manuscript No.
(will be inserted by the editor)

Exploiting sparsity for the min k-partition problem

Guanglei Wang · **Hassan Hijazi**

Received: date / Accepted: date

Abstract The minimum k-partition problem is a challenging combinatorial problem with a diverse set of applications ranging from telecommunications to sports scheduling. It generalizes the max-cut problem and has been extensively studied since the late sixties. Strong integer formulations proposed in the literature suffer from a prohibitive number of valid inequalities and integer variables. In this work, we introduce two compact integer linear and semidefinite reformulations that exploit the sparsity of the underlying graph and develop fundamental results leveraging the power of chordal decomposition. Numerical experiments show that the new formulations improve upon state-of-the-art.

1 Introduction

Given an undirected weighted graph $G = (V, E)$, the minimum k-partition (MkP) problem consists of partitioning V into at most k disjoint subsets, minimizing the total weight of the edges joining vertices in the same partitions. This problem has been shown to be strongly $\mathcal{NP}$-hard [30]. MkP was firstly defined in [8] in the context of scheduling. The authors formulate the problem as a quadratic program and suggest a method returning local minima. The MkP has applications in statistical clustering, telecommunications, VLSI layout, sports team scheduling and statistical physics (see, e.g., [17]).

The complement of MkP is a max-k-cut problem where one is seeking to maximize the sum of weights corresponding to intra-partition edges. Related investigations can be found in [14, 15, 20]. The well-known integer linear programming formulation of max-k-cut can be obtained by complementing the variables in the integer formulation of MkP. Thus solving max-k-cut is equivalent to solving MkP problem. However, these problems have different approximability results. While the max-k-cut can be solved $(1 - k^{-1})$-approximately in polynomial time, the

G. Wang · H. Hijazi
The Australian National University, ACTON 2601, Canberra, Australia.

H. Hijazi
Los Alamos National Laboratory, New Mexico, USA.

MkP problem is not $\mathcal{O}(|E|)$-approximable in polynomial time unless $\mathcal{P} = \mathcal{NP}$ [18]. When $k = 2$, the minimum 2-partition is equivalent to the max-cut problem, which has been extensively studied in the literature. Among the numerous references for max-cut, we point out Barahona and Mahjoub [3], Goemans and Williamson [24], Deza and Laurent [16].

Chopra and Rao [11] present two different Integer Linear Programming (ILP) formulations for the k-partition problem. One has node and edge variables, whereas the other has only edge variables. For each formulation, several families of strong valid linear inequalities have been discovered (see for instance, [11,10,14,19,33]). For some of these families, there are efficient separation heuristics (e.g., [15,7,29,19]). There is also a literature concerned with Integer Semidefinite Programming (ISDP) formulations of the k-partition problem [18,23,1,31,33].

Contributions: In this paper, we propose two compact reformulations of the MkP problem. One is an ILP formulation based on Chopra and Rao's [11] edge-based model. The other is an ISDP formulation based on the Eisenblätter's [18] model. Both models exploit the sparsity of the graph and thus have less integer variables than their counterparts. In [11], Chopra and Rao state that they do not know of any formulation with $|E|$ variables for the MkP problem, for $2 < k < n$. We show that if G is chordal, our proposed ILP formulation is one. To validate the proposed two formulations, several theoretical results are established. The computational experiments show that the proposed formulations are able to improve computational efficiency by several orders of magnitude.

Outline: The rest of the paper is organized as follows. In Section 2, the literature on the existing formulations of MkP is reviewed. In Section 3, we provide our compact formulations which are based on the maximal clique set. Some fundamental results are established. In Section 4, we describe some computational experiments and analyse the results. Finally, some concluding remarks are made in Section 5.

Notation: For a finite set S, $|S|$ represents its cardinality. $\mathbb{S}^n$ represents the set of real symmetric matrices in $\mathbb{R}^{n\times n}$. $\mathbb{S}^n_+$ denotes the cone of real positive semidefinite matrices in $\mathbb{S}^n$, i.e., $\mathbb{S}^n_+ = \{S \in \mathbb{S}^n : S \succeq 0\}$. For a matrix M, M_C represents the principle sub-matrix composed by set C of columns and set C of rows of M. We also use (C, D) to denote the matrix where C and D are concatenated by rows assuming they have the same number of columns. The inner product between two matrices $A, B \in \mathbb{R}^{mn}$ is denoted by $\langle A, B\rangle = \sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}B_{ij}$.

2 Literature Review

In this section, we review some existing formulations of the MkP problem. To facilitate the development of our main results, we also review some basic graph terminology.

2.1 Formulations of MkP

Following [11,10], the MkP problem can be expressed in Model 1. Binary variable x_{ij} is 1 if and only if i and j are in the same partition, 0 otherwise. The *triangle*

Model 1 The edge formulation of the MkP problem

$$\textbf{variables: } x \in \{0,1\}^{\frac{n(n-1)}{2}} \tag{1a}$$

$$\textbf{minimize: } \sum_{(i,j)\in E} w_{ij}x_{ij} \tag{1b}$$

$$\textbf{s.t.: } x_{ih} + x_{hj} - x_{ij} \leqslant 1,\ \forall i,j,h \in V,\ i < h < j \tag{1c}$$

$$\sum_{i,j\in Q: i<j} x_{ij} \geqslant 1,\ \forall Q \subseteq V, \text{where } |Q| = k+1 \tag{1d}$$

constraint (1c) enforces consistency with respect to partition membership. Namely, if $x_{ih} = 1$ and $x_{jh} = 1$, indicating that i, h and j are in the same partition, this implies $x_{ij} = 1$. For every subset of $k+1$ vertices, the *clique* constraint (1d) forces at least two vertices to be in the same partition. Together with constraints (1c), this implies that there are at most k partitions. In total, there are $3\binom{|V|}{3}$ triangle inequalities and $\binom{|V|}{k+1}$ clique inequalities. We refer to this formulation as the edge formulation.

When $3 \leqslant k \leqslant |V|-1$, we denote by P^k the set of feasible solutions of Model 1 as:

$$P^k = \left\{x \in \{0,1\}^{V^2} : x \text{ satisfies (1c) and (1d)}\right\}, \tag{2}$$

where $V^2 := \{(i,j) \in V \times V : i < j\}$ represents the edges of the complete graph induced by vertices V. It is known that the convex hull of P^k is a polytope [11]. Moreover, it is fully-dimensional in the space spanned by the edge variables (see, e.g., [10,16]).

In their seminal paper [11], Chopra and Rao give several valid and facet defining inequalities for P_k. These include *general clique, wheel* and *bicycle* inequalities. Among others, we remark that the triangle (1c) and clique inequalities (1d) can make the continuous relaxation of Model 1 hard to deal with. In fact, it has been noticed that the exact separation of the clique inequalities is $\mathcal{NP}$-hard in general, and the complete enumeration is intractable even for small values of k [23].

Chopra and Rao [11] also propose an alternative ILP formulation, presented in Model 2. It has $kn+m$ variables, where n and m are the respective numbers of nodes and edges of graph G. Binary variable x_{ic} is 1 if node i lies in the cth subset. Binary variable y_{ij} is 1 if i and j are in the same partition. Constraints (3c) require that each node must be assigned to exactly one subset. Constraints (3d) imply that if (i,j) is an edge and i and j are in the same subset in a feasible partition then the edge (i,j) cannot cut, i.e., $y_{ij} = 1$. On the other hand, if i and j are in different subsets in a feasible partition, the edge (i,j) must be cut, i.e., $y_{ij} = 0$. This is implied by (3e) and (3f). We refer to Model 2 as the node-edge formulation. Some valid inequalities have been proposed in the literature to strengthen Model 2 (see, e.g., [11,19]).

