The Edge-Weighted Clique Problem: valid inequalities, facets and polyhedral computations

Elder Magalhães Macambira

Cid Carvalho de Souza

Relatório Técnico IC–97-14

Outubro de 1997

The Edge-Weighted Clique Problem: valid inequalities, facets and polyhedral computations †

Elder Magalhães Macambira‡ *Cid Carvalho de Souza*§

Abstract. Let $K_n = (V, E)$ be the complete undirected graph with weights c_e associated to the edges in E. We consider the problem of finding the subclique $C = (U, F)$ of K_n such that the sum of the weights of the edges in F is maximized and $|U| \leq b$, for some $b \in [1, \ldots, n]$. This problem is called the Maximum Edge-Weighted Clique Problem (MEWCP) and is NP-hard. In this paper we investigate the facial structure of the polytope associated to the MEWCP and introduce new classes of facets for this polytope. Computational experiments with a branch-and-cut algorithm are reported confirming the strength of these inequalities. All instances with up to 48 nodes could be solved without entering into the branching phase. Moreover, we show that some of these new inequalities also define facets of the Boolean Quadric Polytope and generalize previously known inequalities for this polytope.

Key Words: Edge-Weighted Cliques, Polyhedral Combinatorics, Branch-and-Cut, Boolean Quadric Polytope

1 Introduction

Let $K_n = (V, E)$ be the complete undirected graph with weights c_e associated to the edges in E. We consider the problem of finding the subclique $C = (U, F)$ of K_n such that the sum of the weights of the edges in F is maximized and $|U| \leq b$, for some integer $b \in [1, \ldots, n]$. This problem is called the Maximum Edge-Weighted Clique Problem (MEWCP).

The MWECP can be easily seen to be NP-hard, since the usual MAX-CLIQUE problem reduces polynomially to it. Heuristic algorithms based on local search have been proposed in [13] to find good suboptimal solutions for this problem.

Exact algorithms based on Integer Programming formulations have been proposed in [5], [6] and [11]. The natural formulation presented in [5] uses only binary variables corresponding to the edges of K_n. The authors investigate the problem from a polyhedral point of view. Several facet defining inequalities are introduced and computational results obtained by a cutting-plane algorithm using these inequalities are reported. From their computational experiments, the authors conclude that the cutting-plane approach was not suitable to solve the MEWCP even for moderate sized instances. The largest instance they

†This research was supported by FAPESP (grant number 96/0884-9) and by CNPq (grant number 300883/94-3)

‡Universidade Estadual do Ceará, Depto. de Ciência da Computação, Fortaleza/CE, Brazil (*e-mail: elder@uece.br*)

§corresponding author: Universidade Estadual de Campinas, Instituto de Computação, C.P. 6176, 13083-970, Campinas/SP,Brazil (*e-mail: cid@dcc.unicamp.br*)

solve refers to a graph on 25 nodes but extremely poor performances are reported for quite smaller instances.

In [6], an extended formulation is proposed that includes binary variables not only for the edges but also for the nodes in K_n. In this paper, no polyhedral investigation on this new model has been conducted. A polyhedral investigation of the extended formulation is done in [11] where several classes of facet defining inequalities for the associated polytope are presented. The authors also proved that the lower bounds provided by the extended formulation are better than those coming from the natural formulation on the edge variables. The computational results reported by the authors are much more encouraging than those reported in [5]. The instances tested include graphs with up to 30 nodes and most of them have been solved to optimality by pure cutting-planes (no branching was necessary).

Many facet defining inequalities introduced in [11] are based upon facet defining inequalities for the Boolean Quadric Polytope (BQP) investigated in [10]. In fact, the polytope associated to the extended formulation of MEWCP is contained in the BQP and, therefore, any inequality valid for the BQP is also valid for the polytope associated to MEWCP.

In this paper we go further in investigating the facial structure of the polytope associated to the extended formulation of the MEWCP in order to have a better understanding of it. For this, we introduce new classes of valid and facet defining inequalities for this polytope. In the first class of valid inequalities proposed here, we start by an inequality defining facet for the BQP. We generalize this inequality and we prove that it belongs to a more general class of inequalities defining facets for the MEWCP polytope. Moreover, we show that the inequalities in this new class are also valid for the BQP and generalize previously known classes of facets of the latter polytope.

A second class of inequalities generalize the tree inequalities originally introduced in [8] and further studied in [11]. The generalization goes in two different directions. In both of them we have been able to proof that some special cases correspond to facet defining inequalities for the MEWCP.

Besides the search for new classes of facet defining inequalities, we also have carried out computational experiments with a branch-and-cut algorithm that we have implemented. The main goals with these experiments were to evaluate the strength of the new inequalities introduced here and to compare different cutting-plane strategies based on the inequalities found in the literature. Instances with up to 48 nodes were tested and the results confirm that at least one class of inequalities we introduce is computationally useful. If the algorithm is restricted to use cutting-planes (without branching) the strategy using this new class of inequalities was the only one able to solve all the instances in the sample.

The paper is organized as follows. In the next section we give the extended Integer Programming formulation for the MEWCP and summarize the main polyhedral results from the literature which are important for our work. Section 3 describes the first class of inequalities we propose, namely, the (α, β)-inequalities. Section 4 discusses two possible generalizations of the tree inequalities leading to two distinct classes of valid inequalities for the MEWCP. In Section 5 we describe a branch-and-cut algorithm that uses some of the inequalities introduced in the previous section and report our computational results.

2 An Extended Integer Programming Formulation for MEWCP

In this section we describe the Integer Programming formulation for MEWCP. Given the complete undirected graph $K_n = (V, E)$, the variables in the formulation are divided into two sets: the edge variables, denoted by y_{ij} for each $(i, j) \in E$, and the node variables, denoted by x_i for each $i \in V$. More formally, if $C = (U, F)$ is a clique in K_n, we have that:

$$x_i = \begin{cases} 1, & \text{if node } i \in U, \\ 0, & \text{otherwise.} \end{cases}$$

$$y_{ij} = \begin{cases} 1, & \text{if edge } (i,j) \in F, \\ 0, & \text{otherwise.} \end{cases}$$

(IP)

$$\max \sum_{i,j,i<j} c_{ij} y_{ij}$$

Subject to:

$$\begin{array}{llr}
y_{ij} \leq x_i & \forall\, (i,j) \in E \;\; i<j & \text{(I)} \\
y_{ij} \leq x_j & \forall\, (i,j) \in E \;\; i<j & \text{(II)} \\
x_i + x_j - y_{ij} \leq 1 & \forall\, (i,j) \in E \;\; i<j & \text{(III)} \\
\sum_{e \in \delta(i)} y_e - (b-1)x_i \leq 0 & \forall\, i \in V & \text{(IV)} \\
y_{ij} \in \{0,1\}^m, & & \text{where } m = \binom{n}{2} \\
x_i \in \{0,1\}^n & &
\end{array}$$

Inequalities (I) and (II) ensure that an edge is not in the clique if one of its endnodes is not in the clique. Inequality (III) says that the edge is in the clique if both its endnodes are in the clique and, finally, inequality (IV) limits the number of edges incident to each node i to either 0 or $(b-1)$ depending if node i is in the clique or is not in the clique, respectively. Note that by dropping the constraints in (IV), we obtain the linearization for formulating the Boolean Quadric Problem [10].

Let us denote by $\mathcal{P}_C(b)$ the convex hull of all incidence vectors of cliques in K_n with at most b nodes and by $\mathcal{P}_B$ the convex hull of 0-1 vectors satisfying constraints (I), (II) and (III). Therefore, $\mathcal{P}_B$ is the Boolean Quadric Polytope and we have that $\mathcal{P}_C(b) \subseteq \mathcal{P}_B$. From this observation, it turns out that any valid inequality for $\mathcal{P}_B$ is also valid for $\mathcal{P}_C(b)$ but, clearly, the facetness property may be lost.

Below we summarize some results that are known for $\mathcal{P}_C(b)$ which we will use later. Proofs of these results can be found in [11].

The convention adopted to represent the support graphs of the inequalities given in this text is the following: (i) dashed circles indicate nodes with negative coefficients, (ii) circles filled in grey indicate nodes with null coefficient, (iii) full-line circles indicate nodes with positive coefficients, (iv) dash lines indicate edges with negative coefficients and (v) full lines indicate edges with positive coefficients. For sake of brevity, we shall use simply the term *support* to refer to the support graph of an inequality.

Proposition 2.1 *Given the complete undirected graph* $K_n = (V, E)$ *with* $|V| = n$ *and* $|E| = m = \binom{n}{2}$, *the dimension of* $\mathcal{P}_C(b)$ *is given by:*

$$\dim(\mathcal{P}_C(b)) = \begin{cases} n & \text{if } b = 1, \\ n + m & \text{if } 2 \leq b \leq n. \end{cases}$$

Thus, for nontrivial instances of the problem, the polytope is full-dimensional. It follows that, if we want to prove that the face $\mathcal{F}$ defined by a valid inequality $\pi(x, y) \leq \pi_0$ for $\mathcal{P}_C(b)$ is a facet, we only have to show that any other inequality defining a face that contains $\mathcal{F}$ is a (positive) scalar multiple of $\pi(x, y) \leq \pi_0$ (cf,[9]). The proofs in this text are based on this technique.

The next two propositions establish the conditions under which the inequalities in the original formulation are facet defining for $\mathcal{P}_C(b)$.

Proposition 2.2 *Let* $n \geq 3$. *For every two distinct nodes* $i, j \in V$, *the inequalities* $y_{ij} - x_i \leq 0$ *and* $x_i + x_j - y_{ij} \leq 1$ *define facets of* $\mathcal{P}_C(b)$ *if and only if* $b \geq 3$.

