MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

On the Minimum Chordal Completion Polytope

Bergman, D.; Cardonha, C.; Cire, A.; Raghunathan, A.U.

TR2018-095 July 13, 2018

Abstract

A graph is chordal if every cycle of length at least four contains a chord, that is, an edge connecting two nonconsecutive vertices of the cycle. Several classical applications in sparse linear systems, database management, computer vision, and semidefinite programming can be reduced to finding the minimum number of edges to add to a graph so that it becomes chordal, known as the minimum chordal completion problem (MCCP). We propose a new formulation for the MCCP that does not rely on finding perfect elimination orderings of the graph, as has been considered in previous work. We introduce several families of facet-defining inequalities for cycle subgraphs and investigate the underlying separation problems, showing that some key inequalities are NP-Hard to separate. We also identify conditions through which facets and inequalities associated with the polytope of a certain graph can be adapted in order to become facet defining for some of its subgraphs or supergraphs. Numerical studies combining heuristic separation methods and lazy-constraint generation indicate that our approach substantially outperforms existing methods for the MCCP.

Operations Research

On the Minimum Chordal Completion Polytope

David Bergman
Operations and Information Management, University of Connecticut, Storrs, Connecticut 06260
david.bergman@business.uconn.edu

Carlos H. Cardonha
IBM Research, Brazil, São Paulo 04007-900
carloscardonha@br.ibm.com

Andre A. Cire
Department of Management, University of Toronto Scarborough, Toronto, Ontario M1C-1A4, Canada,
acire@utsc.utoronto.ca

Arvind U. Raghunathan
Mitsubishi Electric Research Labs, 201 Broadway, Cambridge,
raghunathan@merl.com

A graph is chordal if every cycle of length at least four contains a *chord*, that is, an edge connecting two nonconsecutive vertices of the cycle. Several classical applications in sparse linear systems, database management, computer vision, and semidefinite programming can be reduced to finding the minimum number of edges to add to a graph so that it becomes chordal, known as the *minimum chordal completion problem* (MCCP). We propose a new formulation for the MCCP that does not rely on finding perfect elimination orderings of the graph, as has been considered in previous work. We introduce several families of facet-defining inequalities for cycle subgraphs and investigate the underlying separation problems, showing that some key inequalities are NP-Hard to separate. We also identify conditions through which facets and inequalities associated with the polytope of a certain graph can be adapted in order to become facet defining for some of its subgraphs or supergraphs. Numerical studies combining heuristic separation methods and lazy-constraint generation indicate that our approach substantially outperforms existing methods for the MCCP.

Key words: Networks/graphs; Applications: Networks/graphs ; Programming: Integer: Algorithms: Cutting plane/facet

History:

1. Introduction

Given a simple graph $G = (V, E)$, the *minimum chordal completion problem* (MCCP) asks for the minimum number of edges to add to E so that the graph becomes *chordal*; that is, every cycle of length at least four in G has an edge connecting two non-consecutive vertices (i.e., a *chord*). Figure

1(b) depicts an example of a minimum chordal completion of the graph in Figure 1(a), where a chord $\{v_1, v_3\}$ is added because of the chordless cycle (v_0, v_1, v_4, v_3). The problem is also referred to as the *minimum triangulation problem* or the *minimum fill-in problem*.

The MCCP is a classical combinatorial optimization problem with a variety of applications spanning both the operations research and the computer science literature. Initially motivated by problems arising in Gaussian elimination of sparse linear equality systems (Parter 1961), chordalization methods have established an active research field, with applications in database management (Beeri et al. 1983, Tarjan and Yannakakis 1984), sparse matrix computation (Grone et al. 1984, Fomin et al. 2013), artificial intelligence (Lauritzen and Spiegelhalter 1990), computer vision (Chung and Mumford 1994), and in several other contexts (Heggernes 2006). Most recently, heuristic solution methods for the MCCP have gained a central role in semidefinite and nonlinear optimization, in particular for exploiting sparsity of nonlinear constraint matrices (Nakata et al. 2003, Kim et al. 2011, Vandenberghe and Andersen 2015).

The literature on exact computational approaches for the MCCP is, however, surprisingly scarce. To the best of our knowledge, the first mathematical programming model for the MCCP is derived from a simple modification of the formulation by Feremans et al. (2002) for determining the *tree-widths* of graphs. That model is based on a result by Fulkerson and Gross (1965), stating that a graph is chordal if and only it has a *perfect elimination ordering*, which is an ordering of vertices such that any vertex v forms a clique with its succeeding neighbours in the ordering. Yüceoğlu (2015)

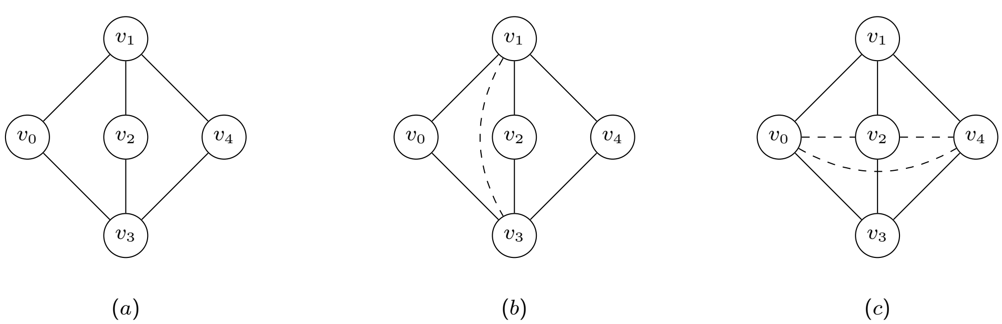


Figure 1 (a) An example graph by Heggernes (2006). (b) A minimum chordal completion of the graph. (c) A minimal, non-optimal chordal completion of the graph.

has recently provided the first polyhedral analysis and computational testing of this formulation, deriving facets and other valid inequalities for specific classes of graphs. Alternatively, Bergman and Raghunathan (2015) introduced a Benders approach to the MCCP that relies on a simple class of valid inequalities, outperforming a backtracking search algorithm. In all such cases, theoretical results have focused on specific families of graphs, and several benchmark graphs with fewer than 30 vertices were still not solved to optimality within reasonable computational times.

Our Contributions. In this paper we investigate a novel mathematical programming model for the MCCP that extends the preliminary work of Bergman and Raghunathan (2015). Our formulation is composed of exponentially many constraints – the *chordal inequalities* – that are defined directly on the edge space of the graph and do not depend on the perfect elimination ordering property, as opposed to earlier formulations. We investigate the polyhedral structure of such a model, which reveals that the proposed inequalities are part of a special class of constraints that induce exponentially many facets for cycle subgraphs. This technique can be generalized to lift other inequalities and strengthen the corresponding linear programming relaxation. In particular, we propose three additional families of valid inequalities and provide a study on the computational complexity of the associated separation problems.

Building on these theoretical results, we propose a hybrid solution method for the MCCP that alternates a lazy-constraint generation with a heuristic separation procedure. The resulting approach is compared to the current state-of-the-art models in the literature, and it is empirically shown to improve solution times and known optimality gaps for standard graph benchmarks, often by orders of magnitude. Our numerical results also indicate that the optimal completions can be significantly smaller than the ones provided by state-of-the-art heuristic methods.

Organization of the paper. We start by discussing previous works related to the MCCP in Section 2 and introducing our notation in Section 3. In Section 4 we describe a new integer programming (IP) formulation for the MCCP and characterize its polytope in Section 5, also proving dimensionality results and simple upper bound facets. Section 6 provides an in-depth analysis of the polyhedral structure of cycle graphs, introducing four classes of facet-defining inequalities. In Section 7 we prove general properties of the polytope, including results concerning lifting of facets. Finally, in Section 8 we propose a hybrid solution technique that considers both a lazy-constraint generation and a heuristic-separation method based on a threshold-rounding procedure, and also present a simple primal heuristic for the problem. We provide a numerical study in Section 9, indicating that our approach substantially outperforms existing methods, in particular solving many benchmark graphs to optimality for the first time.

2. Previous Works.

The first graph-theoretical aspects of the MCCP were introduced by Parter (1961) and Rose (1973) in the context of sparse matrix computations. The authors simulated the Gaussian elimination process of a linear system of equations as an *elimination game* in a graph constructed according to the coefficient matrix of the system. In such a game, vertices are removed one at a time and, for each removed vertex v, edges are added so that v forms a clique with its neighbors in the remaining graph. Each new edge corresponds with adding a non-zero entry (i.e., a *fill-in*) to the original matrix, which impacts the total time and storage requirements of the Gaussian elimination process. The total number of edges added, in turn, was shown to be heavily dependent on the order in which vertices are removed during the elimination game.

The MCCP was then formally proposed by Rose et al. (1976), Ohtsuki (1976), and Ohtsuki et al. (1976) simultaneously. This formalization builds on a result by Fulkerson and Gross (1965), which indicates that the family of chordal graphs coincides with the graphs produced by the elimination game. That is, finding an ordering with minimum fill-in corresponds to adding the minimum number of edges so that the graph becomes chordal. Rose et al. (1976) and Ohtsuki (1976) also proposed the first algorithms for finding *minimal* chordalizations in linear time on the size of the graph. A follow-up work by Rose and Tarjan (1978) proved that the MCCP was NP-Hard for directed graphs, but the undirected case was only stated as a conjecture. Later, Garey and Johnson (1979) posed the MCCP as problem #4 among the 12 major open problems in computational complexity, sparking new studies on both theoretical and computational aspects of the problem. The complexity question was settled by Yannakakis (1981), who showed that the MCCP was NP-hard using a reduction from the optimal linear arrangement problem.

In regards to the theoretical aspects of the problem, a large stream of research has focused on alternative characterizations based on *graph separators* as opposed to vertex orderings. A (u, v)-graph separator is a vertex subset that disconnects vertices u and v when removed. Kloks et al. (1993) showed that, if the set of minimal separators of a graph G can be enumerated efficiently, the MCCP and the related *tree-width* problem are solvable in polynomial time. This includes a large variety of graph classes such as permutation graphs, trapezoid graphs, and circle graphs, to name a few. Parra and Scheffler (1995, 1997) and Kloks et al. (1999), with results later completed by Kratsch and Müller (2009), demonstrated that one could obtain a minimal chordal completion by obtaining a maximal set of parallel (i.e., non-crossing) separators and completing them into cliques. Finally, the influential work by Bouchitté and Todinca (2001) provided key results concerning these questions. Namely, it showed that the MCCP is tractable in polynomial time if the set of *potential maximal cliques*, i.e., vertex subsets that induce cliques in some minimal chordalization, could be listed in polynomial time. The result also answers a conjecture posed by Kloks et al. (1993), showing that the MCCP is solvable efficiently for *weakly triangulated graphs*, i.e., graphs with a polynomial number of separators (not necessarily minimal), which generalizes previous work considering specialized graphs (Chang 1996, Kloks et al. 1998).

There also has been an extensive literature on approximation algorithms and fixed-parameter tractable (FPT) procedures for the MCCP. Agrawal et al. (1993) presented the first polynomial-time algorithm with a guaranteed approximation ratio specifically dependent on the size of the graph and on the maximum number of edges k that can be added. Natanzon et al. (2000) later proposed a new algorithm with an approximation ratio of $\mathcal{O}(8k^2)$, i.e., at most $8k$ from the optimal solution. The first FPT algorithm was proposed by Kaplan et al. (1999), with complexity exponential on k. Fomin and Villanger (2012) later proposed a substantially faster FPT that was subexponential on k and based on the results by Bouchitté and Todinca (2001). Most recently, Bliznets et al. (2016) prove lower bounds on the complexity of FPTs under the Exponential Time Hypothesis. Cao and Sandeep (2016) and Cao and Marx (2016) extend these results by producing new parameterized lower bounds, while also excluding possible approximation schemes through a new reduction from vertex cover.

For computational approaches, the primary focus has been on heuristic methodologies with no optimality guarantees, including techniques by Mezzini and Moscarini (2010), Rollon and Larrosa (2011), and Berry et al. (2006, 2003). The state-of-the-art heuristic, proposed by George and Liu (1989), is a simple and efficient algorithm based on ordering the vertices by their degree. Previous articles have also developed methodologies for finding a *minimal* chordal completion. Note that a minimal chordal completion is not necessarily *minimum*, as depicted in Figure 1(c), but does provide a heuristic solution to the MCCP.

Finally, exact computational methodologies for the MCCP have been limited, to the best of our knowledge. A valid mathematical programming model for the MCCP can be obtained by modifying the formulation by Feremans et al. (2002) for determining the *tree-widths* of graphs based on perfect elimination orderings. Yüceoğlu (2015) has recently provided a first polyhedral analysis and computational testing of this formulation, deriving facets and other valid inequalities for specific classes of graphs. Alternatively, Bergman and Raghunathan (2015) introduced a Benders approach to the MCCP, outperforming a simple constraint programming backtracking search algorithm.

We note in passing that the MCCP literature is extensive; we refer to the survey by Heggernes (2006) for additional details and references. We contribute to this body of work by investigating the polyhedral structure of a new mathematical programming formulation for the problem. Besides the underlying structural results from this study, we derive a computational approach that has been effective in providing new solutions and improved bounds for benchmark problems.

3. Notation and Terminology

For the remainder of the paper, we assume that each graph $G = (V, E)$ is connected, undirected, and does not contain self-loops or multi-edges. For any set S, $\binom{S}{2}$ denotes the family of two-element subsets of S. Each edge $e \in E \subseteq \binom{V}{2}$ is a two-element subset of vertices in V. The *complement edge set* (or *fill edges*) E^c of G is the set of edges missing from G; that is, $E^c = \binom{V}{2} \setminus E$. We denote by m and m^c the cardinality of the edge set and of the complement edge set of G, respectively (i.e., $m = |E|$ and $m^c = |E^c|$). The graph *induced* by a set $V' \subseteq V$ is the graph $G[V'] = (V', E')$ whose edge set is such that $E' = E \cap \binom{V'}{2}$. If multiple graphs are considered in a context, we include "(G)" in the notation to avoid ambiguity; e.g., $V(G')$ and $E(G')$ represent the vertex and edge set of a graph G', respectively. Moreover, for every integer $k \geq 0$, we let $[k] := \{1, 2, \ldots, k\}$.

For any ordered list $C = (v_0, v_1, \ldots, v_{k-1})$ of k distinct vertices of V, let $V(C) = \{v_0, v_1, \ldots, v_{k-1}\}$ be the set of vertices composing C, with $|C| = |V(C)|$. The *exterior* of C is the family

$$\xi(C) = \{\{v_{k-1}, v_0\}\} \cup \bigcup_{i \in [k-1]} \{\{v_{i-1}, v_i\}\},$$

and the *interior* of C is the family of two-element subsets of $V(C)$ that do not belong to $\xi(C)$,

$$\iota(C) = \binom{V(C)}{2} \setminus \xi(C).$$

An ordered list C is a *cycle* if $\xi(C) \subseteq E$. If C is a cycle, an element of $\iota(C)$ is referred to as a *chord*. A cycle C for which the induced graph $G[V(C)]$ contains no chords is a *chordless* cycle. G is said to be *chordal* if the maximum size of any chordless cycle is three. A chordless cycle with k vertices is denoted by *k-chordless cycle*.

Let $G = (V, E)$. Any subset of fill edges $F \subseteq E^c$ is a *completion* of G, and $G \cup F$ represents the graph that results from the addition of edges in F to E; that is, $G \cup F := (V, E \cup F)$. A *chordal completion* of G is any set of fill edges $F \subseteq E^c$ for which the completion $G \cup F$ is chordal. A *minimal chordal completion* F is a chordal completion such that, for any proper subset $F' \subset F$, F' is not a chordal completion of G. A *minimum chordal completion* is a minimal chordal completion of minimum cardinality, and the minimum chordal completion problem (MCCP) is concerned with the identification of a minimum chordal completion.

EXAMPLE 1. The graph in Figure 1(a) has three chordless cycles, $C_1 = (v_0, v_1, v_2, v_3)$, $C_2 = (v_1, v_2, v_3, v_4)$ and $C_3 = (v_0, v_1, v_4, v_3)$. Figure 1(c) shows a chordal completion consisting of edges

$\{v_0, v_2\}, \{v_2, v_4\}$, and $\{v_0, v_4\}$. Removing any of these edges will result in a graph that is not chordal, and hence this chordal completion is minimal. Figure 1(b) shows a a *minimum* chordal completion consisting only of edge $\{v_1, v_3\}$.

4. IP Formulation for the MCCP

We now describe the IP formulation investigated in this paper. Given a graph $G = (V, E)$, each binary variable x_f in our model indicates whether the fill edge $f \in E^c$ is part of a chordal completion for G. That is, if we define the set $E(x) := \{f \in E^c : x_f = 1\}$ for each vector $x \in [0,1]^{m^c}$, the set of feasible solutions to our model is given by

$$X(G) := \left\{ x \in \{0,1\}^{m^c} : G \cup E(x) \text{ is chordal} \right\}.$$

Thus, each $x \in X(G)$ equivalently represents the characteristic vector of a chordal completion of G. We use $G(x)$ to denote $G \cup E(x)$, i.e., $G(x) = (V, E \cup E(x))$, and $x(F)$ to represent the characteristic vector of completion $F \subseteq E^c$, i.e., $x(F)_f = 1$ if $f \in F$ and $x(F)_f = 0$ otherwise.