Eisenblätter [18] provides an exact ISDP reformulation of Model 1. We introduce variables $X_{ij} \in \left\{\frac{-1}{k-1}, 1\right\}$, where $X_{ij} = 1$ if and only if node i and j are in the same partition. This formulation is presented in Model 3.

Model 2 The node-edge formulation of the MkP problem

$$
\begin{aligned}
\textbf{variables:}\ & x \in \{0,1\}^{kn}, y \in \{0,1\}^{m} && \text{(3a)}\\
\textbf{minimize:}\ & \sum_{(i,j)\in E} w_{ij} y_{ij} && \text{(3b)}\\
\textbf{s.t.:}\ & \sum_{c=1}^{k} x_{ic} = 1 \; i \in V && \text{(3c)}\\
& x_{ic} + x_{jc} - y_{ij} \leqslant 1 \;\; (i,j) \in E, c = 1,\dots,k && \text{(3d)}\\
& x_{ic} - x_{jc} + y_{ij} \leqslant 1 \;\; (i,j) \in E, c = 1,\dots,k && \text{(3e)}\\
& -x_{ic} + x_{jc} + y_{ij} \leqslant 1 \;\; (i,j) \in E, c = 1,\dots,k. && \text{(3f)}
\end{aligned}
$$

Model 3 The integer SDP formulation of the MkP problem

$$
\begin{aligned}
\textbf{variables:}\ & X_{ij} \in \left\{\frac{-1}{k-1}, 1\right\} \;\; (\forall i,j \in V : i < j). && \text{(4a)}\\
\min\ & \sum_{(i,j)\in E} w_{ij} \frac{(k-1)X_{ij}+1}{k} && \text{(4b)}\\
\text{s.t.}\ & X \in \mathbb{S}^n_+ && \text{(4c)}\\
& X_{ii} = 1, \;\; \forall i \in V. && \text{(4d)}
\end{aligned}
$$

Replacing the constraint $X_{ij} \in \left\{\frac{-1}{k-1}, 1\right\}$ with $\frac{-1}{k-1} \leqslant X_{ij} \leqslant 1$ yields a Semidefinite Programming (SDP) relaxation. Notice that $X_{ij} \leqslant 1$ can be dropped since it is implicitly enforced by X being positive semidefinite and $X_{ii} = 1$. This SDP relaxation was used in combination with randomized rounding to obtain a polynomial time approximation algorithm for the max k-cut problem by Freize and Jerrum [20].

As proposed by Eisenblätter [18], one can recast Model 3 using binary variables. Indeed, we introduce binary variables Y_{ij} such that

$$
X_{ij} = \frac{-1}{k-1} + \frac{k}{k-1} Y_{ij}. \tag{5}
$$

Thus, $X_{ij} = \frac{-1}{k-1}$ if $Y_{ij} = 0$ and $X_{ij} = 1$ if $Y_{ij} = 1$. This leads to

$$
\begin{aligned}
\min\ & \sum_{(i,j)\in E} w_{ij} Y_{ij} \\
\text{s.t.}\ & \frac{-1}{k-1} J + \frac{k}{k-1} Y \in \mathbb{S}^n_+, && \text{(6a)}\\
& Y_{ii} = 1, \quad \forall i \in V, && \text{(6b)}\\
& Y_{ij} \in \{0,1\} \;\; \forall i,j \in V : i < j. && \text{(6c)}
\end{aligned}
$$

where J is the all-one matrix. Note that the mapping function (5) is bijective or one-to-one. This bijective relationship will be used in Section 3.2.

Several SDP based branch-and-bound frameworks for MkP problem have appeared in the literature [23,1,33]. Typically, these approaches identify several families of inequalities to tighten the SDP relaxation during the solution procedure.

However, solving large scale SDP problems is less efficient than their LP counterparts [34,4].

2.2 Graph terminology

We assume familiarity with elementary concepts and definitions from graph theory, such as tree, edge, connected components, etc. In several cases, we shall use results from the algorithmic graph theory for chordal graphs; for this, we refer readers to [25,5]. Here we introduce some of the graph notation and terminology that will be used throughout the paper. Other concepts from graph theory will be introduced as needed in later sections of the paper.

Let $G = (V, E)$ denote an *undirected graph* with vertex set V and edge set V. The number of vertices is denoted by $n = |V|$ and the number of edges by $m = |E|$. A graph is called *complete* if every pair of vertices are adjacent. A *clique* of a graph is an induced subgraph which is complete, and a clique is *maximal* if its vertices do not constitute a proper subset of another clique. A set $C \subseteq E$ is a *cycle* if it induces a connected subgraph of G in which every node has degree 2.

A graph is said to be *chordal* if every cycle $C \subseteq E$ with $|C| \geqslant 4$ has a *chord* (an edge joining two nonconsecutive vertices of the cycle). Given a graph $G = (V, E)$, we say that a graph $G_F = (V, F)$ is a *chordal extension* of G if G_F is chordal and $E \subseteq F$. Throughout this article, we assume that the graph G is connected.

3 Main results

Observe that the objective function (1b) is determined by variables $x_{ij}, \forall (i,j) \in E$. Let $\mathcal{K}$ be the set of all maximal cliques in G_F, representing the chordal extension of G, i.e., $\mathcal{K} = \{C_1, \ldots, C_l\}$, $C_i \subseteq V \ \forall i \in \{1, \ldots, l\}$, such that $F = \bigcup_{r=1}^{l} C_r \times C_r \supseteq E$. We will next show that the following property holds:

Property 1 (Completion Property) For any vector $x_F \in \mathbb{R}_+^{|F|}$, if x_F satisfies

$$x_{ih} + x_{hj} - x_{ij} \leqslant 1, \ \forall i,j,h \in C_r, r = 1, \ldots, l, \tag{7a}$$

$$\sum_{i,j \in Q} x_{ij} \geqslant 1, \ \forall Q \subseteq C_r, \text{where } |Q| = k+1, r = 1, \ldots, l, \tag{7b}$$

$$x_F \in \{0,1\}^{|F|}, \tag{7c}$$

then there exists $x_T \in \{0,1\}^{|V^2 \setminus F|}$ such that $x = (x_F, x_T)$ satisfies (1c) and (1d).

Under the conditions above, we observe that the value of the objective (1b) can be determined by values of entries x_F and independently of values in x_T. In addition, the resulting formulation has $\sum_{r=1:|C_r| \geqslant 3}^{l} 3\binom{|C_r|}{3} + \sum_{r=1:|C_r| \geqslant k+1}^{l} \binom{|C_r|}{k+1}$ constraints and $|F|$ binary variables. Thus it becomes more compact if this number is less than $3\binom{|V|}{3} + \binom{|V|}{k+1}$.