Proposition 2.3 *For every node* $i \in V$, *the* **star inequality**,

$$\sum_{j \in V - \{i\}} y_{ij} - (b - 1)x_i \leq 0$$

defines a facet of $\mathcal{P}_C(b)$ *if and only if* $b \leq n - 1$.

The above inequality is called a *star inequality* since its support graph is a tree with a single node of degree higher than one (which by definition is a star graph).

Proposition 2.4 *Let* $T \subseteq V$ *be a subset of nodes in* K_n. *If* $|T| \geq 3$ *and* $1 \leq \beta \leq |T| - 2$, *the* **clique inequality**

$$\beta \sum_{i \in T} x_i - \sum_{e \in E(T)} y_e \leq \beta(\beta + 1)/2 \tag{1}$$

defines a facet of $\mathcal{P}_C(b)$ *if and only* $T = V$ *or* $\beta \leq b - 2$.

A special case of the clique inequalities is given when $|T| = 3$ and $\beta = 1$ which are called the *clique triangle inequalities*. The support graph of such an inequality is given in Figure 1(a).

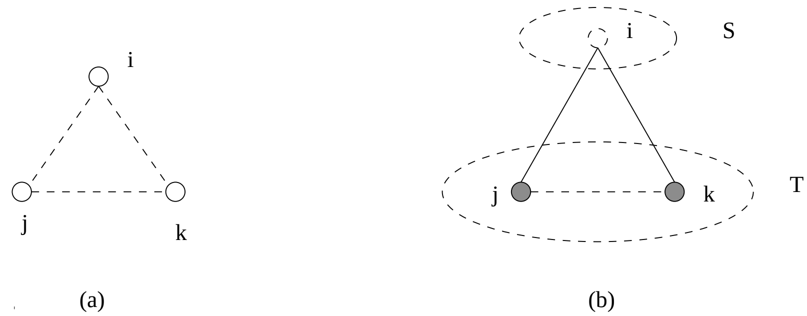

Figure 1: (a) Support graph of clique triangle inequality with $T = \{i, j, k\}$ e $\beta = 1$. The clique triangle inequality corresponding is $x_i + x_j + x_k - y_{ij} - y_{ik} - y_{jk} \leq 1$. (b) Support graph of cut triangle inequality with $S = \{i\}$ e T $= \{j, k\}$. The cut triangle inequality corresponding is $y_{ij} + y_{ik} - y_{jk} - x_i \leq 0$.

Proposition 2.5 *Let $S \subseteq V$ and $T \subseteq V - S$ be two disjoint subset of nodes in K_n. For $|S| = s \geq 1$ and $|T| = t \geq 2$, the* **cut inequality**

$$\sum_{e \in E(S:T)} y_e - \sum_{e \in E(S)} y_e - \sum_{e \in E(T)} y_e - \sum_{i \in S} x_i \leq 0 \tag{2}$$

defines a facet of $\mathcal{P}_C(b)$ if and only $|S| = 1$ and $b \geq 3$ or $|S| \geq 2$ and $b \geq 4$.

When $s = 1$ and $t = 2$, we have the *cut triangle inequalities* whose support is shown in Figure 1(b). Like the clique inequalities, the cut inequalities were introduced for the Boolean Quadric Polytope in [10].

In the next section we generalize the cut inequalities and obtain a large class of facet defining inequalities for $\mathcal{P}_C(b)$. Moreover all the inequalities in this new class are shown to be valid for $\mathcal{P}_B$ and include both the clique and the cut inequalities.

We close this section by presenting the *tree inequalities*. They have been first introduced in [8] and further studied in [11]. The proposition below appears in the latter paper.

Proposition 2.6 *Let $T = (W, H)$ be a tree in $K_n = (V, E)$ such that, $|W| = b+1$. If $b \geq 3$ then the* **tree inequality**

$$\sum_{e \in H} y_e - \sum_{i \in W} (d_i - 1) x_i \leq 0 \tag{3}$$

where d_i is the degree of node i in T, defines a facet of $\mathcal{P}_C(b)$ if and only if $b = n - 1$ or T is not a star.

Note that, among the inequalities presented here, *tree* and *star inequalities* are the only ones who deal with the cardinality b of the largest feasible clique.

In Section 4 we generalize the tree inequalities in two different ways and obtain large classes of valid inequalities of $\mathcal{P}_C(b)$ which are also related to b, the maximum cardinality of a feasible clique. We show that, at least for some special cases, the generalized inequalities define facets of $\mathcal{P}_C(b)$.

3 Generalization of the *Cut Inequalities*

In this section we present a new class of facet defining inequalities for $\mathcal{P}_C(b)$ which can be viewed as a generalization of the *cut inequalities* given in Proposition 2.5. We start by introducing the (s,t)-*cut inequalities* which were originally proposed in [10] for the polytope $\mathcal{P}_B$. The proposition below gives necessary conditions under which these inequalities are valid for $\mathcal{P}_C(b)$.

Proposition 3.1 *Let $S \subseteq V$ and $T \subseteq V - S$ be two disjoint subset of nodes in K_n, where $|S| = s$ and $|T| = t$. The (s,t)-***cut inequality**

$$\sum_{e \in E(S:T)} y_e - \sum_{e \in E(T)} y_e - \sum_{e \in E(S)} y_e + (s-t)\sum_{i \in S} x_i + (t-s-1)\sum_{i \in T} x_i \leq \frac{1}{2}(t-s)(t-s-1) \quad (4)$$

is valid for $\mathcal{P}_C(b)$.

Proof: Let $C = (U, F)$ be any feasible clique in K_n (i.e., $|U| \leq b$). Suppose that $|U \cap S| = \tilde{s}$ and $|U \cap T| = \tilde{t}$. Thus, for the incidence vector of C inequality (4) gives:

$$\tilde{t}\tilde{s} - \tilde{t}(\tilde{t}-1)/2 - \tilde{s}(\tilde{s}-1)/2 + (s-t)\tilde{s} + (t-s-1)\tilde{t} \leq (t-s)(t-s-1)/2.$$

$$\begin{aligned}
\tilde{t}\tilde{s} - \tilde{t}(\tilde{t}-1)/2 - \tilde{s}(\tilde{s}-1)/2 + (s-t)\tilde{s} + (t-s-1)\tilde{t} &\leq (t-s)(t-s-1)/2 \\
-\frac{1}{2}[\tilde{t}^2 - 2\tilde{t}\tilde{s} - \tilde{t} - 2(t-s-1)\tilde{t} + \tilde{s}^2 - \tilde{s} - 2(s-t)\tilde{s} + (t-s)(t-s-1)] &\leq 0 \\
-\frac{1}{2}[\tilde{t}^2 - 2\tilde{t}\tilde{s} - \tilde{t} - 2(t-s-1)\tilde{t} + \tilde{s}^2 + \tilde{s}(-2(s-t)-1) + (t-s)(t-s-1)] &\leq 0
\end{aligned}$$

After factorizing the second order polynomium in $\tilde{s}$ and the terms in $\tilde{t}$, we get:

$$-\frac{1}{2}[\tilde{t}^2 - \tilde{t}(2\tilde{s} + 2t - 2s - 1) + (\tilde{s} + t - s)(\tilde{s} + t - s - 1)) \quad \leq \quad 0$$

By factorizing the second order polynomium in $\tilde{t}$ we obtain:

$$\begin{aligned}
-\frac{1}{2}[(\tilde{t} - (\tilde{s} + t - s))(\tilde{t} - (\tilde{s} + t - s - 1))] &\leq 0 \\
-\frac{1}{2}(\tilde{t} - \tilde{s} - t + s)(\tilde{t} - \tilde{s} - t + s + 1) &\leq 0
\end{aligned}$$

Since $\tilde{t} - \tilde{s} - t + s$ and $\tilde{t} - \tilde{s} - t + s + 1$ are consecutive integers, the inequality must hold. $\square$

Before proving that the (s,t)-*cut inequalities* define facets of $\mathcal{P}_C(b)$, we give the lemma below characterizing the vectors which are roots of such inequalities (i.e, the vectors liying on the face defined by this inequality in $\mathcal{P}_C(b)$). This will be useful in proving the facetness property.

Lemma 3.1 *Let $C = (U, F)$ be a clique of $K_n = (V, E)$ with $|U| \leq b$. Moreover, let $S \subseteq V$ and $T \subseteq V - S$ be two disjoint subsets of nodes defining an (s,t)-cut inequality where $|S| = s$ and $|T| = t$. Assume that $\tilde{s} = |U \cap S|$ and $\tilde{t} = |U \cap T|$. Then,*

(i) If $t \geq s$, the roots of (4) are incidence vectors of cliques C such that $\tilde{t} = (t - s) + \tilde{s}$ or $\tilde{t} = (t - s) + \tilde{s} - 1$.

(ii) If $s > t$, the roots of (4) are incidence vectors of cliques C such that $\tilde{s} = (s - t + 1) + \tilde{t}$ or $\tilde{s} = (s - t + 1) + \tilde{t} - 1$.

Proof: Immediate from the previous proof.

□

The next proposition states that the (s,t)-cut inequalities introduced earlier are facet defining for $\mathcal{P}_C(b)$.

Proposition 3.2 *Let $S \subseteq V$ and $T \subseteq V - S$, with $|S| = s \geq 1$ and $|T| = t \geq 2$, two disjoint subsets of nodes in K_n. If $b \geq 3$ the (s,t)-cut inequality defines a facet of $\mathcal{P}_C(b)$.*

Proof: Since we have already proved the validity of the inequality, we can concentrate on proving that it is facet defining. For this, consider the face $\mathcal{F}$ defined by inequality (4) in $\mathcal{P}_C(b)$. Assume that there exists a valid inequality $\pi x + \beta y \leq \alpha_0$ for $\mathcal{P}_C(b)$ which defines the face $F_{(\pi,\beta)} = \{(x,y) \in \mathcal{P}_C(b) : \pi x + \beta y = \alpha_0\}$. Moreover, assume that $\mathcal{F} \subseteq \mathcal{F}_{(\pi,\beta)}$. Since $\mathcal{F} \neq \emptyset$, if we show that $\pi x + \beta y \leq \alpha_0$ is a positive scalar multiple of the (s,t)-cut inequality, the proof will be completed.

