Les Cahiers du GERAD ISSN: 0711–2440

Exact separation of k-projection polytope constraints

E. Adams
M.F. Anjos

G–2015–20

March 2015

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Exact separation of k-projection polytope constraints

Elspeth Adams [a]

Miguel F. Anjos [b]

[a] *GERAD & Polytechnique Montréal, Montréal (Québec) Canda, H3C 3A7*

[b] *Canada Research Chair in Discrete Nonlinear Optimization in Engineering, GERAD & Polytechnique Montréal, Montréal (Québec) Canda, H3C 3A7*

elspeth.adams@polymtl.ca
anjos@stanfordalumni.org

March 2015

Les Cahiers du GERAD

G–2015–20

Abstract: A critical step of any cutting plane algorithm is to find valid inequalities, or cuts, that improve the current relaxation of the integer-constrained problem. The maximally violated valid inequality problem aims to find the most violated inequality from a given family of cuts. k-projection polytope constraints are a family of cuts that are based on an inner description of the cut polytope of size k and are applied to $k \times k$ principal minors of a semidefinite optimization relaxation. We propose a bilevel second order cone optimization approach to solve the maximally violated k projection polytope constraint problem. We reformulate the bilevel problem as a single-level mixed binary second order cone optimization problem that can be solved using off-the-shelf conic optimization software. Additionally we present two methods for improving the computational performance of our approach, namely lexicographical ordering and a reformulation with fewer binary variables. All of the proposed formulations are exact. Preliminary results show the computational time and number of branch-and-bound nodes required to solve small instances in the context of the max-cut problem.

Acknowledgments: The second author acknowledges the support of this research by the Natural Sciences and Engineering Research Council of Canada (NSERC).

1 Introduction

Cutting planes are often used as an efficient means to tighten continuous relaxations of mixed-integer optimization problems and are a vital component of branch-and-cut algorithms. A critical step of any cutting plane method is solving the separation problem to find valid inequalities, or cuts, that are violated by the current solution but are satisfied by every feasible integer solution. The problem of finding a cut that achieves maximal violation over all possible cuts for a given solution to the relaxation is called the maximally violated valid inequality problem (MVVIP) [18].

This paper is concerned with finding the most violated k-projection polytope constraints (kPPC). Because kPPCs are not inequalities we refer to the problem as the maximally violated k-projection polytope constraint problem (MVkPPC). This class of constraints was introduced in [1] and is defined for NP-hard combinatorial problems based on graphs for which the projection of the problem onto a subgraph shares the same structure as the original problem. Examples of such problems include the well-known max-cut and stable-set problems. While originally introduced as a means to define a new hierarchy of semidefinite optimization relaxations for this type of NP-hard problem, these constraints can be used individually to tighten any semidefinite relaxation.

This paper explores the exact separation of maximally violated k-projection polytope constraints. Specifically we present a bilevel optimization model that fits into the MVVIP framework and finds the maximally violated k-projection polytope constraint. We also show how to reformulate the model as a single-level mixed integer second order cone optimization problem, and how the single-level model can be strengthened by reformulating it using a different set of binary variables and by breaking symmetry.

This paper is organized as follows. In Section 2 we provide a literature review. Section 3 introduces the bilevel model for finding maximally violated kPPCs . Section 4 presents the single-level model for separating a maximally violated kPPC and shows how the single-level problem is a reformulation of the bilevel model. We divide the reformulation into three steps for clarity (Subsections 4.1.1–4.1.3). In Section 5 we show how the single-level model can be strengthened by adding symmetry-breaking constraints (Subsection 5.1) and by changing the binary variables (Subsection 5.2). Finally Section 6 presents computational results comparing the different models with respect to both computational time and the size of the enumeration tree.

2 Literature review

Separation procedures (or constraint identification problems) are defined as follows: given a point x and a family of valid constraints $\mathcal{L}$, identify one or more constraints in $\mathcal{L}$ violated by x, or prove that no such constraint exists [23]. Note that although separation procedures are typically used to identify inequalities, the framework is identical for any set of valid constraints.

Separation procedures have been studied from both the practical and theoretical perspectives and are often discussed in the context of cuts which are used to tighten relaxations. Cuts that share a special structure can be categorized into a specific family or class. Applegate, Bixby, Chvt́al and Cook [2] called the paradigm of generating cuts from a given family the template paradigm. Different types of cuts include Chvátal cuts [8], Chvátal-Gomory [22], $\{0, \frac{1}{2}\}$-Chvátal-Gomory cuts [4], split cuts [9], MIR-inequalities [22] and lift-and-project cuts [3].

Practically, the separation of valid inequalities (i.e. cuts) is a key component of cutting plane algorithms. Cutting plane algorithms have been well studied and are fundamental in solving integer (and mixed integer) optimization problems. For early research see Dantzig, Fulkerson and Johnson [10], Gomory [14] and Grötschel, Lovász and Schrijver [16]. For more recent advances see [21, 20].

Different separation procedures are used for different families of cuts. Caprara and Letchford [7] examined the complexity of the separation procedure for various inequalities. They proved strong $\mathcal{NP}$-completeness for the separation of split cuts and strengthened $\mathcal{NP}$-completeness results for $\{0, \frac{1}{2}\}$-cuts (initially in [4]) and Chvàtal-Gomory cuts (initially in [12]). Matrix cuts and lift-and-project cuts have been shown to be separable in polynomial time [3, 19]. We discuss the MVVIP in Section 2.3.

This paper examines the MVkPPCP for the max-cut problem. The subsequent subsections examine the max-cut problem, kPPCs , maximally violated cuts and the concepts of validity and membership.

2.1 Max-cut problem

We use the following notation. Let $\mathcal{S}_n$ be the set of symmetric matrices of size n. Let $N := \{1, \ldots, n\}$ be the vertex set of a graph G. The max-cut problem is defined by an undirected graph G with n vertices and the weighted adjacency matrix A. It is assumed that the graph contains no loops. Feasible solutions to the max-cut problem are called 'cuts' and are defined by partitioning the n nodes into two parts. Let each of the 2^{n-1} cut matrices be denoted $C_i \in \mathcal{S}_n$ where $C_i = c_r c_r^T$ and $c_r \in \{-1, 1\}^n$. The term 'cut' in cut matrix refers to feasible max-cut solutions. The cut polytope, CUT_n, is the convex hull of all 2^{n-1} feasible solutions C_i.

The max-cut problem is

$$z_{\text{max-cut}} = \max\{\langle L, X\rangle : \ X \in \mathrm{CUT}_n\}$$

where L is the Laplacian associated to the weighted adjacency matrix A such that $L = \mathrm{Diag}(Ae) - A$ where e is the vector of all ones.

