104 | 0.064 | **0.043** | 0.056 |
| map2_9 | 8.373 | 0.128 | 0.096 | 0.154 | 0.106 | **0.03** | 0.044 |
| map2_10 | 34.23 | 0.254 | 0.193 | 0.376 | 0.189 | **0.091** | 0.324 |
| map2_11 | 143.8 | 0.464 | 0.512 | 0.671 | 0.227 | 0.127 | **0.094** |
| map2_12 | 436.4 | 0.944 | 0.99 | 1.316 | 0.628 | **0.171** | 0.233 |
| map2_13 | 1393 | 1.751 | 1.268 | 3.007 | 0.507 | **0.149** | 0.189 |
| map2_14 | 5342 | 2.718 | 1.532 | 9.388 | 0.887 | **0.546** | 0.856 |
| map2_15 | 21851 | 4.57 | 2.691 | 8.088 | 1.778 | **0.742** | 1.17 |
| map2_16 | - | 11.95 | 4.292 | 16.69 | 2.826 | **0.985** | 1.627 |
| map2_17 | - | 14.2 | 6.987 | 17.79 | 9.434 | **1.654** | 2.622 |
| map2_18 | - | 29.77 | 10.03 | 49.97 | 12.14 | **1.24** | 4.798 |
| map2_19 | - | 106.1 | 15.15 | 62.60 | 16.11 | 1.913 | **1.266** |
| map2_20 | - | 70.09 | 22.55 | 120.70 | 18.61 | **2.621** | 4.935 |
| map2_21 | - | 3831 | 29.79 | 162.26 | 23.55 | 3.384 | **1.943** |
| map2_22 | - | 1891 | 57.3 | 533.68 | 53.73 | **3.495** | 6.949 |
| map2_23 | - | 7268 | 107.5 | 847.80 | 108.8 | **3.334** | 8.991 |
| map2_24 | - | 24633 | 214.3 | 1346 | 80.45 | **5.487** | 39.06 |
| map2_25 | - | - | 517.4 | 1159 | 187.9 | 14.19 | **12.18** |
| map2_26 | - | - | 1097 | 1874 | 280.3 | **12.67** | 16.07 |
| map2_27 | - | - | 675.5 | 13952 | 373.0 | **1.951** | 22.09 |
| map2_28 | - | - | 3809 | - | 415.5 | **12.58** | 98.8 |
| map2_29 | - | - | 1724 | - | 740.6 | **15.76** | 84.06 |
| map2_30 | - | - | 25983 | - | 1236 | **15.13** | 236.9 |
| map2_31 | - | - | 3090 | - | 1109 | **41.91** | 74.52 |
| map2_32 | - | - | 15660 | - | 8951 | **36.15** | 200.8 |
| map2_33 | - | - | - | - | 14630 | **51.88** | 239.5 |
| map2_34 | - | - | - | - | - | **80.5** | 452.2 |
| map2_35 | - | - | - | - | - | **12.68** | 35.15 |
| map2_36 | - | - | - | - | - | 154.6 | **16.87** |
| map2_37 | - | - | - | - | - | **66.77** | 990.2 |
| map2_38 | - | - | - | - | - | **200.2** | 2564 |
| map2_39 | - | - | - | - | - | **78.28** | 1333 |
| map2_40 | - | - | - | - | - | **160.1** | 1135 |

Table 8: Computation times (s) for bioinformatics instances from the 'map' class.

