Projection Results for the k-Partition Problem

Jamie Fairbrother[*]   Adam N. Letchford[†]

Abstract

The *k-partition problem* is an $\mathcal{NP}$-hard combinatorial optimisation problem with many applications. Chopra and Rao introduced two integer programming formulations of this problem, one having both node and edge variables, and the other having only edge variables. We show that, if we take the polytopes associated with the 'edge-only' formulation, and project them into a suitable subspace, we obtain the polytopes associated with the 'node-and-edge' formulation. This result enables us to derive new valid inequalities and separation algorithms, and also to shed new light on certain SDP relaxations. Computational results are also presented.

Keywords: graph partitioning; polyhedral combinatorics; branch-and-cut; semidefinite programming.

1   Introduction

The *k-partition problem* (k-PP) is a strongly $\mathcal{NP}$-hard combinatorial optimisation problem, first defined in [5]. We are given an undirected graph G, with vertex set V and edge set E, a rational weight w_e for each edge $e \in E$, and an integer k with $2 \le k \le |V|$. The task is to partition V into k or fewer subsets (called "clusters"), such that the sum of the weights of the edges that have both end-vertices in the same cluster is minimised. The k-PP has applications in statistical clustering, numerical linear algebra, telecommunications, VLSI layout, sports team scheduling and statistical physics (see, e.g., [14, 18, 31]).

Note that the k-PP is equivalent to the *max-k-cut* problem, in which one wishes to maximise the sum of the weights of the edges that have *exactly one* end-vertex in the same cluster (see [15]). In particular, when $k = 2$, we have the well-known *max-cut* problem, which is known to be strongly $\mathcal{NP}$-hard

---

[*]STOR-i Centre for Doctoral Training, Lancaster University, Lancaster LA1 4YF, United Kingdom. E-mail: `J.Fairbrother@lancaster.ac.uk`

[†]Department of Management Science, Lancaster University, Lancaster LA1 4YW, United Kingdom. E-mail: `A.N.Letchford@lancaster.ac.uk`

(see [16]). Moreover, the problem of checking whether G is k-colourable can be reduced to the k-PP. Thus, the k-PP is strongly $\mathcal{NP}$-hard for all fixed k, and this is so even when $k = 3$ and G is planar (see again [16]). Not only that, but the special case of the k-PP in which G is a complete graph and $k = |V|$, called the *clique partitioning problem* (CPP), is strongly $\mathcal{NP}$-hard as well [21].

In their seminal paper [7], Chopra and Rao presented two different *0-1 linear programming* (0-1 LP) formulations of the k-PP. One of these formulations has both node and edge variables, whereas the other has only edge variables. For each formulation, several families of strong valid linear inequalities (a.k.a. cutting planes) have been discovered (e.g., [7, 8, 11, 21]). For some of these families, we also have efficient separation algorithms (e.g., [3, 4, 12, 27]). There is also a parallel literature concerned with *semidefinite programming* (SDP) relaxations of the k-PP (e.g., [14, 15, 18, 25, 31, 33, 34]).

The main result in this paper is the following. If we take the polytopes associated with the 'edge-only' formulations, and project them into a suitable subspace, we obtain the polytopes associated with the 'node-and-edge' formulations. Although this result is fairly easy to derive, it is very useful. Specifically, it leads to new valid inequalities and separation algorithms for the 'node-and-edge' formulations, and it sheds new light on the SDP relaxations given in [15, 25, 31].

The paper is structured as follows. A literature review is given in Section 2. The projection results are given in Section 3. The new inequalities and separation algorithms, along with our remarks on SDP relaxations, are presented in Section 4. Some computational results are given in Section 5. Finally, some concluding remarks are given in Section 6.

We use the following (standard) notation throughout the paper. The number of nodes and edges in G is denoted by n and m, respectively. For a given positive integer p, we let K_p denote the complete graph on p nodes. Its vertex set is $\{1, \dots, p\}$, which we denote by V_p. Its edge set is denoted by E_p. We also let I_p, e_p and J_p denote the identity matrix of order p, the all-ones vector with p components, and the (square) all-ones matrix of order p, respectively. Given a real symmetric matrix M, we write $M \succeq 0$ to indicate that M is positive semidefinite (psd).

We also use the following (again standard) terminology. A *clique* is a set of pairwise adjacent nodes. A set $C \subseteq E$ is a *cycle* if it induces a connected subgraph of G in which every node has degree 2. The nodes in the subgraph are denoted by $V(C)$. Two disjoint sets $R, S \subset E$ form a *wheel* if R is a cycle and there exists a node $h \in V \setminus V(R)$ such that $S = \big\{ \{v, h\} : v \in V(R) \big\}$. (The set R is called the *rim*, the edges in S are called *spokes*, and h is called the *hub*.) Two disjoint sets $R, S \subset E$ and an edge $\{h, h'\} \in E \setminus R$ form a *bicycle wheel* if R is a cycle and $S = \big\{ \{v, h\}, \{v, h'\} : v \in V(R) \big\}$.

2

2 Literature Review

We now review the relevant literature. We cover the two 0-1 LP formulations in Subsections 2.1 and 2.2, separation algorithms in Subsection 2.3, and SDP relaxations in Subsection 2.4.

2.1 The node-and-edge formulation

Chopra & Rao [7] present the following 0-1 LP formulation of the k-PP, which has $nk + m$ variables. For each $v \in V$ and for $c = 1, \ldots, k$, let x_{vc} be a binary variable, indicating whether node v lies in the cth cluster. Also, for each $e \in E$, let y_e be a binary variable, indicating whether the end-nodes of e lie in the same cluster. Then:

$$
\begin{align}
\min \quad & \sum_{e \in E} w_e y_e \tag{1} \\
\text{s.t.} \quad & \sum_{c=1}^{k} x_{vc} = 1 && (v \in V) \tag{2} \\
& y_{uv} \geq x_{uc} + x_{vc} - 1 && (\{u, v\} \in E,\ c = 1, \ldots, k) \tag{3} \\
& x_{uc} \geq x_{vc} + y_{uv} - 1 && (\{u, v\} \in E,\ c = 1, \ldots, k) \tag{4} \\
& x_{vc} \geq x_{uc} + y_{uv} - 1 && (\{u, v\} \in E,\ c = 1, \ldots, k) \tag{5} \\
& x_{vc} \in \{0, 1\} && (v \in V,\ c = 1, \ldots, k) \tag{6} \\
& y_{uv} \in \{0, 1\} && (\{u, v\} \in E). \tag{7}
\end{align}
$$

We will let $P^{xy}(G, k)$ denote the associated polytope, i.e., the convex hull in $\mathbb{R}^{nk+m}$ of pairs (x, y) satisfying (2)–(7). We remark that the inequalities (4) and (5) can be removed when all of the edge weights are non-negative, but this leads to a different polytope, which has not been studied.