The set $\mathcal{C}$ of correlation matrices is

$$\mathcal{C} := \{X \in \mathcal{S}_n : \ \mathrm{diag}(X) = e, \ X \succeq 0\}$$

and the metric polytope $\mathcal{M}$ is the set of all symmetric matrices with diagonal equal one and satisfying the triangle inequalities,

$$\mathcal{M} := X \in \mathcal{S}_n : \ \mathrm{diag}(X) = e, \ x_{ij} + x_{ik} + x_{jk} \geq -1, x_{ij} - x_{ik} - x_{jk} \geq -1 \ \forall i, j, k\}$$

These two sets yield two popular semidefinite optimization relaxations of the max-cut problem:

$$z_{\mathcal{C}} := \max\{\langle L, X\rangle : \ X \in \mathcal{C}\}$$
$$z_{\mathcal{C}\cap\mathcal{M}} := \max\{\langle L, X\rangle : \ X \in \mathcal{C} \cap \mathcal{M}\}$$

Delorme and Poljak [11] introduced the relaxation optimizing over $\mathcal{C}$ and Helmberg, Rendl, Vanderbei and Wolkowicz [17] solved this relaxation with an interior point method. Fischer, Gruber, Rendl and Sotirov [13] presented a computational efficient way to solve the relaxation over $\mathcal{C} \cap \mathcal{M}$ and Rendl, Rinaldi and Wiegele [24] proposed an exact method that begins with the relaxation over $\mathcal{C} \cap \mathcal{M}$ and use branch-and-bound to solve the max-cut problem.

2.2 k-projection polytope constraints for max-cut

Recall that $N := \{1, \ldots, n\}$ is the vertex set of a graph. For $I \subseteq N$, X_I denotes the principal submatrix of X indexed by I. Then using the definitions of cut matrices and the cut polytope from the previous section we can denote the k-projection polytope constraint $X_I \in \mathrm{CUT}_k$ corresponding to the k-subset I as:

$$X_I = \sum_{i=1}^{2^{k-1}} \lambda_i^I C_i \text{ with } \lambda_i^I \geq 0, \ \sum_{i=1}^{2^{k-1}} \lambda_i^I = 1$$

where $\lambda^I \in \mathbb{R}^{2^{k-1}}$.

Projection polytope constraints can be used to tighten semidefinite relaxations. Specifically we will show how kPPCs can be used to tighten the max-cut relaxation. For additional details regarding how and when kPPCs can be used see Adams et al. [1]. Let $\widehat{\mathcal{I}}_k$ be the set of all k-index sets,

$$\widehat{\mathcal{I}}_k := \{I \subseteq V : \ |I| = k\}$$

It is possible to strengthen the relaxation over $\mathcal{C} \cap \mathcal{M}$ by including k-projection polytope constraints over some set $\mathcal{I}_k \subseteq \widehat{\mathcal{I}}_k$,

$$z_{\bar{\mathcal{I}}_k} := \max\{\langle L, X\rangle : \ X \in \mathcal{C} \cap \mathcal{M} \cap \mathcal{I}_k\} \text{ where } \mathcal{I}_k \subseteq \widehat{\mathcal{I}}_k$$

This paper addresses the question of how to find if there is a k-projection polytope constraint that is not satisfied by a given solution X^* and if so, how to find the most violated kPPC. This question is formulated as a separation problem.

2.3 Maximally violated inequalities

Optimization models have been proposed that look for maximally violated cuts. Caprara, Fischetti and Letchford [6] proposed a model that finds the mod-k cut that is maximally violated for a given point x^*. They also show that for any given k for which a prime factorization is known maximally violated mod-k cuts can be found efficiently in $O(mn \min\{m, n\})$ time.

Lodi, Ralphs and Woeginger [18] propose a mixed-integer bilevel model for a general separation problem which finds the maximally violated valid inequality. They emphasize the conceptual nature of this formulation as challenges exist to explicitly write a compact description of the inner problem in addition to the practical issues surrounding solving bilevel problems. However for certain examples (split cuts, generalized subtour elimination constraints (GSECs) for the capacitated vehicle routing problem) the bilevel model can be converted to a single-level linear optimization problem. Two key components in MVVIPs are validity and membership. We address them in turn.

The validity verification problem determines if all points in a polyhedron satisfy the constraint. Lodi et al. [18] formalize this concept for linear inequalities. For a given polyhedron $\mathcal{P} = \{x \in \mathcal{R}^n_+ \mid Ax \geq b\}$, (α, β) defines a valid inequality if and only if there exists $u \in \mathcal{R}^m_+$ such that $\alpha \geq u^T A$ and $\beta \leq u^T b$. The inequality $\alpha^T u \geq \beta$ is valid for the polyhedron $\mathcal{P}$ since all $x \in \mathcal{P}$ satisfy $\alpha^T x \geq \beta$.

The membership problem is a decision problem that asks whether a given point $\hat{X}$ is contained in a polyhedron $\mathcal{P}$ or the intersection of the polyhedron $\mathcal{P}$ and a given cut. This problem has been looked at in the context of many different families of cuts, for example Gomory-Chvátal cuts [12] and $\{0, \frac{1}{2}\}$ cuts [5].

3 Finding maximally violated k-projection polytope constraints

3.1 Validity

The kPPCs are always valid because they satisfy the projection property, therefore the validity verification problem is not addressed for kPPCs .

3.2 Membership

The membership problem for a kPPC is: for a given $k \times k$ submatrix $X_\mathcal{I}$ of X, where $|\mathcal{I}| = k$, is $X_\mathcal{I} \in \mathrm{CUT}_\mathcal{I}$? If $X_\mathcal{I} \notin \mathrm{CUT}_\mathcal{I}$ then adding the kPPC for set $\mathcal{I}$ will tighten the relaxation.

The following problem, denoted *distance-to-polytope (D2P)*, not only determines if the point X^* is a member of the polyhedron $\mathrm{CUT}_{|I|}$ but also quantifies the separation if X^* is not a member of $\mathrm{CUT}_{|I|}$:

$$d^* = \min \ \left\{||\mathrm{triu}\,(X^*_I) - Q\lambda|| \ : \ e^T\lambda = 1, \ \lambda \geq 0\right\} \tag{D2P}$$

where X^*_I is the principal submatrix indexed by I of X^*, e is the vector of all ones of the appropriate size, $\mathrm{triu}(X)$ is the vector formed from the elements in the strictly upper triangular part of matrix X taken column-wise and Q is a $\binom{|I|}{2} \times 2^{|I|-1}$ matrix with the column, Q_i, being the feasible cut solution in vector form, $\mathrm{triu}(C_i)$.