3.1 A compact ILP reformulation

Let P_F^k represent the set of feasible solutions satisfying (7a), (7b), (7c), i.e.,

$$P_F^k = \left\{ x \in \mathbb{R}_+^F : (7a), (7b), (7c) \right\}. \tag{8}$$

We first show that the $|F|$ components corresponding to index set F in $x \in P^k$ represent a member of P_F^k. We denote by $\mathrm{Proj}_{x_F} S$ the projection of S into the space defined by the components in F, i.e., for a given set $S \subset \mathbb{R}^I, F \subset I \subseteq V^2$,

$$\mathrm{Proj}_{x_F} S = \left\{ x_F \in \mathbb{R}^F : \exists x_T \in \mathbb{R}^{I \setminus F}, (x_F, x_T) \in S \right\} \tag{9}$$

Lemma 1 $\forall k,\ 2 \leqslant k \leqslant n$, $\mathrm{Proj}_{x_F} P^k \subseteq P_F^k$

Proof Let $\bar{x}$ be an arbitrary point in P^k. It is easy to verify that entries of $\bar{x}_F$ satisfy constraints in P_F^k and therefore $\mathrm{Proj}_{x_F} P^k \subseteq P_F^k$. □

Lemma 2 (Lemma 3, [26]) *Given that G_F is chordal, for any pair of vertices u and v with $u \neq v, \{u, v\} \notin E$, the graph $G_F + \{u, v\}$ has a unique maximal clique which contains both u and v.*

Lemma 3 (Lemma 4, [26]) *Given that G_F is chordal, there exists a sequence of chordal graphs*

$$G_i = (V, F_i), i = 0, \ldots, s, \tag{10}$$

such that $G_0 = G_F$, G_s is the complete graph, and G_i is obtained by adding an edge to G_{i-1} for all $i = 1, \ldots, s$.

For sake of further development, let us represent the neighborhood of a vertex $v \in V$ as $N_G(v) = \{w \in V : (v, w) \in E\}$.

Proposition 1 *Given a chordal graph $G = (V, E)$, for every edge $(u, v) \in E$, the unique maximal clique C containing both u and v is the union of $\{u, v\}$ and $N_G(u) \bigcap N_G(v)$, i.e., $C = \{u, v\} \bigcup (N_G(u) \bigcap N_G(v))$.*

Proof If $N_G(u) \bigcap N_G(v) = \varnothing$, the result holds. When $N_G(u) \bigcap N_G(v)$ is nonempty, we observe that $N_G(u) \bigcap N_G(v)$ contains all cliques involving $\{u, v\}$, minus $\{u, v\}$. Indeed, if L is a clique containing both u and v, then any element (other than u and v) in C is a neighbor of u and v, which means $L \subseteq N_G(u) \bigcap N_G(v)$. Hence if we can show that $C = \{u, v\} \bigcup (N_G(u) \bigcap N_G(v))$ is a clique, then by definition, C is the maximal clique containing $\{u, v\}$. Indeed, suppose there exist $l_1, l_2 \in N_G(u) \bigcap N_G(v)$ such that l_1 and l_2 are disconnected, then the ordered set $\{u, l_1, v, l_2, u\}$ is a cycle of length 4 without a chord, contradicting with the fact that G is chordal. The uniqueness comes from the maximality of C. □

Theorem 1 $\forall k,\ 2 \leqslant k \leqslant n$, $P_F^k = \mathrm{Proj}_{x_F} P^k$.

Proof By Lemma 1, $\mathrm{Proj}_{x_F} P^k \subseteq P^k_F$. We now show that the reverse also holds. Given G_F, let $\{G_0, G_1, \ldots G_s\}$ be a sequence of choral graphs satisfying Lemma 3. Denote by (i_1, j_1) the edge of G_1 that is not a member of G_0 (w.l.o.g., we assume $i_1 < j_1$). Let $\bar{x}_F$ be a point in P^k_F. If we can show that there exists a vector $\bar{x}_1 = (\bar{x}_F, x_{i_1 j_1})$ satisfying the constraints in $P^k_{G_1}$, then by induction, we can show the existence of a point $x_s \in P^k$.

By Lemma 2, there is a unique maximal clique C of G_1 containing $\{i_1, j_1\}$ and it can be identified by Propoistion 1. Without loss of generality, we may reorder indices of the nodes in C and let $C = \{l_0, \ldots, l_p, i_1, j_1\}$ with $l_0 < \cdots < l_p < i_1 < j_1$. Let x_C be the vector corresponding to the maximal clique C. Since any clique containing $\{i_1, j_1\}$ is a subset of C, $x_1 \in P^k_{G_1}$ iff $x_C \in P^k_C$, where P^k_C is formed by replacing V with C in (2).

Hence we only need to show that $x_C \in P^k_C$ for some value of $x_{i_1 j_1}$. Given x_F, we construct $x_{i_1 j_1}$ as follows: if $C = \{i_1, j_1\}$, let $x_{i_1 j_1} = 1$ and the solution is feasible. Otherwise, three cases can occur:

1. $(x_{hi_1}, x_{hj_1}) = (0, 0), \forall h \in \{l_0, \ldots, l_p\}$.
2. $\exists h \in \{l_0, \ldots, l_p\}$ such that $x_{hi_1} + x_{hj_1} = 1$.
3. $\exists h \in \{l_0, \ldots, l_p\}$ such that $x_{hi_1} + x_{hj_1} = 2$.

We now construct the solution as follows:

1. If case 1 or case 3 occur, let $x_{i_1 j_1} = 1$.
2. otherwise (i.e., case 2 occurs), let $x_{i_1 j_1} = 0$.

In order to show that the constructed solution is unique, we next show that case 1 and case 3 are exclusive with respect to case 2. It is obvious that case 1 is exclusive with case 2. We now show that case 2 and case 3 are exclusive:

- The result is straightforward when $p = 0$.
- When $p \geqslant 1$, we proceed by contradiction: suppose there exists $h, h' \in V : h \neq h'$ such that $x_{hi_1} + x_{hj_1} = 1$ and $x_{h'i_1} + x_{h'j_1} = 2$. Let us consider $(x_{hi_1}, x_{hj_1}) = (0, 1)$. The other case $(x_{hi_1}, x_{hj_1}) = (1, 0)$ will follow symmetrically. Note that (h, h', i_1) and (h, h', j_1) are cliques of length 3 in G_F. Thus by (7a), we have, on one hand, $x_{hh'} + x_{h'i_1} - x_{hi_1} \leqslant 1$ leading to $x_{hh'} \leqslant 0$. On the other hand, we have $x_{h'j_1} + x_{hj_1} - x_{hh'} \leqslant 1$ leading to $x_{hh'} \geqslant 1$, contradiction.

We now verify that the extended solution $x_1 = (x_F, x_{i_1 j_1})$ satisfies constraints in P^k_C. Since $\{l_0, \ldots, l_p, i_1\}$ and $\{l_0, \ldots, l_p, j_1\}$ are cliques in G_F, the associated constraints have been imposed by P^k_F. Thus, we just need to verify that x_1 satisfies

$$x_{hi_1} + x_{hj_1} - x_{i_1 j_1} \leqslant 1, \forall h \in \{l_0, \ldots, l_p\} \tag{11a}$$
$$x_{hi_1} + x_{i_1 j_1} - x_{hj_1} \leqslant 1, \forall h \in \{l_0, \ldots, l_p\} \tag{11b}$$
$$x_{hj_1} + x_{i_1 j_1} - x_{hi_1} \leqslant 1, \forall h \in \{l_0, \ldots, l_p\} \tag{11c}$$
$$\sum_{(h,f) \in C_q} x_{hf} + x_{i_1 j_1} \geqslant 1, \ \forall Q \subseteq C, |Q| = k + 1, \ i_1, j_1 \in Q \tag{11d}$$

where $C_q = \big\{(h, f) \in Q^2 : \ h < f, \ (h, f) \neq (i_1, j_1)\big\}$.