Chopra and Rao show that the inequalities (3)–(5), together with non-negativity constraints, define facets of $P^{xy}(G)$. They also show that the following inequalities define facets for $k \geq 3$:

- *Clique* inequalities, which take the form:

$$
\sum_{u, v \in C} y_{uv} \geq \binom{t+1}{2} r + \binom{t}{2}(k - r), \tag{8}
$$

  for all $C \subseteq V$ inducing a clique in G with $t = \lfloor |C|/k \rfloor \geq 1$ and $r = |C| \bmod k \neq 0$.

- *Cycle* inequalities, which take the form:

$$
y_f \geq \sum_{e \in C \setminus \{f\}} y_e - |C| + 2, \tag{9}
$$

  for all chordless cycles $C \subseteq E$ and for all $f \in C$.

- *Odd wheel* inequalities, which take the form:

$$\sum_{e \in R} y_e \geq \sum_{e \in S} y_e - \lfloor |R|/2 \rfloor \,, \tag{10}$$

  for all R, S forming a wheel in G with $|R| \geq 3$ and odd.

- *Odd bicycle wheel* inequalities, which take the form:

$$y_{hh'} + \sum_{e \in S} y_e \geq \sum_{e \in R} y_e - (|R| - 1), \tag{11}$$

  for all $R, S, \{h, h'\}$ forming a bicycle wheel in G with $|R|$ odd.

- *Odd cycle* inequalities:

$$\sum_{e \in C} y_e \geq \sum_{v \in V(C)} x_{ic} - \lfloor |C|/2 \rfloor \,, \tag{12}$$

  for all cycles $C \subseteq E$ with $|C|$ odd, and for $c = 1, \ldots, k$.

2.2 The edge formulation

Chopra & Rao [7] also presented the following 0-1 LP formulation. Add dummy edges (of zero weight) so that G becomes K_n. Have one y variable for each $e \in E_n$, but no x variables. Then:

$$\min \quad \sum_{e \in E} w_e y_e \tag{13}$$
$$\text{s.t.} \quad y_{uv} \geq y_{uw} + y_{vw} - 1 \quad (\{u, v, w\} \subset V_n) \tag{14}$$
$$\sum_{u,v \in C} y_{uv} \geq 1 \quad (C \subset V_n : |C| = k + 1) \tag{15}$$
$$y_e \in \{0, 1\} \quad (e \in E_n). \tag{16}$$

Note that the constraints (14) are equivalent to the inequalities (9) when $G = K_n$, and the constraints (15) are a special case of the inequalities (8). We will let $P^y(K_n, k)$ denote the associated polytope.

By definition, any valid inequality for $P^{xy}(K_n, k)$ that involves only y variables is valid also for $P^y(K_n, k)$. This includes not only the inequalities (14), (15), but also inequalities (8), (10) and (11). Still more inequalities for $P^y(K_n, k)$ are presented in [8, 11].

The polytope $P^y(K_n, n)$, called the *clique partitioning polytope*, has been studied in depth [1, 11, 12, 20, 21, 22, 28, 29, 30, 32]. Among the many families of strong valid inequalities known for it are, e.g., the *2-partition* and *2-chorded odd cycle* inequalities [21] and the *weighted (s,T)* inequalities [30]. Note that any valid inequality for $P^y(K_n, n)$ is valid also for $P^y(K_n, k)$ and $P^{xy}(K_n, k)$ for all $k \leq n$.

It would be attractive to have a 0-1 LP formulation with only m variables, i.e., one y variable for each edge in E. Chopra & Rao [7] state that they

do not know of such a formulation for $2 < k < n$. (When $k = 2$, one can use the standard formulation of the max-cut problem [2]. When $k = n$, the cycle inequalities are enough to get a formulation [6].) We return to this issue in Subsection 3.1.

2.3 Separation algorithms

For a given family of inequalities, a *separation algorithm* is a routine that takes a solution of an LP relaxation as input, and searches for violated inequalities in that family [19]. Chopra and Rao [7] point out that separation can be done in polynomial time for the cycle inequalities (9) and odd cycle inequalities (12) using the approach in [2]. Separation is $\mathcal{NP}$-hard for the clique inequalities (8), by a trivial reduction from the maximum clique problem [13]. Heuristics for clique separation are presented in [13, 24]. Deza *et al.* [12] show that separation for the odd wheel inequalities (10) can be done in polynomial time, and point out that separation for the odd bicycle wheel inequalities (11) can also be done in polynomial time, by adapting the approach in [17].

The complexity of separation for the other known families of inequalities is unknown. Separation heuristics for 2-partition and weighted (s, T) inequalities are given in [20] and [30], respectively. It is shown in [3, 4, 27, 29] that one can separate in polynomial time over various families of valid inequalities that *include* all 2-chorded odd cycle inequalities.

2.4 SDP relaxations

Finally, we briefly review SDP relaxations of the k-PP. Freize & Jerrum [15] use the relaxation:

$$
\begin{aligned}
\min \quad & \tfrac{k-1}{k} \sum_{e \in E} w_e \left(Y_e + \tfrac{1}{k-1} \right) \\
\text{s.t.} \quad & \operatorname{diag}(Y) = e_n \\
& Y_e \geq \tfrac{-1}{k-1} \qquad (e \in E_n) \\
& Y \succeq 0.
\end{aligned}
$$

The idea here is that Y represents a feasible solution if Y_{uv} takes the value 1 when nodes u and v are in the same cluster, and $-1/(k-1)$ otherwise.

Observe that Y is related to the vector y in the edge formulation via the identities $y_e = \tfrac{k-1}{k} Y_e + \tfrac{1}{k}$ and $Y_e = \tfrac{k}{k-1} y_e - \tfrac{1}{k-1}$ for all $e \in E_n$. Using these mappings, valid inequalities for $P^y(K_n, k)$ can be used to strengthen the SDP [14, 18, 34]. The same observation led Rendl [31] to propose the

following equivalent, yet more natural SDP:

$$\begin{aligned} \min \quad & \sum_{e \in E} w_e \tilde{Y}_e \\ \text{s.t.} \quad & \text{diag}(\tilde{Y}) = e_n \\ & \tilde{Y}_{uv} \geq 0 \qquad (\{u, v\} \in E_n) \\ & k\tilde{Y} - J_n \succeq 0. \end{aligned}$$

Here, $\tilde{Y}_e$ plays the same role as y_e in the edge formulation.