Let $\mathcal{C}$ be the family of all possible ordered lists composed of distinct vertices of V, i.e., every $C \in \mathcal{C}$ can be written as a sequence $C = (v_0, v_1, \ldots, v_{k-1})$ for some $k \leq |V|$. Also, let $F_G(C) = F(C) \subseteq E^c$ be the set of fill edges that are missing in $\xi(C)$ for C to induce a cycle in G, that is,

$$F(C) := \xi(C) \setminus E(G).$$

We propose the following model for the MCCP:

$$\begin{aligned}
\text{minimize} \quad & \sum_{f \in E^c} x_f && \text{(IPC)} \\
\text{s.t.} \quad & \sum_{f \in \iota(C)} x_f - (|C| - 3) \left(\sum_{f \in F(C)} x_f - |F(C)| + 1 \right) \geq 0, \text{ for all } C \in \mathcal{C}, && \text{(I1)} \\
& \qquad \qquad \text{with } \iota(C) \cap E = \emptyset \\
& x \in \{0,1\}^{m^c}.
\end{aligned}$$

The set of inequalities (I1) will also be referred to as the *chordal inequalities*. Note that every sequence C such that $F(C) = \emptyset$ and $\iota(C) \cap E = \emptyset$ describes a cycle in G, and its associated inequality (I1) simplifies to

$$\sum_{f \in \iota(C)} x_f \geq |C| - 3.$$

The following lemma is a reinstatement of Fact 2.1 in Natanzon et al. (2000). It shows that inequalities (I1) are valid in this special case.

Lemma 1. *If C is a chordless cycle of G such that $|C| \geq 4$, then any chordal completion of G contains at least $|C| - 3$ edges that belong to $\iota(C)$.*

Based on the previous lemma, we show below that the set of inequalities (I1) is valid for all sequences in $\mathcal{C}$ and, as a consequence, that (IPC) is a valid formulation for the MCCP.

Proposition 1. *The model (IPC) is a valid formulation for the MCCP.*

Proof. We first show that there is an one-to-one correspondence between $X(G)$ and feasible points of (IPC). Let x^* be a feasible solution to (IPC) and suppose that the graph $G \cup E(x^*)$ contains a chordless cycle C with more than 3 vertices. Then $\sum_{f \in \iota(C)} x^*_f = 0 < |C| - 3$, thus contradicting the feasibility of x^*.

Conversely, let $E^* \subseteq E^c$ be such that $G \cup E^*$ is chordal, and suppose $x^* = \{x \in \{0,1\}^{m^c} : x_f = 1 \Leftrightarrow f \in E^*\}$ is infeasible to (IPC), i.e., x^* violates at least one inequality of type (I1). Let C^* be a sequence associated with a violated inequality $I(C^*)$. We can assume that $|C^*| \geq 4$, as the inequality would be trivially satisfied otherwise. If $I(C^*)$ is violated, the expression multiplying $(|C^*| - 3)$ in the inequality must be equal to 1; otherwise, the sum would be less than or equal to 0, making the inequality trivially satisfied. Consequently, each edge in $\xi(C^*)$ must belong to $G \cup E^*$, i.e., C^* is a cycle in $G \cup E^*$. Moreover, by the definition of (I1), $\iota(C^*) \cap E = \emptyset$, i.e., C^* is chordless in $G + F(C^*)$. Finally, a violation of $I(C^*)$ takes place if $\sum_{f \in \iota(C^*)} x_f < |C^*| - 3$, a condition that, according to Lemma 1, cannot hold if $G \cup E^*$ is chordal, so we have a contradiction. Finally, since the one-to-one correspondence holds and the objective function of (IPC) minimizes the number of added edges, the result follows. □

5. MCCP Polytope Dimension and Simple Upper Bound Facets

This section begins our investigation of the convex hull of the feasible set of chordal completions $X(G)$, which will lead to special properties that can be exploited by computational methods for the MCCP. We identify the dimension of the polytope and provide a proof that the simple upper bound inequalities $x_f \leq 1$ are facet defining.

THEOREM 1. *If $G = (V, E)$ is not a complete graph (and hence not trivially chordal),*

a. $\text{conv}(X(G))$ *is full-dimensional;*

b. $x_f \leq 1$ *is facet-defining for all* $f \in E^c$.

Proof. We first show (a). Let $e \in \{0,1\}^{m^c}$ be the vector consisting only of ones and $e^j \in \{0,1\}^{m^c}$ be the unit vector for coordinate j. By definition, $G \cup E(e)$ is the complete graph, which is trivially chordal. By removing the edge associated with coordinate j, we obtain graph $G \cup E(e - e^j)$, which is also chordal. The set of $m^c + 1$ vectors $\{e\} \cup \{e - e^1, e - e^2, \ldots, e - e^{m^c}\}$ is affinely independent and contained in the set $X(G)$, and so it follows that $\text{conv}(X(G))$ is a full-dimensional polytope.

For (b), let $f \in E^c$, and notice that the set of m^c vectors $\{e\} \cup \{e - e^{f'} : f' \in E^c \setminus \{f\}\}$ is affinely independent and satisfies $x_f = 1$. □

6. Cycle Graph Facets

We restrict our attention now to *cycle graphs*, i.e., graphs consisting of a single cycle. Cycle graphs are the building blocks of computational methodologies for the MCCP, since finding a chordal completion of a graph naturally concerns identifying chordless cycles and eliminating them by adding chords. We present four classes of facet-defining inequalities for cycle graphs in this section.

Let $G = (V, E)$ be a cycle graph associated with a k-vertex chordless cycle $C = (v_0, \ldots, v_{k-1})$, i.e., $V = V(C)$ and $E = \xi(C)$. Assume all additions and subtractions involving indices of vertices are modulo-k. The proofs presented in this section show only the validity of the inequalities; arguments proving that they are facet defining for cycle graphs are presented in Section EC.1.

Proposition 2. *Let $G = (V, E)$ be a cycle graph associated with cycle $C = (v_0, \ldots, v_{k-1})$, $k \geq 4$. The chordal inequality (I1) associated with C, which in this case simplifies to*

$$\sum_{f \in \iota(C)} x_f \geq |C| - 3,$$

is valid and facet defining for $\mathrm{conv}(X(G))$. □

The proof of the validity of the inequality in Proposition 2 follows directly from Lemma 1.

Proposition 3. *If $k \geq 4$, the inequality*

$$x_{\{v_{i-1}, v_{i+1}\}} + \sum_{f : v_i \in f, \{v_{i-1}, v_{i+1}\} \cap f = \emptyset} x_f \geq 1, \quad \text{for all } i \in \{1, \ldots, k\} \tag{I2}$$

is valid and facet defining for $\mathrm{conv}(X(G))$.

Proof. Suppose that (I2) is violated by some $x \in \mathrm{conv}(X(G))$, i.e., that for some $\{v_{i-1}, v_i, v_{i+1}\} \subset V$, $x_{\{v_{i-1}, v_{i+1}\}} + \sum_{f : v_i \in f, \{v_{i-1}, v_{i+1}\} \cap f = \emptyset} x_f = 0$. As $k \geq 4$, a shortest path P from v_{i-1} to v_{i+1} in $G(x)$ that does not include v_i traverses at least two edges. The sequence defined by the concatenation of P with (v_{i-1}, v_i, v_{i+1}) thus defines a k'-chordless cycle of $G(x)$ for $k' \geq 4$, a contradiction. □

For the next proposition, some additional notation is in order. For any two vertices v_i and v_j with $i < j$, let $d_C(v_i, v_j) := \min\{j - i, k - j + i\}$ be the distance between v_i and v_j in C, and assume $d_C(v_i, v_j) := d_C(v_j, v_i)$ if $i > j$.

Proposition 4. *If $k \geq 5$, the inequality*

$$\sum_{f \in \left\{ \{v_i, v_j\} \in E^c \,:\, d_C(v_i, v_j) = 2 \right\}} x_f \geq 2 \tag{I3}$$

is valid and facet defining for $\mathrm{conv}(X(G))$.

Proof. Inequality (I3) states that at least two out of the $|C|$ pairs of vertices of distance 2 must appear in any chordal completion of G. Given completion F of G, without loss of generality, let $f' = \{v_0, v_{j_1}\}$ be the edge of $\iota(C)$ that connects the "closest" vertices with respect to d_C. If $j_1 \geq 3$, then $C' = (v_0, v_1, \ldots, v_{j_1})$ is a chordless cycle in $G \cup F$, a contradiction; therefore, $j_1 = 2$.

As $k \geq 5$, $C' = (v_0, v_2, v_3, \ldots, v_{k-1})$ is a cycle in $G \cup F$ with at least 4 vertices, so at least one edge of $\iota(C')$ must be present in $G \cup F$. Let $f'' = \{v'_i, v'_{j_2}\}$ be the edge of $\iota(C')$ that connects the "closest" vertices with respect to $d_{C'}$. An argument similar to the one used to show $j_1 = 2$ shows that f'' connects two vertices of distance 2 in C', so we have two cases to analyze. First, if $f'' \neq \{v_0, v_3\}$, the result follows directly. Otherwise, we either have $|C'| = 4$, in which case $d_{C'}(v_0, v_3) = d_C(v_0, v_3) = 2$, as desired, or we have a chordless cycle $C'' = (v_0, v_3, \ldots, v_{k-1})$ in $G \cup F$ with at least 4 vertices, on which we can apply the same arguments; as a sequence of cycles emerging from this construction will lead to a cycle of length 4, the result holds. $\square$

PROPOSITION 5. *If $k \geq 5$, the inequality*

$$\sum_{f \in \iota(C) \setminus \{\{v_{j-1}, v_{j+1}\}, \{v_j, v_i\}\}} x_f \geq |C| - 4, \quad \text{for all } i, j \in \{1, \ldots, k\}, d_C(v_j, v_i) \geq 2 \tag{I4}$$

is valid and facet defining for $\operatorname{conv}(X(G))$.

Proof. Given vertices v_i and v_j such that $d_C(v_j, v_i) \geq 2$, inequality (I4) enforces the inclusion of at least $|C| - 4$ edges of $\iota(C) \setminus \{\{v_{j-1}, v_{j+1}\}, \{v_j, v_i\}\}$ in any chordal completion of G. Without loss of generality, let $j = 0$ and let i be any value in $[2, k-3]$. Suppose by contradiction that there exists $x^0 \in X(G)$ such that

$$\sum_{f \in \iota(C) \setminus \{\{v_{k-1}, v_1\}, \{v_0, v_i\}\}} x^0_f < |C| - 4. \tag{1}$$

By Lemma 1, we have that

$$\sum_{f \in \iota(C)} x^0_f \geq |C| - 3.$$

This implies that $x^0_{v_{k-1}, v_1} = x^0_{v_0, v_i} = 1$, for otherwise inequality 1 would be violated. Thus, the sequences $C^1 = (v_0, v_1, \ldots, v_i)$ and $C^2 = (v_0, v_i, v_{i+1}, \ldots, v_{k-1})$ are cycles in $G(x^0)$. Again, by Lemma 1, at least $|C^\ell| - 3$ fill edges must be present in $\iota(C^\ell)$, $\ell = 1, 2$. This is only possible if at least $i - 2$ edges of $\iota(C^1)$ and at least $|C| - i - 2$ edges of $\iota(C^2)$ belong to the set of fill edges described by x^0. As $\iota(C^1) \cap \iota(C^2) = \emptyset$ and $\iota(C^1) \cup \iota(C^2) \subseteq \iota(C)$, we have that

$$\sum_{f \in \iota(C) \setminus \{\{v_{i-1}, v_{i+1}\}, \{v_0, v_i\}\}} x^0_f \geq \sum_{f \in \operatorname{int}(C^1)} x^0_f + \sum_{f \in \operatorname{int}(C^2)} x^0_f \geq |C| - 4,$$

thus contradicting inequality (1).

Example 2. Let G be a cycle graph associated with 6-cycle $C=(v_0,v_1,v_2,v_3,v_4,v_5)$. An example of inequality (I2) with $i=1$ is $x_{\{v_0,v_2\}}+\left(x_{\{v_1,v_3\}}+x_{\{v_1,v_4\}}+x_{\{v_1,v_5\}}\right)\geq 1$, which enforces the inclusion of at least one edge $f=\{v_1,v_k\}$, $k\in\{3,4,5\}$, for every chordal completion without edge (v_0,v_2).

Inequality (I3) translates to $x_{\{v_5,v_1\}}+x_{\{v_0,v_2\}}+x_{\{v_1,v_3\}}+x_{\{v_2,v_4\}}+x_{\{v_3,v_5\}}+x_{\{v_4,v_0\}}\geq 2$, which enforces that at least 2 of the 6 pairs of vertices that are separated by one vertex must appear in any chordal completion of G. Finally, inequality (I4) for $i=4$ and $j=1$ is given by

$$x_{\{v_0,v_3\}}+x_{\{v_0,v_4\}}+x_{\{v_1,v_3\}}+x_{\{v_1,v_5\}}+x_{\{v_2,v_4\}}+x_{\{v_2,v_5\}}+x_{\{v_3,v_5\}}\geq 2,$$

which enforces that at least 2 of the edges in $\iota(C)\setminus\{\{v_0,v_2\},\{v_1,v_4\}\}$ must be included in any chordal completion of G.

7. General Polyhedral Properties

This section provides theoretical insights into the polyhedral structure of the MCCP polytope. The first result, provided in Theorem 2, shows that any inequality proven to be valid on an induced subgraph can be extended into a valid inequality for the original graph. This result is relevant since it shows that finding valid inequalities/facets on particular substructures, such as cycles, helps in the generation of valid inequalities for larger graphs containing these substructures. Theorem 3 shows how facets for a cycle graph can be lifted to inequalities that, in turn, are facets for its subgraphs. This leads to the result in Corollary 1 concerning when the inequalities (I1) in their general form are facet defining. The final result of the section, Theorem 4, proves and describes how facets for small cycles can be lifted to facets of larger cycles.

First, we show a lemma that will be used in the proof of Theorem 2.

Lemma 2. *If $G=(V,E)$ is chordal, then $G[W]$ is chordal for any $W\subseteq V$.*

Proof. It follows since a chordless cycle C in $G[W]$ must be a chordless cycle in G. □

Theorem 2. *Let $G=(V,E)$ be an arbitrary graph and $W\subseteq V$ be any subset of vertices. If $a'x\geq b$ is a valid inequality for $X(G[W])$, then $ax\geq b$ is a valid inequality for $X(G)$, where $a_f=a'_f$ if $f\in E^c(G[W])$ and $a_f=0$ otherwise.*

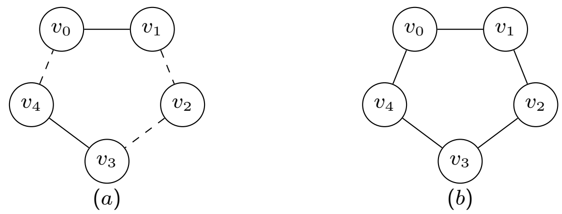


Figure 2 (a) A graph G with $V = \{v_0, v_1, v_2, v_3, v_4\}$ and $E = \{\{v_0, v_1\}, \{v_3, v_4\}\}$. Solid lines represent graph edges and dashed lines are fill edges whose addition to G makes a chordless cycle of length 5. (b) Cycle graph with 5 vertices.

Proof. By way of contradiction, suppose that $a'x \geq b$ is valid for $X(G[W])$ and let F be a chordal completion of G such that $ax(F) < b$. From the construction of a, we have $ax(F) = a'x(F \cap E^c(G[W])) < b$. By Lemma 2, $G[W] + F \cap E^c(G[W])$ must be chordal and, consequently, we must have $a'x(F \cap E^c(G[W])) \geq b$, establishing thus a contradiction. □

We now present a result that goes in the opposite direction of Theorem 2. Namely, it shows how facet-defining inequalities for a cycle graph G can be transformed into facet-defining inequalities for subgraphs of G; note that subgraphs of cycle graphs consist of collections of paths. This result allows us to show that inequality (I1) is facet defining for subgraphs of cycle graphs.

THEOREM 3. *Let $G' = (V, E')$ be a cycle graph associated with the cycle $C = (v_0, v_1, \ldots, v_{k-1}) \in \mathcal{C}$ and $G = (V, E)$ be a subgraph of G' such that $G' = G \cup F_G(C)$, $F_G(C) = E' \setminus E$. If $ax \geq b$ is facet defining for* $\text{conv}(X(G'))$, *$a \geq \mathbf{0}$, and $a' \in \mathbb{R}^{|E^c|}$, with $a'_f = a_f$ if $f \in E^c \setminus F_G(C)$ and $a'_f = 0$ otherwise, the inequality*

$$a'x \geq b\left(\sum_{f \in F_G(C)} x_f - |F_G(C)| + 1\right)$$

is facet defining for $\text{conv}(X(G))$. □

Theorem 3 (proved in Section EC.2) immediately leads to the following result:

COROLLARY 1. *For any graph $G = (V, E)$, and for any sequence $C \in \mathcal{C}$ such that $\iota(C) \cap E = \emptyset$, the chordal inequality (I1) is facet defining for the MCCP polytope of $G[V(C)]$.*

EXAMPLE 3. Consider the graph G in Figure 2(a); solid lines represent graph edges in this example. As in the statement of Theorem 3, we have $C = (v_0, v_1, v_2, v_3, v_4)$, $E(G) =$

$\{\{v_0,v_1\},\{v_3,v_4\}\}$, and $F_G(C)=\{\{v_1,v_2\},\{v_2,v_3\},\{v_4,v_0\}\}$. Graph $G\cup F(C)$ is 5-chordless cycle. One facet-defining inequality for this cycle, according to Proposition 2, is the simplified version of the chordal inequality, given by $x_{\{v_0,v_2\}}+x_{\{v_0,v_3\}}+x_{\{v_1,v_3\}}+x_{\{v_1,v_4\}}+x_{\{v_2,v_4\}}\geq 2$. Corollary 1 stipulates that the inequality below is facet defining for G:

$$x_{\{v_0,v_2\}}+x_{\{v_0,v_3\}}+x_{\{v_1,v_3\}}+x_{\{v_1,v_4\}}+x_{\{v_2,v_4\}}\geq 2\cdot\left(x_{\{v_1,v_2\}}+x_{\{v_2,v_3\}}+x_{\{v_4,v_0\}}-3+1\right).$$

By Theorem 2, these inequalities will also be valid even if G is a subgraph of a larger graph.

We now define a method for lifting facet-defining inequalities defined on smaller cycles into facet-defining inequalities for large cycles. This is done by considering the inclusion of chords into the inequalities, which reveals a lifting property of MCCPs that can be used to strengthen known inequalities. We present this result in Theorem 4, which is proved in Section EC.2.