The optimal objective value d^* equals the Euclidean distance between X_I^* and $\mathrm{CUT}_{|I|}$. Therefore

$$\begin{aligned} &\text{If } d^* = 0 \text{ then } X_I^* \in \mathrm{CUT}_{|I|} \\ &\text{If } d^* > 0 \text{ then } X_I^* \notin \mathrm{CUT}_{|I|} \end{aligned}$$

To illustrate the definition we look at the following example. Grishukhin [15] showed that $\langle Q, X\rangle \geq 5$ defines a facet of CUT_7. X^* is the optimal solution when optimized over $\mathcal{C} \cap \mathcal{M}$. For $k = 5$ we can find the distance-to-polytope (d^*) for each $I \in V$ such that $|I| = 5$. The following table shows the results of adding projection polytope constraints to the initial relaxation. The 3rd column shows the bound when the k-projection polytope constraint associated with the single index I is added to $\mathcal{C} \cap \mathcal{M}$. The 4$^{\text{th}}$ and 5$^{\text{th}}$ column show the bound when multiple kPPCs are added.

Note that for any set of indices I where the distance-to-polytope is equal to (approximately) 0 adding the corresponding PPC does not change the optimal objective function. We observe that even if we add all the PPCs corresponding to the 14 sets of indices with $d^* \leq .0008$ the optimal objective function does not change since for each of these sets of indices $X_I^* \in \mathrm{CUT}_{|I|}$ and the optimal solution X^* is still feasible. Adding all $k = 5$ projection polytope constraints (i.e. optimizing over $\mathcal{C} \cap \mathcal{M} \cap \widehat{\mathcal{I}}_5$) improves the bound to 5.8000.

Table 1: Bounds for adding different $k = 5$ PPCs to the SDP relaxation $\mathcal{C} \cap \mathcal{M}$

<table>
<tr><th>d^*</th><th>I</th><th>bound</th><th>bound</th><th>bound</th></tr>
<tr><td rowspan="2">0.1274</td><td>[1 3 5 6 7]</td><td>5.9800</td><td rowspan="2">5.9000</td><td rowspan="8">5.8000</td></tr>
<tr><td>[2 4 5 6 7]</td><td>5.9800</td></tr>
<tr><td rowspan="4">0.0800</td><td>[1 2 3 5 6]</td><td>6.0371</td><td rowspan="4">5.9412</td></tr>
<tr><td>[1 2 4 5 7]</td><td>6.0371</td></tr>
<tr><td>[1 3 4 5 6]</td><td>6.0371</td></tr>
<tr><td>[2 3 4 5 7]</td><td>6.0371</td></tr>
<tr><td>0.0563</td><td>[1 2 3 4 5]</td><td>6.0485</td><td></td></tr>
<tr><td>$\leq$ 0.0008</td><td>the remaining 14 subsets</td><td></td><td>6.0584</td></tr>
</table>

3.3 Formulation of the MVkPPCP as a bilevel problem

The MVVIP can now be easily extended to find the maximally violated kPPC. The following is the MVkPPCP formulation:

$$\begin{aligned} (\mathrm{DP}_{\text{Bilevel}}) \qquad \max \quad & d \\ \text{s.t.} \quad & Be_n = e_k && (1) \\ & Be_k \leq e_n && (2) \\ & B \in \{0,1\}^{n\times k} && (3) \\ & d = \left\{ \min_{e^T\lambda=1,\ \lambda\geq 0} ||\mathrm{triu}\left(B^T X B\right) - Q\lambda|| \right\} && (4) \end{aligned}$$

where e_n is a $n \times 1$ vector of all ones; $\mathrm{triu}(X)$ is the vector formed from the elements in the strictly upper triangular part of matrix X taken column-wise. The inner problem solves the D2P problem for a certain k-projection polytope constraint. The variables of the outer problem define the parameters of the kPPC, namely the k indices that define the submatrix $X_{\mathcal{I}}$.

In $\mathrm{DP}_{\text{Bilevel}}$ the outer problem variable B is a $n \times k$ row selection matrix. Constraints (1)–(3) require that exactly k rows are selected, these rows correspond to the indices of a kPPC. The inner problem (4) is D2P with the principal submatrix X_I is written more generally as $B^T XB$. Since the outer variable B is assumed to be given in the inner problem (hence assumed to be a row selection matrix) this product selects the principal submatrix indexed by the rows selected in B. Namely,

$$X_I = B^T X B \text{ where (1)–(3) are satisfied and } \sum_{j=1}^{k} B_{ij} = \begin{cases} 1 & \text{if } i \in I \\ 0 & \text{if } i \notin I \end{cases}$$

4 Reformulation of the MVkPPCP as a single-level problem

In this section we reformulate $\mathrm{DP}_{\mathrm{Bilevel}}$ to a single-level mixed binary second order cone optimization problem. We begin by presenting the full single-level problem and in the remainder of this section we show the equivalence between the two problems.

$$
\begin{aligned}
(\mathrm{DP}_{\mathrm{single}}) \qquad & \max \qquad d \\
& \text{s.t. } (1)\text{–}(3) && && (5)\\
& e^T\lambda = 1 && && (6)\\
& \mu_{jt} + Q\lambda - \sum_{i=1\dots n}\ \sum_{s=1\dots n\,:\,s\neq i} X_{is}\beta_{ijst} = 0 && \forall 1 \le j < t \le k && (7)\\
& \lambda \ge 0 && && (8)\\
& \begin{bmatrix} d \\ \mu \end{bmatrix} \in \mathrm{SOC}^{1+\binom{k}{2}} && && (9)\\
& ye + Q^T z \le 0 && && (10)\\
& \begin{bmatrix} 1 \\ -z \end{bmatrix} \in \mathrm{SOC}^{1+\binom{k}{2}} && && (11)\\
& d - y - \sum_{ijst\in\mathcal{S}} X_{is}\gamma_{ijst} = 0 && && (12)\\
& \alpha \in \{0,1\}^{2^{k-1}} && && (13)\\
& \alpha_i - \lambda_i \ge 0 && \forall i = 1\dots 2^{k-1} && (14)\\
& \left(2^{\frac{k+1}{2}}\right)\alpha_i - y - Q_i^T z \le \left(2^{\frac{k+1}{2}}\right) && \forall i = 1\dots 2^{k-1} && (15)\\
& \beta_{ijst} - b_{ij} \le 0 && \forall\, ijst \in \mathcal{S} && (16)\\
& \beta_{ijst} - b_{st} \le 0 && \forall\, ijst \in \mathcal{S} && (17)\\
& \sum_{ijst\in\mathcal{S}} \beta_{ijst} = \binom{k}{2} && && (18)\\
& 0 \le \beta_{ijst} \le 1 && \forall\, ijst \in \mathcal{S} && (19)\\
& \gamma_{ijst} + \beta_{ijst} \ge 0 && \forall\, ijst \in \mathcal{S} && (20)\\
& \gamma_{ijst} - \beta_{ijst} \le 0 && \forall\, ijst \in \mathcal{S} && (21)\\
& \gamma_{ijst} - z_{jt} - \beta_{ijst} \ge -1 && \forall\, ijst \in \mathcal{S} && (22)\\
& \gamma_{ijst} - z_{jt} + \beta_{ijst} \le 1 && \forall\, ijst \in \mathcal{S} && (23)
\end{aligned}
$$

where $\mathcal{S} = \{ijst \mid i, s = 1, \dots, n, i \neq s, 1 \le j < t \le k\}$.