De Klerk *et al.* [25] consider an alternative approach. Let $z \in \{0, 1\}^{nk}$ be a vector such that, for $v = 1, \ldots, n$ and $c = 1, \ldots, k$, node n lies in cluster c if and only if the component of z in position $n(c-1) + v$ takes the value 1. From a consideration of the matrix

$$\begin{pmatrix} 1 \\ z \end{pmatrix} \begin{pmatrix} 1 \\ z \end{pmatrix}^T = \begin{pmatrix} 1 & z^t \\ z & zz^T \end{pmatrix},$$

they derive an SDP relaxation in which the matrix variable is of order $nk+1$. It is shown in [25] that this relaxation gives the same lower bound as the relaxations mentioned above, but provides extra information that can be exploited in an approximation algorithm.

Finally, we mention two recent papers. Van Dam & Sotirov [10] examine an SDP relaxation whose solution has a closed form (involving Eigenvalues). De Sousa *et al.* [34] show empirically that odd wheel and odd bicycle wheel inequalities can be useful for strengthening SDP relaxations.

3 The Projection Results

In this section, we present our projection results. We proceed in two steps, which are covered in the following two subsections.

3.1 From $P^{xy}(G, k)$ and $P^y(K_n, k)$ to $P^y(G, k)$

We start with the following definition.

Definition 1 *Let $G = (V, E)$ be an undirected graph with n nodes and m edges. The projection of $P^{xy}(G, k)$ into y-space, which is also the projection of $P^y(K_n, k)$ into $\mathbb{R}^m$, will be denoted by $P^y(G, k)$.*

Finding valid and even facet-defining inequalities for $P^y(G, k)$ is rather easy, as shown by the following two lemmas.

Lemma 1 *The clique inequalities (8), cycle inequalities (9), odd wheel inequalities (10) and odd bicycle wheel inequalities (11) are valid for $P^y(G, k)$, and they define facets of $P^y(G, k)$ if and only if they define facets of $P^{xy}(G, k)$.*

Proof. These inequalities are valid for $P^{xy}(G, k)$ and have zero coefficients for all x variables. $\square$

Lemma 2 *Any valid inequality for $P^y(K_n, k)$ is valid also for $P^y(G, k)$, provided that the variables with non-zero coefficients correspond to edges in E.*

Proof. This follows directly from the fact that $P^y(G, k)$ is the projection of $P^y(K_n, k)$ into $\mathbb{R}^m$. $\square$

So, for example, the 2-partition, 2-chorded odd cycle and weighted (s, T) inequalities are valid for $P^y(G, k)$ whenever G contains the corresponding graph as a subgraph.

The main reason that we are interested in $P^y(G, k)$ is that, as we will see in the next subsection, valid inequalities for $P^y(G', k)$, where G' is a suitable graph, can be used to derive new valid inequalities for $P^{xy}(G)$. There is however another motivation for studying $P^y(G, k)$: it is possible to compute lower bounds for the k-PP with a cutting-plane algorithm that has only m variables, in which the cutting planes are valid inequalities for $P^y(G, k)$.

Now, recall from the end of Subsection 2.2 that, for $2 < k < n$, Chopra and Rao were unable to find a *formulation* of the k-PP that uses only m variables. In fact, a simple formulation is:

$$
\begin{aligned}
\min \quad & \sum_{e \in E} w_e y_e \\
\text{s.t.} \quad & y_f \geq \sum_{e \in C \setminus \{f\}} y_e - |C| + 2 \quad (C \in \mathcal{C}, f \in C) \\
& \sum_{e \in F} y_e \geq 1 \quad\quad\quad\quad (F \in \mathcal{F}) \\
& y_e \in \{0, 1\} \quad\quad\quad\quad (e \in E),
\end{aligned}
$$

where $\mathcal{C}$ contains all chordless cycles $C \subset E$, and $\mathcal{F}$ contains all minimal sets $F \subset E$ that induce a subgraph of G that is not k-colourable. Unfortunately, this formulation seems to be of little practical use, since testing k-colourability is $\mathcal{NP}$-hard.

To close this subsection, we remark that, when $k = 2$, the polytope $P^y(G, k)$ reduces to the *cut polytope* of G (or, more precisely, a reflection of the cut polytope, obtained by complementing all variables). In [2], it is shown how to obtain new facets of the cut polytope from old ones, by applying graph operations such as "edge subdivision" and "node splitting". We suspect that one could obtain new facets of $P^y(G, k)$ in a similar way, but we do not explore this here, for brevity.

3.2 From $P^y(G, k)$ (almost) to $P^{xy}(G, k)$

To proceed further, we will need the following definition and lemma.

Definition 2 *Given a graph $G = (V, E)$ and a positive integer k, the "k-augmented graph", denoted by G^k, is a graph with node set $V \cup \{d_1, \ldots, d_k\}$ and edge set $D \cup E \cup F$, where $D = \{\{v, d_c\} : v \in V, c = 1, \ldots, k\}$ and $F = \{\{d_c, d_{c'}\} : 1 \le c < c' \le k\}$. The nodes $d_1, \ldots, d_k$ are called "dummy" nodes. The edges in D and F are called "dummy" edges and "forbidden" edges, respectively.*

Lemma 3 *Suppose that $y \in \{0, 1\}^{|D|+|E|+|F|}$ is an extreme point of $P^y(G^k, k)$ such that $y_e = 0$ for all forbidden edges $e \in F$. Then, in the corresponding k-PP solution, there are exactly k clusters, and exactly one dummy node lies in each cluster.*

Proof. From the definition of F, we have $y_e = 0$ for every edge connecting two dummy nodes. This means that no two dummy nodes can lie in the same cluster, which implies in turn that each dummy node lies in a separate cluster. The result then follows from the fact that there are k dummy nodes, and the fact that a feasible k-PP solution has no more than k clusters. $\square$

The main result in this section is then the following.

Theorem 1 *For a given G and k, let G^k be the corresponding augmented graph. Suppose we take $P^y(G^k, k)$ and perform the following operations:*

1. *Take the face of $P^y(G^k, k)$ induced by the hyperplanes $y_e = 0$ for all $e \in F$.*

2. *Project the face into $\mathbb{R}^{D+E}$.*

3. *For each dummy edge $e = \{v, d_c\}$, change the name of the variable y_e to x_{vc}.*

The resulting polytope is $P^{xy}(G, k)$.