| Instance | SCIP-SDP | KT | CG1 | CG2 | SEC-simple | SEC | SEC-CG |
|---|---|---|---|---|---|---|---|
| map2_3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| map2_4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| map2_5 | 11 | 1 | 1 | 1 | 1 | 1 | 1 |
| map2_6 | 40 | 14 | 17 | 8 | 5 | 1 | 6 |
| map2_7 | 109 | 89 | 89 | 35 | 25 | 1 | 1 |
| map2_8 | 232 | 189 | 160 | 90 | 93 | 1 | 1 |
| map2_9 | 571 | 457 | 405 | 262 | 113 | 1 | 1 |
| map2_10 | 1304 | 1195 | 823 | 964 | 122 | 1 | 508 |
| map2_11 | 3935 | 2322 | 3696 | 1760 | 278 | 8 | 1 |
| map2_12 | 13169 | 5393 | 4928 | 2531 | 1203 | 1 | 1 |
| map2_13 | 26604 | 5467 | 2650 | 5373 | 875 | 3 | 1 |
| map2_14 | 93888 | 5014 | 2548 | 7493 | 1982 | 57 | 60 |
| map2_15 | 255144 | 6920 | 5496 | 3707 | 4982 | 18 | 108 |
| map2_16 | - | 26031 | 5539 | 8806 | 5520 | 80 | 51 |
| map2_17 | - | 22364 | 8195 | 6804 | 12704 | 272 | 125 |
| map2_18 | - | 48020 | 14204 | 11778 | 14115 | 144 | 196 |
| map2_19 | - | 183060 | 14306 | 19651 | 10824 | 260 | 1 |
| map2_20 | - | 92733 | 22274 | 26159 | 12710 | 17 | 58 |
| map2_21 | - | 4794948 | 23121 | 19926 | 10604 | 337 | 1 |
| map2_22 | - | 1224783 | 41964 | 74130 | 22431 | 118 | 63 |
| map2_23 | - | 3642274 | 57896 | 180470 | 48598 | 161 | 90 |
| map2_24 | - | 9658211 | 137416 | 200038 | 31283 | 229 | 334 |
| map2_25 | - | - | 288931 | 114603 | 30430 | 937 | 232 |
| map2_26 | - | - | 500653 | 121344 | 86160 | 1077 | 66 |
| map2_27 | - | - | 177516 | 599002 | 36414 | 1 | 136 |
| map2_28 | - | - | 1132717 | - | 42713 | 322 | 408 |
| map2_29 | - | - | 290590 | - | 107281 | 187 | 226 |
| map2_30 | - | - | 2812907 | - | 194767 | 112 | 545 |
| map2_31 | - | - | 384952 | - | 66366 | 735 | 128 |
| map2_32 | - | - | 1437957 | - | 691596 | 470 | 784 |
| map2_33 | - | - | - | - | 1041749 | 238 | 165 |
| map2_34 | - | - | - | - | - | 537 | 307 |
| map2_35 | - | - | - | - | - | 1 | 1 |
| map2_36 | - | - | - | - | - | 1121 | 1 |
| map2_37 | - | - | - | - | - | 149 | 366 |
| map2_38 | - | - | - | - | - | 1145 | 908 |
| map2_39 | - | - | - | - | - | 212 | 461 |
| map2_40 | - | - | - | - | - | 313 | 215 |

Table 9: Number of branching nodes for bioinformatics instances from the 'map' class.