4.1 Reformulating steps

In this section we present the steps to transform $\mathrm{DP}_{\mathrm{Bilevel}}$ into $\mathrm{DP}_{\mathrm{single}}$. For clarity we split the reformulation into the following three steps: 1) replace the inner problem by its optimality conditions 2) rewrite the complementary slackness conditions and 3) linearize the nonlinear terms. We consider the steps in turn.

4.1.1 Step 1: Rewrite the inner problem

The first step in the reformulation is to transform the bilevel problem to a single-level problem. Recall the definition of second order cones (SOC):

$$
\begin{bmatrix} x_o \\ \bar{x} \end{bmatrix} \in \mathrm{SOC}^{1+n} \Leftrightarrow x_o \ge ||\bar{x}||
$$

where x_o is a scalar and $\bar{x}$ is a vector of length n.

Using this property we can reformulate the inner problem (4) to the following SOC problem:

$$
\begin{aligned}
(\mathrm{P}_{\mathrm{Inner}}) \qquad & \min_{d,\lambda,\mu} && d \\
& \text{s.t.} && (6),(8),(9) \\
& && \mu_{jt} + Q\lambda - \sum_{i=1}^{n-1}\sum_{s=i+1}^{n} X_{is}b_{ij}b_{st} = 0 \qquad \forall 1 \le j < t \le k
\end{aligned} \tag{24}
$$

where constraint (24) ensures that $\mu = \operatorname{triu}\left(B^T X B\right) - Q\lambda$, and the minimization of d implies $d = ||\mu||$ at optimality. Recall that b is given (and not a variable) in this formulation. The dual of $\mathrm{P}_{\mathrm{Inner}}$ is:

$$
\begin{aligned}
\max_{y,z} \quad & y + \sum_{ijst\in\mathcal{S}} X_{is}b_{ij}b_{st}z_{jt} \\
\text{s.t.} \quad & (10),(11)
\end{aligned}
$$

where $y \in \mathbb{R}$ and $z \in \mathbb{R}^{\binom{k}{2}}$ are variables.

The inner problem $\mathrm{P}_{\mathrm{Inner}}$ is convex (since it is a SOC problem) and both the objective function and constraints are differentiable. In addition the problem satisfies Slater's Condition: an example of a strictly feasible primal solution is given by μ, λ and d that satisfy the following:

$$
\lambda_i = \frac{1}{2^{k-1}}, \;\; \mu_{jt} = -Q\lambda + \sum_{i=1}^{n-1}\sum_{s=i+1}^{n} X_{is}b_{ij}b_{st} \; \forall 1 \le j < t \le k; \;\; d > ||\mu||
$$

and a strictly feasible dual solution is given by the following y and z:

$$
z_{jt} = 0 \, \forall 1 \le j < t \le k; \;\; y < 0
$$

Therefore the KKT conditions are both necessary and sufficient for the optimality of $\mathrm{P}_{\mathrm{Inner}}$, and we can replace the rewritten inner problem (4) with the primal feasibility constraints (6), (8), (9) and (24); the dual feasibility constraints (10) and (11) and the complementary slackness constraints:

$$
\lambda_i(y + Q_i^T z) = 0 \quad \forall i = 1, \ldots, \binom{k}{2} \tag{25}
$$

$$
d - \mu^T z = 0 \tag{26}
$$

The resulting problem is no longer bilevel but is exact (and not a relaxation).

4.1.2 Step 2: Rewrite complementary slackness conditions

The next step in the reformulation process is to substitute linear/binary constraints for the quadratic complementary slackness constraints (25) and (26). Constraint (25) implies $\lambda_i = 0$ or $y + Q_i^T z = 0$ for each i. Lemma (1) shows how this disjunction is modeled with the binary variables α.

Lemma 1 *There exists a feasible solution to constraints (8), (10), (13)–(15) if and only if there exists a feasible solution to (25).*

Proof.

($\Rightarrow$) Assume constraints (8), (10), (13)–(15) are satisfied. For each i (13) implies $\alpha_i = 0$ or 1. Consider these cases in turn.
If $\alpha_i = 0$ then (8) and (14) imply $\lambda_i = 0$.
If $\alpha_i = 1$ then (10) and (15) imply $y + Q_i^T z = 0$.

($\Leftarrow$) Assume there exists a feasible solution (λ, y, z) such that (25) is satisfied. Then either $\lambda_i = 0$ or $y + Q_i^T z = 0$. For all i such that $\lambda_i = 0$ set $\alpha_i = 0$ and for all i such that $y + Q_i^T z = 0$ set $\alpha_i = 1$. If $\lambda_i = 0$ and $y + Q_i^T z = 0$ then $\alpha_i = 0$ or 1 will satisfy the constraints. □

Moreover constraints (14) and (15) are constructed so that they do not introduce any additional restrictions on λ_i or $y + Q_i^T z$. If $\alpha_i = 0$ then (15) implies $y + Q_i^T z \geq -\left(2^{\frac{k+1}{2}}\right)$ which is unrestrictive and if $\alpha_i = 1$ then (14) implies $\lambda \leq 1$ which is also unrestrictive.

The following constraint is added to the model so that the quadratic constraint (26) is implicitly enforced.

$$d - y - \sum_{ijst \in \mathcal{S}} X_{is} b_{ij} b_{st} z_{jt} = 0 \tag{27}$$

Lemma 2 details how (27) plus constraints already in the model imply (26), however we begin by discussing the motivation for this constraint. Constraint (27) states strong duality of the inner problem, namely that the gap between the primal and dual objective values is 0. For strong duality to hold for a pair of conic optimization problems a constraint qualification must be satisfied (Slater's condition was verified in the previous step). It is important to note the difference between the observation that Slaters condition implies that strong duality holds and Lemma 2. Lemma 2 shows that the strong duality equation with primal and dual feasibility constraints together imply the complementary slackness constraint (26).

Lemma 2 *If a feasible solution exists for (8), (10), (13)–(15) and (24) then (26) is satisfied if and only if (27) is satisfied.*

Proof. Assume constraints (8), (10) (13)–(15) and (24) are satisfied. By Lemma 1 we know that $\lambda_i(y + Q_i^T z) = 0\ \forall i$. Summing over all i we get that $\lambda^T(ye + Q^T z) = 0$. We use this fact in the following equivalences:

$$\begin{aligned}
(27) \text{ is satisfied} &\Leftrightarrow d - ye^T\lambda - (\mu + Q\lambda)^T z = 0 \text{ (since (6) and (24) are satisfied)} \\
&\Leftrightarrow d - \mu^T z - \lambda^T(ye + Q^T z) = 0 \\
&\Leftrightarrow (26) \text{ is satisfied } \text{ (since } \lambda^T(ye + Q^T z) = 0)
\end{aligned}$$

□

4.1.3 Step 3: Linearization

The final step of the reformulation is to linearize $b_{ij}b_{st}$ in (24) with the variable β_{ijst} to get (7) and to linearize $b_{ij}b_{st}z_{jt}$ in (27) with the variable γ_{ijst} to get (12). We consider these in turn.