Proof. From Lemma 3, every extreme point of $P^y(G^k, k)$ lying on the stated face corresponds to a k-partition in which exactly one dummy node lies in each cluster. For $c = 1, \ldots, k$, let us call the cluster containing node d_c "cluster c". Then, for any given dummy edge $e = \{v, d_c\} \in D$, $y_e = 1$ if and only if node v lies in cluster c, i.e., if and only if $x_{vc} = 1$ in the node and edge formulation. Moreover, $|D| = nk$, and therefore the projection lies in $\mathbb{R}^{nk+m}$, the same space in which $P^{xy}(G, k)$ lies. $\square$

Theorem 1 has the following useful corollary:

Corollary 1 *If the inequality $\alpha^T y \le \beta$ is valid for $P^y(G^k, k)$, then the "projected" inequality*

$$\sum_{e=\{v,d_c\}\in D} \alpha_e x_{vc} + \sum_{e\in E} \alpha_e y_e \le \beta$$

is valid for $P^{xy}(G, k)$.

Corollary 1 sheds light on the valid inequalities for $P^{xy}(G, k)$ given in Chopra & Rao [7]. In particular:

- The inequalities (14) are valid for $P^y(G^k, k)$, provided that $\{u, v, w\}$ induces a triangle in G^k. If we assume that $\{u, v\} \in E$ and identify node w with the dummy node d_c, we obtain the inequalities (3).

- If instead we assume that $\{u, w\} \in E$ and identify node v with the dummy node d_c, we obtain (up to a relabelling of the remaining nodes) the inequalities (4) and (5).

- The odd wheel inequalities (10) are valid for $P^y(G^k, k)$, provided that G^k contains the odd wheel as a subgraph. If we assume that $R \subseteq E$ and identify the hub h with the dummy node d_c, we obtain the odd cycle inequalities (12).

Further implications of Theorem 1 and Corollary 1 are given in the next section.

We remark that a necessary condition for a "projected" inequality to define a facet of $P^{xy}(G, k)$ is that the original inequality defines a facet of $P^y(G^k, k)$. We do not know whether this condition is also sufficient.

4 Implications

In this section, we show how the results of the previous subsection lead to new valid inequalities and separation algorithms for the node-and-edge formulation. We also make some remarks about SDP relaxations.

4.1 New valid inequalities

One way to derive new valid inequalities for $P^{xy}(G, k)$ is to use Lemma 1 to project from $P^{xy}(G^k, k)$ to $P^y(G^k, k)$, and then use Corollary 1 to project from $P^y(G^k, k)$ to $P^{xy}(G, k)$. Here is an example. Let $T \subseteq V$ be a clique in G, let S be any subset of $\{1, \ldots, k\}$, and let $S' = \{d_c : c \in T\}$ be the corresponding set of dummy nodes in G^k. By construction, $C = T \cup S'$ forms a clique in G^k. Then, provided that $t = \lfloor |C|/k \rfloor \geq 1$ and $r = |C| \bmod k \neq 0$, the clique inequality (8) defines a facet of $P^{xy}(G^k, k)$. By Lemma 1, it also defines a facet of $P^y(G^k, k)$. Corollary 1 then yields the following valid inequality for $P^{xy}(G, k)$:

$$\sum_{v \in T} \sum_{c \in S} x_{vc} + \sum_{u,v \in T} y_{uv} \geq \binom{t+1}{2} r + \binom{t}{2}(k - r). \qquad (17)$$

We call inequalities of this type *projected clique* inequalities.

Note that projected clique inequalities reduce to (standard) clique inequalities if $S = S' = \emptyset$. Moreover, we have the following lemma:

9

Lemma 4 *If $S = \{1, \ldots, k\}$, then the projected inequality (17) is equivalent to the clique inequality on T.*

Proof. Consider what would happen if we changed S from $\{1, \ldots, k\}$ to $\emptyset$. The effect on the left-hand side of (17) is that the term involving x variables would disappear. Due to the equations (2), the net decrease in the left-hand side would be $|S|$. As for the right-hand side of (17), note that the stated change in S causes us to remove the elements of S' from C, which in turn causes t to decrease by 1. As a result, the net decrease in the right-hand side is $tr + (t-1)(k-r) = (tk + r) - k = |C| - k = |S|$. $\qquad \square$

For $1 \leq |S| < k$, the projected clique inequalities are new. Moreover, we have the following result.

Theorem 2 *Projected clique inequalities define facets of $P^{xy}(G, k)$.*

Proof. See A. $\qquad \square$

In a similar way, one can derive projected versions of the cycle, odd wheel and odd bicycle wheel inequalities. For the sake of brevity, we do not explore this in detail here. We note however that:

- The only projected cycle inequalities that define facets are the cycle inequalities themselves. (This is because a cycle that passes through a dummy node in G^k can never be chordless.)

- The odd cycle inequalities (12) are an example of facet-defining projected odd wheel inequalities. (This follows from the remark at the end of Section 3.)

Another way to derive new valid inequalities for $P^{xy}(G, k)$ is to take a family of valid inequalities for the clique partitioning polytope $P^y(K_{n+k}, n)$, note that they are valid also for $P^y(K_{n+k}, k)$, use Lemma 2 to find out when they are valid also for $P^y(G^k, k)$, and finally use Corollary 1 to project them from $P^y(G^k, k)$ to $P^{xy}(G, k)$. Here is an example. Oosten *et al.* [30] showed that, for any set $T \subset V$ with $|T| \geq 3$, any node $s \in V \setminus T$, and any integer α between 1 and $|T| - 2$, the following *weighted (s, T)* inequality defines a facet of $P^y(K_n, n)$:

$$\sum_{u,v \in T} y_{uv} \geq \alpha \sum_{v \in T} y_{sv} - \binom{\alpha + 1}{2}.$$

It is valid, though not necessarily facet-defining, also for $P^y(K_{n+k}, k)$. By Lemma 2, it is valid also for $P^y(G^k, k)$, provided that T is a clique in G^k.

Now, if we identify s with a dummy node, say d_c, Corollary 1 yields the following valid inequality for $P^{xy}(G, k)$:

$$\sum_{u,v \in T} y_{uv} \geq \alpha \sum_{v \in T} x_{vc} - \binom{\alpha + 1}{2}. \tag{18}$$

We call inequalities of this type *weighted clique* inequalities.

The weighted clique inequalities are valid for all $k \geq 2$, all cliques $T \subseteq V$ with $|T| \geq 3$, all cluster indices $c \in \{1, \ldots, k\}$, and all α between 1 and $|T| - 2$. It turns out that they define facets of $P^{xy}(G, k)$ when α is sufficiently large.