To establish the relations between the coefficients in π, β and π_0 we have to find incidence vectors that lie on the face $\mathcal{F}$ (and conseqüently in $\mathcal{F}_{(\pi,\beta)}$). The incidence vectors given here can be easily checked to satisfy this condition by noting that they are in the form described in Lemma 3.1. We assume that these vectors represent a feasible clique $C = (U, F)$ such that $\widetilde{S} = S \cup U$ and $\widetilde{T} = T \cup U$.

Claim 1. $\beta(a,b) = -\alpha$ and $\beta(a,c) = \alpha$ for all $a, b \in S$ e $c \in T$.

Proof. Consider $\widetilde{T} \subseteq T$, such that $|\widetilde{T}| = t - s$. Let $a \in S$, $b \in S - \{a\}$ and $c \in T - \widetilde{T}$ be three nodes arbitrarily chosen. Since $\mathcal{X}^{\widetilde{T}} \in \mathcal{F}$, $\mathcal{X}^{\widetilde{T} \cup \{a\}} \in \mathcal{F}$ and $\mathcal{X}^{\widetilde{T} \cup \{b\}} \in \mathcal{F}$ we obtain that:

$$\begin{aligned} (\pi,\beta)\widetilde{T} &= \alpha_0 \\ (\pi,\beta)\widetilde{T} &+ \pi_a + \beta(a,\widetilde{T}) = \alpha_0 \\ (\pi,\beta)\widetilde{T} &+ \pi_b + \beta(b,\widetilde{T}) = \alpha_0 \end{aligned}$$

Thus, we have that $\pi_a + \beta(a,\widetilde{T}) = 0$ and $\pi_b + \beta(b,\widetilde{T}) = 0$.

Now, since $\mathcal{X}^{\widetilde{T} \cup \{a\} \cup \{c\}} \in \mathcal{F}$, we conclude that $\pi_c + \beta(c,\widetilde{T}) + \beta(a,c) = 0$. Moreover, since $\mathcal{X}^{\widetilde{T} \cup \{a\} \cup \{b\} \cup \{c\}} \in \mathcal{F}$, we obtain $\pi_a + \pi_b + \pi_c + \beta(a,\widetilde{T}) + \beta(b,\widetilde{T}) + \beta(c,\widetilde{T}) + \beta(a,b) + \beta(a,c) + \beta(b,c) = 0$. By combining the previous results we get $\beta(a,b) = -\beta(b,c)$.

Consider now a new set $\widetilde{T} \subseteq T$, with $|\widetilde{T}| = t-s-1$. Let a, b and d be any nodes chosen such that $a \in S$, $c \in T - \widetilde{T}$ and $d \in T - (\widetilde{T} \cup \{c\})$. The following incidence vectors are in $\mathcal{F}$: $\mathcal{X}^{\widetilde{T}}$, $\mathcal{X}^{\widetilde{T} \cup \{c\}}$ and $\mathcal{X}^{\widetilde{T} \cup \{d\}}$. Thus, we can deduce that $\pi_c + \beta(c, \widetilde{T}) = 0$ and $\pi_d + \beta(d, \widetilde{T}) = 0$.

Noting that $\mathcal{X}^{\widetilde{T}} \in \mathcal{F}$, $\mathcal{X}^{\widetilde{T} \cup \{a\} \cup \{c\}} \in \mathcal{F}$ and $\mathcal{X}^{\widetilde{T} \cup \{a\} \cup \{d\}} \in \mathcal{F}$, it follows that $\pi_a + \beta(a, \widetilde{T}) + \beta(a, c) = 0$ and $\pi_a + \beta(a, \widetilde{T}) + \beta(a, d) = 0$, respectively.

Therefore, from the previous results we conclude that $\beta(a, c) = \beta(a, d) = \alpha_a$. If we take b in $S - \{a\}$, in an analogous way we have that $\beta(b, c) = \beta(b, d) = \alpha_b$. Thus, since $\beta(a, b) = -\beta(b, c)$ we conclude that $\beta(a, b) = -\beta(a, c) = -\beta(b, c) = -\alpha_a = -\alpha_b$, that is , $\beta(a, b) = -\alpha$. Since the nodes were chosen arbitrarily, the result is proved.

From now on, assume that $\widetilde{T} \subseteq T$ is a subset of nodes satisfying $|\widetilde{T}| = t - s$.

Claim 2. $\beta(z, w) = 0$ for all $z, w \in V - (S \cup T)$.

Proof. Let z, w be two arbitrary nodes in $V - (S \cup T)$. Since $\mathcal{X}^{\widetilde{T}} \in \mathcal{F}$, $\mathcal{X}^{\widetilde{T} \cup \{z\}} \in \mathcal{F}$ and $\mathcal{X}^{\widetilde{T} \cup \{w\}} \in \mathcal{F}$, we can conclude that $\pi_z + \beta(z, \widetilde{T}) = 0$ and $\pi_w + \beta(w, \widetilde{T}) = 0$. But since $\mathcal{X}^{\widetilde{T}} \in \mathcal{F}$ and $\mathcal{X}^{\widetilde{T} \cup \{z\} \cup \{w\}} \in \mathcal{F}$, we have that $\beta(z, w) = 0$.

Claim 3. $\beta(u, z) = 0$ and $\beta(v, z) = 0$ for all $u \in S$, for all $v \in T - \widetilde{T}$ and for all $z \in V - (S \cup T)$.

Proof. Take any three nodes u, v and z such that $u \in S$, $v \in T - \widetilde{T}$ and $z \in V - (S \cup T)$. The following incidence vectors are in $\mathcal{F}$: $\mathcal{X}^{\widetilde{T}}$, $\mathcal{X}^{\widetilde{T} \cup \{u\}}$, $\mathcal{X}^{\widetilde{T} \cup \{z\}}$ and $\mathcal{X}^{\widetilde{T} \cup \{u\} \cup \{z\}} \in \mathcal{F}$. This implies that $\beta(u, z) = 0$.

By considering the incidence vectors in $\mathcal{F}$ given by $\mathcal{X}^{\widetilde{T}}$, $\mathcal{X}^{\widetilde{T} \cup \{z\}}$, $\mathcal{X}^{\widetilde{T} \cup \{v\} \cup \{u\}}$ and $\mathcal{X}^{\widetilde{T} \cup \{v\} \cup \{u\} \cup \{z\}}$, and using the fact that $\beta(u, z) = 0$, we obtain that $\beta(v, z) = 0$.

Claim 4. $\pi_z = 0$ for all $z \in V - (S \cup T)$.

Proof. Let z be an arbitrary node in $V - (S \cup T)$. By noting that the incidence vectors $\mathcal{X}^{\widetilde{T}}$ and $\mathcal{X}^{\widetilde{T} \cup \{z\}}$ are in $\mathcal{F}$, and using that $\beta(z, \widetilde{T}) = 0$, we show that $\pi_z = 0$.

Claim 5. $\beta(b, d) = -\alpha$ for all $b, d \in T$.

Proof. Let a, b, c and d be any nodes in S, $S - \{a\}$, $T - \widetilde{T}$ and $T - (\widetilde{T} \cup \{b\})$, respectively. We have that: $\mathcal{X}^{\widetilde{T}}$, $\mathcal{X}^{\widetilde{T} \cup \{a,c\}}$ $\mathcal{X}^{\widetilde{T} \cup \{a,d\}}$, $\mathcal{X}^{\widetilde{T} \cup \{b,c\}}$, $\mathcal{X}^{\widetilde{T} \cup \{b,d\}}$ and $\mathcal{X}^{\widetilde{T} \cup \{a,b,c,d\}}$ are in $\mathcal{F}$. Therefore, since $\beta(a, d) = \beta(b, c) = \alpha$ and $\beta(a, c) = \beta(b, d) = -\alpha$, it follows that $\beta(b, d) = -\alpha$.

Claim 6. $\pi_a = \alpha(s - t)$ for all $a \in S$.

Proof. Let a be an arbitrary node in S. The vectors $\mathcal{X}^{\widetilde{T}}$ and $\mathcal{X}^{\widetilde{T} \cup \{a\}}$ are in $\mathcal{F}$ and since $\beta(a, u) = -\alpha, \forall u \in T$, we obtain that $\pi_a + \beta(a, \widetilde{T}) = 0$, that is, $\pi_a = \alpha(t - s)$.

Claim 7. $\pi_c = \alpha(t - s - 1)$ for all $c \in T$.

Proof. Let a and c be arbitrary nodes in S and $T - \widetilde{T}$, respectively. From the incidence vectors $\mathcal{X}^{\widetilde{T}}$, $\mathcal{X}^{\widetilde{T} \cup \{a\}}$ and $\mathcal{X}^{\widetilde{T} \cup \{a\} \cup \{c\}}$ that are in $\mathcal{F}$, we obtain that $\pi_c + \beta(c, \widetilde{T}) + \beta(a, c) = 0$. Since $\beta(c, \widetilde{T}) = -\alpha(t - s)$ and $\beta(a, c) = \alpha$, we conclude that $\pi_c = \alpha(t - s - 1)$. This completes the proof.

□

Below we give the main results of this section which prove that the (s, t)-cut inequalities belong to a larger class of valid and facet defining inequalities of $\mathcal{P}_C(b)$. The next theorem introduces this class of inequalities, which we call the (α, β)-*inequalities*.