THEOREM 4. *Let $G=(V,E)$ be a cycle graph associated with the cycle $C=(v_0,v_1,\ldots,v_{k-1})$ and let $f^*=\{v_s,v_t\}\in E^c$, $0\leq s<t\leq k-1$, be any chord of C. If the inequality $ax\geq b$, $a\geq \mathbf{0}$, is facet-defining for the MCCP polytope of cycle graph $G'=(V',E')$ associated with the cycle $C'=(v_s,v_{s+1},\ldots,v_t)$ (i.e., $G'=G[V(C')]+\{f^*\}$), then*

$$a'x\geq b\cdot x_{f^*}$$

is facet defining for $\operatorname{conv}(X(G))$*, where $a'_f=a_f$ if $f\in\iota(C')$ and $a'_f=0$ otherwise.* □

If $ax\geq b$ is facet defining for any induced subcycle obtained by adding edge f^*, then $ax\geq b\cdot x_{f^*}$ will be facet defining for the original cycle graph.

EXAMPLE 4. Consider the graph in Figure 2(b), and let $G'=G[\{v_1,v_2,v_3,v_4\}]+\{\{v_1,v_4\}\}$ be a cycle graph associated with cycle $C'=(v_1,v_2,v_3,v_4)$. From Proposition 2, we have that the following chordal inequality is facet defining for $\operatorname{conv}(X(G'))$:

$$\sum_{f\in\iota(C')}x_f=x_{\{v_1,v_3\}}+x_{\{v_2,v_4\}}\geq|C'|-3=1.$$

This inequality can be modified in order to become facet defining for $\operatorname{conv}(X(G))$ through Theorem 4, yielding $x_{\{v_1,v_3\}}+x_{\{v_2,v_4\}}\geq x_{\{v_1,v_4\}}\implies x_{\{v_1,v_3\}}+x_{\{v_2,v_4\}}-x_{\{v_1,v_4\}}\geq 0$.

8. Solution Method for the MCCP

We now describe a procedure to solve formulation (IPC) for general graphs based on the structural results showed in the previous sections. Since the model has exponentially many constraints, our solution technique is based on a hybrid branch-and-bound procedure that applies separation, lazy-constraint generation, and a primal heuristic to the problem.

8.1. Separation complexity

The problem of separating inequalities (I1)-(I4) over an *integer* point x^* is equivalent to finding a chordless cycle on the graph completion $G' = (V, E') = G \cup E(x^*)$. In our computational experiments, we employed an adapted version of the $O(|V| + |E'|^2)$ procedure proposed by Nikolopoulos and Palios (2007) for this task. Namely, we removed the condition which forces the algorithm to stop as soon as a first cycle is detected, thereby enabling the procedure to potentially identify several cycles every time it is executed. This change does not impact the worst-case complexity of the algorithm. We also note in passing that the recent work by Uno and Satoh (2014) enumerates all chordless cycles of G' in time $O(|E'|)$ per cycle. Nonetheless, the execution of such procedure at every integer node was far too computationally expensive in our analysis.

The separation problem of (I1)-(I4) over *fractional* points is, however, much more challenging. We state the results below concerning this question.

Theorem 5. *Given a fractional point* $x^* \in [0,1]^{m^c}$*:*

a. The separation problem of (I1) *is NP-Complete.*

b. Inequalities (I2) can be separated in $O(|V|^5)$*.*

c. Inequalities (I3) can be separated in $O(|V|^8)$*.*

d. The separation problem of (I4) is NP-Complete.

Proof. Due to space limitations, we present below only proof sketches for these results. The full version of each proof is presented in Section EC.3 of the online supplement.

a. The proof reduces the *quadratic assignment problem* (QAP), a classical and well-studied NP-hard problem, to the α*-quadratic shortest cycle problem* (α-QSCP), introduced in this paper.

In the QSCP, we are given a graph $G=(V,E)$ and a quadratic cost function $q:V\times V\to[0,1]$, with $q(u,v)=0$ if $(u,v)\in E$. A feasible solution of QSCP is a simple chordless cycle $C=(v_1,v_2,\ldots,v_{|C|})$ whose cost is $p(C)=\sum_{\{u,v\}\in E(G[C])^C} q(u,v)-|C|$. The α-QSCP is the decision version of QSCP in which the goal is to decide whether G has a simple chordless cycle C such that $p(C)<\alpha$. We employ a reduction of the quadratic assignment problem to (-3)-QSCP that resembles the ones used by Rostami et al. (2015) for the quadratic shortest path problem. Finally, the -3-QSCP is reduced to the problem of separating the inequality (I1), completing the proof.

b. An auxiliary graph G', a complete digraph on $|V(G)|$ nodes, is constructed for which the separation problem is reduced to finding, for every triple of vertices (v_1,v_2,v_3), the shortest path from v_1 to v_3 that does not include v_2. The number of sequences for which this verification needs to be performed is $O(|V(G)|^3)$, and the identification of such a path can be made in time $O(|V(G)|)^2$.

c. As in **b**, an auxiliary graph G', specifically a complete digraph on $|V(G)|^2$ nodes, is constructed for which the separation problem is reduced to finding at most $O(|V(G)|^4)$ shortest paths, each of which can be performed in polynomial time.

d. The proof is similar to that of **a**, except that we use a reduction from -4-QSCP*, a slight variant of -3-QSCP. □

8.2. Heuristic separation algorithms

In view of Theorem 5, we tackle model (IPC) by applying a branch-and-bound procedure that alternates between heuristic separation and lazy-constraint generation.

For the lazy constraint generation part, at every integer node of the branching tree we apply the procedure presented in Section 8.1 to separate at least one violated inequality (I1)-(I4), similar to a combinatorial Benders methodology (Codato and Fischetti 2006). Proposition 1 ensures that this approach yields an optimal (and feasible) solution to (IPC), since a violated inequality is not found if and only if the resulting graph is chordal.

Nonetheless, adding violated inequalities only at integer points typically yield weak bounds at intermediate nodes of the branching tree. Since a complete separation of fractional points is not viable due to Theorem 5, we consider a heuristic *threshold* procedure. Given a point $x^* \in [0,1]^{m^c}$ and a threshold $\delta \in (0,1)$, let $E^\delta(x) := \{f \in E^c : x_f \geq \delta\}$, we can use the procedure from Section 8.1 to find violating inequalities for the graph $G \cup E^\delta(x^*)$. Such inequalities may not be necessarily violated by x^*, and thus require a (simple) extra verification testing step. The threshold policy does not guarantee that at least one violated inequality is found, but it can be performed efficiently and, as our numerical experiments indicate, it is a fundamental component for good performance.

8.3. Primal Heuristic

We have also incorporated a primal heuristic to be applied at infeasible integer nodes of the branching tree. The method is based on the state-of-the-art heuristic for the problem, designed by George and Liu (1989). Specifically, the vertices of the graph are sorted in ascending order according to their degree, thereby defining a sequence $S = (v_1, v_2, \ldots, v_{|V|})$. The vertices are then picked one at a time, in the order indicated by S. For each vertex v_i, edges are added to G so that S defines a *perfect elimination ordering*, i.e., v_i and its neighbours on set $\{v_{i+1}, v_{i+2}, \ldots, v_{|V|}\}$ induce a clique, which makes G chordal. This procedure has complexity $O(|V|^2\,|E|)$.

For any integer point $x \in \{0,1\}^{m^c}$ found during the branch-and-bound procedure, if $G' := G \cup E(x)$ is not chordal, we can apply George and Liu (1989)'s heuristic in order to chordalize G' and obtain a feasible solution to the problem. The application of this procedure at the root node ensures we can identify solutions which are at least as good as those provided by the heuristic.

9. Numerical Experiments

In this section we present an experimental evaluation of the solution methods introduced in this paper. The experiments ran on an Intel(R) Xeon(R) CPU E5-2640 v2 at 2.80GHz with 100 GB RAM. We used the integer programming solver IBM ILOG CPLEX 12.7.1 (IBM ILOG 2017) in all experiments, which were limited to one thread and 3,600 seconds of execution. Experiments that exceeded the memory limit had their time limits reduced by 5 or 10 minutes in order to enable the extraction of an optimality gap.

Instances. We considered five families of instances for the experimental evaluation: four structured benchmarks (relaxed caveman graphs, grid graphs, queen graphs, and DIMACS), and one new family derived from an application in knowledge-based systems (RCC-8).

- *Relaxed caveman graphs* (Judd et al. 2011) are used in the analysis of social network models, where small pockets of individuals are tightly connected and have sporadic connections to individuals in other groups. Each instance is randomly generated based on three parameters, $\alpha, \beta \in \mathbb{Z}^+$, and $\gamma \in (0,1)$. Starting from a set of β disjoint cliques each of size α, each edge is examined and, with probability γ, one of its endpoints is uniformly at random switched to a vertex belonging to another clique. The structure of relaxed caveman graphs is particularly useful for evaluating algorithms for chordal completions. Namely, the endpoint switches lead to large chordless cycles, thus enforcing the inclusion of several edges in chordal completions. For our experiments, ten instances of each configuration of $\alpha, \beta \in \{4,5,6,7,8\}$ and $\gamma = 0.30$ were generated. This set of graphs will be henceforth be referred to as caveman instances.

- *Grid graphs* are graphs whose vertex sets can be partitioned into a set of r *rows* $R_1, \ldots, R_r$ and c *columns* $C_1, \ldots, C_c$. Each pair (r', c') in $[r] \times [c]$ is associated with a single vertex $v_{r',c'} \in R_{r'} \cap C_{c'}$. Vertices $v_{i,j}$ and $v_{k,l}$ are adjacent if and only if either $i = k$ and $|j-l| = 1$, or $j = l$ and $|i-k| = 1$. These graphs have been used in previous computational studies of the MCCP because they contain large chordless cycles (Yüceoğlu 2015).

- *Queen graphs* are extensions of grid graphs with additional edges representing longer hops as well as diagonal movements. More precisely, there exists an edge connecting $v_{i,j}$ to $v_{i',j'}$ if and only if one of the three following conditions is satisfied for some k: (1) $i = i' \pm k$ and $j = j' \pm k$, (2) $i = i'$ and $j = j' \pm k$, or (3) $i = i' \pm k$ and $j = j'$. The configurations of grid graphs and of queen graphs used in our experiments are equivalent to those used by Yüceoğlu (2015).

- *DIMACS* is a classical graph coloring benchmark frequently used in computational evaluations of graph algorithms. The dataset can be downloaded from http://dimacs.rutgers.edu/Challenges/.

- *RCC-8* are graphs derived from real-world region-connection calculus networks (Renz 2002). RCC-8 is the state-of-the-art approach for qualitative topological reasoning with applications, e.g., in the Semantic Web (Zhang et al. 2015). In an RCC-8 network, nodes correspond to regions (in some topology) and a connection between nodes represents a relationship the regions share. For instance, regions could be associated with countries and connections with trade agreements.

The main use of RCC-8 networks is to answer qualitative *queries*, e.g., what countries may be involved with a particular type of trade. To perform these queries, the networks must be modified by enforcing a notion of consistency (i.e., *path-consistency*), which is achieved by chordalizing the graph. The size of the chordal completion plays a critical role on the efficiency of the consistency methods for RCC-8 (Sioutis and Koubarakis 2012, Sioutis 2014).

For our experiments, we extracted subgraphs of an RCC-8 network associated with administrative regions of the United Kingdom. The graph that must be subjected to path consistency was obtained from Nikolaou and Koubarakis (2014) (`adm1`). For each $n \in \{25, 50, 75, 100, 125, 150\}$, we chose n vertices by picking a vertex v uniformly at random, setting $S = \{v\}$, and increasing S by picking vertices that are adjacent to at least one vertex in S uniformly at random until $|S| = n$. Finally, we generated the graph induced by S for our tests. This was repeated 10 times per n.

Other Approaches. The state-of-the-art techniques to the MCCP reported in the literature are a branch-and-cut approach by Yüceoğlu (2015) and a Benders decomposition approach by Bergman and Raghunathan (2015), henceforth denoted by `YUC` and `BEN`, respectively. The complete branch-and-cut algorithm proposed in this paper will be denoted by `BC`.

`YUC` employs a branch-and-cut approach based on a polyhedral analysis of a *perfect elimination ordering* (PEO) model for the MCCP. The PEO model is presented in Section EC.4. `YUC` incorporates valid inequalities for grid and queen graphs. `BEN` is the precursor of the approach described in the present work, in that it solves a formulation consisting only of inequalities (I1) with a pure Benders decomposition approach; i.e., `BEN` solves to optimality an MILP using the current set of inequalities (I1) found up to the current iteration. If the solution contains no chordless cycles, then it is optimal. Otherwise, a collection of chordless cycles is found, new inequalities (I1) are added to the model, and the procedure repeats.

Also of interest is to compare our approach with state-of-the-art heuristics in terms of solution quality, thus assessing how significant the differences between exact and heuristic solutions are. We consider the state-of-the-art heuristic developed by George and Liu (1989) (described in Section 8.3), which will henceforth be denoted by `MDO`.

9.1. Algorithmic Enhancements

We first evaluate the impact of individual algorithmic enhancements on `BEN`. The following enhancements are considered: (1) using all inequalities (I1)-(I4); (2) separating the inequalities at fractional nodes based on our polyhedral results; and (3) invoking `MDO` as a primal heuristic. `BC` is an implementation where all enhancements are applied, i.e., `BEN`+(1)+(2)+(3).

Figure 3 presents a cumulative distribution plot of performance considering all instances classes. The horizontal axis is divided into two areas: the left portion denotes solution time (in seconds), and the right portion denotes relative optimality gap (in this case, computed as ${}^{ub-lb}/_{ub+10^{-8}}$, where ub and lb represent the resulting upper and lower bounds, respectively). The vertical axis denotes the number of solutions with that solution time or relative gap. Note that instances solved in up to 3,600s have a final gap of 0%, and hence the curves are non-decreasing. Furthermore, each instance of *caveman* was counted as 0.1 because of its randomized process.

The plot indicates that separating at fractional nodes (`BEN`+(2)) significantly improves the optimality proof time for instances that were solved within the time limit. Adding a primal heuristic (`BEN`+(3)) was crucial in obtaining better relative gaps, especially since `MDO` typically provides good solutions quickly, as exhibited later in this section. In contrast, additional cuts without fractional separation (`BEN`+(1)) did not provide significant enhancements. The complete method, `BC`, outperformed all other approaches both in terms of solution times and relative gaps, thus suggesting a positive effect of combining the individual enhancements. The same behavior was observed when plotting each benchmark separately. Other possible combinations were dominated by `BC` as well.

In order to verify whether an individual enhancement leads to a statistically significant improvement in performance, a two-tailed paired t-test was employed. Specifically, the null hypothesis

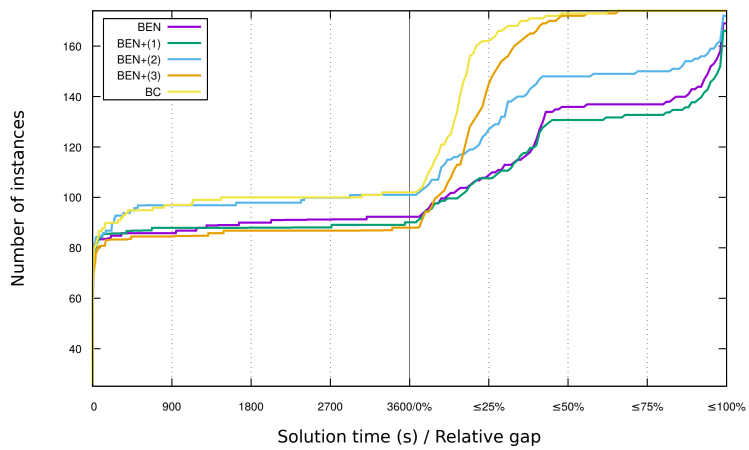


Figure 3 Cumulative distribution plot evaluating individual algorithmic enhancements.

indicates whether the solutions times or gaps are equivalent with or without an enhancement. We first consider the solution times, restricting to instances that were solved within 3,600s for all methodologies. For algorithms BEN+(1), BEN+(2), BEN+(3), and BC, the tests resulted in a p-value of 0.57, 0.00024, 0.088, and 7.6×10^{-5}, respectively, in comparison to BEN. Notice that this is consistent with Figure 3, since only BEN+(2) and BC provided significant and consistent improvements to BEN. For the relative optimality gap, we consider all tested instances. The p-values for BEN+(1), BEN+(2), BEN+(3), and BC were then 0.016, 5.7×10^{-7}, 2.0×10^{-12}, and 2.5×10^{-14}, respectively. This is again consistent with Figure 3 and shows that the relative gaps for BEN+(3) and BC are significantly superior to BEN.

Table 1 provides further details on the impact of the enhancements on each dataset. Column $\#I$ reports the total number of instances in the dataset; columns t provides the average solution time for instances that were solved within 3,600 seconds, with the number in parenthesis representing the number of such instances; and columns $\%Gap$ provides the average gap for instances that were not solved within the time limit. BC outperforms the other configurations, with respect to both the relative gap and the number of instances solved within the time limit. There were no instances solved by other configurations that BC was unable to solve.

Cut Analysis. We also experiment with enabling and disabling distinct combinations of valid inequalities (I2)-(I4) in the presence of the primal heuristic and separation at fractional nodes. BC

Table 1 Relative performance of individual improvement per benchmark.