Theorem 3 *Weighted clique inequalities define facets of $P^{xy}(G, k)$ if and only if $k \geq 3$ and $|T| - k < \alpha \leq |T| - 2$.*

Proof. See B. $\qquad \square$

We remark that the inequalities (14) can be regarded as 'degenerate' weighted (s, T) inequalities with $|T| = 2$ and $\alpha = 1$, and the inequalities (3) can be regarded as 'degenerate' weighted clique inequalities with $|T| = 2$ and $\alpha = 1$.

4.2 New separation algorithms

In addition to new inequalities for the node-and-edge formulation, we obtain new separation algorithms. The key is the following proposition.

Proposition 1 *Let F be a family of valid inequalities for $P^y(G^k, k)$, and let F' be the corresponding family of projected inequalities for $P^{xy}(G, k)$. Also let $(x^*, y^*) \in [0, 1]^{nk+m}$ be a solution to an LP relaxation of the node-and-edge formulation. We construct a point $\tilde{y} \in [0, 1]^{nk+m+\binom{k}{2}}$ by setting y_e^* to:*

- *x_{vc}^* if $e = \{v, d_c\}$ for some $v \in V$ and $c \in \{1, \ldots, k\}$,*

- *y_e^* if $e \in E$,*

- *0 if $e = \{d_c, d_{c'}\}$ for some c, c' with $1 \leq c < c' \leq k$.*

Then $\tilde{y}$ violates an inequality in F if and only if (x^*, y^*) violates an inequality in F'.

Proof. This follows from Theorem 1 and the definition of "projected" inequalities in Corollary 1. $\qquad \square$

Example: Suppose that $k = 3$ and $G = K_3$. Setting x^*_{12}, x^*_{13}, x^*_{22}, x^*_{23}, x^*_{32}, x^*_{33} to $1/2$ and all other variables to zero, we obtain a pair (x^*, y^*) that satisfies all of the inequalities mentioned in Subsection 2.1. Using the mapping in the proposition, we obtain a point $\tilde{y}$ with

$$\tilde{y}_{1,d_2} = \tilde{y}_{1,d_3} = \tilde{y}_{2,d_2} = \tilde{y}_{2,d_3} = \tilde{y}_{3,d_2} = \tilde{y}_{3,d_3} = 1/2$$

and all other variables equal to zero. This point violates the following clique inequality by 1:

$$y_{1,d_1} + y_{2,d_1} + y_{3,d_1} + y_{12} + y_{13} + y_{23} \geq 1.$$

The corresponding projected clique inequality is

$$x_{11} + x_{21} + x_{31} + y_{12} + y_{13} + y_{23} \geq 1.$$

The point (x^*, y^*) violates it by 1. $\qquad\square$

We have the following corollary.

Corollary 2 *There exist polynomial-time separation algorithms for projected odd wheel and projected odd bicycle wheel inequalities, and for a family of inequalities that includes all projected 2-chorded odd cycle inequalities.*

Proof. This follows from Proposition 1 and the known results mentioned in Subsection 2.3. There is however one minor complication in the case of the third family of inequalities mentioned: the separation algorithms given in [3, 4, 27] assume that the underlying graph is complete, but the graph G^k need not be complete. Fortunately, the separation algorithm given in [29] works on general graphs, and the algorithms in [3, 4, 27] can be easily adapted to the case of general graphs. $\qquad\square$

In a similar way, the heuristics for clique, 2-partition and weighted (s, T) separation, mentioned in Subsection 2.3, can be used to derive heuristics for the corresponding projected inequalities.

4.3 On SDP relaxations

We now present a modified SDP relaxation for the k-PP. Although we are not yet sure whether it is of any practical value in itself, we will see that it yields a new family of valid inequalities for $P^{xy}(K_n, k)$, along with an efficient separation algorithm for them.

Let $X \in \{0, 1\}^{nk}$ be a matrix in which $X_{vc} = 1$ if and only if node v lies in cluster c. (Rendl [31] calls X a *partition matrix*.) Note that X_{vc} plays the same role as x_{vk} in the node-and-edge formulation, and that $Xe_k = e_n$. Now consider the matrix:

$$Z = \begin{pmatrix} I_k \\ X \end{pmatrix} \begin{pmatrix} I_k \\ X \end{pmatrix}^T = \begin{pmatrix} I_k & X^T \\ X & XX^T \end{pmatrix}.$$

By definition, this matrix is psd. Moreover, the submatrix XX^T is nothing but the matrix $\tilde{Y}$ used in Rendl's SDP relaxation. This leads naturally to the following SDP relaxation:

$$
\begin{aligned}
\min \quad & \sum_{e \in E} w_e \tilde{Y}_e \\
\text{s.t.} \quad & \text{diag}(\tilde{Y}) = e_n \\
& \tilde{Y}_{uv} \geq 0 && (\{u, v\} \in E_n) \\
& X e_k = e_n \\
& Z = \begin{pmatrix} I_k & X^T \\ X & \tilde{Y} \end{pmatrix} \succeq 0.
\end{aligned}
$$

Note that this relaxation involves a matrix variable of order $k + n$.

We have the following proposition:

Proposition 2 *The new SDP relaxation gives the same lower bound as the Freize–Jerrum SDP relaxation (and therefore also the Rendl and De Klerk* et al. *relaxations).*

Proof. As mentioned in Subsection 2.4, the Freize–Jerrum, Rendl and De Klerk *et al.* relaxations all give the same bound. Now, let $\tilde{Y}^*$ be a feasible solution to Rendl's relaxation. Suppose we set X_{vc}^* to $1/k$ for all v and c. Then our claim is that the pair $(X^*, \tilde{Y}^*)$ is a feasible solution to the new relaxation. To see this, note that all linear constraints are satisfied. As for the psd constraint on Z, Schur complement implies that $Z \succeq 0$ if and only if $\tilde{Y} - XX^T \succeq 0$. But this latter constraint is satisfied by $(X^*, \tilde{Y}^*)$, since $k\tilde{Y}^* - J_n \succeq 0$ and $X^*(X^*)^T = J_n/k$ by construction.