| Instance | SCIP-SDP | KT | CG1 | CG2 | SEC-simple | SEC | SEC-CG |
|---|---|---|---|---|---|---|---|
| ml2_3 | **0** | 0.001 | **0** | 0.001 | **0** | **0** | 0.001 |
| ml2_4 | 0.056 | 0.002 | **0.001** | 0.002 | **0.001** | **0.001** | **0.001** |
| ml2_5 | 0.089 | 0.004 | 0.004 | 0.027 | 0.007 | **0.003** | 0.019 |
| ml2_6 | 0.182 | 0.032 | 0.034 | 0.035 | 0.044 | **0.015** | **0.015** |
| ml2_7 | 0.402 | 0.064 | 0.057 | 0.063 | 0.056 | **0.024** | 0.04 |
| ml2_8 | 1.769 | 0.075 | 0.075 | 0.100 | 0.077 | **0.039** | 0.089 |
| ml2_9 | 4.528 | 0.118 | 0.116 | 0.172 | 0.112 | **0.026** | 0.04 |
| ml2_10 | 11.19 | 0.196 | 0.225 | 0.419 | 0.151 | **0.109** | 0.302 |
| ml2_11 | 115.9 | 0.56 | 0.407 | 0.753 | 0.243 | 0.193 | **0.19** |
| ml2_12 | 734.2 | 1.094 | 1.019 | 1.530 | 0.331 | 0.208 | **0.176** |
| ml2_13 | 2544 | 1.379 | 1.032 | 4.594 | 0.615 | **0.302** | 0.334 |
| ml2_14 | 9235 | 3.404 | 2.218 | 7.348 | 1.117 | **0.455** | 0.647 |
| ml2_15 | 24930 | 3.827 | 2.397 | 7.858 | 1.468 | **0.633** | 1.888 |
| ml2_16 | - | 7.03 | 7.557 | 14.35 | 1.817 | **0.993** | 1.684 |
| ml2_17 | - | 11.08 | 8.268 | 33.36 | 5.035 | **1.282** | 2.008 |
| ml2_18 | - | 34.76 | 13.61 | 54.37 | 7.403 | **0.988** | 3.594 |
| ml2_19 | - | 83.41 | 20.42 | 57.72 | 9.350 | 1.337 | **0.611** |
| ml2_20 | - | 313.1 | 22.72 | 90.01 | 14.42 | **2.731** | 4.522 |
| ml2_21 | - | 820.5 | 26.01 | 139.9 | 27.21 | 2.82 | **1.226** |
| ml2_22 | - | 2159 | 82.52 | 483.6 | 59.51 | **4.997** | 6.803 |
| ml2_23 | - | 15380 | 105.3 | 627.5 | 77.46 | **4.556** | 13.15 |
| ml2_24 | - | 10106 | 170.2 | 871.8 | 106.2 | **6.188** | 28.29 |
| ml2_25 | - | - | 497.3 | 1496 | 257.5 | **8.106** | 11.83 |
| ml2_26 | - | - | 339.2 | 1959 | 303.7 | **7.327** | 42.75 |
| ml2_27 | - | - | 834.4 | 14290 | 332.8 | **4.965** | 39.99 |
| ml2_28 | - | - | 799.3 | - | 1273 | **24.31** | 92.09 |
| ml2_29 | - | - | 1905 | - | 1363 | **15.8** | 70.04 |
| ml2_30 | - | - | 2552 | - | 794.5 | **45.74** | 113.8 |
| ml2_31 | - | - | 5976 | - | 894.0 | **27.43** | 48.51 |
| ml2_32 | - | - | - | - | 1281 | **29.85** | 394.9 |
| ml2_33 | - | - | - | - | 9315 | **39.37** | 460.7 |
| ml2_34 | - | - | - | - | - | **96.96** | 273.3 |
| ml2_35 | - | - | - | - | - | 99.8 | **69.91** |
| ml2_36 | - | - | - | - | - | 53.87 | **21.26** |
| ml2_37 | - | - | - | - | - | **110.2** | 866.7 |
| ml2_38 | - | - | - | - | - | **83.92** | 921.9 |
| ml2_39 | - | - | - | - | - | **124.6** | 688.3 |
| ml2_40 | - | - | - | - | - | **238.6** | 4244 |

Table 10: Computation times (s) for bioinformatics instances from the ‘ml’ class.