		BEN		BEN+(1)		BEN+(2)		BEN+(3)		BC	
Set	*#I*	*t*	*%Gap*	*t*	*%Gap*	*t*	*%Gap*	*t*	*%Gap*	*t*	*%Gap*
queen	32	$107.2^{(15)}$	32	$120.9^{(15)}$	13	$55.1^{(15)}$	24	$77.5^{(15)}$	31	$109.2^{(15)}$	14
grid	23	$286.9^{(11)}$	54	$9.4^{(10)}$	25	$68.6^{(13)}$	34	$7.6^{(10)}$	51	$45.1^{(13)}$	13
caveman	25	$45.0^{(24)}$	22	$47.3^{(24)}$	7	$17.0^{(25)}$	0	$24.8^{(24)}$	38	$5.9^{(25)}$	0
rcc-8	60	$134.6^{(24)}$	59	$175.1^{(22)}$	21	$354.2^{(30)}$	55	$12.2^{(23)}$	61	$346.4^{(31)}$	14
DIMACS	34	$167.0^{(18)}$	83	$91.5^{(17)}$	31	$34.1^{(18)}$	61	$350.9^{(18)}$	83	$64.6^{(18)}$	26
Total	174	$120.4^{(92)}$	56	$84.4^{(88)}$	22	$128.7^{(101)}$	46	$87.6^{(90)}$	57	$138.6^{(102)}$	17

had a higher average running time than the algorithms with no extra inequalities or with a single family of cuts (I2), (I3), (I4). Nevertheless, BC solved to optimality either one or two instances more than each of the other algorithms; in particular, BC solves all instances that were solved by the other combinations. Average relative gaps varied by at most 1% between combinations. Finally, two-tailed paired t-tests indicate that there is no significant difference in solution times or relative gaps between combinations. We therefore use all inequalities (I2)-(I4) in the subsequent experiments.

9.2. Comparison with Other Approaches

This section provides a comparison of BC with YUC, as well as with the MDO heuristic. We first focus on the instances reported upon in Yüceoğlu (2015) and Bergman and Raghunathan (2015)—grid, queen, and DIMACS graphs—and compare solution times and objective function bounds. The reported numbers for YUC were obtained directly from Yüceoğlu (2015), which were run with IBM ILOG CPLEX 12.2 and a processor with similar clock (2.53 GHz), but using the parallel version of the solver, with 4 cores, and not with 1 core, as was set in our experiments.

The data for grid graphs, queen graphs, and DIMACS graphs are presented in Tables 2, 3, and 4, respectively. These tables report, for each instance, the number of vertices, the number of edges, and, for all algorithms, the resulting lower bounds, upper bounds, and solution times (in seconds if solved to optimality in 3,600s, or a ”-” mark otherwise). An upper bound presented in bold face indicates that the algorithm found the best-known solution for the instance.

As a summary of our results, BC finds the best known solutions in 85 out of all 89 of these instances, and improves upon the best lower bound in 14 of the 64 instances (13 by more than the least integer greater than the bound) that have previously been reported on in the literature. In particular, the optimality gap in 6 of the instances was closed for the first time. We now discuss the results for each class of graph in detail.

For grid graphs, Yüceoğlu (2015) enhances `PEO` with cuts tailored for graphs containing grid structures, so these instances are particularly well-suited for `YUC`. The results are presented in Table 2. `BC` always finds the best-known solutions, and only in 4 cases out of 23 the relaxation bound for `YUC` outperforms that of `BC` (3 by more than the last integer greater than the bound). Also, `BC` improves upon the solutions by `MDO` in 9 instances, proving optimality for 3 new cases.

Table 2 Results for grid graphs.* Note: grid5_5 was not included in the results by Yüceoğlu (2015).

Instance			YUC			BC			MDO
name	$\|V\|$	$\|E\|$	*LB*	*UB*	*t*	*LB*	*UB*	*t*	*UB*
grid3_3	9	12	5	**5**	0.01	5	**5**	0.01	**5**
grid3_4	12	17	9	**9**	0	9	**9**	0.01	**9**
grid3_5	15	22	13	**13**	0.02	13	**13**	0.04	**13**
grid3_6	18	27	17	**17**	0.02	17	**17**	0.09	**17**
grid3_7	21	32	21	**21**	0.01	21	**21**	0.12	**21**
grid3_8	24	37	25	**25**	0.02	25	**25**	0.69	**25**
grid3_9	27	42	29	**29**	0.02	29	**29**	0.87	33
grid3_10	30	47	33	**33**	0.03	33	**33**	2.6	37
grid4_4	16	24	18	**18**	1.23	18	**18**	0.58	**18**
grid4_5	20	31	25	**25**	18.11	25	**25**	3.95	**25**
grid4_6	24	38	32.2	**34**	-	34	**34**	71.26	**34**
grid4_7	28	45	39	**41**	-	41	**41**	370.9	**41**
grid4_8	32	52	45.5	52	-	47.97	**50**	-	**50**
grid4_9	36	59	52.5	58	-	52.89	**57**	-	**57**
grid4_10	40	66	59.3	**66**	-	58.58	**66**	-	**66**
grid5_5	25	40	*	*	*	37	**37**	135.32	**37**
grid5_6	30	49	46.2	53	-	47.31	**50**	-	52
grid5_7	35	58	56.9	65	-	56.03	**62**	-	68
grid5_8	40	67	67.5	77	-	63.30	**75**	-	80
grid5_9	45	76	33.3	90	-	70.33	**87**	-	93
grid6_6	36	60	60.9	77	-	57.03	**69**	-	71
grid6_7	42	71	31	94	-	68.22	**86**	-	92
grid7_7	49	84	37	125	-	80.82	**111**	-	119

* Numerical results for the `YUC` columns were taken directly from Yüceoğlu's Thesis.

Next, Table 3 reports on queen graphs. As previously mentioned, these instances are also well-suited to `YUC` because of their grid-like structures. The results show that `BC` typically delivers better upper bounds, whereas `YUC` frequently returns smaller optimality gaps and solution times

Table 3 Results for queen graphs.* Note: queen9_9 was not included in the results by Yüceoğlu (2015).

Instance			YUC			BC			MDO
name	$\vert V\vert$	$\vert E\vert$	LB	UB	t	LB	UB	t	UB
queen3_3	9	28	5	**5**	0	5	**5**	0.01	**5**
queen3_4	12	46	12	**12**	0.01	12	**12**	0.01	**12**
queen3_5	15	67	22	**22**	0.31	22	**22**	0.01	**22**
queen3_6	18	91	36	**36**	1.03	36	**36**	0.07	**36**
queen3_7	21	118	53	**53**	2.17	53	**53**	0.06	**53**
queen3_8	24	148	74	**74**	8.49	74	**74**	2.58	**74**
queen3_9	27	181	98	**98**	15.77	98	**98**	9.37	**98**
queen3_10	30	217	126	**126**	65.91	126	**126**	43.08	**126**
queen4_4	16	76	26	**26**	0.19	26	**26**	0.02	28
queen4_5	20	110	51	**51**	4.54	51	**51**	0.54	53
queen4_6	24	148	83	**83**	16.54	83	**83**	13.25	**83**
queen4_7	28	190	119	**119**	68.22	119	**119**	73.56	121
queen4_8	32	236	164	**164**	636.28	164	**164**	1129.7	167
queen4_9	36	286	209.8	**217**	-	208.5	218	-	222
queen4_10	40	340	255.5	**278**	-	254.72	279	-	286
queen5_5	25	160	93	**93**	41.03	93	**93**	50.06	94
queen5_6	30	215	144	**144**	185.81	144	**144**	315.97	154
queen5_7	35	275	203.1	**214**	-	200.25	215	-	223
queen5_8	40	340	265.8	**293**	-	261.49	294	-	306
queen5_9	45	410	339.8	393	-	335.27	**392**	-	398
queen5_10	50	485	424.9	501	-	422.31	**497**	-	503
queen6_6	36	290	214.9	232	-	210.13	**231**	-	244
queen6_7	42	371	299.2	351	-	292.59	**340**	-	352
queen6_8	48	458	400.7	481	-	394.47	**461**	-	482
queen6_9	54	551	521.4	622	-	512.35	**615**	-	633
queen6_10	60	650	656.7	**786**	-	642.16	792	-	826
queen7_7	49	476	423.7	520	-	418.85	**502**	-	515
queen7_8	56	588	577.6	710	-	566.06	**684**	-	687
queen7_9	63	707	751.8	935	-	733.14	**905**	-	919
queen7_10	70	833	948.5	1177	-	924.79	**1141**	-	1149
queen8_8	64	728	782.1	965	-	767.02	**938**	-	970
queen9_9	81	1056	*	*	*	1289.17	**1641**	-	1664

* Numerical results for the YUC columns were taken directly from Yüceoğlu's Thesis.

for harder instances. We remark that different configurations of BC (e.g., the one using only family of cuts (I4)) delivered the best-known upper bounds for all instances of this class of graphs.

Table 4 reports on DIMACS graphs. The 11 instances above the double horizontal line are those reported on in Yüceoğlu (2015), whereas the others are the remaining graphs in the benchmark set consisting of fewer than 150 vertices. Our results show that instances of the first group are solved orders of magnitude faster by BC and, for those in which YUC was not able to prove optimality, substantially better objective function bounds are obtained. In particular, BC was able to close the

Table 4 Results for DIMACS graphs.*

Instance name	$\lvert V\rvert$	$\lvert E\rvert$	YUC LB	YUC UB	YUC t	BC LB	BC UB	BC t	MDO UB
anna	138	493	47	**47**	1386.04	47	**47**	0.64	**47**
david	87	406	59.5	65	-	64	**64**	1.84	66
games120	120	638	496.4	1626	-	902.48	**1452**	-	1513
huck	74	301	5	**5**	2.92	5	**5**	0.03	9
jean	80	254	16	**16**	6.13	16	**16**	0.06	19
miles250	128	387	45.7	61	-	53	**53**	4.95	61
miles500	128	1170	196.4	447	-	337.68	**404**	-	446
miles750	128	2113	352.1	954	-	471	**471**	812.59	723
myciel3	11	20	10	**10**	0	10	**10**	0	**10**
myciel4	23	71	46	**46**	0.06	46	**46**	0.01	**46**
myciel5	47	236	189.7	197	-	196	**196**	17.04	197
1-FullIns_3	30	100				80	**80**	1.22	**80**
1-FullIns_4	93	593				652.75	**776**	-	839
1-Insertions_4	67	232				294.99	**364**	-	394
2-FullIns_3	52	201				238.65	**248**	-	273
2-Insertions_3	37	72				87.46	**99**	-	103
2-Insertions_4	149	541				599.21	**1585**	-	1588
3-FullIns_3	80	346				427.18	**575**	-	661
3-Insertions_3	56	110				124.96	**191**	-	198
4-FullIns_3	114	541				639.86	**1110**	-	1274
4-Insertions_3	79	156				170.54	**322**	-	331
DSJC125.1	125	736				1714.13	**2599**	-	2618
DSJC125.5	125	3891				2367.78	**3240**	-	**3240**
DSJC125.9	125	6961				583.08	**734**	-	**734**
miles1000	128	3216				535	**535**	286.98	700
miles1500	128	5198				218	**218**	1.13	308
mug100_1	100	166				64	**64**	0.48	91
mug100_25	100	166				64	**64**	0.90	93
mug88_1	88	146				56	**56**	0.33	82
mug88_25	88	146				56	**56**	0.68	84
myciel6	95	755				730.97	**753**	-	**753**
r125.1	125	209				11	**11**	1.53	15
r125.1c	125	7501				207	**207**	33.83	**207**
r125.5	125	3838				1006.78	**1212**	-	1231

* Numerical results for the YUC columns were taken directly from Yüceoğlu's Thesis.

optimality gap of four instances of this dataset that were still open: `david`, `miles250`, `miles750`, and `myciel5`. These results can be explained by the fact that DIMACS graphs do not necessarily have grid-like structures, which makes them more challenging for YUC. For the remaining instances, 9 are solved to optimality and for many of the other instances, the best solutions obtained by BC employed substantially fewer fill edges than those obtained by the traditional heuristic MDO. Namely, BC improved upon the solution obtained by MOD in 26 of the 34 DIMACS graphs.

Table 5 Comparison between MDO and BC on caveman graphs.

α	β	BC %Gap	BC t	BC UB	MDO UB	% Dec	Min % Dec	Max % Dec
4	4	0.0	0.0	0.9	1.4	15.0	0.0	50.0
4	5	0.0	0.0	1.7	2.8	35.11	0.0	75.0
4	6	0.0	0.0	5.3	7.4	42.75	0.0	66.66
4	7	0.0	0.03	13.2	17.3	25.1	8.33	56.25
4	8	0.0	0.07	19.0	26.3	29.27	6.66	64.28
5	4	0.0	0.0	1.6	1.8	6.25	0.0	50.0
5	5	0.0	0.0	1.7	4.4	51.77	12.5	80.0
5	6	0.0	0.01	7.7	10.3	40.3	0.0	66.66
5	7	0.0	0.17	15.5	21.7	35.67	0.0	54.54
5	8	0.0	0.27	25.1	37.1	34.76	13.79	61.9
6	4	0.0	0.0	0.6	1.7	32.14	0.0	71.42
6	5	0.0	0.0	4.1	8.4	54.92	16.66	80.0
6	6	0.0	0.15	12.2	16.9	35.41	0.0	61.53
6	7	0.0	0.35	18.1	24.2	25.28	6.12	50.0
6	8	0.0	1.47	28.9	43.0	35.76	8.82	59.25
7	4	0.0	0.0	1.1	2.5	50.83	0.0	75.0
7	5	0.0	0.0	3.6	6.5	42.85	0.0	83.33
7	6	0.0	0.32	17.1	23.6	35.79	7.31	38.46
7	7	0.0	2.05	24.8	35.4	37.63	9.37	55.81
7	8	0.0	35.53	55.9	74.1	27.61	11.39	70.96
8	4	0.0	0.0	0.3	3.3	70.83	33.33	75.0
8	5	0.0	0.01	5.5	9.9	58.66	0.0	80.0
8	6	0.0	3.96	14.6	23.3	39.92	0.0	92.85
8	7	0.0	3.76	29.7	45.0	37.64	8.0	81.25
8	8	0.0	101.44	52.6	69.8	26.67	13.88	56.81

We now evaluate the solution quality provided by BC on the remaining datasets. Table 5 presents the results for relaxed caveman graphs. The columns indicate the parameters α and β, and, for each configuration, the average optimality gap, solution time, upper and lower bounds provides by BC, as well as the the average upper bound provided by MDO and the average, minimum, and maximum percentage decrease in the upper bound from MDO to BC over each individual instance in that configuration. All instances have been solved to optimality by BC, so all gaps are equal to zero. As MDO runs in under a hundredth of a second in all cases, running times are not reported. The results show the advantage of seeking optimal solutions. The heuristic can be far from the optimum (up to 70% on average for some configurations), showing a trade-off between computational effort and solution quality.

Finally, Table 6 shows the detailed results for RCC-8 graphs. The results are aggregated by the number of vertices of the instances (n), where the first column indicates n and the remaining columns are presented as they are in Table 5. We observe that MDO, which is typically employed

Table 6 Comparison between MDO and BC on RCC-8 graphs.

n	BC %Gap	BC t	BC UB	MDO UB	% Dec	Min % Dec	Max % Dec
25	0.0	0.01	10.6	10.7	1.0	0.0	10.0
50	0.0	16.56	30.4	31.4	3.36	0.0	8.1
75	3.27	1362.64	72.4	77.6	6.97	0.0	30.3
100	7.77	2934.65	101.5	109.1	7.28	0.0	27.84
125	14.08	3600.0	148.6	157.8	5.98	0.79	17.39
150	17.23	3600.0	190.8	202.1	5.59	2.96	9.5

for RCC-8, is highly effective for $n \in \{25, 50\}$. For larger instances with $n \geq 75$, we see that BC can identify substantially better solutions. The reduction in the number of edges in the obtained solution can be as high as 30% and is on average approximately 5%. Finding the solutions requires substantial computational time, but since this application does not require expedience in identifying solutions, using BC is particularly attractive.

10. Conclusion

In this paper we described a new mathematical programming formulation for the MCCP and investigated some key properties of its polytope. The constraints employed in our model correspond to lifted inequalities of induced cycle graphs. Our theoretical results show that this lifting procedure can be generalized to derive other facets of the MCCP polytope of cycle graphs. Finally, we proposed a hybrid solution technique that considers both a lazy-constraint generation and a heuristic separation method based on a threshold rounding, and also presented a simple primal heuristic for the problem. A numerical study indicates that our approach substantially outperforms existing methods, often by orders of magnitude.

References

Agrawal A, Klein P, Ravi R (1993) Cutting down on fill using nested dissection: Provably good elimination orderings. *Graph Theory and Sparse Matrix Computation*, 31–55 (Springer New York).

Beeri C, Fagin R, Maier D, Yannakakis M (1983) On the desirability of acyclic database schemes. *J. ACM* 30(3):479–513.

Bergman D, Raghunathan AU (2015) A Benders approach to the minimum chordal completion problem. *Principles and Practice of Constraint Programming CP 2015*, volume 6308 of *Lecture Notes in Computer Science*, 47–64 (Springer Berlin Heidelberg).

Berry A, Bordat JP, Heggernes P, Simonet G, Villanger Y (2006) A wide-range algorithm for minimal triangulation from an arbitrary ordering. *Journal of Algorithms* 58(1):33 – 66.

Berry A, Heggernes P, Simonet G (2003) The minimum degree heuristic and the minimal triangulation process. Bodlaender H, ed., *Graph-Theoretic Concepts in Computer Science*, volume 2880 of *Lecture Notes in Computer Science*, 58–70 (Springer Berlin Heidelberg).

Bliznets I, Cygan M, Komosa P, Mach L, Pilipczuk M (2016) Lower bounds for the parameterized complexity of minimum fill-in and other completion problems. *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, 1132–1151.

Bouchitté V, Todinca I (2001) Treewidth and minimum fill-in: Grouping the minimal separators. *SIAM Journal on Computing* 31(1):212–232.

Cao Y, Marx D (2016) Chordal editing is fixed-parameter tractable. *Algorithmica* 75(1):118–137.

Cao Y, Sandeep RB (2016) Minimum fill-in: Inapproximability and almost tight lower bounds. *Proc. of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, 875–880.

Chang MS (1996) Algorithms for maximum matching and minimum fill-in on chordal bipartite graphs. Asano T, Igarashi Y, Nagamochi H, Miyano S, Suri S, eds., *Algorithms and Computation*, volume 1178 of *Lecture Notes in Computer Science*, 146–155 (Springer Berlin Heidelberg).

Chung F, Mumford D (1994) Chordal completions of planar graphs. *Journal of Combinatorial Theory, Series B* 62(1):96 – 106.

Codato G, Fischetti M (2006) Combinatorial Benders' cuts for mixed-integer linear programming. *Operations Research* 54(4):756–766.