Theorem 3.1 *Let $S \subseteq V$ and $T \subseteq V - S$ be two disjoint subsets of nodes in K_n, with $|S| = s \geq 1$ e $|T| = t \geq 2$. For $b \geq 1$ and $\alpha - \beta = 1$, where α and $\beta \in \mathbb{Z}$, the (α, β)-inequality given by*

$$\sum_{e \in E(S:T)} y_e - \sum_{e \in E(T)} y_e - \sum_{e \in E(S)} y_e \; - \alpha \sum_{i \in S} x_i + \; \beta \sum_{i \in T} x_i \leq \frac{1}{2} \alpha \beta \tag{5}$$

is valid $\mathcal{P}_C(b)$.

Proof: Simply replace $t - s$ by α in the proof of Lemma 3.1

$\square$

The roots of the inequality (5) are defined exactly as in Lemma 3.1 by replacing $t - s$ by α. This shows that the (s,t)-cut inequalities are special cases of (α, β)-inequalities when $\alpha = t - s$. Moreover, the clique and cut inequalities form particular subclasses of the (α, β)-inequalities for when $|S| = 0$ and $\beta = 0$, respectively.

Finally, the theorem below provide some necessary conditions under which inequality (5) defines facet of $\mathcal{P}_C(b)$.

Theorem 3.2 *Let $S \subseteq V$ and $T \subseteq V - S$ be two disjoint subsets of nodes in K_n, with $|S| = s$ and $|T| = t$. For positive integers α and β satisfying $\alpha - \beta = 1$, and $t \geq \alpha + 1$, the (α, β)-inequality given by (5) defines a facet of $\mathcal{P}_C(b)$ when $\alpha \leq b - 4$.*

Proof: The proof is analogous to that of Proposition 3.2. It suffices to replace $t - s$ by α and use the same incidence vectors.

$\square$

Note that, Lemma 3.1 also holds for the incidence vectors of feasible solutions of the Boolean Quadric Problem. Thus, the (α, β)-inequality is valid for $\mathcal{P}_B$. Moreover, since $\mathcal{P}_C(b) \subseteq \mathcal{P}_B$ and $\mathcal{P}_C(b)$ is full dimension, the same set of affinely independent points used to prove that (5) defines a facet of $\mathcal{P}_C(b)$ can be used to prove that this inequality also defines facet for $\mathcal{P}_B$. The theorem below formalizes this result.

Theorem 3.3 *Let $S \subseteq V$ and $T \subseteq V - S$ be two disjoint subsets of nodes in K_n, with $|S| = s$ e $|T| = t \geq 2$. For positive integers α and β satisfying $\alpha - \beta = 1$, and $t \geq \alpha + 1$, the (α, β)-inequality given by (5) defines a facet of the* **Boolean Quadric Polytope** *$\mathcal{P}_B$.*

4 Generalization of *Tree Inequalities*

In this section we generalize the tree inequalities given in Proposition 2.6. This generalization goes in two different ways. In the first generalization we obtain new valid inequalities for $\mathcal{P}_C(b)$ by decreasing the size of the tree. To avoid infeasibility, the support of the inequality is enlarged via a lifting operation (cf, [9]). The support graph we obtain is disconnected and, to the best of our knowledge, these are the first inequalities for $\mathcal{P}_C(b)$ with this feature.

We note that a very similar result is presented in [3] for the equicut problem. In the latter problem, cycles seem to play a role which is very similar to that of trees in the MEWCP. By exploring this idea we have been able to derive a second generalization for the tree inequalities which can be viewed as special combinations of originally nonvalid inequalities for $\mathcal{P}_C(b)$.

These two generalizations are discussed in the following subsections. The main result of this section is that both generalizations lead to valid inequalities for $\mathcal{P}_C(b)$ and, for some special cases, they can be shown to be facet defining for $\mathcal{P}_C(b)$.

4.1 Small *Tree Inequalities*

Given a tree $T = (W, H)$ in K_n consider the value of $\omega_T(x, y)$ given by $\omega_T(x, y) = \sum_{e \in H} y_e - \sum_{i \in W}(d_i - 1)x_i$. The inequalitity in Proposition 2.6 can be rewritten as

$$\omega_T(x, y) = \sum_{e \in H} y_e - \sum_{i \in W}(d_i - 1)x_i \leq 1 \tag{6}$$

for $|W| = b + 1$. The validity of this inequality is based on the following lemma taken from [2].

Lemma 4.1 *Let $C = (U, F)$ and $T = (W, H)$ be a clique and a tree of K_n, respectively. Define (x^C, y^C) to be the incidence vector of C and assume that $A = U \cap W$. If $c(A)$ is the number of connected components induced by A in T and $\delta_T(A)$ is the cutset of A in T, then $\omega_T(x^C, y^C) = c(A) - |\delta_T(A)|$.*

Proof: We compute:

$$\begin{aligned} \omega_T(x^C, y^C) &= \textstyle\sum_{e \in H} y_e + \sum_{u \in W}(1 - d_u)x_u \\ &= \textstyle\sum_{e \in E(A)} y_e + \sum_{u \in A}(1 - d_u) \\ &= \textstyle\sum_{e \in E(A)} y_e + |A| - \sum_{u \in A} d_u \end{aligned}$$

The first summation corresponds to the number of edges in the forest induced by A in T, therefore: $\sum_{e \in A} y_e = |A| - c(A)$. In the second summation each edge joining two nodes i and j in A is counted twice (in d_i and d_j) and the edges going from A to $W \setminus A$ (those in $\delta_T(A)$) appear only once. This yields: $\sum_{u \in A} d_u = 2(|A| - c(A)) + |\delta_T(A)|$.

Therefore:

$$\begin{aligned} \omega_T(x^C, y^C) &= |A| - c(A) - 2|A| + 2c(A) - |\delta_T(A)| \\ &= c(A) - |\delta_T(A)| \end{aligned}$$

□

From the lemma above, we have that $\omega_T(x, y)$ is never greater than one, except when $A = W$ (A is the set of all nodes in the tree), in which case we have $c(A) = 1$ and $|\delta_T(A)| = 0$, implying that $\omega_T(x, y) = 1$. A root of $\omega_T(x, y)$ is obtained when $c(A) = |\delta_T(A)|$, meaning that each of the subtrees in the forest induced by A in T can be disconnected from the remaining nodes of T by removing a single edge (that is, the nodes in $W \setminus A$ are all contained in a single subtree of T).

Clearly, this lemma proves the validity of the *tree inequalities* whenever we can ensure that not all nodes of the tree belong simultaneously to a feasible clique. This is the case when $|W| > b+1$. But, what if $|W| \leq b$? Below we focus on the case when $|W| = b$.

From Lemma 4.1, the only incidence vector of a feasible clique that violates the corresponding tree inequality is the one associated to the clique $C = (W, E(W))$, that is, the clique is composed by the nodes of the tree $T = (W, H)$. Thus, the tree inequality $\omega_T(x, y) \leq 0$ is not valid for $\mathcal{P}_C(b)$ when $|W| = b$.

To recover the validity of the inequality we make a *lifting*. For this consider the (α, β)-inequalities for $\alpha = \beta + 1 = 2$ which is given by:

$$\omega_1(x,y) = \sum_{e \in E(S:S')} y_e - \sum_{e \in E(S')} y_e - \sum_{e \in E(S)} y_e - 2\sum_{i \in S} x_i + \sum_{i \in S'} x_i \leq 1 \tag{7}$$

where $S \subseteq V - W$ and $S' \subseteq V - (W \cup S)$.

Note that if $x_i = 0$ for all nodes in $S \cup S'$ then $\omega_1(x,y) = 0$ and $\omega_T(x,y)$ can take the value one if the feasible clique has all its nodes in W. On the other hand, if $x_i = 1$ for some i in $S \cup S'$, $\omega_1(x, y)$ can be one but, in this case, $\omega_T(x,y) \leq 0$. These arguments prove the validity of the inequality given in the next theorem.

Theorem 4.1 *Let $T = (W, H)$ be a tree in $K_n = (V, E)$ such that $|W| = b$, $S \subseteq V - W$ and $S' \subseteq V - (W \cup S)$. For $b \geq 2$, $|S|$ and $|S'| \geq 1$, the* **tree+(2,1) inequality**

$$\omega_T(x,y) + \omega_1(x,y) \leq 1 \tag{8}$$

where $\omega_T(x,y)$ and $\omega_1(x,y)$ are defined as in (6) and (7), respectively, is valid for $\mathcal{P}_C(b)$

Though validity can be easily proved for the inequalities presented in the Theorem 4.1, we have not been able to establish necessary and sufficient conditions to ensure the facetness property. However, the next theorem presents a case in which these inequalities define facets of $\mathcal{P}_C(b)$.

Theorem 4.2 *If $T = (W, H)$ is a* **path** *then the inequality in Theorem 4.1 defines a facet for $\mathcal{P}_C(b)$.*

Proof: Let $\mathcal{F}$ be the face defined in $\mathcal{P}_C(b)$ by $\omega_T(x,y) + \omega_1(x,y) \leq 1$. Moreover, assume that $\pi x + \gamma y \leq \pi_0$ is a valid inequality for $\mathcal{P}_C(b)$ such that $\mathcal{F} \subseteq \mathcal{F}_{(\pi,\gamma)} = \{(x,y) \in \mathcal{P}_C(b) : \pi x + \gamma y = \pi_0\}$. Since $\mathcal{F} \neq \emptyset$, if we prove that $\pi x + \gamma y \leq \pi_0$ is a scalar multiple of $\omega_T(x,y) + \omega_1(x,y) \leq 1$ we are done.

Again we assume that $\mathcal{X}^C$ is the incidence vector of a feasible clique C. If Z is a set of nodes in V and H a set of edges in E, we use the notation $\pi(Z)$ and $\gamma(H)$ to denote respectively $\sum_{i \in Z} \pi_i$ and $\sum_{e \in H} \gamma_e$. We abuse notation and denote by $\gamma(Z)$ the value of $\sum_{(i,j): i,j \in Z} \gamma_{i,j}$. Finally, we denote by $Z|Z'$ the set of edges in the cutset $\delta(Z, Z')$.