- First, let us show that the the constructed solution satisfies the above triangle inequalities (11a)–(11c). For case 1, for each $h, i_1, j_1 \in C$, the unique solution $(x_{hx_1}, x_{hj_1}, x_{i_1j_1}) = (0,0,0)$ is feasible. For case 2, both solution $(x_{hi_1}, x_{hj_1}, x_{i_1j_1}) = (1,0,0)$ and $(x_{hi_1}, x_{hj_1}, x_{i_1j_1}) = (0,1,0)$ are feasible. For case 3, the unique solution $(x_{hi_1}, x_{hj_1}, x_{i_1j_1}) = (1,1,1)$ is feasible.

- Second, we need to verify that the constructed solution is feasible for the clique inequalities (11d) when $|C| \geqslant k+1$. It is easy to see that this is true for case 1 and case 3. For case 2, we consider $|C| = k+1$ and $C \geqslant |k+2|$. Let us denote by h^* the index such that $x_{h^*i_1} + x_{h^*j_1} = 1$. Recall that $x_{i_1j_1} = 0$.
  1. When $|C| = k+1$, it holds that $Q = C$, $(h^*, i_1) \in C_q$ and $(h^*, j_1) \in C_q$. Then we have

$$\sum_{(h,f)\in C_q} x_{hf} + x_{i_1j_1} \geqslant 1,$$

  2. When $|C| \geqslant k+2$, there are multiple subsets Q. If Q contains h^*, the desired result follows. If $h^* \notin Q$, we have

$$\sum_{(h,f)\in C_q} x_{hf} + x_{i_1j_1} = \sum_{(h,f)\in C_q: f<i_1} x_{hf} + \sum_{(h,i_1)\in C_q} x_{hi_1} + \sum_{(h,j_1)\in C_q} x_{hj_1}$$

Note that if the following disjunction is true:

$$\sum_{(h,f)\in C_q: f<i_1} x_{hf} \geqslant 1 \vee \sum_{(h,i_1)\in C_q} x_{hi_1} \geqslant 1 \vee \sum_{(h,j_1)\in C_q} x_{hj_1} \geqslant 1$$

then we have $\sum_{(h,f)\in C_q} x_{hf} + x_{i_1j_1} \geqslant 1$, the result follows. We now show that it is not possible to have a solution $x_F \in P_F^k$ satisfying

$$\sum_{(h,f)\in C_q: f<i_1} x_{hf} = 0 \tag{12a}$$

$$\sum_{(h,i_1)\in C_q} x_{hi_1} = 0 \tag{12b}$$

$$\sum_{(h,j_1)\in C_q} x_{hj_1} = 0. \tag{12c}$$

We proceed by contradiction. Assume that (12a)-(12c) hold. Then we have:

$$\begin{aligned}
0 = & \sum_{(h,f)\in C_q: f<i_1} x_{hf} + \sum_{(h,i_1)\in C_q} x_{hi_1} + \sum_{(h,j_1)\in C_q} x_{hj_1} \\
= & \sum_{(h,i_1)\in C_q} x_{hi_1} + \sum_{(h,j_1)\in C_q} x_{hj_1} \\
\geqslant & 2 - 2 \sum_{(h,f)\in C_q: f<i_1} x_{hf} - 2 \sum_{(h,h^*)\in C_q} x_{hh^*} - x_{h^*i_1} - x_{h^*j_1} \\
\geqslant & 1 - 2 \sum_{(h,h^*)\in C_q} x_{hh^*} \\
= & 1,
\end{aligned}$$

which is a contradiction. The first "$\geqslant$" comes from the fact that $Q \setminus \{i_1\} \cup h^*$ and $Q \setminus \{j_1\} \cup h^*$ are cliques of size $k+1$. Thus by clique inequalities (7b) enforced in P_F^k, we have

$$\sum_{(h,f)\in C_q : f<i_1} x_{hf} + \sum_{(h,i_1)\in C_q} x_{hi_1} + \sum_{(h,h^*)\in C_q} x_{hh^*} + x_{h^* i_1} \geqslant 1$$
$$\sum_{(h,f)\in C_q : f<i_1} x_{hf} + \sum_{(h,j_1)\in C_q} x_{hj_1} + \sum_{(h,h^*)\in C_q} x_{hh^*} + x_{h^* j_1} \geqslant 1.$$

The second "$\geqslant$" comes from (12a) and $x_{h^* i_1} + x_{h^* j_1} = 1$. The last equality comes from the fact that for each $h \in Q : h < i_1, h \neq h^*$, (h, h^*, i_1) and (h, h^*, j_1) are cliques in G_F. Thus, by (12b)-(12c) and the triangle inequality (7a) in P_F^k, we have

$$x_{hh^*} \leqslant 1 + x_{hi_1} - x_{h^* i_1} \text{ and } x_{hh^*} \leqslant 1 + x_{hj_1} - x_{h^* j_1}, \forall h \in Q : h < i_1.$$

which leads to $x_{hh^*} = 0, \forall h \in Q : h < i_1, h \neq h^*$.

□

This immediately yields the following desired result.

Corollary 1 *Model 1 is equivalent to Model 4 in the sense that for any feasible solution for Model 1, there exists a feasible solution for 4 such that their objective values are equal; conversely, for any feasible solution x_F for Model 1, there also exists a vector extending x_F such that it is feasible for Model 4.*

Model 4 The clique-based reformulation of the MkP problem

variables: $x \in \{0,1\}^{|F|}$
minimize: (1b)
s.t.: (7a), (7b).

Proof Observe that the objective (1b) is determined by variables $x_{ij}, \forall (i,j) \in E$. Given that $E \subseteq F$ and based on Theorem 1, the result follows. □

Observe that Model 4 has less binary variables than Model 1 when G is not complete. In the most favorable case, where the given graph $G = (V, E)$ is chordal, Model 4 involves no additional binary variables and thus becomes more attractive than Model 2. Recall that Chopra and Rao [11] state that they do not know of any formulation that uses only $|E|$ variables for the MkP problem, for $2 < k < n$ (When $k = 2$, one can use the standard formulation of the max-cut problem [3]. When $k = n$, the cycle inequalities are enough to get a formulation [9]). Here, we are readily to see that if G is chordal, Model 4 uses exactly $|E|$ variables.

3.2 A compact ISDP reformulation

As remarked before, the compact reformulation 4 of Model 1 exploits the sparsity of the chordal extension of the original graph G. Similarly, we show in this section that there exists a clique-based reformulation of Model 3. It is presented in Model 5.

Model 5 The clique-based integer SDP formulation of the MkP problem

$$\textbf{variables: } X_{ij} \in \left\{\frac{-1}{k-1}, 1\right\} \quad (\forall (i,j) \in F) \tag{13a}$$

$$\min \ (4b)$$

$$\text{s.t. } X_{ii} = 1, \quad \forall i \in V. \tag{13b}$$

$$X_{C_r} \in \mathbb{S}^n_+, \ \forall C_r \in \mathcal{K}. \tag{13c}$$

To show that Model 5 is a valid formulation of MkP problem, it is sufficient to show that

$$\mathrm{Proj}_{x_F} \left\{ x \in \left\{\frac{-1}{k-1}, 1\right\}^{n(n-1)/2} : (4c), (4d) \right\} = \left\{ x \in \left\{\frac{-1}{k-1}, 1\right\}^F : (13b), (13c) \right\},$$

where the projection operator $\mathrm{Proj}_{x_F} S$ has been defined in (9). With the bijective mapping defined in (5), we see that the left-hand side and the right-hand side of the above equation correspond to the respective $\mathcal{F}$ and $\mathcal{F}'$ presented below,

$$\mathcal{F} = \left\{ y \in \{0,1\}^{\frac{n(n-1)}{2}} : \ (6a), (6b) \right\},$$

$$\mathcal{F}' = \left\{ y \in \{0,1\}^F : \frac{-1}{k-1} J_{C_r} + \frac{k}{k-1} Y_{C_r} \in \mathbb{S}^n_+, \ \forall C_r \in \mathcal{K}, \ Y_{ii} = 1, \ \forall i \in V \right\}.$$

Thus we just need to prove that $\mathrm{Proj}_{x_F} \mathcal{F} = \mathcal{F}'$. To this end, we exploit a technical lemma that was proposed in [20]. A similar lemma has been used by Eisenblätter's [18] to prove the equivalence of Model 3 and Model 1.