Now, let $(X^*, \tilde{Y}^*)$ be a feasible solution to the new relaxation. Since the corresponding matrix Z^* is psd, we have

$$
\begin{pmatrix} a \\ b \end{pmatrix} \begin{pmatrix} I_k & (X^*)^T \\ X^* & \tilde{Y}^* \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix}^T \geq 0 \tag{19}
$$

for all $a \in \mathbb{R}^k$ and $b \in \mathbb{R}^n$. Now consider what happens if we set a to $\epsilon\, e_k$ for some $\epsilon \in \mathbb{R}$. Since $e_k I_k e_k^T = k$ and $X^* e_k = e_n$, the inequality (19) then reduces to:

$$
\begin{pmatrix} \epsilon \\ b \end{pmatrix} \begin{pmatrix} k & e_n \\ e_n & \tilde{Y}^* \end{pmatrix} \begin{pmatrix} \epsilon \\ b \end{pmatrix}^T \geq 0.
$$

Since this true for all ϵ and all b, we have:

$$
\begin{pmatrix} k & e_n \\ e_n & \tilde{Y}^* \end{pmatrix} \succeq 0.
$$

By Schur complement, this is equivalent to $k\tilde{Y}^* - J_n \succeq 0$. Thus, $\tilde{Y}^*$ is feasible for Rendl's relaxation. $\qquad\square$

As mentioned above, it is not clear to us whether the new relaxation could be of practical use in itself. Although it gives the same bound as Rendl's relaxation, the matrix X^* gives additional information, which could perhaps be exploited in a randomised rounding heuristic for the k-PP. We leave this as a possible topic for future research. Our main interest in the new relaxation is that it leads to new valid inequalities for $P^{xy}(K_n, k)$, along with an efficient separation algorithm for them.

Proposition 3 *The following "psd" inequalities are valid for $P^{xy}(K_n, k)$, for all $a \in \mathbb{R}^k$ and $b \in \mathbb{R}^n$:*

$$\sum_{c=1}^{k} \sum_{v \in V} a_c b_v x_{vc} + \sum_{\{u,v\} \in E} b_u b_v y_{uv} \geq -\left(||a||_2^2 + ||b||_2^2\right)/2.$$

Proof. This follows from the inequalities (19), the definition of Z, the fact that X encodes the x variables, the fact that $\mathrm{diag}(\tilde{Y}) = e_n$, and the fact that the off-diagonal elements of $\tilde{Y}$ encode the y variables. $\square$

Proposition 4 *The separation problem for the psd inequalities can be solved in polynomial time (to arbitrary precision).*

Proof. Let (x^*, y^*) be the point to be separated, with $x^* \in [0,1]^{kn}$ and $y^* \in [0,1]^{\binom{n}{2}}$. Construct the corresponding matrix Z^*, and compute its minimum Eigenvalue to the desired precision. If the Eigenvalue is positive, Z^* is psd, and therefore no psd inequality is violated. Otherwise, let $\binom{a}{b}$ be the associated Eigenvector, where $a \in \mathbb{R}^k$ and $b \in \mathbb{R}^n$. The inequality (19) is violated by Z^*, and therefore the corresponding psd inequality is violated by (x^*, y^*). $\square$

We remark that a similar argument was used in [26] to derive valid inequalities for the cut polytope, or, equivalently, for $P^y(K_n, 2)$, along with an efficient separation algorithm for them.

We close this section with two further remarks. First, the psd inequalities are valid for $P^{xy}(G, k)$ even when G is not complete, as long as the nodes with non-zero b_v values form a clique in G. Second, if one takes the so-called *hypermetric* inequalities for $P^y(K_n, k)$, presented in [8], and then applies our projection procedure, the resulting *projected* hypermetric inequalities for $P^{xy}(K_n, k)$ dominate the psd inequalities. We skip the details, partly for brevity, and partly because the complexity of separation for the hypermetric inequalities is unknown.

5 Computational Experiments

In this section, we describe some preliminary computational experiments with projected clique inequalities. The experiments were run on a desktop

computer with a 4-core Intel Core i5-4690 CPU clocked at 3.50GHz and running under Linux with kernel version 4.4. We used the software package `igraph` [9] to construct graphs and enumerate maximal cliques, and `Gurobi` version 7 [23] to solve all LPs and 0-1 LPs.

5.1 Test instances

Recall that the "node-and-edge" formulation is most suitable for k-PP instances in which G is sparse. We therefore constructed ten sparse disk graphs as follows. We take the unit torus and place 50 nodes at random. We then include an edge between a given pair of nodes if and only if their Euclidean distance exceeds a constant r. (We chose $r = 0.25$, which led to graphs with average degree around 9.5.) Each edge in the resulting graph is given a random integer weight between 1 and 10. For each of the ten graphs, we then considered $k \in \{2, 3, 4\}$. This makes 30 k-PP instances in total.

The motivation for restricting attention to k-PP instances with positive edge weights is that clique inequalities work well for such instances. For instances with a mixture of positive and negative weights, other families of inequalities (such as odd wheel inequalities) are more useful [34].

5.2 Experimental setup

For each instance, we compare three algorithmic settings:

1. Solve the 0-1 LP problem (1)–(7) by plain branch-and-bound.

2. Run a cutting plane algorithm, in which the separation problem for the clique inequalities (8) is solved by brute-force enumeration. At termination, delete all clique inequalities with slack larger than 0.1 and run branch-and-bound.

3. As in option 2, except that, between the termination of the cutting-plane algorithm and the start of branch-and-bound, we run a second cutting-plane algorithm, in which the separation problem for the *projected* clique inequalities (17) is solved by enumeration.

For the instances considered, the time taken in the cutting-plane phases was negligible compared to the branch-and-bound time. (For larger instances and/or larger values of k, one would probably have to separate over the clique and projected clique inequalities heuristically, rather than exactly.)

5.3 Results

Table 1 shows, for each value of k and each of the three settings, the average branch-and-bound time and the average number of branch-and-bound nodes. We see that the addition of clique inequalities leads to a dramatic reduction

	No Cuts		Clique		Gen. Clq.	
k	nodes	time	nodes	time	nodes	time
2	5410.10	15.10	17.30	0.19	15.60	0.20
3	38665.90	157.72	3117.70	3.68	4650.90	9.05
4	210040.60	183.78	3973.80	3.20	30665.40	33.60

Table 1: Branch-and-bound results for random k-PP instances.

	No Cuts		Clique		Gen. Clq.	
k	nodes	time	nodes	time	nodes	time
2	5079.60	6.80	5.60	0.17	3.60	0.12
3	24217.90	202.45	2715.60	7.68	1004.30	3.30
4	7478.20	77.01	1870.30	8.13	780.00	3.86

Table 2: Branch-and-bound results for node-weighted variant.

in both measures. Surprisingly, however, the projected clique inequalities do not lead to any further reduction. In fact they sometimes make things worse.