Claim 1. For all $u \in S'$, $\mathcal{X}^{\{u\}} \in \mathcal{F}$. Thus, $\pi_u = \pi_0$.

Claim 2. For all $u, v \in S'$, $\mathcal{X}^{\{u,v\}} \in \mathcal{F}$. Thus, $\pi_u + \pi_v + \gamma_{u,v} = \pi_0$, and then $\beta_{u,v} = -\pi_0$.
Claim 3. For all $u, v, z \in S'$ and $w \in S$, $\mathcal{X}^{\{u,v,z,w\}}$ and $\mathcal{X}^{\{u,v,w\}}$ are in $\mathcal{F}$. Therefore, $\pi_w + \gamma(w, \{u, v, z\}) = \pi_w + \gamma(w, \{u, v\}) = \pi_0$. We conclude that $\gamma_{w,z} = \pi_0$ and $\pi_w = -2\pi_0$.
Claim 4. For all $w_1, w_2 \in S$ and $u, v, z_1, z_2 \in S'$, the vector $\mathcal{X}^{\{u,v,z_1,z_2,w_1,w_2\}}$ is in $\mathcal{F}$. From the previous results we get that $\gamma_{w_1,w_2} = -\pi_0$.
Claim 5. For all $z \in V \setminus (W \cup S \cup S')$ and $u, v \in S'$, the vectors $\mathcal{X}^{\{u,v,z\}}$ and $\mathcal{X}^{\{u,z\}}$ are in $\mathcal{F}$. Therefore, $\gamma_{z,u} = 0$ and, consequently, $\pi_z = 0$.
Claim 6. Let u, z, v and w be any four distinct nodes such that $u \in S$ and $z, v \in S'$ and $w \in V \setminus (W \cup S \cup S')$. The incidence vectors $\mathcal{X}^{\{u,v,z,w\}}$ and $\mathcal{X}^{\{u,v,z\}}$ are in $\mathcal{F}$. Thus, $\pi_w + \gamma_{u,w} + \gamma_{v,w} + \gamma_{z,w} = 0$ and, we conclude that $\gamma_{u,w} = 0$.

Up to now we have proved that all coefficients of π and γ related to nodes and edges not involving nodes of the path T (nodes in W) satisfy the desired conditions. In the next steps we deal with the remaining coefficients.

Initially, let $W = \{u_1, \ldots, u_b\}$ and $F = \{(u_1, u_2), \ldots, (u_{b-1}, u_b)\}$ be the nodes and edges of path T respectively. Note that symmetry implies that proofs involving nodes $(u_1, \ldots, u_p)$, for $p \leq b - 1$, also hold for nodes $(u_{b-p+1}, \ldots, u_b)$.
Claim 7. Since $\mathcal{X}^{\{j,u_1,\ldots,u_p\}}$ and $\mathcal{X}^{\{j,z,u_1,\ldots,u_p\}}$ for all $j \in S'$, $z \in V \setminus (W \cup S \cup S')$ and $p < b - 2$, we have that $\gamma(z|\{u_1, \ldots, u_p\}) = 0$. Now, varying p from 1 to $(b - 2)$ and using the symmetry of the path, we conclude that $\gamma_{z,u_i} = 0$ for all $i = 1, \ldots, b$.
Claim 8. Since $\mathcal{X}^{\{j,u_1,\ldots,u_p\}}$ and $\mathcal{X}^{\{j,k,u_1,\ldots,u_p\}}$ for all $j, k \in S'$ and $p < b - 2$, we have that $\gamma(k|\{u_1, \ldots, u_p\}) = 0$. Now, varying p from 1 to $(b - 2)$ and using the symmetry of the path, we conclude that $\gamma_{k,u_i} = 0$ for all $i = 1, \ldots, b$.
Claim 9. Let i, j, k be any arbitrary (distinct) nodes such that $i \in S$ and $j, k \in S'$. For $p < b - 3$, the following incidence vectors are in $\mathcal{F}$: $\mathcal{X}^{\{j,u_1,\ldots,u_p\}}$, $\mathcal{X}^{\{i,j,k,u_1,\ldots,u_p\}}$. From the previous results we can conclude the following. The first incidence vector implies that: $\pi(u_1, \ldots, u_p) + \gamma(u_1, \ldots, u_p) = 0$. The second incidence vector and the latter equation imply that $\gamma(i|\{u_1, \ldots, u_p\}) = 0$. Thus, for $p = 1, \ldots, p - 3$, $\gamma_{i,u_p} = 0$ and, by symmetry, we can extend this result so that it holds for all $p \in \{1, \ldots, b\}$.

Besides the coefficients of nodes and edges that are internal to the path, it remains to prove that the coefficients for all chords in the path are null. This is done in the next step.
Claim 10. Let $p < b - 3$, $b \geq r > p + 2$ and z be an arbitrary node in S'. The following cliques are feasible and their incidence vectors can be easily verified to be in $\mathcal{F}$: $\{z, u_1, \ldots, u_p, u_r, \ldots, u_b\}$, $\{z, u_1, \ldots, u_{p+1}, u_r, \ldots, u_b\}$, $\{z, u_1, \ldots, u_{p+1}\}$ and $\{z, u_r, \ldots, u_b\}$. This implies that:

$$\begin{aligned}
\pi(u_1, \ldots, u_p) + \pi(u_r, \ldots, u_b) + \gamma(u_1, \ldots, u_p) + \gamma(u_r, \ldots, u_b) \quad &+ \\
+\gamma(u_1, \ldots, u_p|u_r, \ldots, u_b) \quad &= \quad 0 \\
\pi(u_1, \ldots, u_{p+1}) + \pi(u_r, \ldots, u_b) + \gamma(u_1, \ldots, u_{p+1}) + \gamma(u_r, \ldots, u_b) \quad &+ \\
+\gamma(u_1, \ldots, u_{p+1}|u_r, \ldots, u_b) \quad &= \quad 0 \\
\pi(u_1, \ldots, u_{p+1}) + \gamma(u_1, \ldots, u_{p+1}) = 0& \\
\pi(u_r, \ldots, u_b) + \gamma(u_r, \ldots, u_b) = 0&
\end{aligned}$$

The last three equations imply that $\gamma(u_1, \ldots, u_{p+1} | u_r, \ldots, u_b) = 0$. Starting with $p = 0$ and $r = b$ and alternately incrementing p and decrementing r by one unit, we can prove that $\gamma_{u_i,u_j} = 0$, for all $1 \leq i < j - 1$ and $j \leq b$.

Now comparing the first two equations and using the last result for the coefficients of chords, we get that $\gamma_{u_{p+1},u_p} + \pi_{u_{p+1}} = 0$ for all $2 \leq p \leq b-2$ and $\pi_{u_1} = 0$.

Using the symmetry of the path we can also conclude that $\gamma_{u_{p+1},u_p} + \pi_{u_p} = 0$ for all $2 \leq p \leq b-2$ and $\pi_{u_b} = 0$. This implies that, for a constant λ, $\gamma_e = -\pi_u = \lambda$ for all edges e of the path and all nodes u which are not endnodes of the path.

Finally, the incidence vector of the (feasible) clique formed by the nodes of the path lies on $\mathcal{F}$. Therefore, $\lambda(b-1) - \lambda(b-2) = \pi_0$, and we must have $\lambda = \pi_0$ which completes the proof.

□

4.2 *Block-Tree Inequalities*

Consider the tree inequality and the result given in Lemma 4.1. Clearly, $c(A)$ is always less than or equal to $|\delta_T(A)|$ except if $A = V(T)$ in which case $c(A) = 1$ and $|\delta_T(A)| = 0$. Therefore, the tree inequality is valid for $\mathcal{P}_C(b)$ if and only if $|W| > b$. We now show that when suitable conditions are satisfied, the sum of nonvalid tree inequalities also gives rise to strong valid inequalities for $\mathcal{P}_C(b)$. This idea was first investigated in [2], [3], and is also described in [4] in the context of the equicut polytope. Due to the notation used in these previous works and the analogy between the roles played by cycles in cut problems and those played by trees in clustering problems, we call these inequalities *block-tree inequalities.*

Let Υ be a collection of trees $((V(T_1), T_1), \ldots, (V(T_t), T_t)$ of a graph G satisfying the following property: if $u \in V(T_i) \cap V(T_j)$ for some $i, j \in \{1, \ldots, t\}$, then $u \in V(T_k) \cap V(T_\ell)$ for all $k, \ell \in \{1, \ldots, t\}$.

For each tree T_i of Υ, we can define the tree inequality as before, i.e., $\omega(T_i)(x,y) \leq 0$. The tree combination inequality for Υ is given by the sum of these inequalities, that is:

$$\omega(\Upsilon)(z,y) = \sum_{T_i \in \Upsilon} \omega(T_i)(x,y) \leq 0 \tag{9}$$

Consider the subgraph of G formed by the edges and nodes in Υ. Let N be defined as the set of nodes that are common to all trees in Υ. If the nodes in N are removed from this subgraph, what remains is a forest of subtrees. Let $(S_1, S_2, \ldots)$ be the ordered set of these subtrees where $q_1 \geq q_2 \geq \ldots$ and $q_i = |S_i|$. We define $\overline{Q} = \sum_{i=1}^{t-1} q_i$. In Figure 2 it is shown the support of a tree combination inequality for which $t = 3$ and $\overline{Q} = 6$.

Theorem 4.3 *Let Υ be a tree combination in $G = (V,E)$ such that $V(\Upsilon) = \cup_{i=1}^{t} V(T_i)$. Suppose that $|V(\Upsilon)| = b + W$, where $W \geq 0$. Then, the* **block-tree inequality** *(9) is valid for $\mathcal{P}_C(b)$ if and only if $\overline{Q} < W$.*

Proof:

Necessity: Suppose that $\overline{Q} \geq W$. This implies that there exists a set Λ of $\lambda \leq t-1$ subtrees such that the number of nodes in all subtrees in Λ is greater than or equal to W.