| Instance | SCIP-SDP | KT | CG1 | CG2 | SEC-simple | SEC | SEC-CG |
|---|---|---|---|---|---|---|---|
| ml2_3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| ml2_4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| ml2_5 | 11 | 1 | 1 | 1 | 1 | 1 | 1 |
| ml2_6 | 31 | 17 | 17 | 1 | 16 | 1 | 1 |
| ml2_7 | 79 | 68 | 68 | 66 | 28 | 1 | 1 |
| ml2_8 | 232 | 198 | 187 | 163 | 85 | 1 | 18 |
| ml2_9 | 505 | 408 | 464 | 360 | 106 | 1 | 1 |
| ml2_10 | 1209 | 902 | 1050 | 1260 | 122 | 3 | 297 |
| ml2_11 | 3828 | 3552 | 2775 | 2358 | 231 | 26 | 1 |
| ml2_12 | 13214 | 5766 | 4496 | 4108 | 525 | 42 | 1 |
| ml2_13 | 30886 | 3443 | 2201 | 5289 | 2047 | 49 | 3 |
| ml2_14 | 103844 | 11042 | 7749 | 7670 | 2794 | 85 | 1 |
| ml2_15 | 277563 | 6909 | 3672 | 3841 | 3306 | 67 | 67 |
| ml2_16 | - | 13789 | 16980 | 7213 | 2834 | 381 | 62 |
| ml2_17 | - | 13032 | 14854 | 13186 | 7307 | 40 | 32 |
| ml2_18 | - | 71319 | 15265 | 15073 | 8484 | 25 | 73 |
| ml2_19 | - | 152418 | 20636 | 18234 | 8754 | 46 | 1 |
| ml2_20 | - | 543672 | 25188 | 21432 | 9287 | 95 | 58 |
| ml2_21 | - | 870889 | 14851 | 19920 | 18928 | 125 | 1 |
| ml2_22 | - | 1555614 | 54545 | 89797 | 35269 | 490 | 53 |
| ml2_23 | - | 6063406 | 62828 | 85769 | 38643 | 394 | 224 |
| ml2_24 | - | 5173626 | 81607 | 119736 | 37759 | 133 | 442 |
| ml2_25 | - | - | 224467 | 123644 | 37777 | 259 | 77 |
| ml2_26 | - | - | 117689 | 123695 | 30987 | 71 | 170 |
| ml2_27 | - | - | 236341 | 445741 | 36693 | 42 | 130 |
| ml2_28 | - | - | 225144 | - | 276524 | 659 | 299 |
| ml2_29 | - | - | 355239 | - | 261368 | 220 | 221 |
| ml2_30 | - | - | 391804 | - | 49445 | 1266 | 237 |
| ml2_31 | - | - | 635442 | - | 44693 | 576 | 48 |
| ml2_32 | - | - | - | - | 87210 | 201 | 2380 |
| ml2_33 | - | - | - | - | 595119 | 612 | 405 |
| ml2_34 | - | - | - | - | - | 1054 | 194 |
| ml2_35 | - | - | - | - | - | 486 | 140 |
| ml2_36 | - | - | - | - | - | 981 | **7** |
| ml2_37 | - | - | - | - | - | 319 | 148 |
| ml2_38 | - | - | - | - | - | 123 | 176 |
| ml2_39 | - | - | - | - | - | 256 | 104 |
| ml2_40 | - | - | - | - | - | 446 | 1007 |

Table 11: Number of branching nodes for bioinformatics instances from the ‘ml’ class.