Lemma 4 (Lemma 1, [20]) *For all integers n and k satisfying $2 \leqslant k \leqslant n+1$, there exist k unit vectors $\{u_1, \ldots, u_k\} \in \mathbb{R}^n$ such that $\langle u_l, u_h \rangle = \frac{-1}{k-1}$, for $l \neq h$.*

Theorem 2 *Given integer $2 \leqslant k \leqslant n$, $\mathrm{Proj}_{x_F} \mathcal{F} = \mathcal{F}'$.*

Proof As remarked in Section 2, $\mathcal{F} = P^k$. Thus it holds that $\mathrm{Proj}_{x_F} \mathcal{F} = \mathrm{Proj}_{x_F} P^k$. By Theorem 1, $\mathrm{Proj}_{x_F} P^k = P^k_F$. Thus, it is sufficient to show that $P^k_F = \mathcal{F}'$.

- $\mathcal{F}' \subseteq P^k_F$: Let $\bar{y}$ be any binary vector in $\mathcal{F}'$. We show that $\bar{y}$ satisfies the triangle inequalities (7a) and clique inequalities (7b).
  1. Triangle inequalities (7a): Suppose there exists $i, j, h \in \mathcal{C}_r$ such that (y_{ij}, y_{ih}, y_{jh}) violates (7a). Then, (y_{ij}, y_{ih}, y_{jh}) can only be assigned the values $(1, 1, 0)$,

$$
\begin{tikzcd}
P^k \arrow[r, equal] \arrow[d, dotted] & \mathcal{F} \arrow[d, dotted] \\
P_F^k \arrow[r, equal] & \mathcal{F}'
\end{tikzcd}
$$

Fig. 1 The relationship between the four constraint sets

$(1,0,1)$ or $(0,1,1)$. Since $\bar{y} \in \mathcal{F}'$, the principle sub-matrix of $X_{C_r} = \frac{-1}{k-1} J_{C_r} + \frac{k}{k-1} Y_{C_r}$ corresponding to indices (i,j,h) in the following form

$$\begin{pmatrix} 1 & x_{ij} & x_{ih} \\ x_{ij} & 1 & x_{jh} \\ x_{ih} & x_{jh} & 1 \end{pmatrix} \tag{14}$$

is positive semidefinite. This implies its determinant $1 + 2x_{ij}x_{ih}x_{jh} - x_{ij}^2 - x_{ih}^2 - x_{ij}^2 \geqslant 0$. One can verify that if (y_{ij}, y_{ih}, y_{jh}) is assigned $(1,1,0)$, $(1,0,1)$ or $(0,1,1)$, the determinant becomes $-(\frac{k}{k-1})^2 < 0$, contradiction.

2. Clique inequalities (7b): Suppose there exists $Q \subseteq \mathcal{C}_r : |Q| = k+1$ such that y_Q violates the clique constraint (7b). This implies that $x_{ij} = \frac{-1}{k-1}$, $\forall (i,j) \in Q$, leading to $\sum_{i,j \in Q : i<j} x_{ij} = \frac{-k(k+1)}{2(k-1)}$. Note that $X_Q = \frac{-1}{k-1} J_Q + \frac{k}{k-1} Y_Q \in \mathbb{S}_+^{|k+1|}$. This implies $\mathbf{1}^T X_Q \mathbf{1} = \sum_{i,j \in Q} 2x_{ij} + k + 1 \geqslant 0$, leading to $\sum_{ij} x_{ij} \geqslant \frac{-(k+1)}{2} > \frac{-(k+1)}{2} \times \frac{k}{k-1}$, contradiction.

- $P_F^k \subseteq \mathcal{F}'$: for any $\bar{y} \in P_F^k$, we show that for each $C_r \in \mathcal{K}$ ($|C_r| \geqslant 3$), the matrix $X_{C_r} = \frac{-1}{k-1} J_{C_r} + \frac{k}{k-1} Y_{C_r}$ formed by vector y_{C_r} is positive semi-definite. Given k, let $\{u_1, \dots, u_k\}$ be a set of unit vectors satisfying Lemma 4. Now we construct a real matrix $B \in \mathbb{R}^{n \times |C_r|}$ in the following way. For each $i, j \in C_r$,
  1. If $y_{ij} = 0$, then assign column i, j of matrix B with different unit vectors from set $\{u_1, \dots, u_k\}$.
  2. Otherwise, assign column i, j with the same unit vector from set $\{u_1, \dots, u_k\}$.

  One can now verify that matrix $X_{C_r} = B^T B$, showing that X_{C_r} is positive semidefinite.

□

In summary, the relationship between the aforementioned four constraint sets, i.e., P^k, $\mathcal{F}$, P_F^k and $\mathcal{F}'$, can be depicted in Figure 1, where the dotted arrow represents the projection operator Proj_{x_F}.

Note that the number of integer variables in Model 5 is generally smaller than that of Model 3, making it attractive. The number of constraints in Model 5 grows linearly with respect to the size of the clique set $\mathcal{K}$. The value $|\mathcal{K}|$, as mentioned in Section 3.3, is bounded by $(n-2)$. Reader familiars with SDP with sparsity may realize that Model 5 is closely related to positive semidefinite matrix completion techniques (see, e.g., [26,21,13]). These techniques exploit a common chordal sparsity pattern of all the data matrices. We emphasis that Model 5 cannot be implied by the existing results, due to the binary constraints in Model 3.

Analogously, four continuous relaxations of the four constraint sets can be formed by relaxing the respective integrality constraints. We denote by $\overline{P^k}, \overline{\mathcal{F}}, \overline{P_F^k}$

and $\overline{\mathcal{F}'}$ the respective continuous relaxation sets of $P^k, \mathcal{F}, P_F^k$ and $\mathcal{F}'$. It has been shown in [18] that $\overline{P^k}$ and $\overline{P_{sdp}^k}$ are not contained in one another. We remark that similar results also hold for $\overline{P_F^k}$ and $\overline{\mathcal{F}'}$. This will be illustrated numerically in the subsequent section.

3.3 The construction of the clique set

Model 4 and Model 5 are attractive when they involve less constraints and integer variables than the respective Models 1 and 3. Hence one would like to find an "optimal" (maximal) clique set $\mathcal{K}$ that minimizes the number of constraints and variables in these models. In general, it is hard to determine such a chordal extension G_F and consequently the clique set $\mathcal{K}$ (e.g., [28]).

Nevertheless, for many practical purposes, there are good methods to find chordal extensions such that the size of $\mathcal{K}$ (measured by $\max_{C_r \in \mathcal{K}} |C_r|$) is small. We refer readers to [6] for an excellent survey. Here, we employ the `greedy fill-in heuristic` [28] to obtain G_F and $\mathcal{K}$. The greedy fill-in heuristic attempts to create few new edges, which leads to less integer variables in Models 4 and 5. To make the text self-contained, we present the algorithm below. The term *fill-in* of a vertex refers to the number of pairs of its non-adjacent neighbors. Note that we terminate the algorithm when the number of nodes in H is less than 3 as all valid inequalities (7a)-(7b) are based on maximal clique sets with size $\geqslant 3$.