Feremans C, Oswald M, Reinelt G (2002) A y-formulation for the treewidth. Technical report, Heidelberg University.

Fomin FV, Philip G, Villanger Y (2013) Minimum fill-in of sparse graphs: Kernelization and approximation. *Algorithmica* 71(1):1–20.

Fomin FV, Villanger Y (2012) Subexponential parameterized algorithm for minimum fill-in. *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms*, 1737–1746 (SIAM).

Fulkerson DR, Gross OA (1965) Incidence matrices and interval graphs. *Pacific J. Math.* 15(3):835–855.

Garey MR, Johnson DS (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness* (New York, NY, USA: W. H. Freeman & Co.), ISBN 0716710447.

George A, Liu WH (1989) The evolution of the minimum degree ordering algorithm. *SIAM Rev.* 31(1):1–19.

Grone R, Johnson CR, S EM, Wolkowicz H (1984) Positive definite completions of partial hermitian matrices. *Linear Algebra and its Applications* 58(0):109 – 124.

Heggernes P (2006) Minimal triangulations of graphs: A survey. *Discrete Mathematics* 306(3):297 – 317.

IBM ILOG (2017) CPLEX Optimization Studio 12.7.1 user manual.

Judd S, Kearns M, Vorobeychik Y (2011) Behavioral conflict and fairness in social networks. *Internet and Network Economics*, volume 7090 of *Lecture Notes in Computer Science*, 242–253 (Springer).

Kaplan H, Shamir R, Tarjan RE (1999) Tractability of parameterized completion problems on chordal, strongly chordal, and proper interval graphs. *SIAM Journal on Computing* 28(5):1906–1922.

Kim S, Kojima M, Mevissen M, Yamashita M (2011) Exploiting sparsity in linear and nonlinear matrix inequalities via positive semidefinite matrix completion. *Math. Program.* 129(1):33–68.

Kloks T, Bodlaender H, Müller H, Kratsch D (1993) *Computing treewidth and minimum fill-in: All you need are the minimal separators*, 260–271 (Berlin, Heidelberg: Springer Berlin Heidelberg).

Kloks T, Kratsch D, Müller H (1999) Approximating the bandwidth for asteroidal triple-free graphs. *Journal of Algorithms* 32(1):41 – 57.

Kloks T, Kratsch D, Wong C (1998) Minimum fill-in on circle and circular-arc graphs. *Journal of Algorithms* 28(2):272 – 289.

Kratsch D, Müller H (2009) On a property of minimal triangulations. *Discrete Math.* 309(6):1724 – 1729.

Lauritzen SL, Spiegelhalter DJ (1990) Local computations with probabilities on graphical structures and their application to expert systems. Shafer G, Pearl J, eds., *Readings in Uncertain Reasoning*, 415–448 (San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.).

Mezzini M, Moscarini M (2010) Simple algorithms for minimal triangulation of a graph and backward selection of a decomposable Markov network. *Theoretical Computer Science* 411(79):958 – 966.

Nakata K, Fujisawa K, Fukuda M, Kojima M, Murota K (2003) Exploiting sparsity in semidefinite programming via matrix completion II: implementation and numerical results. *Math. Program.* 95(2):303–327.

Natanzon A, Shamir R, Sharan R (2000) A polynomial approximation algorithm for the minimum fill-in problem. *SIAM Journal on Computing* 30(4):1067–1079.

Nikolaou C, Koubarakis M (2014) Fast consistency checking of very large real-world RCC-8 constraint networks using graph partitioning. *Proc. of Twenty-Eighth AAAI Conf. on A.I.*, 2724–2730.

Nikolopoulos SD, Palios L (2007) Detecting holes and antiholes in graphs. *Algorithmica* 47(2):119–138.

Ohtsuki T (1976) A fast algorithm for finding an optimal ordering for vertex elimination on a graph. *SIAM Journal on Computing* 5(1):133–145.

Ohtsuki T, Cheung LK, Fujisawa T (1976) Minimal triangulation of a graph and optimal pivoting order in a sparse matrix. *Journal of Mathematical Analysis and Applications* 54(3):622 – 633.

Parra A, Scheffler P (1995) How to use the minimal separators of a graph for its chordal triangulation. *Automata, Languages and Programming: 22nd International Colloquium*, 123–134 (Berlin, Heidelberg).

Parra A, Scheffler P (1997) Characterizations and algorithmic applications of chordal graph embeddings. *Discrete Applied Mathematics* 79(1):171 – 188.

Parter S (1961) The use of linear graphs in gauss elimination. *SIAM Review* 3(2):119–130.

Renz J (2002) The region connection calculus. *Qualitative Spatial Reasoning with Topological Information*, 41–50 (Berlin, Heidelberg: Springer Berlin Heidelberg).

Rollon E, Larrosa J (2011) On mini-buckets and the min-fill elimination ordering. *Principles and Practice of Constraint Programming (CP)*, 759–773 (Berlin, Heidelberg: Springer Berlin Heidelberg).

Rose DJ (1973) A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations. *Graph Theory and Computing* 183–217.

Rose DJ, Tarjan RE (1978) Algorithmic aspects of vertex elimination on directed graphs. *SIAM Journal on Applied Mathematics* 34(1):176–197.

Rose DJ, Tarjan RE, Lueker GS (1976) Algorithmic aspects of vertex elimination on graphs. *SIAM Journal on Computing* 5(2):266–283.

Rostami B, Malucelli F, Frey D, Buchheim C (2015) On the quadratic shortest path problem. *International Symposium on Experimental Algorithms*, 379–390 (Springer).

Sioutis M (2014) Triangulation versus graph partitioning for tackling large real world qualitative spatial networks. *2014 IEEE 26th International Conference on Tools with Artificial Intelligence*, 194–201.

Sioutis M, Koubarakis M (2012) Consistency of chordal rcc-8 networks. *2012 IEEE 24th International Conference on Tools with Artificial Intelligence*, volume 1, 436–443.

Tarjan RE, Yannakakis M (1984) Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM J. Comput.* 13(3):566–579.

Uno T, Satoh H (2014) An efficient algorithm for enumerating chordless cycles and chordless paths. *International Conference on Discovery Science*, 313–324 (Springer).

Vandenberghe L, Andersen MS (2015) Chordal graphs and semidefinite optimization. *Foundations and Trends in Optimization* 1(4):241–433.

Yannakakis M (1981) Computing the minimum fill-in is NP-Complete. *SIAM J. Alg. Discr. Meth.* 2(1):77–79.

Yüceoğlu B (2015) *Branch-and-cut algorithms for graph problems.* Ph.D. thesis, Maastricht University.

Zhang C, Zhao T, Li W (2015) *Ontology Data Query in Geospatial Semantic Web*, chapter 3, 57–87 (Cham: Springer International Publishing).

a *b*

Figure EC.1 The graph (a) $G(\tilde{x}^i)$ and the graph (b) $G(\hat{x}^i)$ defined in the proof of Proposition 2.

Online Supplement—Proofs of Statements

EC.1. Additional Proofs for Section 6

Facet-defining proof of Proposition 2. Let $F^I = \left\{x \in X(G) : \sum_{f \in \iota(C)} x_f = |C| - 3\right\}$ and $\mu x \geq \mu_0$ be a valid inequality for $\text{conv}(X(G))$ that is satisfied at equality by each $x \in F^I$. It suffices to show that there exists some λ for which $\mu_f = \lambda$ and $\mu_0 = (|C| - 3)\lambda$.

Let $x' \in \{0,1\}^{m^c}$ be such that $x'_f = 1$ if $f = \{v_0, v_j\}, j = 2, \ldots, k-2$, and $x'_f = 0$ otherwise. We claim that $x' \in X(G)$, i.e., $G(x')$ is chordal, and $x' \in F^I$. First, let $\bar{V}_j = \{v_0, v_1, \ldots, v_j\}$ for every $j \in [2, k-1]$. By construction, set $N_{G[\bar{V}_j]}(v_j) = \{v_0, v_{j-1}\}$ induces a clique in $G[\bar{V}_j]$. Therefore, $v_0, v_1, \ldots, v_{k-1}$ is a perfect elimination ordering of $V(G(x'))$, thereby proving that $G(x')$ is chordal. Additionally, since exactly $|C| - 3$ edges in $\iota(C)$ are in $G(x'), x' \in F^I$.

Consider now the solutions $\tilde{x}^i \in \{0,1\}^{m^c}$ for $i = 3, \ldots, k-1$, such that $\tilde{x}^i_f = 1$ if $f = \{v_1, v_j\}, j = 3, \ldots, i$, or if $f = \{v_0, v_j\}, j = i, i+1, \ldots, k-2$; otherwise, $\tilde{x}^i_f = 0$ (see Figure EC.1 (a) for an example of $G(\tilde{x}^i)$). For every $j \in [2, k-1]$, let $\bar{V}_j$ be the set of vertices belonging to the subsequence of $(v_1, \ldots, v_i, v_0, v_{i+1}, \ldots, v_{k-1})$ finishing at element v_j. By construction, $N_{G[\bar{V}_j]}(v_j)$ is given by $\{v_1, v_i\}$ if $j = 0$, $\{v_1\}$ if $j = 2$, $\{v_1, v_{j-1}\}$ if $3 \leq j \leq i$, and $\{v_0, v_{j-1}\}$ if $i+1 \leq j \leq k-1$, which in each case is a clique. Therefore, $v_1, \ldots, v_i, v_0, v_{i+1}, \ldots, v_{k-1}$ is a perfect elimination ordering of $V(G(\tilde{x}^i))$, thus showing that $G(\tilde{x}^i)$ is chordal. Moreover, exactly $|C| - 3$ edges of $\iota(C)$ are in $G(\tilde{x}^i)$, so $\tilde{x}^i \in F^I$.

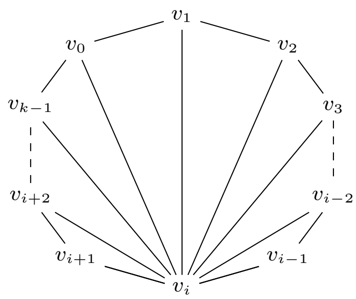
Let $\lambda_2 = \mu_{\{v_0, v_2\}}$. Solutions $\tilde{x}^i$ and x' belong to F_I, so $\mu\tilde{x}^i = \mu x' = \mu_0$. By subtracting equation $\mu\tilde{x}^3 = \mu_0$ from $\mu x' = \mu_0$, we obtain $\mu_{\{v_0, v_2\}} = \mu_{\{v_1, v_3\}}$. Additionally, shift operations on the order of the vertices (to the left or to the right) lead to the same cycle C. Therefore, $\mu_{\{v_0,v_2\}} = \mu_{\{v_j,v_{(j+2) \mod k}\}}$ for any $j \in [k-1]$, implying thus that $\mu_{\{v_0,v_2\}} = \lambda_2$ for every $f \in \iota(C)$ containing vertices whose indices in C differ by 2. The same operation involving $\tilde{x}^{i-1}$ and $\tilde{x}^i$ for $i = 4,\dots,k-1$ yields $\mu_{\{v_0,v_{i-1}\}} = \mu_{\{v_1,v_i\}}$. Again, as the ordering around C can be arbitrarily shifted to the left and to the right, all edges $f \in \iota(C)$ containing vertices whose indices in C differ by $i-1$ have the same coefficient in μ; let λ_{i-1} be this common value. We thus conclude that $\mu_{\{v_j,v_{j'}\}} = \lambda_{j'-j}$ for any $f = \{v_j, v_{j'}\}$ (assuming $j < j'$).

Consider now the solutions $\hat{x}^i \in \{0,1\}^{m^c}$ for $i = 2,\dots,k-2$, where $\hat{x}^i_f = 1$ if $f = \{v_{i-1}, v_{i+1}\}$ or if $f = \{v_0, v_j\}, j = 2,\dots,i-1,i+1,\dots,k-2$ and $\hat{x}^i_f = 0$ otherwise (see Figure EC.1 (b) for an example of $G(\hat{x^i})$.). For every $j \in [2,k-1]$, let $\bar{V}_j$ be the set of vertices belonging to the subsequence of $(v_0, v_1,\dots,v_{i-1},v_{i+1},\dots,v_{k-1},v_i)$ finishing at element v_j. By construction, we have that $N_{G[\bar{V}_j]}(v_j)$ is equal to $\{v_0\}$ if $j = 2$, $\{v_0, v_{j-1}\}$ if $1 \le j \le i-1$, and $\{v_{i-1}, v_{i+1}\}$ if $j = i$, which, in each case, is a clique. Consequently, $(v_0, v_1,\dots,v_{i-1},v_{i+1},\dots,v_{k-1},v_i)$ is a perfect elimination ordering, so $\hat{x}^i \in X(G)$. Finally, as $|C|-3$ edges from $\iota(C)$ are included in $G(\hat{x}^i), \hat{x}^i \in F^I$.

By subtracting $\mu x' = \mu_0$ from $\mu\hat{x}^i = \mu_0$ for any $i = 2,\dots,k-2$, we obtain $\mu_{\{v_0,v_i\}} = \mu_{\{v_{i-1},v_{i+1}\}}$. Therefore, we have that $\lambda_i = \lambda_2$ for any $i = 2,\dots,k-2$. If $\lambda = \lambda_2$, $\mu x = \mu_0$ can be rewritten $\sum_{f\in\iota(C)} \lambda x_f = \mu_0$. Finally, substituting x' in this equation yields $\mu_0 = (|C|-3)\,\lambda$, as desired. □

Facet-defining proof of Proposition 3. Let $I := ax \ge b$ be the inequality of type (I2) associated with $i = 1$, F^I be the set of points in $\text{conv}(X(G))$ that satisfy I at equality, and $\mu x \ge \mu_0$ be a valid inequality for $\text{conv}(X(G))$ satisfied at equality for each $x \in F^I$. Let $\mu_{v_0,v_2} = \lambda$.

For $i \in \{3,4,\dots,k-1\}$, let $\check{x}^i$ be such that $\check{x}^i_f = 1$ if $f = \{v_i, v_j\}, j \in [k-1]\backslash\{i-1,i,i+1\}$, and $\check{x}^i_f = 0$ otherwise, and let $\check{y}^i$ be such that $\check{y}^i_{\{v_1,v_i\}} = 0$, $\check{y}^i_{\{v_0,v_2\}} = 1$, and $\check{y}^i_f = \check{x}^i_f$ for the remaining edges in $E(G)^c$; both families of solutions are depicted in Figure EC.2. We have that both $\check{x}^i$ and $\check{y}^i$ belong to $X(G)$, as $G(\check{x}^i)$ is isomorphic to $G(x')$ and $\check{y}^i$ is isomorphic to $G(\hat{x}^{i-1})$; note that $G(\check{x}^i)$ and $G(\hat{x}^{i-1})$ were defined and shown to be associated with chordal completions in the proof of Proposition 2. Additionally, note that $a_f\check{x}^i_f = 1$ only for $f = \{v_1, v_i\}$ and $a_f\check{y}^i_f = 1$ only for



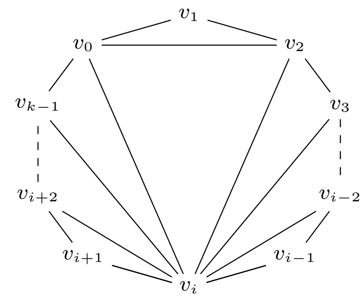


Figure EC.2 The graph (a) $G(\breve{x}^i)$ and the graph (b) $G(\breve{y}^i)$ defined in the proof of Proposition 3.

$f = \{v_0, v_2\}$, so both solutions satisfy I at equality and, by definition, $\mu\breve{x}^i = \mu_0 = \mu\breve{y}^i$. Therefore, we must have $\mu_{\{v_1,v_i\}} = \mu_{\{v_0,v_2\}} = \lambda$ for $i \in \{3,4,\ldots,k-1\}$.

We claim now that $\mu_{f'} = 0$ if $f' \in V_{0,2} = \iota(C)\setminus\left(\{v_0,v_2\} \cup \bigcup_{i=3,\ldots,k-1}\{v_1,v_i\}\right)$. First, note that $V_{0,2} \neq \emptyset$ only if $k \geq 5$. Let $f' = \{v_j, v_{j'}\}$ be such an edge, and assume without loss of generality that $v_j \neq v_0$ (i.e., $v_{j'}$ can be equal to v_2). Let $z(f')$ be the solution in $\{0,1\}^{m^c}$ presented in Figure EC.3 (part a) defined by

$$z(f')_f = \begin{cases} 1, & f = \{v_i, v_{j'}\}, k = 0,\ 2 \leq i \leq j'-2, \text{ and } j+1 \leq i \leq k-1, \\ 1, & f = \{v_{j+1}, v_{j'+1}\}, \\ 1, & f = \{v_j, v_i\}, j'+1 \leq i \leq j-2, \\ 0, & \text{otherwise.} \end{cases}$$

Solution $z(f')$ satisfies I at equality, as $a_f z(f')_f = 1$ for $f = \{v_1, v_{j'}\}$ and $a_f z(f')_f = 0$ for all the other edges in $G(z(f'))$. Moreover, $G(z(f'))$ is isomorphic to the graph presented in Figure EC.1 (part a), so $z(f') \in X(G)$.

Let now $z'(f')$ be the solution in $\{0,1\}^{m^c}$ such that $z'(f')_{f'} = 1$ and $z'(f')_f = z(f')_f$ for the remaining edges; this solution is presented in Figure EC.3 (part b). The same argument used for $z(f')$ shows that $z'(f')$ satisfies I at equality, and sequence $(v_j, v_{j'}, v_{j+1}, v_{j+2}, \ldots, v_{k-1}, v_0, v_1, \ldots, v_{j'-1}, v_{j'+1}, \ldots, v_{j-1})$ is a perfect elimination ordering of $V(G)$ for $G(z'(f'))$, which shows that $z'(f') \in X(G)$. Finally, because $\mu z'(f') = \mu_0 = \mu z(f')$, it follows that $\mu_{f'} = 0$, as desired.