| Instance | | | | Average computation times (s) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Class** | ***n*** | **$(p,c)$** | **OPT** | **SCIP-SDP** | **KT** | **CG1** | **CG2** | **SEC-simple** | **SEC** | **SEC-CG** |
| 1 | 10 | (0.5,5) | 5.2 | 0.131 | 0.049 | 0.025 | 0.029 | 0.018 | 0.020 | 0.030 |
| | 10 | (0.5,10) | 5.8 | 0.135 | 0.034 | 0.038 | 0.042 | 0.029 | 0.020 | 0.031 |
| | 10 | (0.5,20) | 7.7 | 0.218 | 0.022 | 0.023 | 0.035 | 0.024 | 0.017 | 0.033 |
| | 10 | (1,5) | 2 | 2.135 | 0.177 | 0.160 | 0.210 | 0.163 | 0.152 | 0.236 |
| | 10 | (1,10) | 3.4 | 1.888 | 0.121 | 0.135 | 0.176 | 0.129 | 0.121 | 0.182 |
| | 10 | (1,20) | 4.5 | 0.857 | 0.100 | 0.086 | 0.108 | 0.092 | 0.065 | 0.095 |
| | | *Average* | *4.8* | *0.894* | *0.084* | *0.078* | *0.100* | *0.076* | *0.066* | *0.101* |
| | 15 | (0.5,5) | 4.1 | 3.187 | 0.211 | 0.202 | 0.386 | 0.185 | 0.185 | 0.280 |
| | 15 | (0.5,10) | 6.5 | 2.131 | 0.160 | 0.182 | 0.197 | 0.146 | 0.134 | 0.235 |
| | 15 | (0.5,20) | 8.5 | 1.450 | 0.102 | 0.097 | 0.171 | 0.095 | 0.098 | 0.153 |
| | 15 | (1,5) | 0.4 | 450.1 | 2.657 | 1.720 | 9.566 | 1.884 | 1.835 | 5.209 |
| | 15 | (1,10) | 2.9 | 204.4 | 1.065 | 1.064 | 2.697 | 0.887 | 0.788 | 2.243 |
| | 15 | (1,20) | 5.1 | 77.44 | 0.557 | 0.587 | 1.246 | 0.514 | 0.497 | 1.022 |
| | | *Average* | *4.583* | *123.1* | *0.792* | *0.642* | *2.377* | *0.619* | *0.589* | *1.524* |
| | 20 | (0.5,5) | 3.2 | 169.6 | 0.869 | 0.938 | 2.521 | 0.690 | 0.629 | 1.994 |
| | 20 | (0.5,10) | 6.1 | 19.67 | 0.543 | 0.541 | 1.111 | 0.424 | 0.473 | 0.957 |
| | 20 | (0.5,20) | 9.3 | 56.92 | 0.419 | 0.396 | 0.897 | 0.335 | 0.294 | 0.759 |
| | 20 | (1,5) | 0 | 1985 | 496.3 | 235.2 | 5464 | 458.0 | 102.1 | 9648 |
| | 20 | (1,10) | 2.143 | 6549 | 48.71 | 32.95 | 245.7 | 24.89 | 23.69 | 174.2 |
| | 20 | (1,20) | 4.8 | 3189 | 6.223 | 6.422 | 19.26 | 4.733 | 3.808 | 12.04 |
| | | *Average* | *4.257* | *1995* | *92.19* | *46.09* | *955.7* | *81.51* | *21.83* | *1639* |
| 2 | 10 | (0.5,5) | 16.1 | 0.185 | 0.036 | 0.038 | 0.061 | 0.028 | 0.031 | 0.043 |
| | 10 | (0.5,10) | 22 | 0.117 | 0.031 | 0.038 | 0.043 | 0.025 | 0.023 | 0.040 |
| | 10 | (0.5,20) | 30.11 | 0.252 | 0.040 | 0.042 | 0.044 | 0.034 | 0.033 | 0.048 |
| | 10 | (1,5) | 4.6 | 1.316 | 0.260 | 0.227 | 0.166 | 0.110 | 0.201 | 0.200 |
| | 10 | (1,10) | 8.4 | 0.833 | 0.111 | 0.106 | 0.151 | 0.108 | 0.117 | 0.136 |