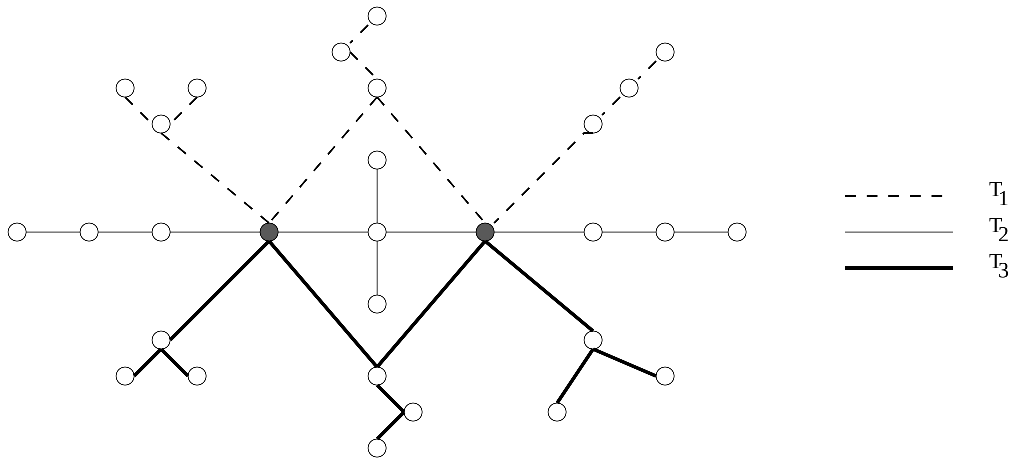

Figure 2: Support graph of the tree combination inequality for $t = 3$.

Define S to be the set of all nodes of $V(\Upsilon)$ that are not in $V(\Lambda)$ (= { nodes of subtrees in Λ }). The incidence vector (x^S, y^S) of S can be easily checked to be in $\mathcal{P}_C(b)$.

From Lemma 4.1, $\omega(T_i)(x^S, y^S) = c(S \cap V(T_i)) - |\delta_T(S \cap V(T_i))|$. However, $|\delta_T(S \cap V(T_i))|$ can be computed as $c(S \cap V(T_i)) + c(V(T_i) \setminus S) - 1$ which yields $\omega(T_i)(x^S, y^S) = 1 - c(V(T_i) \setminus S)$. Thus,

$$\begin{array}{rcl} \omega(\Upsilon)(x^S, y^S) & = & \sum_{i=1}^{t} \omega(T_i)(x^S, y^S) \\ & = & \sum_{i=1}^{t} (1 - c(V(T_i) \setminus S)) \\ & = & t - \sum_{i=1}^{t} c(V(T_i) \setminus S) \\ & = & t - \lambda \\ & \geq & t - (t-1) \geq 1 \end{array}$$

The left-hand side of inequality (9) is positive for this feasible solution and therefore the inequality is not valid. Necessity is proved.

Sufficiency: We assume that (9) is not valid and we end up with the conclusion that $\overline{Q}$ must be greater than or equal to W. For this, let S be a subset of V such that (x^S, y^S) violates (9), that is:

$$\omega(\Upsilon)(x^S, y^S) = t - \sum_{i=1}^{t} c(V(T_i) \setminus S) \geq 1$$

which implies that:

$$\sum_{i=1}^{t} c(V(T_i) \setminus S) \leq t - 1$$

Now, since (x^S, y^S) violates (9), there exists at least one tree in Υ, say T_j, such that $\omega(T_j)(x^S, y^S) = 1$. This implies that all nodes in $V(T_j)$ and, consequently, all nodes in N are in S. Thus, from the last expression above, we can conclude that there are at most $t - 1$ subtrees in Υ (obtained by removing the nodes in N) that do not contain nodes in S.

If $\overline{Q} < W$, the previous observation implies that $|S| > b$ and, therefore, S cannot be a feasible clique. We conclude that, if (9) is not valid, then $\overline{Q} \geq W$ and this completes the proof.

□

Again, it is hard to find necessary and sufficient conditions for inequality (9) to be facet defining for $\mathcal{P}_C(b)$. Nevertheless, we have found one case for which facet defining inequalities can be obtained by combining nonvalid tree inequalities. This case is described below.

Suppose that Υ is a collection of two trees T_1 and T_2 that have one node v_c in common ($N = \{v_c\}$) and the degree of any node i in trees T_1 and T_2 is not greater than 2. In this case, each of the trees of the combination reduces to a path and the support graph of the tree combination inequality looks like a cross centered at node v_c (see Figure 3).

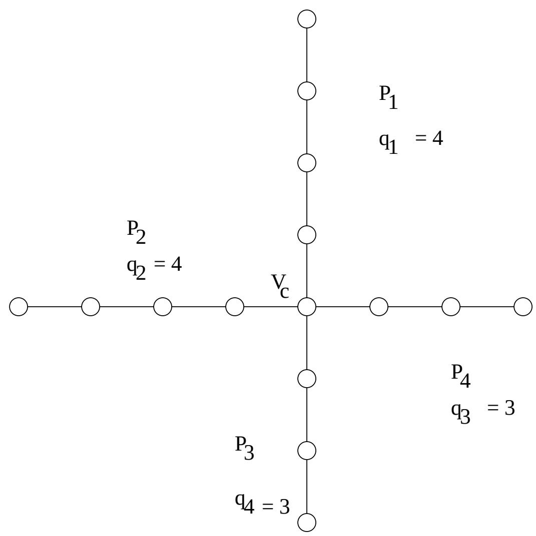

Figure 3: Support graph of the tree combination inequality with cross centered at node v_c.

Removing node v_c from the support graph, what remains is a forest composed of 4 paths. Let $(V(P_1), P_1), \ldots, (V(P_4), P_4)$ be those paths, $q_1 \geq q_2 \geq q_3 \geq q_4$ ($q_i = |V(P_i)|$) and $v_1^i, \ldots, v_{q_i}^i$ denote the nodes of $V(P_i)$. Moreover, define p_j to be the minimum of $q_k + q_\ell$ for k and ℓ in $\{1, 2, 3, 4\} \setminus \{j\}$. The tree combination inequality corresponds to:

$$\sum_{e \in P_1 \cup P_2 \cup P_3 \cup P_4} y_e - 2x_{v_c} - \sum_{i=1}^{4} \sum_{j=1}^{q_j - 1} x_{v_j^i} \leq 0 \tag{10}$$

The following result can be proved.

Theorem 4.4 *Let Υ be cross centered at a node v_c and spanning the graph $G = (V, E)$. Suppose that $q_2 + q_3 + q_4 \geq b$ and $|V(\Upsilon)| = b + W$. Then, for $b > W \geq 3$, the following holds:*

(i) Inequality (10) is valid for $\mathcal{P}_C(b)$.

(ii) Inequality (10) is facet defining for $\mathcal{P}_C(b)$ if and only if $q_j + p_j = b$ for all $j = 1, 2, 3, 4$.

5 Computational Results

We now describe the computational experiments that we have carried out. Our primary goal is to confirm that the inequalities that we have introduced here provide a better description of the polytope $\mathcal{P}_C(b)$. A second goal is to compare different cutting plane strategies. A large variety of facet defining inequalities for $\mathcal{P}_C(b)$ are found in the literature and all these inequalities can be considered in designing a branch-and-cut algorithm for MEWCP. In the literature, for the extended formulation, only one strategy have been tested in [11].

To achieve our goals, we have implemented a branch-and-cut algorithm whose main features are: **(i)** use of primal heuristic to provide *a priori* lower bounds; **(ii)** exact separation routines for the inequalities used in tightening the formulation, excepting tree inequalities; **(iii)** branching on the *most fractional* variable if no violated inequality is found or tailling-off has been detected (objective function change less than 0.0001 after 30 LPs) ; **(iv)** the strategy for node selection in the branch-and-bound tree is *Best Bound*; **(v)** the LP solver is CPLEX 3.0 (see [1]) and all tests have been done in a Sun SPARC 1000 machine.

We now describe in more details some of the features listed above.

Primal Heuristics

To derive an initial lower bounds, we use an algorithm which fits in the framework of *Greedy Randomized Adaptative Local Search Procedures*, **GRASP** for short. The main ideas of GRASP are discussed in [7].

Essentially, our heurisitic starts by building a feasible solution which is then given as an input for the local search heuristic *Greedy All* described in [13]. The initial solution is built as follows. First we pick one node of the graph and add it to the clique. At each iteration, a list of k nodes providing the largest augmentations in the objective function is constructed. Then, a randomly chosen node of the list is added to the current clique and the procedure keeps repeating these steps until b nodes have been added to the clique or no more nodes are to be examined.

For all test instances this easy-to-implement heuristic reaches the optimal solution. Thus, if good upper bounds come up from the formulations the branch-and-bound process presumably will stop soon.

Separation Routines

We have conducted several computational experiments comparing both CPU times and upper bounds obtained by cutting-planes (without entering the branching phase) for exact and heuristic separation routines for various classes of inequalities.

We have not been able to design good heuristics for separation. The best performances of the code for all inequalities were always achieved when exact separation was used. However, many of these separation algorithms have complexity of $\Theta(n^4)$ or even $\Theta(n^5)$, which is too much time consuming. Our conclusion is that this is certainly a topic that deserves more attention: one should look for a better compromise in which fast heuristics are designed so as to provide a considerable amount of violated inequalities when they exist.