Algorithm 1: Heuristic to find a chordal extension [28]

```
input  : Graph G = (V, E)
output: A chordal graph G_F = (V, F) of G
Initialisation: H = G, i = 0, G_F = (V, F) with F = E
while # Vertices of H ⩾ 3 do
    if all fill-in values of vertices in H is 0 then
        terminates
    else
        Choose a vertex v with the smallest number of fill-in edge
        Label v with i (i.e., v_i)
        Make the neighbouring vertices N_H(v_i) of v_i a clique
        K_i = N_H(v_i) ∪ {v_i}
        F = F ⋃ {(u, v_i) : u ∈ N_H(v_i)}
        i = i + 1
        Remove v_i from H
    end
end
```

We now need to extract the maximal clique set $\mathcal{K}$ from $\{K_i : i = 1, \ldots, n-2\}$. Due to [22], it is known that $\mathcal{K}$ contains exactly sets K_i for which there exists no $K_i, K_j : i < j$, such that $K_j \subset K_i$. This also shows that the cardinality of $\mathcal{K}$ is bounded by $(n-2)$. It should also be noted that for each two distinct (maximal) cliques $(C_r, C_s) \in \mathcal{K} \times \mathcal{K}$ such that $|C_r \bigcap C_s| \geqslant 3$, there are redundant triangle inequalities (7a) in variables $x_{ij}, (i,j) \in C_r \bigcap C_s$. Similarly, for each two distinct (maximal) cliques $(C_r, C_s) \in \mathcal{K} \times \mathcal{K}$ such that $|C_r \bigcap C_s| \geqslant k+1$, there are redundant clique inequalities (7b). This redundancy can be avoided by checking the occurrence of each associated tuple.

3.4 The separation of valid inequalities

Model 4 can also suffer from a prohibitive number of clique inequalities when the size of some maximal clique set $C_r \in \mathcal{K}$ is large. This issue can be alleviated by a separation algorithm approach. For a maximal clique set $C_r \in \mathcal{K}$ with $|C_r| \geqslant k+1$, the number of clique constraints is $\binom{|C_r|}{k+1}$. This number grows roughly as fast as $|C_r|^k$ as long as $2k \leqslant |C_r|$. Hence, the number of constraints is not bounded by a polynomial in $|C_r|$ and $\log k$. Some heuristics for the separation of cliques and other inequalities are presented in [17,27,33]. We remark that these heuristics can be adapted for both Model 4 and 5 to solve the MkP problem to global optimality. We leave the sufficiently thorough investigation for future research.

4 Numerical experiments

Results in this section illustrate the following key points:

1. Model 4 is more scalable than Model 1 for general graphs.
2. Compared with Model 2, Model 4 is quite attractive when the underlying graph is chordal. For general random graphs, it is less competitive for branch-and-bound although it provides stronger continuous relaxation bounds. This is mainly due to the exponential number of clique inequalities (7b).
3. The continuous relaxation of Model 5 is computationally more scalable than that of Model 3 when the underlying graph has structured sparsity.
4. The continuous relaxation of Model 5 and that of 4 do not dominate each other.

4.1 Test instances

To illustrate the above key points, we randomly generate four sets of sparse graphs. The first set includes chordal graphs called `band`. These instances have been used in [21]. The other sets are generated by the package `rudy` [32]. Similar instances have been used in [23,33] for numerical demonstration.

- `band`: we generate graphs with edges set $E = \{(i,j) \in V \times V : j - i \leqslant \alpha, i < j\}$, where α is 1 plus the partition parameter k, i.e., $\alpha = k+1$. The 50% of edge weights are -1 and the others are 1.
- `springlass2g`: Eleven instances of a toroidal two dimensional grid with gaussian interactions. The graph has n =(rows $\times$ columns) vertices.
- `spinglass2pm`: generates a toroidal two-dimensional grid with ± 1 weights. The grid has size n = (rows $\times$ columns). The percentage of negative weights is 50%.
- `rndgraph`: we generate a random graph of n nodes and density 10%. The edge weights are all 1.

4.2 Implementation and experiments setup

Models 1, 2, 3, 4 and 5 are encoded in C++. MIP problem instances are solved by CPLEX 12.7 [12] with default settings. The SDP relaxation instances are solved by the state-of-the-art solver MOSEK 8 [2] with default tolerance settings, which exploits the sparsity for performance and scalability gains.

The experiments are conducted on a Mac with Inter Core i5 clocked at 2.7 GHz and with 8 GB of RAM. For a fair computational comparison, CPU time is used for all computations. A wall-clock time limit of 10 hours was used for all computations. If no solution is available at solver termination or the solution process is killed by the solver, (−) is reported.

4.3 Analysis of the computational results

Results on Model 1 and Model 4

As remarked in previous sections, both Models 1 and 4 suffer from a prohibitive number of clique inequalities. Thus we fix $k = 3$. For each problem instance, we measure the computational performance of each formulation by computational time and optimality gap (if it has). All Computational time is measure by the CPU time in seconds. As all relaxations lead to a lower bound of the optimum, we quantify optimality gap as

$$\text{gap} = \frac{\text{optimum - lower bound}}{\text{optimum}} \times 100$$

The numerical results are presented in Tables 1. For each problem instance, the statistics on root node relaxation and the full branch-and-bound procedure are reported. These tested cases illustrate the following key points.

1. The compact Model 4 is remarkably more scalable for all tested instances than Model 1.
2. The continuous relaxations of both models are strong. It is also worth mentioning that the optimality gaps at root node for Model 4 are nearly the same with the original model 1.
3. For problem instances with over 100 vertices, the computational time for Model 4 grows exponentially as the problem size increases. This is probably due to the fact that the size of each clique in $\mathcal{K}$ becomes larger, leading to an exponential number of clique inequalities (7b).

Results on Model 4 and Model 2

Let us now compare the results of Models 4, 2 presented in Table 2. First, for `band` instances, the performance of Model 4 is significantly better than Model 2. This is largely because Model 4 has much less binary variables and constraints due the small sizes of its maximal clique sets. Second, for `spinglass2g` problem instances, the performance of Models 4 and 2 are similar. When k or the sizes of instances get larger, Model 4 is less attractive than Model 2. This is probably because the sizes of the maximal clique sets are large, causing a great number of