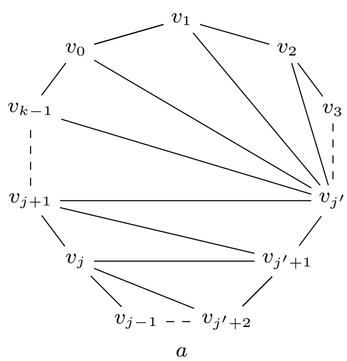


Figure EC.3 The graph (a) $G(z(f'))$ and the graph (b) $G(z'(f'))$ defined in the proof of Proposition 3.

Direct inspection on any solution $\breve{x}$ (e.g., $\breve{x}^3$) allows us to see that $\mu_0 = \lambda$. Therefore, there exists a λ such that $\forall f \in E^c, \mu_f = \lambda a_f$ and $\mu_0 = b\lambda$, completing the proof that I is facet defining. □

Facet-defining proof of Proposition 4. Let $I := ax \geq b$ be any inequality (I3). Moreover, let F^I be the set of points in $\text{conv}(X(G))$ that satisfy I at equality and $\mu x \geq \mu_0$ be a valid inequality for $\text{conv}(X(G))$ satisfied at equality for each $x \in F^I$. Let $\lambda = \mu_{\{v_0,v_2\}}$.

We claim that $\exists \lambda \neq 0$ such that $\forall f \in \iota(C)$ with $d_C(f) = 2, \mu_f = \lambda$. Consider solutions x' and $\tilde{x}^3$ presented in the proof of Proposition 2; graph $G(x')$ is isomorphic to the one shown in Figure EC.2 (a) and has v_0 as the neighbor of all vertices in $V(G)$, whereas graph $G(\tilde{x}^3)$ is shown in Figure EC.1 (a). By construction, both solutions are in F^I. Subtracting $\mu x' = \mu_0$ from $\mu \tilde{x}^3 = \mu_0$ and canceling like terms yields $\mu_{\{v_0,v_2\}} = \mu_{\{v_1,v_3\}}$. By sequentially applying this procedure starting from any fill edge of C with $d_C(v_i, v_j) = 2$, we obtain the desired result.

Now, we show that $\mu_f = 0$ for every f in $\iota(C)$ such that $d_C(f) \geq 3$. Let $\tilde{y}^i$ be the set of solutions given by $\tilde{y}^i = \tilde{x}^i + e^{\{v_1,v_{i+1}\}}$, $3 \leq i \leq k-3$, with $\tilde{x}$ being again the solutions defined in the proof of Proposition 2. Each solution $\tilde{y}^i$ satisfies I at equality. Moreover, $(v_1, v_2, \ldots, v_i, v_0, v_{i+1}, v_{i+2}, \ldots, v_{k-1})$ is a perfect elimination ordering for each $G(\tilde{y}^i)$, thus showing that each $\tilde{y}^i$ is a valid solution. Therefore, $\mu \tilde{y}^i = \mu_0 = \mu \tilde{x}^i = \mu_0$, and, as $\tilde{y}^i$ and $\tilde{x}^i$ only differ on the coordinate corresponding to fill edge $\{v_1, v_{i+1}\}$, we must have $\mu_{\{v_1,v_{i+1}\}} = 0$ for any edge index $i, 3 \leq i \leq k-2$. This implies, due to cyclic symmetry, that $\mu_f = 0$ for any edge $f \in \iota(C)$ such that $d_C(f) \geq 3$.

Finally, we have $\mu_0 = \mu x' = \mu_{\{v_0,v_2\}} + \mu_{\{v_0,v_2\}} = 2\lambda$. Therefore, there exists a λ such that $\mu_0 = b\lambda$ and $\mu_f = \lambda a_f$ for every f in E^c, which shows that I is facet defining. □

Facet-defining proof of Proposition 5. Let $I := ax \geq b$ be any inequality (I4). Without loss of generality, let $j = 0$ and i be any value in $[2, k-3]$. Let F^I be the set of points in $\text{conv}(X(G))$ that satisfy I at equality, and let $\mu x \geq \mu_0$ be a valid inequality for $\text{conv}(X(G))$ satisfied at equality for each $x \in F^I$.

Consider solutions x' and $\hat{x}^i$ presented in the proof of Proposition 2. Direct inspection allows us to see that both belong to F^I and differ only on coordinates $\sigma(\{v_0, v_i\})$ and $\sigma(\{v_{i-1}, v_{i+1}\})$, so that $\mu x' = \mu_0 = \mu \hat{x}^i$ implies that $\mu_{\{v_0, v_i\}} = \mu_{\{v_{i-1}, v_{i+1}\}}$. Additionally, the solution $x' + e^{\left\{v_{i-1}, v_{i+1}\right\}}$ also belongs to F^I: it satisfies I at equality and the sequence $(v_0, v_1, \ldots, v_{k-1})$ is a perfect elimination order of $V(G)$. Therefore, as $\mu\left(x' + e^{\left\{v_{i-1}, v_{i+1}\right\}}\right) = \mu x' + \mu_{\left\{v_{i-1}, v_{i+1}\right\}} = \mu_0$, it follows that $\mu_{\{v_0, v_i\}} = \mu_{\left\{v_{i-1}, v_{i+1}\right\}} = 0$.

Let $C' = (v_0, v_1, \ldots, v_{i-1}, v_{i+1}, v_{i+2}, \ldots, v_{k-1})$ and let $\lambda = \mu_{v_1, v_{k-1}}$. We have that $\mu_f = \lambda$ for each $f \in \iota(C')$. Let x be any feasible solution of $X(C' + \{\{v_{i-1}, v_{i+1}\}\})$ satisfying inequality (I1) at equality. Let y be a solution such that $y_f = x_f$ for $f \in \text{int}(C')$, $x_f = 1$ if $f = \{v_{i-1}, v_{i+1}\}$, and $x_f = 0$ otherwise. Solution y belongs to $X(G)$ because any perfect elimination order of $V(C')$ can be extended into a perfect elimination order for $V(C)$ by putting v_i in the end of the sequence (note that the only neighbors of v_i are v_{i-1} and v_{i+1}, which are necessarily connected). Moreover, by construction, $\sum_{f \in \text{int}(C')} y_f = |C'| - 3 = |C| - 4$, so $y \in F^I$. Finally, as $\sum_{f \in \iota(C) \setminus \{\{v_{i-1}, v_{i+1}\}, \{v_j, v_i\}\}} y_f = \sum_{f \in \iota(C')} y_f$ for all $j \in [k-1] \setminus \{i\}$, it follows from the arguments used in the proof of Proposition 2 that $\mu_f = \lambda$ for each $f \in \iota(C')$.

Now, we show that $\mu_f = \lambda$ for each $f = \{v_i, v_\ell\}, \ell = 1, 2, \ldots, i-2, i+2, i+3, \ldots, k-2$. For an arbitrary value of ℓ', let $\bar{x}$ be such that $\bar{x}_f = 1$ if $v_i \in f$ and $\bar{x}_f = 0$ otherwise. $G(\bar{x})$ is isomorphic to $G(x')$ for the solution x' defined in the proof of Proposition 2, so $\bar{x}$ is feasible. Moreover, $\bar{x}_f = 1$ for $|C| - 4$ edges in $\iota(C) \setminus \{\{v_{i-1}, v_{i+1}\}, \{v_0, v_i\}\}$, so we have that $\bar{x} \in F^I$. Let $\bar{x}^{\ell'}$ be the solution of $X(G)$ such that $\bar{x}^{\ell'}_{\{v_{\ell'-1}, v_{\ell'+1}\}} = 1$, $\bar{x}^{\ell'}_{\{v_i, v_{\ell'}\}} = 0$, and $\bar{x}^{\ell'}_f = \bar{x}_f$ for the remaining edges. The graph $G(\bar{x}^{\ell'})$ is isomorphic to one of the graphs $G(\hat{x}^{i'})$ defined in the proof of Proposition 2, and therefore $\bar{x}^{\ell'} \in X(G)$. Moreover, $\bar{x}^{\ell'}_f = 1$ for $|C| - 4$ edges in $\iota(C) \setminus \{\{v_{i-1}, v_{i+1}\}, \{v_i, v_{l'}\}\}$, so we have that

$\bar{x}^{\ell'} \in F^I$. Finally, we have $\mu\bar{x} = \mu_0$ and $\mu\bar{x}^{\ell'} = \mu_0$, and the subtraction of these two equalities yields $\mu_{\{v_{\ell'-1},v_{\ell'+1}\}} = \mu_{\{v_i,v_{\ell'}\}}$. Since $\{v_{\ell'-1}, v_{\ell'+1}\} \in \text{int}(C'), \lambda = \mu_{\{v_{\ell'-1},v_{\ell'+1}\}}$ and $\mu_{\{v_i,v_{\ell'}\}} = \lambda$.

We conclude thus that any solution in F^I yields $\mu_0 = \lambda(|C| - 4)$, as desired. □

EC.2. Additional Proofs for Section 7

Lemma EC.1. *If $G = (V, E)$ is chordal, then $G' = (V \cup w, E \cup \{(w, v) : \forall\, v \in V\}$ is also chordal.*

Proof. Suppose by contradiction that $C = (v_0, v_1, \ldots, v_{k-1}), k \geq 4$, is a chordless cycle in G'. As G does not contain chordless cycles, $V(C)$ cannot be contained in $V(G)$, so $w \in V(C)$. By construction, w is adjacent to all vertices in $V(G)$ and, in particular, to all vertices in $V(C)$, thus contradicting the hypothesis that C is chordless. □

Corollary EC.1. *If $G = (V, E)$ is a chordal graph, then the graph $G' = (V \cup W, E \cup \{(w, v) : \forall w \in W,\, v \in V \cup W \setminus \{w\}\}$ is also chordal.*

Proof. Lemma EC.1 can be extended to cliques as opposed to single vertices, since this addition can be seen as a inductively adding a single vertex one-by-one. □

Definition EC.1. An edge e is said to be *critical* in a chordal graph $G = (V, E)$ if $G' = (V, E \setminus e)$ is not chordal (i.e., if the removal of e from G creates a chordless cycle).

Lemma EC.2. *Let $G = (V, E)$ be a chordal graph. If $e = \{v, w\}$ is critical, then any chordless cycle C emerging after the deletion of e is such that $\{v, w\} \subset V(C)$ and $|C| = 4$.*

Proof. Let C be a chordless cycle emerging after the deletion of e. If either v or w does not belong to $V(C)$, then C is also a chordless cycle in G, a contradiction. Suppose $|C| > 4$. In this case, C can be written as a sequence $v \sim P_1 \sim w \sim P_2$, where P_1 and P_2 are paths in G such that $V(P_1) \cap V(P_2) = \emptyset$ and $v, w \notin V(P_1) \cup V(P_2)$. Moreover, as $|C| > 4$, at least one of P_1, P_2 contains 2 or more vertices. If $|P_1| > 1$ ($|P_2| > 1$), then the sequence described by path $v \sim P_1 \sim w$ ($w \sim P_2 \sim v$) induces a chordless cycle in G, thereby contradicting the assumption that G is chordal. □

Proof of Theorem 3. This follows directly from Theorem EC.1, presented next. □

Theorem EC.1. *Let $G=(V,E)$ and $E'\subseteq E^c$ be such that $G\cup E'$ is not chordal and $G\cup E'\backslash\{f\}$ is chordal for every $f\in E'$. If $ax\geq b$ is facet defining for* $\text{conv}(X(G\cup E')), a\geq 0$, *and $a'\in\mathbb{R}^{|E^c(G)|}$, with $a'_f=a_f$ if $f\in E^c\backslash E'$ and $a'_f=0$ otherwise, the inequality*

$$a'x\geq b\left(\sum_{f\in F_G(C)} x_f-|E'|+1\right)$$

is facet defining for $\text{conv}(X(G))$.

Proof of Theorem EC.1 Let $ax\geq b$ be a facet-defining inequality for $\text{conv}(X(G\cup E'))$ and I be the corresponding lifted inequality $a'x\geq b\left(\sum_{f\in E'}x_f-|E'|+1\right)$ for $\text{conv}(X(G))$.

First, we show that I is valid for $\text{conv}(X(G))$. Since $a\geq 0$, I can only be violated by a feasible element x of $\text{conv}(X(G))$ if $\sum_{f\in E'}x_f=|E'|$; otherwise, I is trivially satisfied. Moreover, because $ax'\geq b$ is valid for every $x'\in\text{conv}(X(G\cup E'))$, we have

$$a'x=\sum_{f\in E^c\backslash E'} a_f x_f\geq b=b\left(\sum_{f\in E'}x_f-|E'|+1\right),$$

as desired. Now we present a set of $|E^c|$ affinely independent vectors of $\text{conv}(X(G))$ satisfying I at equality. For any facet-defining inequality $ax\geq b$ of $\text{conv}(X(G\cup E'))$, there exists an affinely independent set of vectors $W=\{w^j\}_{j=1}^d\subseteq\{0,1\}^{|E^c\backslash E'|}$ that satisfy $ax=b$. Let $T=\{t^j\}_{j=1}^d\subseteq\{0,1\}^{|E^c|}$ be such that $t^j_f=w^j_f$ for $f\in E^c\setminus E'$ and $t^j_f=1$ for $f\in E'$. That is, t^j is an embedding of w^j in $\{0,1\}^{|E^c|}$ in which coordinates associated with edges in E' are set to 1. Note that every t^j belongs to $\text{conv}(X(G))$ because $G(t^j)$ is isomorphic to $(G\cup E')(w^j)$, which is chordal. Moreover, by construction, $a't^j=\sum_{f\in E^c\backslash E'}a'_f t^j_f=b$ and $\sum_{f\in E'}t^j_f=|E'|$ for each $j=1,\dots,d$, so solutions of T satisfy I at equality. Finally, note that the embedding operation in the elements of W is such that T is also affinely independent.

Let $Z=\{z^f\}_{f\in E'}\subseteq\{0,1\}^{|E^c|}$ be such that $z^f_{f'}=1$ for $f'\in E'\setminus f$ and $z^f_{f'}=0$ otherwise. As $G\cup E'\setminus\{f\}$ is chordal by hypothesis, it follows that each solution z^f belongs to $\text{conv}(X(G))$. Moreover, by construction, $a'z^f=\sum_{f'\in E^c\backslash E'}a'_{f'}t^j_{f'}=0$ and $\sum_{f'\in E'}z^f_{f'}=|E'|-1$ for each $f\in E'$, so each solution of Z satisfies I at equality. Let $\alpha_f, f\in E^c\setminus E'$, and $\beta_f, f\in E'$, be constants for which

$$\sum_{f\in E^c\backslash E'}\alpha_j t^j+\sum_{f\in E'}\beta_f z^f=0,\qquad \sum_{f\in E^c\backslash E'}\alpha_j+\sum_{f\in E'}\beta_f=0.$$

For each $f \in E'$, we have $z_f^f = 0$, whereas $s_f = 1$ for $s \in T \cup Z \setminus \{z^f\}$. Therefore, we must have $\sum_{f' \in E^c \setminus E'} \alpha_{f'} + \sum_{f' \in E' \setminus \{f\}} \beta_{f'} = 0$ and, as a consequence, $\beta_f = 0$ for each $f \in E'$. Finally, as T is affinely independent, we have that $\alpha_f = 0$, $f \in E^c \setminus E'$. It follows that I is a faced-defining inequality for conv(X), as desired. □

Proof of Theorem 4. Without loss of generality, let $f^* = \{v_0, v_{t-1}\}, t < k$, be the chord considered and I be the associated lifted inequality $a'x - bx_{f^*} \geq 0$ for $X(G)$. For any vector $x \in \{0,1\}^{|E^c|}$ and set $E' \subseteq E^c$, let $x[E']$ be the projection of x onto the coordinates corresponding to fill edges in E'. First, we claim that I is valid for $X(G)$. Take any solution $x^0 \in X(G)$. If $x^0_{f^*} = 0$, then I reduces to $a'x^0 \geq 0$, which must be satisfied because $a' \geq 0$ and $x^0 \geq 0$. If $x^0_{f^*} = 1$, then I reduces to $a'x \geq b$. Since $G(x^0)$ is chordal, by Lemma 2 we have that $G(x^0)[C']$ is also chordal, and therefore $x^0[\iota(C')] \in X(G')$. Since I is facet-defining for conv$(X(G'))$, we have $a'x^0 = ax^0[\iota(C')] \geq b$. As x^0 was chosen arbitrarily among all feasible solutions in $X(G)$, it follows that I is valid for $X(G)$.

Let $C' = (v_0, v_1, \ldots, v_{t-1})\,, C'' = (v_0, v_{t-1}, v_t, \ldots, v_{k-1})$, and $\text{Cross}(f^*) = \{f : f \cap \{v_1, v_2, \ldots, v_{t-2}\} \neq \emptyset, f \cap \{v_t, v_{t+1}, \ldots, v_{k-1}\} \neq \emptyset\}$, that is, Cross$(f^*)$ contains all fill edges in $\iota(C)$ containing exactly one vertex incident in $C' \setminus C''$ and one vertex incident in $C'' \setminus C'$. Set $\iota(C)$ can therefore be partitioned as $\iota(C) = \iota(C') \,\dot\cup\, \iota(C'') \,\dot\cup\, f^* \,\dot\cup\, \text{Cross}(f^*)$. Let F^I be the set of points in conv$(X(G))$ that satisfy I at equality, and $\mu x \geq \mu_0$ be a valid inequality for conv$(X(G))$ satisfied at equality by each $x \in F^I$. Inequality $\mu x \geq \mu_0$ can be written as

$$\sum_{f \in \text{int}(C')} \mu_f x_f + \sum_{f \in \text{int}(C'')} \mu_f x_f + \mu_{f^*} x_{f^*} + \sum_{f \in \text{Cross}(f^*)} \mu_f x_f \geq \mu_0$$

Claim EC.1. *For every f in* Cross$(f^*), \mu_f = 0$.

Proof. Take any vector $\tilde{w}^b$ in $X(G')$ such that $a\tilde{w}^b = b$. Moreover, let us assume that the fill-in set associated with $\tilde{w}^b$ is minimal; note that if $\tilde{w}^b$ does not satisfy this condition, then it can be substituted for some other feasible solution w', $aw' = a\tilde{w}^b = b$, associated with a subset of the fill-in edges represented by $\tilde{w}^b$.