The only inequality that we have not separate exactly are the ones having trees as part of their support graphs. For the tree inequality (3), Park et al [11] present a theorem which states that the most violated tree inequality can be found in polynomial time if its node set is fixed in advance otherwise, the problem is NP-hard. They also have developed an heuristic separation for these inequalities which initially chooses the nodes in the tree, that is, the set W. The procedure starts by finding the set $\widetilde{W}$ corresponding to all nodes in V whose variables are not null. If $|\widetilde{W}|$ is less than $b+1$ the procedure fails. Otherwise, suppose that $\widetilde{W} = \{u_1, \ldots, u_p\}$ such that $x_{u_i} \leq x_{u_{i+1}}$ for all $i \in \{1, \ldots, p\}$ where $p \geq b+1$. Then an optimal tree is built for all sets $W = (\{u_1, u_2, \ldots, u_{b+1}\} \cup \{u_j\}) \setminus \{u_i\}$, where $i \in \{1, \ldots, b+1\}$ and $j \in \{b+2, \ldots, p\}$. All violated inequalities found are add to the current formulation ([12]).

We also have implemented our own heuristic which builds the tree in a greedy fashion starting once with each edge of the graph. A few more inequalities could be separated with this heuristic. But in general, the results we have obtained for both heuristics were discouraging as we will show later. Tipically, the number of violated inequalities found is extremely low. We have not been able to understand precisely the causes of this behavior. It may be the case that the heuristics perform badly or, on the other hand, that incidence vectors satisfying the original formulation often satisfy the tree inequalities. Despite of this fact, to test the cutting strategy proposed in [11], we have separated tree inequalities. For the results reported later, we have used our own separation heuristic for trees.

Test Instances

Like in [5] and in [11], the instances are partitioned into two sets. In the first one, all edge weights are positive, while in the second one, positive and negative edge weights are allowed. The weights were randomly generated according to the scheme described in [13], that is:

- $1 \leq c_{ij} \leq \lfloor 10^{w+1} r^k \rfloor$ for positive weights, and
- $-\lfloor 10^{w+1} r^k \rfloor \leq c_{ij} \leq \lfloor 10^{w+1} r^k \rfloor$ for posite and negative weights.

where $k \in \{1, \ldots, 5\}$, $w > 0$ and $r \in (0, 1]$. We have generated instances for $n \in \{30, 40, 42, 44, 45, 46, 48\}$ and for five possible values of k, $w = 2$ and $b = \lfloor \frac{n}{2} \rfloor$. This choice of b was motivated from our preliminary tests which indicated that, for fixed n and edge weights, these are usually the most difficult instances. This is in accordance with the computational experiments reported in the literature ([11]).

Cutting Strategies

Since there are many different families of facet defining inequalities that are known for $\mathcal{P}_C(b)$, several alternative cutting strategies can be applied in a branch-and-cut algorithm for the Maximum Edge-Weighted Clique Problem.

The strategies we have tested in our computational experiments involve the inequalities listed below. The notation in parenthesis is used in the sequel to denote the corresponding inequalities.

- *cut triangle* ($\triangle$);
- *cut* for $|S| = 1$ and $|T| = 3$ ($C_{1,3}$);
- *cut* for $|S| = 2$ and $|T| = 3$ ($C_{2,3}$);
- *tree* (TRE);
- *clique triangle* ($C\triangle$);
- *(s,t)-cut* for $|S| = 1$ and $|T| = 3$ (st$C_{1,3}$);
- *(s,t)-cut* for $|S| = s = 1$ and $|T| = t = 4$ (st$C_{1,4}$);
- *(α, β)* for $\alpha = \beta + 1 = 2$, $|S| = 1$ and $|T| = 4$ ($(\alpha, \beta)_{1,4}$).

In our experiments four different strategies are compared. The choice for these stategies was based on preliminary tests we have done on small instances. Using the notation introduced above, the four strategies can be summarized as follows:

- **Strategy 1**: $\triangle$, $C_{1,3}$ and $C_{2,3}$;
- **Strategy 2**: $\triangle$, $C\triangle$ and TRE,
- **Strategy 3**: $\triangle$, st$C_{1,3}$ and st$C_{1,4}$;
- **Strategy 4**: $\triangle$ and $(\alpha, \beta)_{1,4}$.

It is worth mentioning that **Strategy 2** corresponds to the one used in [11].

We have noticed that the performance of the code is extremely sensitive to the parameters of the branch-and-cut algorithm. In particular, the number of cuts added at each iteration (after each LP) seem to influence a lot the behavior of the algorithm.

We have fixed the maximum number of cuts generated at each iteration to **maxcut** $= \lceil \frac{25n}{2} \rceil$ for each class of inequalities. The separation routines are called in the same order they appear in the description of the strategy. The separation of a new family of inequalities is only executed if the number of violated inequalities found for the previous family is less than **maxcut**/2. It is worth noting that in [11] there is also an analogous parameter for **maxcut** but no reference for the value used in the computations is given.

Comparison between Cutting Strategies

The first test we have done concerns the pure branch-and-bound code using CPLEX 3.0 with default parameters. The larger instances of MEWCP used in computational experiments reported in the literature refer to graphs with up to 30 nodes ([5], [11]). Using branch-and-bound we have been able to solve most of the 30 node graphs in about one hour of CPU. The real challenge seem to solve problems for graphs with more than 40 nodes.

In Tables 1–4 below, we show the results obtained with Strategies 1–4 respectively for ten instances from our sample. In each of these tables we have the columns: **n**: the number of nodes in the graph; **b**: the maximum cardinality of a feasible clique; **k**: the parameter used to generate the edge weights; **# Nodes**: the number of nodes in the enumeration tree; **# LP**: the number of LPs solved; **# Cuts**: number of cuts generated for each family of valid inequalities in the corresponding strategy; **First Node**; the value of the lower bound after the last LP solved in the first node of the enumeration tree and **Time**: the CPU time (in seconds) needed to solve the instance.

Initially, for comparison purposes, we have restricted ourselves to the ten instances in Tables 1–4 for which n lies is in the range $\{40,\ldots,45\}$.

We analyze the quality of the cuts used in each strategy by looking at the number of nodes opened in the enumeration tree. According to that, the first and the last strategies outperform the two other strategies since the instances have been solved without branching. The number of LPs in the last strategy (except for the first instance) is always smaller when compared to the first strategy, while the total number of cuts added to the original formulation remains almost the same.

This indicates that the new inequalities introduced here, namely (α,β) inequalities, are more effective than the $C_{1,3}$ and $C_{2,3}$ inequalities in describing the optimal solution. However, Strategy 1 runs faster than Strategy 4.

Concerning the CPU time, Strategy 2 seem to be the best one, though Strategy 3 have been faster on a few instances. In Strategy 2, the tree inequalities do not seem to help in solving the problem. This is also the case in the computational results reported in [11]. The number of nodes in the enumeration tree is much larger for this strategy contrarily to the CPU times. From this observation one can raise the following question that appears in many branch-and-cut applications: how long should one proceed in a cutting-plane phase before branching?

The cuts of Strategy 2 are not as good as those in Strategies 1 and 4 to describe the optimal solution but this disadvantage is overcame by the fact that branching allows us to obtain the optimal solution quicker.

Tables 5 and 6 summarize the results we have obtained by applying Strategy 1 for the 60 instances we have generated for $n \geq 40$. Strategy 1 was chosen since it gives the best trade-off between the strenght of the cuts and CPU time. We have tried to solve this problems with standard branch-and-bound procedure limiting the number of nodes in the enumeration tree to 20000 and CPU time to one hour. The only instances solved were for the pairs $(n,k) \in \{(40,5)\}$ given by for positive edge weights and $(n,k) \in \{(40,4),(40,5),(42,4),(42,5)\}$ for mixed edge weights.

It is interesting to note that only four instances could not be solved by pure cutting planes. They are given by the pairs $(n,k) = \{(40,4), (44,4), (42,4), (48,5)\}$ with positve and negative weigths (see Table 6). We have tried to solve these instances using Strategy 4 and it turns out that they are solved without any branching which reinforces the conclusion that the (α,β)-inequalities are strong. The optimal values for these four instances are 27758, 32601, 32968 and 31351 respectively.

Positive weights										
					# Cuts					
n	b	k	# Nodes	# LP	# △	# $C_{1,3}$	# $C_{2,3}$	1st. node	Optimal	Time (sec.)
40	20	1	0	17	5289	3000	0	1099346	109346	3719.52
40	20	3	0	18	5075	2407	300	68759	68759	4063.29
40	20	4	0	27	5053	2523	1200	60782	60782	5419.51
44	22	1	0	15	6399	1650	0	136525	136525	3734.54
45	22	5	0	23	8090	3628	226	69563	69563	7873.28
Positive and negative weights										
					# Cuts					
n	b	k	# Nodes	# LP	# △	# $C_{1,3}$	# $C_{2,3}$	1st. node	Optimal	Time (sec.)
40	20	1	0	55	4044	4197	3800	70348	70348	23805.20
40	20	5	0	10	1596	2127	0	27967	27967	522.46
42	21	5	0	8	2738	678	105	35460	35460	295.94
44	22	1	0	63	5604	6471	4510	90620	90620	47799.73
45	22	1	0	29	5987	6351	791	102295	102295	23572.92

Table 1: Computational results for instances using **Strategy 1**.

Positive weights									
					# Cuts				
n	b	k	# Nodes	# LP	# △	# C△	# TRE	1st. node	Time (sec.)
40	20	1	6	20	5216	802	0	110437.83	2851.44
40	20	3	8	27	4862	293	1	69901.67	2929.24
40	20	4	12	40	5033	360	3	61867.32	3917.87
44	22	1	0	14	6328	634	1	136525	2439.72
45	22	5	6	28	8047	356	0	70166.45	6702.60
Positive and negative weights									
					# Cuts				
n	b	k	# Nodes	# LP	# △	# C△	# TRE	1st. node	Time (sec.)
40	20	1	64	107	4082	1407	74	76625.67	22335.44
40	20	5	0	7	1458	524	90	27967	162.77
42	21	5	0	8	2729	534	0	35460	285.00
44	22	1	78	122	5644	1554	74	97368.92	37371.73
45	22	1	10	30	5974	1254	44	105735.75	8717.98

Table 2: Computational results for instances using **Strategy 2**.