Table 1 Computational evaluation for Model 1 and Model 4 with $k = 3$

	Model 1				Model 4			
	Root node		Branch-and-bound		Root node		Branch-and-bound	
spinglass2g								
$(k, \lvert V \rvert)$	Time (s)	Gap (%)	Time (s)	Nodes	Time (s)	Gap (%)	Time (s)	Nodes
$(3, 3 \times 3)$	0.03	0.00	0.03	0	0.03	0.00	0.03	0
$(3, 4 \times 4)$	0.20	0.00	0.34	0	0.02	0.00	0.38	0
$(3, 5 \times 5)$	1.42	0.00	5.54	0	0.09	0.17	0.16	0
$(3, 6 \times 6)$	8.12	0.00	9.69	0	0.17	0.00	0.30	0
$(3, 7 \times 7)$	25.32	0.48	14075.4	4	0.76	0.98	10.80	0
$(3, 8 \times 8)$	127.13	0.00	-	-	2.05	0.00	2.52	0
$(3, 9 \times 9)$	-	-	-	-	4.66	0.91	40.98	0
$(3, 10 \times 10)$	-	-	-	-	6.19	0.00	6.74	0
$(3, 11 \times 11)$	-	-	-	-	5.00	0.00	46.11	0
$(3, 12 \times 12)$	-	-	-	-	7.86	0.00	9.82	0
$(3, 13 \times 13)$	-	-	-	-	14.02	0.00	229.40	0
$(3, 14 \times 14)$	-	-	-	-	16.12	0.00	23.13	0
$(3, 15 \times 15)$	-	-	-	-	19.57	0.03	1420.8	0
$(3, 16 \times 16)$	-	-	-	-	16.01	0.18	58900	28
spinglass2pm								
$(3, 3 \times 3)$	0.02	0.00	0.05	0	0.03	0.00	0.04	0
$(3, 4 \times 4)$	0.20	0.00	0.26	0	0.05	0.00	0.06	0
$(3, 5 \times 5)$	1.72	0.00	1.93	0	0.09	0.00	0.12	0
$(3, 6 \times 6)$	6.89	2.20	8.89	0	0.21	2.31	0.42	0
$(3, 7 \times 7)$	26.53	2.00	43.88	4	0.77	2.10	6.96	0
$(3, 8 \times 8)$	81.55	0.00	-	-	2.05	0.00	4.97	0
$(3, 9 \times 9)$		-	-	-	2.03	1.21	31.52	0
$(3, 10 \times 10)$		-	-	-	2.78	0.16	21.11	0
$(3, 11 \times 11)$		-	-	-	4.28	0.00	6152.6	28
$(3, 12 \times 12)$		-	-	-	6.73	0.00	18.27	0
$(3, 13 \times 13)$		-	-	-	9.61	1.08	859.37	0
$(3, 14 \times 14)$		-	-	-	12.63	1.18	-	-

clique inequalities. Third, the strength of Model 4 is generally much stronger than that of 2. This is illustrated by instances of `spinglass2g`, where the continuous relaxation of Model 4 leads to 0 optimality gap.

Results on Model 3 and Model 5

We now compare the performance of Model 3 and Model 5 with respect to their continuous relaxation values and solution time. Since the problem is a minimisation problem, the higher the value is, the stronger is the lower bound. The numerical results are summarized in Table 3.

Overall, we remark that that the continuous relaxation of Model 5 reduces significantly the computational time compared with that of Model 3, though a bit inferior in the solution quality. In addition, as expected, the solution time for

Table 2 Continuous relaxations of Model 2 and Model 4

	Model 2				Model 4			
	Root node		Branch-and-bound		Root node		Branch-and-bound	
$(k, \|V\|)$	Time (s)	Gap (%)	Time (s)	Nodes	Time (s)	Gap (%)	Time (s)	Nodes
`band`								
$(3, 50)$	0.09	100.00	1588.65	345246	0.03	5.38	0.28	0
$(3, 100)$	0.25	100.00	-	-	0.04	5.47	0.76	0
$(3, 150)$	0.51	98.64	-	-	0.06	4.78	1.58	0
$(3, 200)$	0.35	100.00	-	-	0.07	5.51	5.14	0
$(3, 250)$	0.47	100.00	-	-	0.08	5.52	5.61	0
$(4, 50)$	0.09	107.14	-	-	0.04	12.21	4.86	0
$(4, 100)$	0.25	111.40	-	-	0.08	13.39	13.05	0
$(4, 150)$	0.38	112.79	-	-	0.09	13.78	36.36	0
$(4, 200)$	0.35	112.55	-	-	0.16	13.59	70.74	122
$(4, 250)$	0.44	113.14	-	-	0.21	13.78	112.85	254
`spinglass2g`								
$(3, 10 \times 10)$	0.06	3.79	10.28	58	6.19	0.00	6.74	0
$(3, 11 \times 11)$	0.07	8.5	17.63	145	5.00	0.00	46.11	0
$(3, 12 \times 12)$	0.09	10.61	16.00	89	7.86	0.00	9.82	0
$(3, 13 \times 13)$	0.11	9.41	34.01	584	14.02	0.00	229.40	0
$(3, 14 \times 14)$	0.13	9.01	67.20	2126	16.12	0.00	23.13	0
$(3, 15 \times 15)$	0.15	10.01	77.30	860	19.57	0.03	1420.8	0
$(4, 10 \times 10)$	0.07	10.38	27.00	3075	6.99	0.00	22.70	0
$(4, 11 \times 11)$	0.09	7.50	30.50	3654	15.212	0.00	50.47	0
$(4, 12 \times 12)$	0.11	9.50	75.40	3068	24.696	0.00	144.87	0
$(4, 13 \times 13)$	0.14	9.19	79.03	4211	76.670	0.00	-	-
$(4, 14 \times 14)$	0.17	8.10	42.68	882	76.940	0.00	-	-
$(4, 15 \times 15)$	0.34	7.20	430.39	11314	279.98	0.00	-	-

Model 5 grows linearly with respect to the size of the graph while the computational time for Model 3 grows more significantly as the instance size increases.

Results on Model 4 and Model 5

We compare the strength of Model 4 and Model 5 with respect to their continuous relaxation values and solution time. Again the problem is a minimization problem, the higher the value is, the stronger is the lower bound.

Our previous numerical results show that the computational time and bounds of both Model 4 and Model 5 are quite similar for test instances of **`band`**, **`spinglass2g`**, **`spinglass2pm`**. To contrast these two compact models, we present the numerical results for tests instances of **`rndgraph`**. The numerical results are summarized in Table 4. We outline the following key observations.

1. The strength of the continuous relaxation of Model 5 neither dominates nor is dominated by that of 4. See, for instance, problem instance $(3, 40)$ and $(3, 60)$.
2. For larger problem instances ($|V| \geqslant 100$), Model 5 appear much more attractive than Model 4 in terms of computational scalability and bound quality.

Table 3 Continuous relaxations for Model 3 and Model 5 with $k = 3$

spinglass2g				
	Model 3		Model 5	
$(k, \|V\|)$	Time (s)	Value	Time (s)	Value
$(3, 11 \times 11)$	125.84	-8.41227e+06	1.74	-8.42335e+06
$(3, 12 \times 12)$	359.43	-1.06568e+07	2.29	-1.06777e+07
$(3, 13 \times 13)$	866.25	-1.20763e+07	4.00	-1.2088e+07
$(3, 14 \times 14)$	2215.69	-1.4147e+07	5.64	-1.416e+07
$(3, 15 \times 15)$	5072.07	-1.80742e+07	10.33	-1.80904e+07
$(4, 11 \times 11)$	130.806	-8.44258e+06	1.88	-8.45242e+06
$(4, 12 \times 12)$	343.846	-1.07031e+07	2.43	-1.07238e+07
$(4, 13 \times 13)$	1164.69	-1.2127e+07	4.34	-1.214e+07
$(4, 14 \times 14)$	2820.33	-1.42114e+07	5.59	-1.42234e+07
$(4, 15 \times 15)$	6951.17	-1.81605e+07	10.88	-1.81734e+07
spinglass2pm				
$(3, 11 \times 11)$	876.87	-107.67	1.21	-108.02
$(3, 12 \times 12)$	2227.27	-129.40	1.66	-129.83
$(3, 13 \times 13)$	5500.92	-150.58	2.54	-151.24
$(3, 14 \times 14)$	11349.41	-175.72	3.37	-176.41
$(3, 15 \times 15)$	33188.43	-197.88	5.67	-198.66
$(4, 11 \times 11)$	102.62	-108.70	1.26	-109.16
$(4, 12 \times 12)$	291.59	-130.23	1.73	-130.64
$(4, 13 \times 13)$	729.29	-152.08	2.80	-152.68
$(4, 14 \times 14)$	1905.11	-177.43	3.56	-178.11
$(4, 15 \times 15)$	4791.15	-199.90	6.67	-200.67

Table 4 Continuous relaxations of Model 4 and Model 5 with $k = 3$ (rndgraph).