| | 10 | (1,20) | 11.6 | 0.736 | 0.120 | 0.110 | 0.141 | 0.131 | 0.098 | 0.136 |
| | | *Average* | *15.46* | *0.573* | *0.100* | *0.094* | *0.101* | *0.073* | *0.084* | *0.101* |
| | 15 | (0.5,5) | 17.7 | 1.967 | 0.160 | 0.182 | 0.256 | 0.164 | 0.141 | 0.263 |
| | 15 | (0.5,10) | 23.3 | 2.476 | 0.198 | 0.150 | 0.241 | 0.183 | 0.169 | 0.259 |
| | 15 | (0.5,20) | 27.2 | 4.525 | 0.221 | 0.191 | 0.269 | 0.173 | 0.206 | 0.260 |
| | 15 | (1,5) | 2.1 | 660.9 | 2.440 | 3.639 | 10.80 | 2.430 | 2.051 | 7.077 |
| | 15 | (1,10) | 6.5 | 118.2 | 0.925 | 0.911 | 1.697 | 0.829 | 0.855 | 1.590 |
| | 15 | (1,20) | 11.7 | 53.32 | 0.723 | 0.753 | 1.427 | 0.621 | 0.664 | 1.203 |
| | | *Average* | *14.75* | *140.2* | *0.778* | *0.971* | *2.449* | *0.733* | *0.681* | *1.775* |
| | 20 | (0.5,5) | 8.3 | 72.89 | 0.661 | 0.646 | 1.377 | 0.593 | 0.641 | 1.580 |
| | 20 | (0.5,10) | 19.2 | 68.31 | 0.544 | 0.528 | 1.057 | 0.545 | 0.517 | 1.289 |
| | 20 | (0.5,20) | 26.8 | 34.82 | 0.521 | 0.582 | 1.304 | 0.517 | 0.571 | 1.186 |
| | 20 | (1,5) | 0 | 1313 | 103.2 | 38.98 | 3452 | 16.93 | 33.99 | 2472 |
| | 20 | (1,10) | 4.125 | 5659 | 23.31 | 17.48 | 95.82 | 13.71 | 12.45 | 68.51 |
| | 20 | (1,20) | 10.11 | 1094 | 5.451 | 6.179 | 13.59 | 5.029 | 4.203 | 9.496 |
| | | *Average* | *11.42* | *1374* | *22.28* | *10.73* | *594.2* | *6.223* | *8.730* | *425.7* |

Table 12: Computation times of the reload instances averaged over 10 generated instances for given values of $n, p$ and $c$.

| Instance | | | Number of branching nodes | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Class** | ***n*** | **$(p,c)$** | **SCIP-SDP** | **KT** | **CG1** | **CG2** | **SEC-simple** | **SEC** | **SEC-CG** |
| 1 | 10 | (0.5,5) | 19.7 | 1.1 | 1.7 | 1.4 | 0.7 | 0.7 | 0.8 |
| | 10 | (0.5,10) | 16.6 | 2.2 | 2.1 | 3.8 | 0.8 | 0.7 | 0.8 |
| | 10 | (0.5,20) | 43.4 | 4.3 | 3.9 | 0.7 | 0.7 | 1.8 | 2.6 |
| | 10 | (1,5) | 132 | 125.9 | 173.1 | 168.6 | 74.6 | 69.4 | 99.4 |
| | 10 | (1,10) | 115 | 79 | 82.4 | 72.3 | 43.2 | 32.5 | 43.4 |
| | 10 | (1,20) | 46.5 | 15.4 | 15.4 | 11.6 | 9.5 | 4.6 | 7.1 |
| | | *Average* | *62.2* | *37.98* | *46.433* | *43.06* | *21.58* | *18.28* | *25.68* |
| | 15 | (0.5,5) | 155.9 | 99.4 | 92.9 | 111 | 73.5 | 67 | 53.2 |
| | 15 | (0.5,10) | 91.8 | 70.9 | 105.1 | 76.9 | 45.7 | 28.4 | 46.8 |
| | 15 | (0.5,20) | 48.3 | 34.9 | 17.3 | 28.3 | 11.6 | 12.9 | 7.6 |
| | 15 | (1,5) | 3001 | 3606 | 2506 | 3607 | 2643 | 2215 | 2084 |