Recall that b_{ij} is defined $\forall i = 1, \ldots, n\ \forall j = 1, \ldots, k$. Constraints (1)–(3) imply that exactly k of the nk variables are equal to 1 and that the remaining are equal to 0. These constraints imply certain $b_{ij}b_{st}$ products will always be 0. Namely,

$$\begin{aligned}
(1) &\Rightarrow b_{ij}b_{it} = 0 && \forall i = 1, \ldots, n\ \forall j, t = 1, \ldots, k \\
(2) &\Rightarrow b_{ij}b_{sj} = 0 && \forall i, s = 1, \ldots, n\ \forall j = 1, \ldots, k
\end{aligned}$$

Therefore there is no need to linearize the terms in which $i = s$ or $j = t$. Since $b_{ij}b_{st} = b_{st}b_{ij}$ we can further limit the number of products we linearize to only those with $j < t$. Note that of the $(nk)^2$ products only $n(n-1)\binom{k}{2}$ of them need to be linearized. The indices of the terms that are linearized are denoted by $\mathcal{S}$ where

$$\mathcal{S} = \{ijst \mid i, s = 1, \ldots, n,\ i \neq s,\ 1 \leq j < t \leq t\}$$

Furthermore since exactly k elements of b are 1 then exactly $\binom{k}{2}$ of the products equal 1 with the remaining products equal to 0. Lemma 3 formalizes this idea and show the constraints necessary to enforce it.

Lemma 3 *If constraints (1)–(3) are satisfied then there exists a feasible solution to constraints (16)–(19) if and only if $b_{ij}b_{st} = \beta_{ijst}\ \forall ijst \in \mathcal{S}$.*

Proof. Assume constraints (1)–(3) are satisfied and consider the cases in turn.

($\Rightarrow$) Let $(\hat{b}, \hat{\beta})$ be any feasible solution to constraints (16)–(19). Constraint (3) implies $\hat{b}_{ij}, \hat{b}_{st} \in \{0,1\}$. Consider the cases in turn.

If $\hat{b}_{ij} = 0$ (resp. $\hat{b}_{st} = 0$) then (16) (resp. (17)) and (19) imply $\hat{\beta}_{ijst} = 0$.

If $\hat{b}_{ij} = \hat{b}_{st} = 1$ then constraints (1)–(3) imply that exactly k of the nk elements in $\hat{B}$ will be equal to 1 (with the rest equal to 0). Therefore all but $\binom{k}{2}$ of the terms in $\sum_{ijst \in \mathcal{S}} \hat{\beta}_{ijst}$ will be forced to 0 because $\hat{b}_{ij}$ or $\hat{b}_{st}$ equals 0. Since $\hat{\beta}_{ijst} \leq 1 \, \forall ijst \in \mathcal{S}$ and the sum of the nonzero elements of $\hat{\beta}$ equals $\binom{k}{2}$ then the remaining $\hat{\beta}$'s are forced to 1.

Therefore $\hat{\beta}_{ijst} = \hat{b}_{ij}\hat{b}_{st} \, \forall ijst \in \mathcal{S}$ as required.

($\Leftarrow$) Let $b_{ij}b_{st} = \beta_{ijst} \, \forall ijst \in \mathcal{S}$. Constraint (3) implies $b_{ij}, b_{st} \in \{0,1\}$. Therefore (19) is feasible since $\beta_{ijst} \in \{0,1\}$.

Constraint (16) implies $b_{ij}b_{st} - b_{ij} = b_{ij}(b_{st} - 1) \leq 0 \, \forall b_{ij}, b_{st} \in \{0,1\}$. Constraint (17) follows similarly.

Finally constraint (1) implies that if $b_{ij} = 1$ then $b_{it} = 0 \, \forall t \neq j$ and (2) implies that there exists exactly one i for each $1 \leq j \leq k$ such that $b_{ij} = 1$ and that if $b_{ij} = 1$ then $b_{sj} = 0 \, \forall s \neq i$. Therefore there exists exactly $\binom{k}{2}$ β's equal to 1 and (18) is feasible. $\square$

Note that although β_{ijst} is binary this does not need to be included as an explicit constraint in $\mathrm{DP}_{\mathrm{single}}$.

The final step is to linearize $b_{ij}b_{st}z_{jt}$ in constraint (27) using the variable γ_{ijst}. The linearization happens over the same set $\mathcal{S}$. Lemma (4) and its proof provide the details.

Lemma 4 *If constraints (1)–(3) are satisfied then there exists a feasible solution to (20)–(23) if and only if* $b_{ij}b_{st}z_{jt} = \gamma_{ijst} \, \forall ijst \in \mathcal{S}$.

Proof. Assume constraints (1)–(3) are satisfied. Consider the directions in turn.

($\Rightarrow$) Let there exist a feasible solution to (20)–(23). Constraint (3) implies $b_{ij}, b_{st} \in \{0,1\}$. Consider the cases in turn.

If $b_{ij} = 0$ or $b_{st} = 0$ then $\beta_{ijst} = 0$ (by Lemma 3). Constraints (20) and (21) then imply $\gamma_{ijst} = 0 \, \forall ijst \in \mathcal{S}$.

If $b_{ij} = b_{st} = 1$ then $\beta_{ijst} = 1$ (by Lemma 3). Constraints (22) and (23) then imply $\gamma_{ijst} - z_{jt} = 0 \, \forall ijst \in \mathcal{S}$.

Therefore in both cases $\gamma_{ijst} = b_{ij}b_{st}z_{jt} \, \forall ijst \in \mathcal{S}$ as required.

($\Leftarrow$) Constraints (20)–(23) follow immediately from the fact that $\gamma_{ijst} = b_{ij}b_{st}z_{jt}$, $\beta_{ijst} = b_{ij}b_{st}$ and $b_{ij}, b_{st} \in \{0,1\}$. $\square$

4.2 Equivalence of the bilevel and single-level models

Recall that the purpose of this section was show that the bilevel model $\mathrm{DP}_{\mathrm{Bilevel}}$ can be reformulated to a single-level problem. The following theorem combines the steps and lemmas presented previously to formally prove the equivalence of the $\mathrm{DP}_{\mathrm{Bilevel}}$ and $\mathrm{DP}_{\mathrm{single}}$.