Positive weights

n	b	k	# Nodes	# LP	# Cuts: # $\triangle$	# Cuts: # $stC_{1,3}$	# Cuts: # $stC_{1,4}$	1st. node	Time(sec.)
40	20	1	0	17	5267	3000	0	109346	3777.23
40	20	3	0	16	4922	2451	0	68759	2881.20
40	20	4	2	22	5060	2597	155	60786.21	3154.60
44	22	1	0	15	6360	1650	0	136525	3942.37
45	22	5	2	27	8141	3687	164	69578.13	10775.66

Positive and negative weights

n	b	k	# Nodes	# LP	# Cuts: # $\triangle$	# Cuts: # $stC_{1,3}$	# Cuts: # $stC_{1,4}$	1st. node	Time(sec.)
40	20	1	8	40	4046	4639	433	73004.21	12832.67
40	20	5	0	10	1588	2127	200	27967	489.44
42	21	5	0	8	2739	678	105	35460	287.34
44	22	1	8	44	5543	5763	497	93037.97	35065.61
45	22	1	2	25	5948	4590	83	102994.05	12784.62

Table 3: Computational results for instances using **Strategy 3**.

Positive weights

n	b	k	# Nodes	# LP	# Cuts: # $\triangle$	# Cuts: # $(\alpha,\beta)_{1,4}$	1st. node	Time (sec.)
40	20	1	0	19	5251	4000	109346	5616.77
40	20	3	0	18	4849	3500	68759	3336.02
40	20	4	0	20	5027	4500	60782	6483.45
44	22	1	0	14	6313	1100	136525	3826.79
45	22	5	0	22	8051	3941	69563	9276.42

Positive and negative weights

n	b	k	# Nodes	# LP	# Cuts: # $\triangle$	# Cuts: # $(\alpha,\beta)_{1,4}$	1st. node	Time (sec.)
40	20	1	0	32	4034	11500	70348	23827.81
40	20	5	0	7	1546	1500	27967	214.15
42	21	5	0	8	2726	1050	35460	614.59
44	22	1	0	36	5537	13750	90620	37290.87
45	22	1	0	27	5962	9008	102295	23465.26

Table 4: Computational results for instances using **Strategy 4**.

6 Conclusions

In this paper we have introduced some new classes of valid and facet defining inequalities for $\mathcal{P}_C(b)$, the polytope corresponding to the convex hull of integer solutions for MEWCP. The (α, β)-inequalities were shown to be computationally efective. Moreover we also have been able to show that they generalize previously known classes of facet defining inequalities not only for $\mathcal{P}_C(b)$ but also for the Boolean Quadric Polytope $\mathcal{P}_B$.

The tree inequalities studied in [8] and [11] have also been generalized to include the case where the nodes in the tree do not form a cover for the feasible cliques. A second generalization for the tree inequalities is proposed in which a linear combination of nonvalid tree inequalities gives rise to a valid inequality for $\mathcal{P}_C(b)$ which, for special cases, has been proved to define a facet of this polytope. Computational use of both these generalized inequalities has still to be investigated. But it seems reasonable that one should first address the question of whether or not the tree inequalities are computationally useful.

We also have done several computational experiments both to confirm the strength of the (α, β)-inequalities and to compare different cutting strategies to be used in a cutting-plane framework. The results have confirmed that a quite small subclass of the (α, β)-inequalities are already very good in describing the part of $\mathcal{P}_C(b)$ on which lies the optimal solution.

Comparison between different cutting strategies has indicated that if more branching is allowed then some gain can be obtained in reducing the CPU time. If stronger cuts, like the (α, β)-inequalities, are to be used then efficient heuristics should be design since the exact separation routines spend a considerable amount of time.

References

[1] CPLEX Optimization, Inc. *Using the CPLEX Callable Library including Using the CPLEX Linear Optimizer with CPLEX Barrier and Mixed Integer Solvers*, 1989-1994. Version 3.0.

[2] C. C. de Souza. *The Graph Equipartition Problem: Optimal Solutions, Extensions and Applications*. PhD thesis, Université Catholique de Louvain, Faculté des Sciences Appliqués, Unité de Gestion Industrielle, Louvain-La-Neuve, Belgique, November 1993.

[3] C. C. de Souza and M. Laurent. Some new classes of facets for the equicut polytope. *Discrete Applied Mathematics*, 62:167 – 192, 1995.

[4] M. M. Deza and M. Laurent. *Geometry of cuts and metrics*. Springer Verlag, 1997.

[5] U. Faigle and G. Dijkhuizen. A Cutting-plane approach to the edge-weigthed maximal clique problem. *European Journal of Operational Research*, 69:121 – 130, 1993.

[6] U. Faigle, R. Schrader, and R. Suletzki. A Cutting-plane algorithm for optimal graph partitioning. *Methods of Operations Research*, 57:109 – 116, 1986.

[7] T. A. Feo and M. G. C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:1 – 27, 1995.

[8] E. L. Jonhson, A. Mehrotra, and G. L. Nemhauser. Min-cut clustering. *Mathematical Programming*, 62:133 – 151, 1993.

[9] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization.* John Wiley & Sons, 1988.

[10] M. Padberg. The Boolean quadric polytope: some characteristics, facets and relatives. *Mathematical Programming*, B-45:139 – 172, 1989.

[11] K. Park, K. Lee, and S. S. Park. An Extended formulation approach to the edge-weighted maximal clique problem. *European Journal of Operational Research*, 95:671 – 682, 1996.

[12] S. S. Park. *Private Communication.*

[13] H. Späth. Heuristically determining cliques of given cardinality and with minimal cost within weighted complete graphs. *Zeitschrift für Operations Research*, 29:125 – 131, 1985.

n	k	# Nodes	# LP	# Cuts			1st. node	Time (sec.)
				# $\triangle$	# $C_{1,3}$	# $C_{2,3}$		
40	1	0	17	5289	3000	0	109346	3719.52
	2	0	9	4000	0	0	82451	821.52
	3	0	18	5075	2407	300	68759	4063.29
	4	0	27	5053	2523	1200	60782	5419.51
	5	0	8	3500	0	0	60513	817.41
42	1	0	32	6149	4484	1260	120299	13679.36
	2	0	37	5419	4971	1785	87810	15919.71
	3	0	31	5470	2903	1575	76554	11298.25
	4	0	11	5021	0	0	69482	1666.13
	5	0	9	4200	0	0	76383	823.63
44	1	0	15	6399	1650	0	136525	3734.54
	2	0	31	6346	4628	1210	98186	15532.74
	3	0	14	6814	550	0	84675	2839.20
	4	0	16	6591	1650	0	75274	3584.06
	5	0	12	5703	550	0	69540	2389.17
45	1	0	60	7597	6065	4068	138694	39889.15
	2	0	42	7543	6086	2034	98321	30618.41
	3	0	43	6814	4103	2712	82743	25514.10
	4	0	20	7367	3191	0	77500	6401.01
	5	0	23	8090	3628	226	69563	7873.28
46	1	0	57	8645	7324	3450	142985	41205.22
	2	0	37	8712	5169	1495	108243	20712.69
	3	0	16	7575	1150	0	94859	2773.16
	4	0	26	8750	3890	345	78747	13901.90
	5	0	29	8581	3708	805	72431	14504.89
48	1	0	30	9665	7529	0	163397	25364.59
	2	0	105	9169	8783	9000	115471	103345.14
	3	0	64	8606	6435	4680	96666	51771.51
	4	0	17	8492	1200	0	88728	3920.53
	5	0	16	9000	0	0	82117	3487.15

Table 5: Computational results for instances with positive weights using branch-and-cut algorithm and **Strategy 1**.

n	k	# Nodes	# LP	# Cuts: # $\triangle$	# Cuts: # $C_{1,3}$	# Cuts: # $C_{2,3}$	1st. node	Time (sec.)
40	1	0	55	4044	4197	3800	70348	23805.20
	2	0	13	2233	4000	0	45404	2316.46
	3	0	11	2280	3000	0	34091	922.72
	4	2	34	2345	5212	1281	27772.52	4767.64
	5	0	10	1596	2127	0	27967	522.46
42	1	0	61	4799	5054	4410	81633	38285.58
	2	0	28	3540	4683	1260	46828	5873.03
	3	0	11	2664	2625	0	36689	1067.87
	4	0	5	1586	525	0	35987	54.99
	5	0	8	2739	678	105	35460	295.94
44	1	0	63	5604	6471	4510	90620	47799.73
	2	0	13	3439	3300	0	56960	4360.94
	3	0	12	3250	2750	0	40967	1255.87
	4	2	44	2415	5893	2698	32711.25	13740.63
	5	0	13	2507	4400	0	29407	1228.17
45	1	0	29	5987	6351	791	102295	23572.92
	2	0	30	4014	4805	1582	55103	9190.72
	3	0	8	2937	1100	0	43914	582.80
	4	0	27	2727	6374	1243	33990	6089.39
	5	0	32	3364	6564	1695	30974	10820.77
46	1	0	40	6093	7801	1840	99550	36453.30
	2	0	21	3873	4683	575	58361	6003.70
	3	0	27	3303	6588	1265	43915	9112.73
	4	2	52	3433	6851	3438	33054.06	28122.14
	5	0	13	3413	3450	0	31000	1400.60
48	1	0	99	7527	9368	8520	113478	124615.38
	2	0	62	5170	6197	5280	61768	45361.93
	3	0	17	4107	6000	0	45941	5214.58
	4	0	11	3160	2444	120	36903	1454.62
	5	2	33	3665	8387	1440	31404.64	11199.91

Table 6: Computational results for instances with positive and negative weights using branch-and-cut algorithm and **Strategy 1**.