Examination revealed that the addition of projected clique inequalities that are not ordinary clique inequalities led to no improvement in the lower bound from the LP relaxation. This may be because they involve x variables, which do not appear in the objective function. To test this hypothesis, we modified the 30 k-PP instances by giving each x variable a random integer cost between 1 and 10. The results obtained are shown in Table 2. We see that, as expected, the projected clique inequalities now lead to a significant improvement.

6 Concluding Remarks

We have shown that known results for the "edge-only" formulation of the k-PP yield, via projection, both known and new results for the "node-and-edge" formulation. In particular, we have obtained several new families of valid inequalities for the latter, along with new separation algorithms. As a by-product, we have obtained a new variant of the SDP relaxations given in [15, 25, 31].

One possible issue for future research is to find a rule for deciding when to use the edge-only formulation, when to use the node-and-edge formulation, and when to use an SDP approach. Another interesting question is whether our SDP relaxation can be of practical use. Although we have shown that it gives the same lower bound as the relaxations in [15, 25, 31], it could

perhaps form the basis for a new heuristic. Finally, we believe that more work is needed on valid inequalities and separation algorithms for problems related to the k-PP, such as the clique partitioning problem and other graph partitioning problems.

References

[1] H.-J. Bandelt, M. Oosten, J.H.G.C. Rutten & F.C.R. Spieksma (1999) Lifting theorems and facet characterization for a class of clique partitioning inequalities. *Oper. Res. Lett.*, 24, 235–243.

[2] F. Barahona & A.R. Mahjoub (1986) On the cut polytope. *Math. Program.*, 36, 157–173.

[3] R. Borndörfer & R. Weismantel (2000) Set packing relaxations of integer programs. *Math. Program.*, 88, 425–450.

[4] A. Caprara & M. Fischetti (1996) $\{0, \frac{1}{2}\}$-Chvátal-Gomory cuts. *Math. Program.*, 74, 221–235.

[5] R. Carlson & G.L. Nemhauser (1966) Clustering to minimize interaction costs. *Oper. Res.*, 14, 52–58.

[6] S. Chopra (1994) The graph partitioning polytope on series-parallel and 4-wheel free graphs. *SIAM J. Discr. Math.*, 7, 16–31.

[7] S. Chopra & M.R. Rao (1993) The partition problem. *Math. Program.*, 59, 87–115.

[8] S. Chopra & M.R. Rao (1995) Facets of the k-partition polytope. *Discr. Appl. Math.*, 61, 27–48.

[9] G. Csardi & T. Nepusz (2006) The igraph software package for complex network research. *InterJournal, Complex Systems*, 1695, 1–7.

[10] E.R. van Dam & R. Sotirov (2016) New bounds for the max-k-cut and chromatic number of a graph. *Lin. Alg. Appl.*, 488, 216–234.

[11] M. Deza, M. Grötschel & M. Laurent (1991) Complete descriptions of small multicut polytopes. In P. Gritzmann & B. Sturmfels (eds) *Applied Geometry and Discrete Mathematics*. Providence, RI: AMS.

[12] M. Deza, M. Grötschel & M. Laurent (1992) Clique-web facets for multicut polytopes. *Math. Oper. Res.*, 17, 981–1000.

[13] A. Eisenblätter (2001) *Frequency Assignment in GSM Networks: Models, Heuristics, and Lower Bounds*. PhD Thesis, Technical University of Berlin.

[14] A. Eisenblätter (2002) The semidefinite relaxation of the k-partition polytope is strong. In W.J. Cook & A.S. Schulz (eds.) *Integer Programming and Combinatorial Optimization IX*, pp. 273–290. Lecture Notes in Computer Science, vol. 2337. Berlin: Springer.

[15] A. Frieze & M. Jerrum (1997) Improved approximation algorithms for max-k-cut and max bisection. *Algorithmica*, 18, 67–81.

[16] M.R. Garey, D.S. Johnson & L.J. Stockmeyer (1976) Some simplified $\mathcal{NP}$-complete graph problems. *Theoret. Comput. Sci.*, 1, 237–267.

[17] A.M.H. Gerards (1985) Testing the odd bicycle wheel inequalities for the bipartite subgraph polytope. *Math. Oper. Res.*, 10, 359–360.

[18] B. Ghaddar, M.F. Anjos & F. Liers (2011) A branch-and-cut algorithm based on semidefinite programming for the minimum k-partition problem. *Ann. Oper. Res.*, 188, 155–174.

[19] M. Grötschel, L. Lovász & A.J. Schrijver (1988) *Geometric Algorithms and Combinatorial Optimization*. New York: Wiley.

[20] M. Grötschel & Y. Wakabayashi (1989) A cutting plane algorithm for a clustering problem. *Math. Program.*, 45, 59–96.

[21] M. Grötschel & Y. Wakabayashi (1990) Facets of the clique partitioning polytope. *Math. Program.*, 47, 367–387.

[22] M. Grötschel & Y. Wakabayashi (1990b) Composition of facets of the clique partitioning polytope. In R. Bodendieck & R. Henn (eds.) *Topics in Combinatorics and Graph Theory*, pp. 271–284. Heidelberg: Physica-Verlag.

[23] Gurobi Optimization, Inc. (2017) *Gurobi Optimizer Reference Manual*. Available at `http://www.gurobi.com`

[24] V. Kaibel, M. Peinhardt & M.E. Pfetsch (2011) Orbitopal fixing. *Discr. Optim.*, 8, 595–610.

[25] E. de Klerk, D.V. Pasechnik & J.P. Warners (2004) On approximate graph colouring and max-k-cut algorithms based on the θ-function. *J. Comb. Optim.*, 8, 267–294.

[26] M. Laurent & S. Poljak (1996) Gap inequalities for the cut polytope. *Eur. J. Combinatorics*, 17, 233–254.

[27] A.N. Letchford (2001) On disjunctive cuts for combinatorial optimization. *J. Combin. Optim.*, 5, 299–315.

[28] A.N. Letchford & M.M. Sørensen (2012) Binary positive semidefinite matrices and associated integer polytopes. *Math. Program.*, 131, 253–271.

[29] R. Müller & A.S. Schulz (2002) Transitive packing: a unifying concept in combinatorial optimization. *SIAM J. Optim.*, 13, 335–367.

[30] M. Oosten, J.H.G.C. Rutten & F.C.R. Spieksma (2001) The clique partitioning problem: facets and patching facets. *Networks*, 38, 209–226.

[31] F. Rendl (2012) Semidefinite relaxations for partitioning, assignment and ordering problems. *4OR*, 10, 321–346.

[32] M.M. Sørensen (2002) A note on clique-web facets for multicut polytopes. *Math. Oper. Res.*, 27, 740–742.