Theorem 1 *$DP_{Bilevel}$ is equivalent to DP_{single}.*

Proof. Since the objective function in both models is the same it is sufficient to show that the feasible regions of $\mathrm{DP}_{\mathrm{Bilevel}}$ and $\mathrm{DP}_{\mathrm{single}}$ are equivalent. Namely that (d, b) is a feasible solution to $\mathrm{DP}_{\mathrm{Bilevel}}$ if and only if there exists a $(\mu, \beta, \gamma, \alpha, y, z, \lambda)$ so that (d, b) is a feasible solution to $\mathrm{DP}_{\mathrm{single}}$.

By the KKT theorem and convexity (4) can be replaced by the KKT conditions and the feasible region is the same. Therefore

$$
\begin{aligned}
(4) &\Leftrightarrow (6), (8)\text{–}(11), (24)\text{–}(26).\\
(25) &\Leftrightarrow (8), (10), (13)\text{–}(15)\\
(26) &\Leftrightarrow (27)\\
(24) \text{ and } \beta_{ijst} = b_{ij}b_{st} &\Leftrightarrow (7), (16)\text{–}(19)\\
(27) \text{ and } \gamma_{ijst} = b_{ij}b_{st}z_{jt} &\Leftrightarrow (12), (20)\text{–}(23)\\
\therefore (d, b) \text{is feasible for } (1)\text{–}(4) &\Leftrightarrow \text{there exists a } (\mu, \beta, \gamma, \alpha, y, z, \lambda) \text{ so that}\\
&\quad\ (d, b) \text{ is a feasible solution to}\\
&\quad\ (1)\text{–}(3), (6)\text{–}(23)
\end{aligned}
$$

□

5 Strengthening the single-level model

5.1 Symmetry

Symmetry exists within the exact separation problem because a subset of k indices will induce the same projection polytope regardless on the order. To eliminate this symmetry we can enforce lexicographical order on b with the following set of constraints:

$$b_{s,j-1} + \sum_{i=1}^{s} b_{ij} \leq 1 \ \forall s = 2, \ldots, n, j = 2, \ldots, k. \tag{28}$$

Lemma 5 *If constraints (1)–(3) and (28) are satisfied then lexicographical order must hold.*

Proof. Assume constraints (1)–(3) and (28) are satisfied. Variable b is a row selection matrix in which each column contains exactly 1 element equal to 1, with the rest equal to 0 (constraints (1)–(3)). Going through the columns in order we will show that the index of the row selected can only strictly increase.

Considering $b_{i1} \forall i$ (i.e. column 1 of b) we know there exists an i' such that $b_{i'1} = 1$ and $b_{i1} = 0 \ \forall i \in \{1, \ldots, n\} \setminus \{i'\}$. Therefore (28) implies $b_{i2} = 0 \ \forall i \leq i'$ and since each column sums to 1 (and each row sums to at most 1) then there exists $i'' > i'$ such that $b_{i''2} = 1$. By a similar argument $b_{i3} = 0 \ \forall i \leq i''$ and there exists $i''' > i''$ such that $b_{i'''3} = 0$. Repeating the argument k times implies that if $b_{i'1}, b_{i''2}, \ldots, b_{\bar{i},k}$ are the k elements of b equal to 1 then $i' < i'' < \cdots < \bar{i}$. □

5.2 Reformulation with fewer binary variables

We reduce the number of binary variables from $2^{k-1} + nk$ to $2^{k-1} + n$ by adding constraints (29)–(33). The proof later in this section shows that the feasible region of the model does not change and moreover that we no longer need to require binarity of the variables b_{ij}. The benefit of these constraints is seen in the computational results which are examined in Section 6.

$$\sum_{i=1}^{n} a_i = k \tag{29}$$

$$a_i - \sum_{j=1}^{k} b_{ij} = 0 \tag{30}$$

$$a_i - \sum_{i'=1}^{i=1} a_{i'} - b_{i1} \leq 0 \qquad \forall i = 1 \ldots n \tag{31}$$

$$a_i + \sum_{i'=1}^{i-1} b_{i',j-1} - \sum_{i'=1}^{i-1} b_{i',j} - b_{ij} \leq 1 \qquad \forall i = 2 \ldots n, \forall j = 2 \ldots k \tag{32}$$

$$a_i \in \{0, 1\} \qquad \forall i = 1 \ldots n \tag{33}$$

These constraints along with the symmetry constraints (28) are included in the $\mathrm{DP}_{\mathrm{single}}$ model. Constraint (3) (binarity of b) is removed as it is automatically enforced by constraints (29)–(33). The model $\mathrm{DP}_{\mathrm{fewerBinary}}$ is defined as follows:

$$(\mathrm{DP}_{\mathrm{fewerBinary}}) \qquad \begin{array}{ll} \max & d \\ \text{s.t.} & (1), (2), (6)\text{–}(23), (28)\text{–}(33) \end{array}$$

Lemma 6 certifies that the given set of constraints (including $a \in \{0,1\}^n$) implies that $b \in \{0,1\}^{n\times k}$.

Lemma 6 *If constraints (1), (2), (29)–(33) and $0 \le b_{ij} \le 1\ \forall i = 1\dots n, j = 1\dots k$ are satisfied then* $b_{ij} \in \{0,1\}$.

Proof. Let (1), (2), (29)–(33) and $0 \le b_{ij} \le 1\ \forall i = 1\dots n, j = 1\dots k$ be satisfied.
Constraints (29) and (33) imply there exists exactly k a_i's equal to 1 with the remaining $n-k$ a_i's equal to 0. Let $a_i = 1$ for $i \in \mathcal{A} := \{i_1 < i_2 < \cdots < i_k\}$
For all $i = 1\dots n$, $\sum_{i'=1}^{i-1} a_{i'} \in \{0,1,2,\dots,k\}$ and $a_i \in \{0,1\}$ therefore constraint (31) is unrestrictive (since $b_{ij} \ge 0$ is already enforced) unless $a_i = 1$ and $\sum_{i'=1}^{i-1} a_{i'} = 0$. This is only the case for $i = i_1$. Therefore $b_{i_1,1} = 1$.

$$\text{For } j=2, \sum_{i'=1}^{i-1} b_{i'1} = \begin{cases} 0 & \text{if } i \le i_1 \\ 1 & \text{if } i > i_1 \end{cases} \quad (\text{since } b_{i_1,1} = 1 \text{ and } b_{i,1} = 0\ \forall i \ne i_1)$$

$$a_i = \begin{cases} 1 & \text{if } i \in \mathcal{S} \\ 0 & \text{if } i \notin \mathcal{S} \end{cases} \quad \text{and} \quad \sum_{i'=1}^{i-1} b_{i'2} = \begin{cases} 0 & \text{if } i \le i_2 \\ 1 & \text{otherwise} \end{cases} \quad (\text{since } b_{i2} = 0\ \forall i \le i_1)$$