	Model 4		Model 5	
$(k, \|V\|)$	Time (s)	Value	Time (s)	Value
$(3, 10)$	0.01	0.00	0.06	0.00
$(3, 20)$	0.00	0.00	0.03	0.00
$(3, 30)$	0.01	0.00	0.03	0.00
$(3, 40)$	0.07	0.67	0.05	0.14
$(3, 50)$	0.61	0.00	0.12	0.00
$(3, 60)$	1.97	1.11	0.25	0.41
$(3, 70)$	3.75	4.01	0.49	2.53
$(3, 80)$	12.44	4.94	1.09	3.47
$(3, 90)$	24.46	10.96	1.69	9.86
$(3, 100)$	49.76	14.67	3.73	16.45
$(3, 110)$	85.13	23.53	5.62	26.11
$(3, 120)$	209.22	28.86	12.21	35.53
$(3, 130)$	326.67	37.00	21.07	48.57
$(3, 140)$	-	-	34.40	61.60
$(3, 150)$	-	-	48.33	78.47

5 Conclusion

This work introduces two compact reformulations of the minimum k-partition problem exploiting the structured sparsity of the underlying graph. The first model is a binary linear program while the second is an integer semidefinite program. Both are based on the maximal clique set corresponding to the chordal extension of the original graph. Numerical results show that the proposed models numerically dominate state-of-the-art formulations.

Based on the results presented in this paper, several research directions can be considered. First, alternative algorithms may be implemented to obtain optimal clique sets minimizing the number of integer variables. Second, separation algorithms for valid inequalities can be investigated. Third, specialized branch-and-bound algorithms for the compact SDP Model 5 can be considered. Lastly, combining Models 4 and 2 to obtain a novel integer LP formulation is also left for future research.

Acknowledgements This research was partly funded by the Australia Indonesia Centre.

References

1. Anjos, M.F., Ghaddar, B., Hupp, L., Liers, F., Wiegele, A.: Solving k-way graph partitioning problems to optimality: The impact of semidefinite relaxations and the bundle method. In: Facets of combinatorial optimization, pp. 355–386. Springer (2013)
2. ApS, M.: MOSEK Fusion API for C++ (2017). URL http://docs.mosek.com/8.0/cxxfusion.pdf
3. Barahona, F., Mahjoub, A.R.: On the cut polytope. Mathematical programming **36**(2), 157–173 (1986)
4. Ben-Ameur, W., Ouorou, A., Wang, G.: Convex and concave envelopes: Revisited and new perspectives. Operations Research Letters (2017)
5. Blair, J.R., Peyton, B.: An introduction to chordal graphs and clique trees. In: Graph theory and sparse matrix computation, pp. 1–29. Springer (1993)
6. Bodlaender, H.L., Koster, A.M.: Treewidth computations i. upper bounds. Information and Computation **208**(3), 259–275 (2010)
7. Caprara, A., Fischetti, M.: $\{0, 1/2\}$-chvátal-gomory cuts. Mathematical Programming **74**(3), 221–235 (1996)
8. Carlson, R., Nemhauser, G.L.: Scheduling to minimize interaction cost. Operations Research **14**(1), 52–58 (1966)
9. Chopra, S.: The graph partitioning polytope on series-parallel and 4-wheel free graphs. SIAM journal on discrete mathematics **7**(1), 16–31 (1994)
10. Chopra, S., Rao, M.: Facets of the k-partition polytope. Discrete Applied Mathematics **61**(1), 27–48 (1995)
11. Chopra, S., Rao, M.R.: The partition problem. Mathematical Programming **59**(1-3), 87–115 (1993)
12. Cplex, I.I.: 12.2 users manual. ILOG. (2010)
13. De Klerk, E.: Exploiting special structure in semidefinite programming: A survey of theory and applications. European Journal of Operational Research **201**(1), 1–10 (2010)
14. Deza, M., Grötschel, M., Laurent, M.: Complete descriptions of small multicut polytopes. Inst. für Mathematik (1990)
15. Deza, M., Grötschel, M., Laurent, M.: Clique-web facets for multicut polytopes. Mathematics of Operations Research **17**(4), 981–1000 (1992)
16. Deza, M., Laurent, M.: Geometry of cuts and metric embeddings. Springer-Verlag, New York (1997)
17. Eisenblätter, A.: Frequency assignment in gsm networks: Models, heuristics, and lower bounds. Ph.D. thesis (2001)

18. Eisenblätter, A.: The semidefinite relaxation of the k-partition polytope is strong. In: International Conference on Integer Programming and Combinatorial Optimization, pp. 273–290. Springer (2002)
19. Fairbrother, J., Letchford, A.N.: Projection results for the k-partition problem. In: BOOK OF ABSTRACTS, p. 101 (2016)
20. Frieze, A., Jerrum, M.: Improved approximation algorithms for maxk-cut and max bisection. Algorithmica **18**(1), 67–81 (1997)
21. Fukuda, M., Kojima, M., Murota, K., Nakata, K.: Exploiting sparsity in semidefinite programming via matrix completion i: General framework. SIAM Journal on Optimization **11**(3), 647–674 (2001)
22. Fulkerson, D., Gross, O.: Incidence matrices and interval graphs. Pacific journal of mathematics **15**(3), 835–855 (1965)
23. Ghaddar, B., Anjos, M.F., Liers, F.: A branch-and-cut algorithm based on semidefinite programming for the minimum k-partition problem. Annals of Operations Research **188**(1), 155–174 (2011)
24. Goemans, M.X., Williamson, D.P.: Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. Journal of the ACM (JACM) **42**(6), 1115–1145 (1995)
25. Golumbic, M.C.: Algorithmic graph theory and perfect graphs. Academic Press, New York (1980)
26. Grone, R., Johnson, C.R., Sá, E.M., Wolkowicz, H.: Positive definite completions of partial hermitian matrices. Linear algebra and its applications **58**, 109–124 (1984)
27. Kaibel, V., Peinhardt, M., Pfetsch, M.E.: Orbitopal fixing. Discrete Optimization **8**(4), 595–610 (2011)
28. Koster, A.M., Bodlaender, H.L., Van Hoesel, S.P.: Treewidth: computational experiments. Electronic Notes in Discrete Mathematics **8**, 54–57 (2001)
29. Letchford, A.N.: On disjunctive cuts for combinatorial optimization. Journal of Combinatorial Optimization **5**(3), 299–315 (2001)
30. Michael, R.G., David, S.J.: Computers and intractability: a guide to the theory of np-completeness. WH Free. Co., San Fr pp. 90–91 (1979)
31. Rendl, F.: Semidefinite relaxations for partitioning, assignment and ordering problems. Annals of Operations Research **240**(1), 119–140 (2016)
32. Rinaldi, G.: Rudy, a graph generator (1998)
33. de Sousa, V.J.R., Anjos, M.F., Le Digabel, S.: Computational study of valid inequalities for the maximum k-cut problem. Annals of Operations Research pp. 1–23 (2016)
34. Wang, G.: Relaxations in mixed-integer quadratically constrained programming and robust programming. Ph.D. thesis, Evry, Institut national des télécommunications (2016)