| | 15 | (1,10) | 1692 | 1223 | 1176 | 1293 | 921.2 | 533.8 | 618.6 |
| | 15 | (1,20) | 405.1 | 279.4 | 170.5 | 241.1 | 157.1 | 140.1 | 108.5 |
| | | *Average* | *899.3* | *885.8* | *678.1* | *893.2* | *642.0* | *499.5* | *486.5* |
| | 20 | (0.5,5) | 1929 | 1312.8 | 1522 | 1534 | 991 | 637.5 | 685.4 |
| | 20 | (0.5,10) | 293.6 | 336 | 369.1 | 266.7 | 159.9 | 148.5 | 152 |
| | 20 | (0.5,20) | 459.4 | 188.4 | 175.6 | 165.9 | 97.7 | 50.3 | 75.7 |
| | 20 | (1,5) | 13529 | 65622 | 54077 | 44724 | 125560 | 27878 | 67724 |
| | 20 | (1,10) | 18062 | 29287 | 21583 | 18798 | 15521 | 10308 | 12110 |
| | 20 | (1,20) | 4749 | 3114 | 3400 | 3312 | 2499 | 1525 | 1507 |
| | | *Average* | *6503* | *16643* | *13521* | *11467* | *24138* | *6758* | *13709* |
| 2 | 10 | (0.5,5) | 19 | 5.6 | 5.6 | 0.9 | 3.7 | 4.2 | 0.9 |
| | 10 | (0.5,10) | 7.778 | 2.667 | 1.5556 | 2.333 | 1.556 | 0.778 | 2.444 |
| | 10 | (0.5,20) | 29.56 | 3.889 | 3.556 | 3.333 | 3.222 | 3.888 | 2.667 |
| | 10 | (1,5) | 88.6 | 63.4 | 68.1 | 76.6 | 46.9 | 22.2 | 68.1 |
| | 10 | (1,10) | 35.6 | 27.2 | 35 | 35.4 | 18 | 17 | 16.7 |
| | 10 | (1,20) | 28.5 | 27.3 | 21.6 | 19.5 | 16.2 | 18.1 | 20.3 |
| | | *Average* | *34.84* | *21.68* | *22.59* | *23.01* | *14.93* | *11.03* | *18.52* |
| | 15 | (0.5,5) | 81.2 | 73.7 | 105.1 | 75.3 | 58.2 | 43.1 | 46 |
| | 15 | (0.5,10) | 93.5 | 67.6 | 50.5 | 81.6 | 63.3 | 49.1 | 65.4 |
| | 15 | (0.5,20) | 188.1 | 67.7 | 65.8 | 65.1 | 48.7 | 38.6 | 50.3 |
| | 15 | (1,5) | 5588 | 3306 | 5257 | 4394 | 4016 | 2734 | 2594 |
| | 15 | (1,10) | 715.2 | 1052 | 861.3 | 778.3 | 669 | 619.9 | 502.4 |
| | 15 | (1,20) | 317.8 | 442.3 | 393.3 | 451.3 | 349 | 307.7 | 315.8 |
| | | *Average* | *1164* | *834.9* | *1122* | *974.4* | *867.4* | *632.1* | *595.6* |
| | 20 | (0.5,5) | 664.2 | 689.6 | 596.8 | 469.2 | 335.9 | 335.2 | 426.1 |
| | 20 | (0.5,10) | 746.9 | 317 | 218.8 | 240.3 | 237.3 | 195.1 | 153.7 |
| | 20 | (0.5,20) | 372.8 | 374 | 474.4 | 320.6 | 258.9 | 340.2 | 328.6 |
| | 20 | (1,5) | 7213 | 23766 | 16086 | 44835 | 8717 | 9182 | 20919 |
| | 20 | (1,10) | 13725 | 15129 | 11001 | 10907 | 8124 | 6421 | 5870 |
| | 20 | (1,20) | 2091 | 2995 | 2789 | 3253 | 2564 | 1753 | 1667 |
| | | *Average* | *4136* | *7212* | *5195* | *10004* | *3373* | *3038* | *4894* |

Table 13: Number of branching nodes for the reload instances averaged over 10 generated instances for given values of $n, p$ and $c$.