Combining the above in (32) we get:

$$b_{i2} \ge a_i + \sum_{i'=1}^{i-1} b_{i',j-1} - \sum_{i'=1}^{i-1} b_{i',j} - 1 = \begin{cases} 1+0-0-1=0 & \text{if } i \in \mathcal{A}, i \le i_1 \\ 1+1-H-1=1-H & \text{if } i \in \mathcal{A}, i > i_2 \\ 1+1-0-1=1 & \text{if } i \in \mathcal{A}, i > i_1, i \le i_2 \\ 0+0-0-1=-1 & \text{if } i \notin \mathcal{A}, i \le i_1 \\ 0+1-1-1=-1 & \text{if } i \notin \mathcal{A}, i > i_2 \\ 0+1-0-1=0 & \text{if } i \notin \mathcal{A}, i > i_1, i \le i_2 \end{cases}$$

Therefore case 3 implies $b_{i_2,2} = 1$ (since $i \in \mathcal{S}, i_1 < i \le i_2 \Rightarrow i = i_2$). If $b_{i_2,2} = 1$ then $H = 1\ \forall i > i_2$ and all cases (except 3) do not restrict b_{ij}.
Repeating this process for $j = 3\dots k$ implies $b_{i_1,1} = b_{i_2,2} = \cdots = b_{i_k,k} = 1$ and all other $b_{ij} = 0$. □

6 Computational performance of the formulations

In Section 4 we presented the exact $\mathrm{DP}_{\mathrm{single}}$ model and in Section 5 we showed how to strengthen the model with symmetry breaking constraints and by changing the binary variable. This section presents computational results comparing the three formulations: $\mathrm{DP}_{\mathrm{single}}$; $\mathrm{DP}_{\mathrm{single}}$ + (28); and $\mathrm{DP}_{\mathrm{fewerBinary}}$. The purpose of these results is to see how the different approaches compare with respect to computational time.

The data comes from max-cut problems from the Biq Mac Library [25]. Specifically the g05_80_i and pm1d_100_i examples, each with 10 instances. All models were formulated in MATLAB and solved with MOSEK.

The original solution X^* is the solution of solving the full size n problem over $\mathcal{C} \cap \mathcal{M}$. Computationally finding the deepest cut over a problem of this size is not yet possible. Therefore we consider a principal minors of size $\bar{n}$ and find the deepest cut of size k for this smaller problem. Results are shown for varying $\bar{n}$ and varying k in order to see how the models perform.

6.1 Comparison of the single-level models

We begin by looking at the specific instance pm1d_100_i0 and comparing the three different formulations for different problem sizes (i.e. different values of $\bar{n}$ and k). Figure 1 shows the computational time (in seconds) vs the problem size ($\bar{n}$) for $k = 6$. Note that we only consider $k < \bar{n}$ and only run formulations $\mathrm{DP}_{\mathrm{single}}$, $\mathrm{DP}_{\mathrm{single}}$ + (28) on smaller examples. The results for $k = 5, \ldots, 9$ are similar. We observe that $\mathrm{DP}_{\mathrm{fewerBinary}}$ is faster and able to solve larger problems. This observation is seen in all instances tested. The remaining results in this section focus on the $\mathrm{DP}_{\mathrm{fewerBinary}}$ formulation.

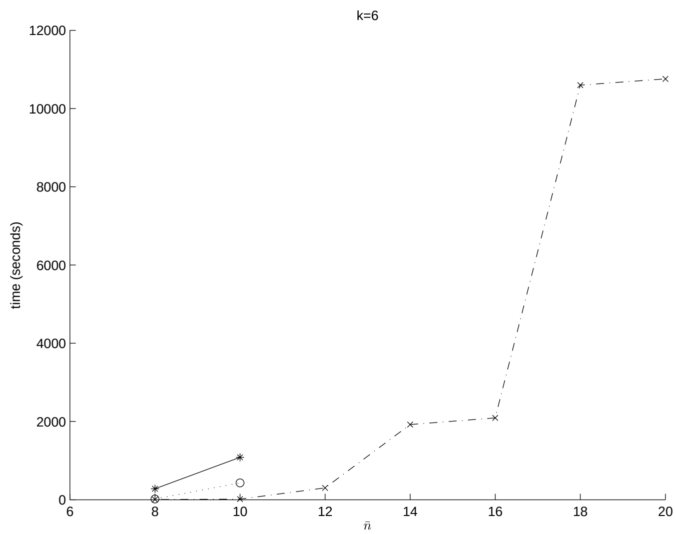

Figure 1: Comparison of computational time for formulations $\mathrm{DP}_{\mathrm{single}}$, $\mathrm{DP}_{\mathrm{single}}$ + (28) and $\mathrm{DP}_{\mathrm{fewerBinary}}$ for max-cut instance pm1d_100_i0 with $k = 6$

6.2 Performance of $\mathrm{DP}_{\mathrm{fewerBinary}}$ formulation

Figure 2 shows the computational time to solve $\mathrm{DP}_{\mathrm{fewerBinary}}$ vs the problem size $\bar{n}$ for $k = 7$. The results are similar for all $k = 5, \ldots, 9$ and for the g05_80 instances. We observe that computational time increases as the size of the problem increases. This is as expected since the size of the problem (both in terms of variables and constraints) increases as n increases.

Figure 3 shows the computational time for the 10 instances of g05_80. This plot shows the instances for $k = 5, \ldots, 9$ and $n = 12$. The key observation is that computational time does not increase as k increases. This is most clearly seen in instances 0, 2, 4 and 6 where a different value of k gives the case which took the most ($k = 7, 8, 9$ and 6, respectively) and least ($k = 5, 9, 6$ & 7 (tied) and 8 respectively) time. The lack of pattern between computational time and k is similar for all $\bar{n}$ tested and both data sets.

7 Conclusion

We have presented an exact formulation to the separation problem of finding the maximally violated valid inequality for a certain class of cuts, namely k-projection polytope constraints. We propose a bilevel second order cone optimization approach and reformulation it into a single-level mixed binary second order cone optimization problem. Furthermore we improve the computational performance by including lexicographical ordering and reducing the number of binary variables. All three formulations are exact. The models were solved with MOSEK for small examples ($n \leq 20, k = 5, \ldots, 9$). We observe that $\mathrm{DP}_{\mathrm{fewerBinary}}$ performed better than $\mathrm{DP}_{\mathrm{single}}$ + (28) which performed better than $\mathrm{DP}_{\mathrm{single}}$, however $\mathrm{DP}_{\mathrm{single}}$ + (28) has more constraints than $\mathrm{DP}_{\mathrm{single}}$ and $\mathrm{DP}_{\mathrm{fewerBinary}}$ has more constraints and more variables than the other models. The