From Proposition 4, it follows that $G'(\tilde{w}^b)$ must contain an edge $\{v_{b-1}, v_{b+1}\}$. Moreover, from Lemma 2, we have that $w' = \tilde{w}^b[E^c(G') \setminus \{\{v_a, v_b\} : v_a \in V(G')\}]$ is associated with a chordal completion of $G'[V(G) \setminus v_b]$. Because v_{b-1} and v_{b+1} are the only neighbours of v_b in G', the edges of w'

are sufficient to make G' chordal; therefore, we have that the neighbours of v_b in $G'[\tilde{w}^b]$ are exactly its neighbours in C'.

Fix $f' \in \text{Cross}(f^*), f' = \{v_a, v_b\}, 1 \leq a \leq t-2, t \leq b \leq k-1$. Let $\tilde{w}$ in $X(G)$ be such that $\tilde{w}_f = \tilde{w}^b_f$ if $f \in \iota(C')$, $\tilde{w}_f = 1$ if $f \in \{f^*\} \cup \iota(C'') \cup \text{Cross}(f^*) \backslash \{v_a, v_b\}$, and $\tilde{w}_f = 0$ if $f = \{v_a, v_b\}$. By Lemma EC.1, $G\left(\tilde{w} + e^{\{v_a,v_b\}}\right)$ is a chordal graph. We claim that $\{v_a, v_b\}$ cannot be critical, and therefore $G\left(T^1\left(\tilde{w}^b\right)\right)$ is chordal. Suppose by contradiction that this is not true. Then, upon the removal of $\{v_a, v_b\}$, by Lemma EC.2 there must exists vertices v', v'' for which (v_a, v', v_b, v'') is a chordless cycle. This can only happen if there exists a pair of vertices in $N(v_b)$ which are not adjacent. However, $N(v_b) = \{v_{b-1}, v_{b+1}\} \cup \{v_t, v_{t+1}, \ldots, v_{k-1}\}$, which, by construction, is a clique.

Therefore, we have that $\tilde{w}$ and $\tilde{w} + e^{\{v_a,v_b\}}$ belong to $\text{conv}(X(G))$. Additionally, both solutions satisfy I at equality and thus belong to F^I. Finally, as $\mu(\tilde{w} + e^{\{v_a,v_b\}} - \tilde{w}) = \mu_{\{v_a,v_b\}} = 0$, it follows that $\mu_{f'} = 0$ for every $f' \in \text{Cross}(f^*)$. □

Claim EC.2. *For every $f \in \text{int}(C'')$, $\mu_f = 0$.*

Proof. Fix $f' = \{z_1, z_2\} \in \iota(C'')$ and any solution $\tilde{w}^b$ in $X(G')$ such that $a\tilde{w}^b = b$. Let $\tilde{w}$ be such that $\tilde{w}_f = \tilde{w}^b_f$ if $f \in \iota(C')$, $\tilde{w}_f = 1$ if $f = f^*$ or $f \in \iota(C'') \backslash f'$, and $\tilde{w}_f = 0$ if $f \in \text{Cross}(f^*)$ or $f = f'$. We claim that $G(\tilde{w} + e^{f'})$ is chordal. Consider the ordering π of the vertices in $V(G)$ consisting of a perfect elimination order of the vertices in $V(C')$ (which must exists because $G[V(C')](\tilde{w}^b)$ is chordal), followed by an arbitrary ordering of the remaining vertices. Because the neighbourhood of each vertex in $V(C'') \setminus V(C')$ is a clique in $V(C'')$, it follows by construction that π is a perfect elimination ordering for the vertices of $G(\tilde{w} + e^{f'})$.

We claim now that $G(\tilde{w})$ is also chordal. If not, by Lemma EC.2 there must exist a chordless cycle (z_1, v', z_2, v'') created upon the removal of f' from $G(\tilde{w} + e^{f'})$. At least one among z_1 and z_2 is contained in $\{v_t, v_{t+1}, \ldots, v_{k-1}\}$; let z_1 be one such vertex. The neighborhood of z_1 in $G(\tilde{w})$ is $V(C'')$, and, as $G(\tilde{w})[V(C'')]$ is a clique, we must have $\{v', v''\} \in E(G(\tilde{w}))$, a contradiction.

Therefore, we have that $\tilde{w}$ and $\tilde{w} + e^{f'}$ belong to $\text{conv}(X(G))$ and, by construction, to F^I. Similar arguments to those used in the previous claim show that $\mu_{f'} = 0$ for every $f' \in \text{int}(C'')$. □

Claim EC.3. $\mu_0 = 0$.

Proof. Consider the solution $\hat{w}$ defined by $\hat{w}_f = 1$ if $v_{k-1} \in f$ and $\hat{w}_f = 0$ otherwise. This solution is isomorphic to the solution x' constructed in the proof of Proposition 2, so $G(\hat{w})$ is chordal. By construction, because $f^* = \{v_0, v_{t-1}\}$ for $t < k$, $\hat{w}_{f^*} = 0$. Moreover, as $a_f = 0$ for $f \notin \iota(C')$, we have $a\hat{w} = 0$, and therefore $a\hat{w} - b\hat{w}_{f^*} = 0$. Substituting $\hat{w}$ into $\mu x = \mu_0$ yields $\mu_0 = \mu\hat{w} = \sum_{f \in \text{int}(C')} \mu_f \hat{w}_f + \mu_{f^*}\hat{w}_{f^*} = 0$, as desired. □

Claim EC.4. *There is a $\lambda \in \mathbb{R}$ such that $\mu_{f^*} = -\lambda b$ and $\mu_f = \lambda a_f$ for every f in $\iota(C')$.*

Proof. Let $\tilde{F}^I$ be the subset of F^I containing only solutions x such that $x_f = 1$ for every edge f which does not belong to $\iota(C')$. For every $x \in \tilde{F}^I$, we have $\mu x = \sum_{f \in \iota(C')} \mu_f x_f + \mu_{f^*} 1 = 0$, which implies that $\sum_{f \in \iota(C')} \mu_f x_f = -\mu_{f^*}$. Consequently, we have that every solution y in $X(G')$ that satisfies $ay = b$ must also satisfy $\mu[\iota(C')]y = -\mu_{f^*}$. As $ay \geq b$ is facet defining for $X(G')$, there exists some λ such that $-\mu_{f^*} = \lambda b$ and $\mu_f = \lambda' a_f$ for every f in $E(G')^c$, as desired. □

From the previous claims, we conclude that I is a facet-defining inequality for $\text{conv}(X(G))$. □

EC.3. Additional Proofs for Section 8.1

Lemma EC.3. *For any fractional point $x \in [0,1]^{m^c}$, if $x \notin \text{conv}(X(G))$, then there is a chordless cycle C in $G \cup E(x)$ whose associated inequality of type (I1) is violated by x.*

Proof. Suppose by contradiction that this claim does not hold, and let C be a cycle in G associated with a violated inequality of type (I1) such that $\iota(C) \cap E(G)$ is minimum. Set $\iota(C)$ must contain at least one edge e in $E(G)$, so let C' and C'' be the subcycles of C such that $V(C') \cap V(C'') = e$, $V(C') \cup V(C'') = V(C)$, and $E(C') \cap E(C'') = \{e\}$; by construction, we have $|C'| + |C''| = |C| + 2$. If x satisfies the inequalities (I1) associated with C' and C'', we have

$$\sum_{e \in \iota(C)} x_e \geq \sum_{e \in \iota(C')} x_e + \sum_{e \in \iota(C'')} x_e + 1 \geq |C'| - 3 + |C''| - 3 + 1 = |C| + 2 - 5 = |C| - 3,$$

contradicting hence the fact that C does not satisfy inequality (I1). Therefore, x must violate inequality (I1) for C' or C''; let us assume that the violation holds for C'. If $\iota(C')$ does not contain any edge in $E(G)$, we have a contradiction. Otherwise, we must have $|\iota(C') \cap E(G)| < |\iota(C) \cap E(G)|$, which contradicts the selection of C. □

Proof of Theorem 5(a). We show this result by proving that the (-3)-Quadratic Shortest Cycle Problem (or (-3)-QSCP), defined below, can be reduced to the the separation of the simplified version of inequalities I1. Lemma EC.3 allows us to conclude that these two problems are equivalent, so the main step of the proof consists of showing that (-3)-QSCP is NP-complete.

We define the Quadratic Shortest Cycle Problem (QSCP) as follows: we are given an undirected graph $G=(V,E)$ and a quadratic cost function $q:V\times V\rightarrow[0,1]$ such that $q(u,v)=0$ if $(u,v)\in E$; that is, the quadratic cost associated with $\{u,v\}$ can be different from zero only if $\{u,v\}\notin E$. For any cycle C in G, let $int^*(C)=E(G[C])^C$, that is, edge $\{u,v\}$ belongs to $int^*(C)$ if $u\neq v$ and $(u,v)\notin E$. A feasible solution for an instance of QSCP consists of a simple chordless cycle $C=(v_1,v_2,\ldots,v_{|C|})$ whose cost $p(C)$ is given by $p(C)=\sum_{\{u,v\}\in int^*(C)} q(u,v)-|C|$. Finally, α-QSCP is the decision version of QSCP in which the goal is to decide whether there is a simple chordless cycle C such that $p(C)<\alpha$.

Our proof employs a reduction of the Quadratic Assignment Problem (QAP) to (-3)-QSCP and is based on a reduction used by Rostami et al. (2015). QAP is known to be NP-complete (see e.g., Garey and Johnson (1979)). For an arbitrary instance I of the QAP, let F and L be the sets, respectively, with $n=|F|=|L|$, and let C, D, and A be $n\times n$ matrices in $\mathbb{R}^+$, with A describing the linear costs and B and C the quadratic costs. We are given a value β, and the goal is to decide whether the instance of the QAP admits an assignment whose cost is smaller than β.

If M is an upper bound on the largest individual penalty (linear or quadratic) that may compose the cost of a feasible assignment, no feasible assignment has an objective value larger than $K=2Mn^2$. Such a bound is given by

$$M=\max\left(\max_{\substack{1\le f,f'\le|F|\\1\le l,l'\le|L|}} C_{f,f'}D_{l,l'},\ \max_{\substack{1\le f\le|F|\\1\le l\le|L|}} A_{f,l}\right).$$

From I, we construct an instance I' of (-3)-QSCP associated with a graph $G=(V,E)$ and a quadratic cost $q:V\times V\rightarrow[0,1]$ such that a cycle C is a feasible solution of I' if and only if I admits an assignment whose cost is inferior to β. Let us assume without loss of generality that there is some (arbitrary) ordering between facilities, that is, $F=\{f_1,f_2,\ldots,f_n\}$.

Set V contains one *assignment vertex* $a_{f,l}$ for each pair $(f,l) \in F \times L$. Each cycle C solving I' is associated with the solution of I containing each assignment (f_i, l_j) such that a_{f_i,l_j} belongs to $V(C)$. A pair of assignment vertices belong to the same *block* if they are associated with the same facility. Moreover, V contains three types of auxiliary variables. We have *type-z* vertices z_1 and z_2 and *type-y* variables $y_1, y_2, \ldots, y_n$, whose usage will become clear next. Additionally, for each pair of assignments (f_i, l) and (f_{i+1}, l'), $1 \leq i < n$ and $l, l' \in L$, we have a *connection vertex* $c_{f_i,f_{i+1},l,l'}$. V also contains connection vertices $c_{z_1,f_1,\emptyset,l}$ and $c_{f_n,y_{n-1},l,\emptyset}$ for all l in L; by an abuse of notation, we might use $c_{f_0,f_1,l,l'}$ instead of $c_{z_1,f_1,\emptyset,l'}$ (i.e., substitute $(z_1, \emptyset)$ for (f_0, l)) and $c_{f_n,f_{n+1},l,l'}$ instead of $c_{f_n,z_n,l,\emptyset}$ ($(z_n, \emptyset)$ for (f_{n+1}, l')) in situations where the correct notation can be easily inferred from the context. A pair of connection vertices is said to belong to the same *block* if they have the same first facility index. Each assignment vertex $a_{f_i,l}$ composes edges with connection vertices $c_{f_i,f_{i+1},l,l'}$ and $c_{f_{i-1},f_i,l'',l}$ for all l', l'' in L. Moreover, z_1 and assignment vertices $a_{f_1,l}$ are connected to $c_{z_1,f_1,\emptyset,l}$, whereas y_n and assignment vertices $a_{f_n,l}$ are connected to $c_{f_n,y_n,l,\emptyset}$, $l \in L$. Note that $a_{f_i,l}$ and $a_{f_{i+1},l'}$ are the only neighbours of $c_{f_i,f_{i+1},l,l'}$, that is, all connection vertices have degree 2. Finally, $\{z_1, z_2\}$, $\{z_2, y_1\}$, $\{y_1, y_2\}, \ldots, \{y_{n-1}, y_n\}$ are also edges of E. An example of graph associated with an instance of the QAP with $n = 3$ is presented in Figure EC.4.

Let `Alg` be an algorithm deciding (-3)-QSCP. `Alg` returns C only if $p(C) = p_L(C) + p_Q(C) + q^*(C) < -3$, where $q^*(C)$ is the sum of $-|C|$ with all additional costs that will be incorporated in our construction and $p_L(C)$ and $p_Q(C)$ denote the linear and the quadratic costs of I mapped into C, respectively. For technical reasons described below, the original (linear and quadratic) costs of I will be divided by K in I'. As a consequence, the assignment associated with C is a solution of I if it is feasible and $p_L(C) + p_Q(C) < \frac{\beta}{K}$, with $\beta \leq K$, so $q^*(C) = -3 - \frac{\beta}{K}$.

Costs composing $q^*(C)$ are constructed in a way that `Alg` can only return *matching cycles*, which are cycles in G of size $3n + 3$ that pass through all type-y and type-z vertices and are associated with a feasible assignment for the QAP instance whose cost is below β.

For each pair of assignment vertices $a_{f,l}$ and $a_{f',l'}$, $f \neq f'$ and $l \neq l'$, we have the *assignment cost* $q(a_{f,l}, a_{f',l'}) = \frac{C_{f,f'} D_{l,l'}}{K}$, which represents the quadratic cost of I, and the *linear cost* $q(z_1, a_{f,l}) =$

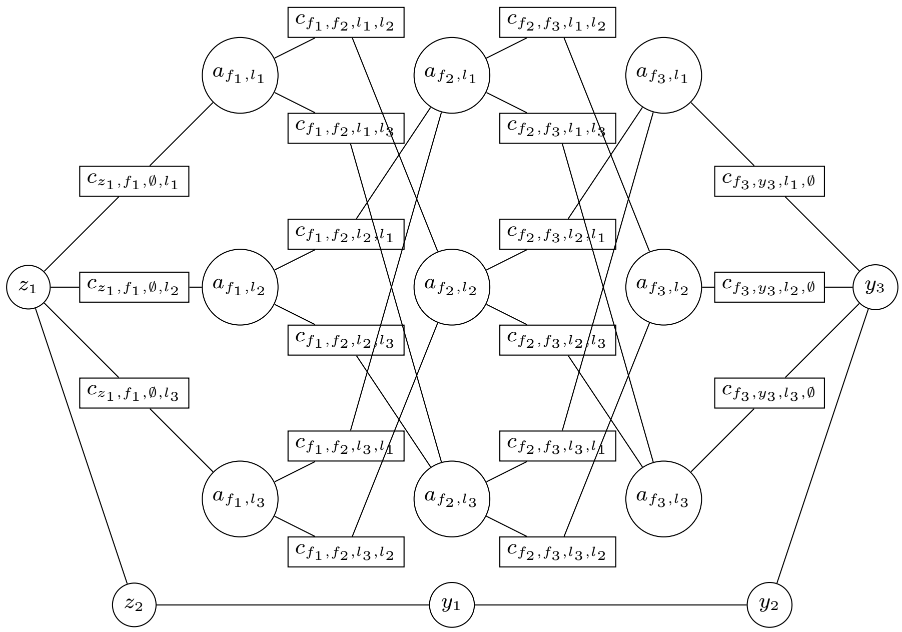


Figure EC.4 Construction for QAP instance with $n = 3$.

$\frac{A_{f,l}}{K}$, representing the linear cost of I. Both values belong to $[0, \frac{1}{2n^2}]$ and, consequently, $0 \le p_L(C) + p_Q(C) \le 1$ for any cycle C of G associated with a feasible assignment in I.

Cycle C cannot be a matching cycle if $V(C)$ contains one or more pairs of assignment vertices sharing the same location or facility. In order to avoid these configurations, we set *assignment conflict costs* $q(a_{f,l}, a_{f',l'}) = 1$ for every pair of assignment vertices $a_{f,l}$ and $a_{f',l'}$ such that either $f = f'$ or $l = l'$. Similar penalties apply to pairs of connection vertices belonging to the same block. That is, given connection vertices $c_{f_i,f_{i+1},l,l'}$ and $c_{f_i,f_{i+1},l'',l'''}$, $1 \le i \le n$ and $l, l', l'', l''' \in L$, we have *transition conflict costs* $q(c_{f_i,f_{i+1},l,l'}, c_{f_i,f_{i+1},l'',l'''}) = 1$.

In order to avoid the selection of cycles which do not pass through type-y and type-z vertices, we penalize pairs of connection vertices belonging to consecutive blocks. That is, given connection vertices $c_{f_i,f_{i+1},l,l'}$ and $c_{f_{i+1},f_{i+2},l'',l'''}$, $0 \le i < n$ and $l, l', l'', l''' \in L$, we have *transition penalties* $q(c_{f_i,f_{i+1},l,l'}, c_{f_{i+1},f_{i+2},l'',l'''}) = 1$. Note that this penalty incurs n times in matching cycles.