[33] R. Sotirov (2014) An efficient semidefinite programming relaxation for the graph partition problem. *INFORMS J. Comput.*, 26, 16–30.

[34] V.J.R. de Sousa, M.F. Anjos & S. Le Digabel (2016) Computational study of valid inequalities for the maximum k-cut problem. *Ann. Oper. Res.*, to appear.

A Proof of Theorem 2

In [7], it is pointed out that a (standard) clique inequality is satisfied at equality by an extreme point of $P^{xy}(G, k)$ if and only if, in the corresponding k-PP solution, each cluster contains either t or $t + 1$ nodes from C. Due to the structure of the augmented graph G^k, this implies that a *projected* clique inequality (17) is satisfied at equality by an extreme point of $P^{xy}(G, k)$ if and only if the following two conditions hold:

- For all $c \in S$, the cth cluster contains either $t - 1$ or t nodes from T.

- For all $c \in \{1, \ldots, k\} \setminus S$, the cth cluster contains either t or $t+1$ nodes from T.

We will call such extreme points *roots*. By abuse of terminology, the corresponding k-PP solutions will also be called *roots*.

Now suppose that all roots satisfy an equation of the form $\alpha^T x + \beta^T y = \gamma$. We will perform a series of exchange arguments to show that the equation is equivalent to a projected clique inequality (in equation form). Throughout, we assume that $1 \leq |S| < k$, since, if $|S| \in \{0, k\}$, we obtain a standard clique inequality (see Lemma 4).

Let c, c' be any two cluster indices, and let W and W' be the corresponding clusters. Consider any root that satisfies the following two conditions:

(i) $|W \cap T|$ is equal to t if $c \in S$, and equal to $t + 1$ otherwise, (ii) $|W' \cap T|$ is equal to $t - 1$ if $c' \in S$, and equal to t otherwise. Let u be any node in W. One can check that, if u is any node in W, we can obtain another root by moving u from W to W'. This yields the equation

$$\alpha_{uc} + \sum_{v \in W \setminus \{u\}: \{u,v\} \in E} \beta_{uv} = \alpha_{uc'} + \sum_{v \in W': \{u,v\} \in E} \beta_{uv}. \tag{20}$$

Now let u' be any node in $W' \setminus T$. If we move u' from W' to W, the equation (20) becomes:

$$\alpha_{uc} + \sum_{v \in (W \setminus \{u\}) \cup \{u'\}: \{u,v\} \in E} \beta_{uv} = \alpha_{uc'} + \sum_{v \in W' \setminus \{u'\}: \{u,v\} \in E} \beta_{uv}.$$

Together with (20), this implies that $\beta_{uu'} = 0$. Applying this argument repeatedly, we see that $\beta_{uu'} = 0$ for all $u \in V$ and all $u' \in V \setminus T$. This in turn implies that $\alpha_{uc} = \alpha_{uc'}$ for all pairs c, c' and for all $u \in V \setminus T$. Thus, for any $u \in V \setminus T$, the coefficients α_{uc} must take a constant value for all $c \in \{1, \ldots, k\}$. Since all feasible k-PP solutions satisfy the equations (2), we can assume that this constant is zero for all $u \in V \setminus T$. In this way, the nodes in $V \setminus T$ can be removed from consideration.

Now let c, c' be cluster indices with $c \in S$ and $c' \notin S$, and let W and W' be the corresponding clusters. Consider any root such that $|W \cap T| = |W' \cap T| = t$. We can obtain another root by taking any node $u \in W \cap T$ and moving it from W to W'. This shows that

$$\alpha_{uc} + \sum_{v \in (W \cap T) \setminus \{u\}} \beta_{uv} = \alpha_{uc'} + \sum_{v \in W' \cap T} \beta_{uv}.$$

Observe that the right-hand side of this equation contains one more β term than the left-hand side. By symmetry, we have $\beta_{uv} = \alpha_{uc} - \alpha_{uc'}$ for all $c \in S$, $c \notin S$ and $\{u, v\} \subset T$. This implies in turn that β_{uv} takes a constant value for all $\{u, v\} \subset T$, that α_{uc} takes a constant value for all $u \in T$ and $c \in S$, and that $\alpha_{uc'}$ takes a constant value for all $c' \notin S$. Since all feasible k-PP solutions satisfy the equations (2), we can assume that the $\alpha_{uc'}$ are zero, which then implies that the α_{uc} are equal to the β_{uv}. Finally, by scaling, we can assume that the α_{uc} and β_{uv} are equal to one. $\qquad \square$

B Proof of Theorem 3

As in the proof of Theorem 2, we call a k-PP solution a *root* if the corresponding extreme point of $P^{xy}(G, k)$ satisfies the weighted clique inequality (18) at equality.

Consider a feasible k-PP solution such that cluster c contains exactly t nodes from T. This solution must use at least $\binom{t}{2}$ of the edges that have

both end-nodes in T, and it will use exactly $\binom{t}{2}$ of those edges if and only if each of the other nodes in T lies in a unique cluster. From this it follows that a k-PP solution is a root if and only if (i) cluster c contains either α or $\alpha+1$ nodes from T, and (ii) each of the other nodes in T lies in a different cluster. This can only happen if there are at least $|T|-\alpha$ clusters, i.e., if $k \geq |T|-\alpha$. Moreover, if there were *exactly* $|T| - \alpha$ clusters, then all roots would satisfy the equation $\sum_{v \in T} x_{vc} = \alpha + 1$, and the weighted clique inequality could not define a facet. This shows α must lie between $|T| - k + 1$ and $|T| - 2$ if we want to obtain a facet. This in turn implies that k must be at least 3. So, we have proved necessity.

The proof of sufficiency is similar to that of Theorem 2. For brevity, we only give a sketch. We suppose that all roots satisfy an equation of the form $\beta^T x + \gamma^T y = \delta$. An exchange argument, in which nodes in $V \setminus T$ are either included in or excluded from cluster c, enables us to show that $\gamma_{uv} = \beta_{vc} = 0$ whenever $v \in V \setminus T$. Another exchange argument, in which a node $u \in T$ is moved between clusters, enables us to show that $\beta_{uc'} = 0$ for all $u \in T$ and for all $c' \in \{1, \ldots, k\} \setminus \{c\}$, and that $\beta_{uc} = -\alpha \gamma_{uv}$ for all $\{u, v\} \subset T$. Finally, by scaling, we can assume that $\gamma_{uv} = 1$ for all $\{u, v\} \subset T$ and that $\beta_{uc} = -\alpha$ for all $u \in T$. $\qquad \square$