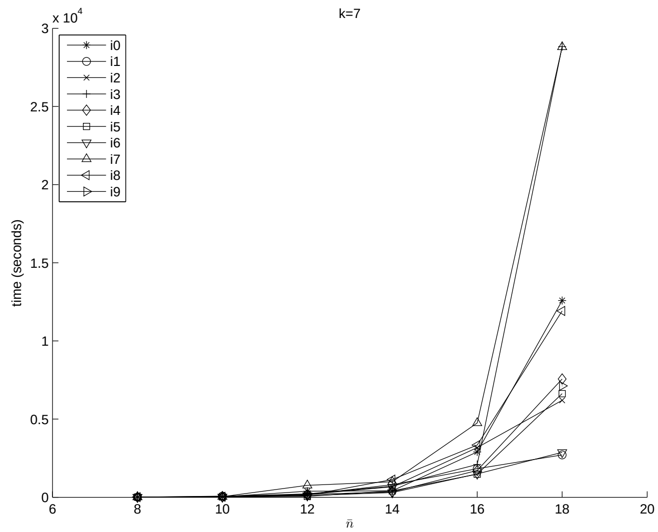

Figure 2: Computation time to solve $\mathrm{DP}_{\mathrm{fewerBinary}}$ for instances pm1d_100_i0,...,_i9 for $k = 7$

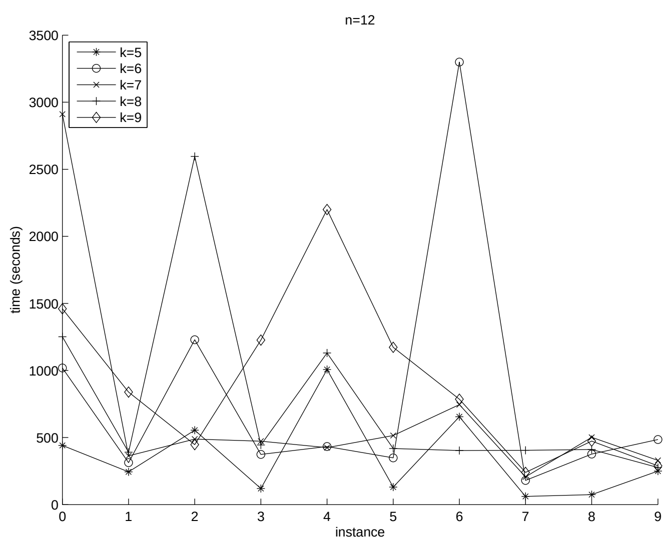

Figure 3: Computational time to solve $\mathrm{DP}_{\mathrm{single}}$ for max-cut instance g05_80 with $n = 12$

improvement in computational performance is a result of having to solve fewer nodes in the branch-and-bound algorithm. Specifically constraint (28) eliminates symmetrical feasible solutions, and the constraints and additional variables in $\mathrm{DP}_{\mathrm{fewerBinary}}$ reduces the number of binary variables.

References

[1] Elspeth Adams, Miguel F Anjos, Franz Rendl, and Angelika Wiegele. A hierarchy of subgraph projection-based semidefnite relaxations for some NP-hard graph optimization problems. (submitted) Journal of Global Optimization, 2014.

[2] David Applegate, Robert Bixby, Vašek Chvàtal, and William Cook. The traveling salesman problem: A computational study, 2006.

[3] Egon Balas, Sebastián Ceria, and Gérard Cornuéjols. A lift-and-project cutting plane algorithm for mixed 0–1 programs. Mathematical programming, 58(1–3):295–324, 1993.

[4] Alberto Caprara and Matteo Fischetti. {0, 1/2}-chvátal-gomory cuts. Mathematical Programming, 74(3):221–235, 1996.

[5] Alberto Caprara and Matteo Fischetti. Branch-and-cut algorithms. Annotated bibliographies in combinatorial optimization, 45–64, 1997.

[6] Alberto Caprara, Matteo Fischetti, and Adam N Letchford. On the separation of maximally violated mod-k cuts. Mathematical Programming, 87(1):37–56, 2000.

[7] Alberto Caprara and Adam N Letchford. On the separation of split cuts and related inequalities. Mathematical Programming, 94(2–3):279–294, 2003.

[8] Vašek Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. Discrete mathematics, 4(4):305–337, 1973.

[9] William Cook, Ravindran Kannan, and Alexander Schrijver. Chvátal closures for mixed integer programming problems. Mathematical Programming, 47(1–3):155–174, 1990.

[10] George Dantzig, Ray Fulkerson, and Selmer Johnson. Solution of a large-scale traveling-salesman problem. Journal of the operations research society of America, 2(4):393–410, 1954.

[11] Charles Delorme and Svatopluk Poljak. Laplacian eigenvalues and the maximum cut problem. Math. Programming, 62(3, Ser. A):557–574, 1993.

[12] Friedrich Eisenbrand. Note–on the membership problem for the elementary closure of a polyhedron. Combinatorica, 19(2):297–300, 1999.

[13] Ilse Fischer, Gerald Gruber, Franz Rendl, and Renata Sotirov. Computational experience with a bundle approach for semidefinite cutting plane relaxations of Max-Cut and equipartition. Math. Programming, 105(2–3, Ser. B):451–469, 2006.

[14] Ralph E Gomory. An algorithm for integer solutions to linear programs. Recent advances in mathematical programming, 64:260–302, 1963.

[15] Vyacheslav P Grishukhin. All facets of the cut cone $\mathbf{C}_n$ for $n = 7$ are know. European Journal of Combinatorics, 11(2):115–117, 1990.

[16] Martin Grötschel, László Lovász, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. Combinatorica, 1(2):169–197, 1981.

[17] Christoph Helmberg, Franz Rendl, Robert J. Vanderbei, and Henry Wolkowicz. An interior-point method for semidefinite programming. SIAM J. Optim., 6(2):342–361, 1996.

[18] Andrea Lodi, Ted K Ralphs, and Gerhard J Woeginger. Bilevel programming and the separation problem. Mathematical Programming, 1–22, 2012.

[19] László Lovász and Alexander Schrijver. Cones of matrices and set-functions and 0-1 optimization. SIAM Journal on Optimization, 1(2):166–190, 1991.

[20] Hugues Marchand, Alexander Martin, Robert Weismantel, and Laurence Wolsey. Cutting planes in integer and mixed integer programming. Discrete Applied Mathematics, 123(1):397–446, 2002.

[21] John E. Mitchell. Branch-and-cut algorithms for combinatorial optimization problems. Handbook of Applied Optimization, 65–77, 2002.

[22] George L Nemhauser and Laurence A Wolsey. *Integer and Combinatorial Optimization*, volume 18. Wiley New York, 1988.

[23] Manfred Padberg and Giovanni Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. SIAM review, 33(1):60–100, 1991.

[24] Franz Rendl, Giovanni Rinaldi, and Angelika Wiegele. Solving max-cut to optimality by intersecting semidefinite and polyhedral relaxations. Mathematical Programming, 121(2):307–335, 2010.

[25] Angelika Wiegele. Biq Mac Library. http://biqmac.uni-klu.ac.at/biqmaclib.