The inclusion of connection vertices is compensated by their quadratic costs with z_2. That is, for every connection vertex $c_{f_i,f_{i+1},l,l'}$, $0 \le i < n-1$ and $l, l' \in L$, we have *connection-covering costs* $q(z_2, c_{f_i,f_{i+1},l,l'}) = 1$. In matching cycles, connection-covering costs incur n times. The costs of type-y vertices are covered by quadratic assignments involving z_1 and connection vertices $c_{f_i,f_{i+1},l,l'}$,

$1 \leq i < n-1$ and $l, l' \in L$. These y-covering costs are given by $q(z_1, c_{f_i, f_{i+1}, l, l'}) = 1$ and incur $n-1$ times in matching cycles. Finally, we compensate the deficit of $1 - \frac{\beta}{K} < 1$ in $p(C)$ by setting $q(z_1, y_n) = 1 - \frac{\beta}{K}$. All the remaining costs that have not been explicitly presented are set to zero.

Lemma EC.4. *Every cycle delivered by* `Alg` *must contain all type-z and type-y vertices.*

Proof. Let us assume that C does not include some type-y or type-z vertex; by construction, a cycle in G contains either all vertices in $\{z_2, y_1, y_2, \ldots, y_{n-1}\}$ or none of them, so C may only contain z_1, y_n, assignment vertices, and connection vertices. Consequently, C belongs to a bipartite region of G (with one part being composed of connection vertices), so $|C|$ must be even and larger than 4. See Figure EC.5 for an example with $|C| = 12$. Set $V(C)$ contains $|C|/2$ connection vertices, which are distributed among $k \geq 2$ (consecutive) blocks with $b_i \geq 2$ elements each, $1 \leq i \leq k$. These vertices are associated with transition conflict penalties and transition penalties, and the sum $p_c(C)$ of all penalties associated with them is

$$p_c(C) = \sum_{1 \leq i \leq k} \frac{b_i(b_i - 1)}{2} + \sum_{1 \leq i < k} b_i b_{i+1}.$$

Because $2 \leq b_i \leq \frac{|C|}{2} - 2$, we have $b_i b_{i+1} \geq b_i + b_{i+1}$ and $b_i^2 \geq 2b_i$. Therefore,

$$\begin{aligned} p_c(C) &\geq \sum_{1 \leq i \leq k} \frac{b_i(b_i - 1)}{2} + \sum_{1 \leq i < k} b_i b_{i+1} \geq \sum_{1 \leq i \leq k} \frac{b_i^2}{2} - \sum_{1 \leq i \leq k} \frac{b_i}{2} + \sum_{1 \leq i < k} (b_i + b_{i+1}) \\ &\geq \sum_{1 \leq i \leq k} 2b_i - \sum_{1 \leq i \leq k} \frac{b_i}{2} + \sum_{1 < i < k} b_i \geq |C| - \frac{b_1 + b_k}{2} \geq |C| - \frac{|C|}{4} = \frac{3|C|}{4}. \end{aligned}$$

$V(C)$ also contains $|C|/2 - 2$ assignment vertices, which are located in the middle of the cycle and stay in the same block with at least one other assignment vertex. Therefore, the assignment conflict costs involving these vertices is at least $|C|/4 - 1$. Finally, $V(C)$ contains 2 more vertices, which are are located in the extremities of the cycle (see Figure EC.5) and may be assignment vertices, y_n, or z_1. By summing all penalties, we have $p(C) \geq -|C| + |C|/4 - 1 + p_c(C) > -1$. Therefore, `Alg` can only return cycles C containing all type-z and type-y vertices. □

Lemma EC.5. *Every cycle delivered by* `Alg` *is associated with a feasible assignment.*

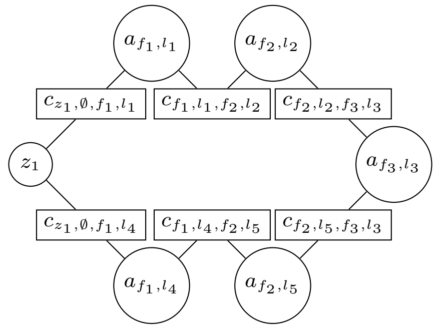


Figure EC.5 Sub-cycle of G with 12 vertices

Proof. Let us suppose by contradiction that `Alg` delivers a cycle C which is not associated with a feasible assignment. From Lemma EC.4, it follows that every cycle delivered by `Alg` necessarily contains at least one assignment vertex containing each facility. Thus, if the assignment associated with C is infeasible, then some location is being assigned to at least two different facilities. Compensation penalties are not affected by this, so $q^*(C) = -3 - \frac{\beta}{K}$. As each assignment conflict cost is equal to 1, we have $p(C) = p_L(C) + p_Q(C) + q^*(C) \geq 1 + q^*(C) \geq 1 - 3 - \frac{\beta}{K}$, and as $\frac{\beta}{K} \leq 1$, $p(C) \geq -3$, and therefore C cannot be delivered by `Alg`. $\square$

The previous lemmas show that `Alg` decides (-3)-QSCP positively on G using only matching cycles. A similar process can be used to construct a solution for (-3)-QSCP on G given a feasible solution for the QAP instance. Therefore, it follows that (-3)-QSCP is NP-complete. $\square$

We conclude by reducing the (-3)-QSCP to the separation of I1. Let I be an instance of (-3)-QSCP associated with graph $G = (V, E)$ and quadratic cost function $q : V \times V \to [0, 1]$. We reduce I to an instance I' of the separation of I1 associated with the same graph $G = (V, E)$. The solution $x \in X(G)$ is derived from the quadratic cost function q of I as follows: if $e = \{u, v\} \in E^c$, $x_e = q(u, v)$. Note that x is valid, since $x_e \in [0, 1]$ for all $e \in E$ and x_e is not defined if $e \in E$.

By construction, any chordless cycle C in G has a cost $c(C)$ in I deciding (-3)-QSCP positively has a cost $c(C)$ such that $c(C) = \sum_{\{u,v\} \in E^c(C)} q(u, v) = \sum_{f \in \iota(C)} x_f < |C| - 3$, that is, if C decides I positively, then C also decides I' positively. The same argument shows that if C decides I' positively, then C is also a valid certificate for I. Finally, from Lemma EC.3, we know that the separation of (I1) can be restricted to cycles which are chordless in G, so we conclude that deciding whether I

has a solution is equivalent to deciding whether I' has a solution. Thus, we conclude that the separation of (I1) is NP-complete. $\square$

Proof of Theorem 5(b). All coefficients of inequalities (I2) are non-negative, so we are able to apply Theorem 3 in order to obtain the following inequalities:

$$x_{\{v_{i-1},v_{i+1}\}} + \sum_{f:v_i\in f,\{v_{i-1},v_{i+1}\}\cap f=\emptyset} x_f \geq \sum_{f\in F(C)} x_f - |F(C)| + 1 \geq 1 - \sum_{f\in F(C)} (1-x_f). \quad \text{(EC.1)}$$

Let $x \in X(G)$ be a fractional solution; for $\{u,v\} \in E(G)$, let $x_{\{u,v\}} = 1$. By construction, solution x violates the inequality (EC.1) associated with cycle $C = (v_{i-1}, v_i, v_{i+1}, v_1, v_2, \ldots, v_n)$ if and only if

$$\begin{aligned} &x_{\{v_{i-1},v_{i+1}\}} + \sum_{t\in C'\setminus\{v_{i-1},v_i,v_{i+1}\}} x_{\{v_i,t\}} < 1 - \sum_{f\in F(C')} (1-x_f) \implies \\ &x_{\{v_{i-1},v_{i+1}\}} + \sum_{v_i\notin\{v_j,v_k\}\in F(C')} \left(\frac{x_{\{v_i,v_j\}} + x_{\{v_i,v_k\}}}{2}\right) + \sum_{v_i\notin\{v_j,v_k\}\in F(C')} (1 - x_{\{v_j,v_k\}}) + \\ &\left(1 - \frac{3x_{\{v_{i-1},v_i\}}}{2}\right) + \left(1 - \frac{3x_{\{v_i,v_{i+1}\}}}{2}\right) \end{aligned}$$

Let $G^i = (V(G), E(G))$ be a complete weighted direct graph such that, for each $e = \{v_j, v_k\} \in E(G)$,

$$w(e) = \begin{cases} 1 - x_{\{v_j,v_k\}} + \frac{x_{\{v_i,v_j\}} + x_{\{v_i,v_k\}}}{2}, & \text{if } v_i \notin \{v_j, v_k\} \\ +\infty, & \text{otherwise.} \end{cases}$$

In order to separate inequalities (EC.1), it suffices to find a path in G^i connecting v_{i+1} to v_{i-1} not passing through v_i whose length is inferior to $1 - x_{\{v_{i-1},v_{i+1}\}} - \left(1 - \frac{3x_{v_{i-1},v_i}}{2}\right) - \left(1 - \frac{3x_{v_i,v_{i+1}}}{2}\right)$. If such a path exists, then, in particular, any shortest path in G^i connecting v_{i+1} to v_{i-1} while avoiding v_i also satisfies this property, so the verification can be done in polynomial time for each sequence (v_{i-1}, v_i, v_{i+1}). The number of sequences for which this verification needs to be performed is $O(|V(G)|^3)$, so we conclude that inequality (EC.1) can be separated in polynomial time. $\square$

Proof of Theorem 5(c). All coefficients of inequalities (I3) are non-negative, so we are able to apply Theorem 3 in order to obtain the following inequalities:

$$\begin{aligned} &\sum_{f:\{\{v_i,v_j\}:d_C(v_i,v_j)=2\}} x_f \geq 2\left(\sum_{f\in F(C)} x_f - |F(C)| + 1\right) \implies \\ &\sum_{f:\{\{v_i,v_j\}:d_C(v_i,v_j)=2\}} x_f + 2\sum_{f\in F(C)} (1-x_f) \geq 2. \end{aligned} \quad \text{(EC.2)}$$

Note that inequality (EC.2) is trivially satisfied if $\sum_{f\in F(C)} x_f < |F(C)|$, as the right-hand side expression becomes zero and all coefficients on the left are non-negative.

Let $x \in X(G)$ be a fractional solution; for $\{u,v\} \in E(G)$, let $x_{\{u,v\}} = 1$. Let D be a weighted directed graph such that, for each two-set $\{v_i, v_j\}$ in $V(G)$, there is one vertex in $V(D)$ labeled by pair (v_i, v_j) and another labeled by pair (v_j, v_i). Moreover, for each pair of vertices (v_i, v_j) and (v_j, v_k) in $V(D)$, $v_i \neq v_k$, we define an arc $a = ((v_i, v_j), (v_j, v_k))$ in $A(D)$ whose weight is given by $w_D(a) = w_D((v_i, v_j), (v_j, v_k)) = (1 - x_{\{v_i, v_j\}}) + x_{\{v_i, v_k\}} + (1 - x_{\{v_j, v_k\}})$.

Let u, v, w, and x be vertices in $V(G)$ such that $P_G = (u, v, w, x)$ is a path in G, and let $P_D = (a_1, a_2, a_3)$ be the associated path in D, with $a_1 = (u, v)$, $a_2 = (v, w)$, and $a_3 = (w, x)$. Let $P'_D = (v_1, v_2, \ldots, v_n)$ be another path in D such that $v_1 = (x, z_1)$, $v_i = (z_i, z_{i+1}), 1 \le i < n$, and $v_n = (z_n, u)$; note that, by construction, arc $((a_3, v_1), (v_n, a_1))$ belongs to $A(D)$.

We claim that C_D is a directed cycle in D if P'_D is a shortest path in D connecting (x, z') to (z'', u) (with z', z'' in $V(G)$) whose internal vertices are not associated with edges in G containing vertices in $\{u, v, w, x\}$ and such that $|P'_D|$ is minimal. As $w_D(a) \ge 0$ for all $a \in A(D)$, it follows from the last condition and from the fact that P'_D is a shortest path that all elements in $\{z_1, \ldots, z_n\}$ are necessarily pairwise different. The sum of the costs of all edges in cycle C_D is given by

$$\sum_{i\in[1,2]} w((a_i, a_{i+1})) + w((a_3, v_1)) + \sum_{i\in[1,n-1]} w((v_i, v_{i+1})) + w((v_n, a_1)) = \sum_{f\in\{\{a,b\}: d_{C_G}(a,b)=2\}} x_f + 2 \sum_{f\in F(C_G)} (1 - x_f).$$

Therefore, solution x does not respect the inequality (EC.2) associated with cycle C_G in G containing path $P = (u, v, w, x)$ if and only if the weight of P'_D is smaller than 2.

The number of tuples for which this verification needs to be performed is $O(|V(G)|^4)$, and the identification (and construction) of a path P'_D with the desired features can be performed in polynomial time.Therefore, Inequalities EC.2 can be separated in polynomial time. □

Proof of Theorem 5(d). The proof of this result is very similar to the one used for Theorem 5(a). More precisely, we introduce QSCP*, a variation of QSCP that is more convenient for proving the hardness of (the simplified version of) (I4), and show that the addition of a single compensation penalty to the construction used in the proof of Theorem 5(a) yields the desired result.

In the α-Adapted Quadratic Shortest Cycle Problem (α-QSCP*) we are given $\alpha \in \mathbb{R}$, an undirected graph $G = (V, E)$, and a quadratic cost function $q : V \times V \to [0, 1]$ such that $q(u, v) = 0$ if $(u, v) \in E$. A feasible solution for an instance of αQSCP* consists of a simple chordless cycle $C = (v_1, v_2, \ldots, v_{|C|})$ whose cost $p^*(C) = \sum_{\{u,v\} \in int^*(C)} q(u, v) - |C| - U$ is smaller than α, where

$$U = \max_{\substack{v_i, v_j \in V(C) \\ v_i \neq v_j \\ d_C(\{v_j, v_i\}) \geq 2}} (q(v_{j-1}, v_{j+1}) + q(v_j, v_i)).\text{`}$$

The present proof also relies on a reduction of QAP to (-4)-QSCP*. Let I be an arbitrary instance of QAP of size n and M be the largest individual (i.e., linear or quadratic) penalty, and $K = 2Mn^2$. If `Alg` is an algorithm that decides (-4)-QSCP*, then it will only return a cycle C containing the linear costs $p_L(C)$ and the quadratic costs $p_Q(C)$ of I if $p^*(C) = p_L(C) + p_Q(C) + q^*(C) - U < -4$, where $q^*(C)$ is the sum of $-|C|$ with additional costs incorporated by our construction. The assignment associated with C is feasible if $p_L(C) + p_Q(C) < \frac{\beta}{K}$, so we have $q^*(C) = -4 - \frac{\beta}{K} + U$. Using the penalties of the construction used in the proof of Theorem 5(a), one can see by inspection that if C is a matching cycle, then $U \leq 1 + \frac{M}{K}$.

In order to guarantee equality in the inequality above for every matching cycle, we set $q(a_{f_1,l_k}, y_1) = \frac{M}{K}$ for all $l_k \in L$. By definition, $M \leq K$, so $q(a_{f_1,l_k}, y_1) \leq 1$. With this, we have $q(v_{j-1}, v_{j+1}) = 1$ and $q(v_j, v_i) = \frac{M}{K}$ for $v_{j-1} = c_{z_1,f_1,\emptyset,l_k}$, $v_j = a_{f_1,l_k}$, $v_{j+1} = c_{f_1,f_2,l_k,l_z}$, and $v_i = y_1$, $l_z, l_k \in L$. Moreover, for every matching cycle C, by direct substitution we have $q^*(C) = -3 - \frac{\beta}{K} + \frac{M}{K}$.

The arguments used in the proof of Lemma EC.4 also apply to the present construction, so `Alg` can only select cycles that include all type-y and type-z vertices. For Lemma EC.5, note that if C contains assignments involving the same location or facility, then $p^*(C) = p_L(C) + p_Q(C) + q^*(C) - U \geq 1 - 3 - \frac{\beta}{K} + \frac{M}{K} > -3$, so every cycle delivered by `Alg` is associated with a feasible assignment.

Finally, the arguments used in the proof of Theorem 5(a) to show that (-3)-QSCP is NP-complete can be used in an identical way in order to show that (-4)-QSCP is NP-complete, and the separation of (I4) can be reduced to (-4)-QSCP in the same way the separation of (I1) was reduced to (-3)-QSCP, so it follows that the separation of (I4) is also NP-complete. $\square$

EC.4. Perfect Elimination Ordering Model

The model finds a perfect elimination ordering that minimizes the number of fill-in edges, and can be written as follows.

$$\min \sum_{\{i,j\}\in E^c} y_{ij} + y_{ji}$$

$$\text{s.t.} \quad x_{ij} + x_{ji} = 1, \qquad \text{for all } \{i,j\} \in E \tag{EC.3}$$

$$x_{ij} + x_{ji} \leq 1, \qquad \text{for all } \{i,j\} \in E^c \tag{EC.4}$$

$$x_{ij} + x_{jk} - x_{ik} \leq 1, \qquad \text{for all } i,j,k \in V, i \neq j, j \neq k, i \neq k \tag{EC.5}$$

$$y_{ij} \leq x_{ij}, y_{ji} \leq x_{ji}, \qquad \text{for all } i,j \in V, i \neq j \tag{EC.6}$$

$$y_{ij} = x_{ij}, y_{ji} = x_{ji}, \qquad \text{for all } \{i,j\} \in E \tag{EC.7}$$

$$x_{jk} + y_{ij} + y_{ik} - y_{jk} \leq 2, \qquad \text{for all } i,j,k \in V, i \neq j, j \neq k, i \neq k \tag{EC.8}$$

$$y_{ij}, x_{ij} \in \{0,1\}, \qquad \text{for all } i,j \in V, i \neq j \tag{EC.9}$$

In the model above, a binary variable y_{ij} indicates whether edge $\{i,j\}$ is added to G and binary variable x_{ij} indicates whether vertex i precedes j in the resulting ordering. Constraints (EC.3) enforce the existence of a precedence relation between vertices i and j if $\{i,j\} \in E$, whereas constraints (EC.4) prevent i and j from preceding each other simultaneously in an elimination ordering. Constraints (EC.5) ensure the transitive closure of precedence relations is satisfied. Constraints (EC.6) and (EC.7) indicate that a precedence relation between edges i and j can exist if and only if $\{i,j\} \in E$. Finally, constraints (EC.8) impose that the final ordering must be a perfect elimination ordering.