Branch-and-Cut for the k-way equipartition problem

John E. Mitchell[1]

Mathematical Sciences
Rensselaer Polytechnic Institute
Troy, NY 12180

January 19, 2001

Abstract

We investigate the polyhedral structure of a formulation of the k-way equipartition problem and a branch-and-cut algorithm for the problem. The k-way equipartition problem requires dividing the vertices of a weighted graph into k equally sized sets, so as to minimize the total weight of edges that have both endpoints in the same set. Applications of the k-way equipartition problem arise in diverse areas including network design and sports scheduling. We describe computational results with a branch-and-cut algorithm.

Keywords: Graph equipartition, branch-and-cut, network design, realignment, clustering.

1 Introduction

We have a graph $G = (V, E)$ with edge weights c_e. The aim of the **k-way equipartition problem** is to divide the graph into k sets of vertices, each of the same size, so as to minimize the total weight of the edges which have both endpoints in one of the sets. We will call these k sets *divisions*. We assume $|V|$ is an integer multiple of k. This can be regarded as a clustering of the vertices, with the additional condition that each cluster must contain the same number $|V|/k =: S$ of elements. We define a binary variable x_{ij}, which takes the value 1 if i and j are in the same division and 0 otherwise. Our formulation is:

$$\begin{array}{lll} \min & \sum_{e \in E} c_e x_e & \\ \text{subject to} & \sum_{e \in \delta(v)} x_e = S - 1 & \forall v \in V \\ & x \text{ is the incidence vector of a clustering} & \end{array}$$

[1]Research supported in part by NSF grant number CCR–9901822

where $\delta(v)$ denotes the set of edges incident to vertex v. In what follows we assume G is the complete graph on $|V|$ vertices; this can be done without loss of generality — any missing edges in a graph can be given weight $c_e = 0$ in the complete graph.

In addition to being a constrained version of the clustering problem, the k-way equipartition problem is closely related to the classical graph partition problem. This requires partitioning the vertices into two equally sized sets U_1 and U_2 so as to minimize the total weight of the edges that either have both endpoints in U_1 or both in U_2.

Approaches to the k-way graph partitioning algorithm include the multilevel approach of Karypis and Kumar [20] and the eigenvalue approach of Donath and Hoffman [11], tested computationally by Areibi and Vannelli [1] and by Falkner *et al.* [12]. If $k = 2$ we have the equipartition problem, which has been studied extensively. Branch and cut approaches to this problem include the work of Conforti *et al* [4, 8, 9] and Chopra [5]. Ferreira *et al.* [13, 14] have developed a branch-and-cut algorithm for the related node capacitated graph partitioning problem. Semidefinite programming approaches for graph equipartition were investigated by Frieze and Jerrum [15] and Ye *et al.* [3, 29]. Semidefinite programming approaches for the k-way equipartition problem were developed by Karisch and Rendl [19] and Wolkowicz and Zhao [28], and we consider the approach of [19] in more detail in §5. Lisser and Rendl [22] have recently described an application of the k-way equipartition problem to network design problems and they investigated both semidefinite and polyhedral relaxations of the problem.

De Souza *et al.* [27] proposed solving this problem by breaking it into a sequence of graph partitioning problems, for the special case of minimizing the frontwidth in finite element calculations. Thus, a graph partitioning problem is solved on the initial graph, giving two sets U_1 and U_2. Graph partitioning problems are then solved recursively on U_1 and U_2 until a k-equipartition is obtained. This approach may not produce the optimal k-partition in the general case.

If the sets are not all constrained to be of the same cardinality, we have a clustering or partition problem. This problem has been investigated by Grötschel and Wakabayashi [17, 18] and we look at their approach in §3. Chopra and Rao [6] have investigated a polyhedral approach for the partition problem for graphs that are sparse; their approach defines variables for both the edges and the vertices.

When $S = 2$, we have a matching problem, so the problem is polynomially solvable. For larger choices of S, the problem is NP-complete, as shown in Garey and Johnson [16].

One motivation for considering this problem arises from consideration of alignment of teams. The National Football League (NFL) in the United States currently comprises 31 teams. It will expand to 32 in 2002 with the addition of a team in Houston. At that point, there is a possibility that the league will be realigned into eight divisions, each containing four teams. Each team plays each other team in its division twice, playing its remaining games against a subset of the teams outside its division. A team's opponents from outside its division depend on the team's results in the previous season and on a rotating choice of one other division. Since the choice of outside teams will therefore vary from season to season, we propose to choose the divisions so as to minimize the total intradivisional travel distance. In this setting, an edge weight will be the distance between the two endpoints of the edge. For more details on this application, see [24].

1.1 Notation

We define the sets $Q(kS)$ and $\bar{Q}(kS)$ as follows:

$$\begin{aligned} Q(kS) &:= \{x \in \{0,1\}^n : \sum_{e\in\delta(v)} x_e = S-1 \;\; \forall\, v \in K_{kS}, \\ &\qquad\qquad x \text{ is the incidence vector of a clustering}\} \\ \bar{Q}(kS) &:= \{x \in [0,1]^n : \sum_{e\in\delta(v)} x_e = S-1 \;\; \forall\, v \in K_{kS}\}, \end{aligned}$$

where $n = kS(kS-1)/2$ and K_q denotes the complete graph on q vertices. We want to minimize the objective function over $Q(kS)$. Our initial linear programming relaxation will have feasible region $\bar{Q}(kS)$.

Given a subset $U \subseteq V$, we define $E(U)$ to be the edges with both endpoints in U, and we define $x(U) := \sum_{e\in E(U)} x_e$. Similarly, we define $\delta(U)$ to be the edges with exactly one endpoint in U. Given two disjoint subsets $U \subseteq V$ and $W \subseteq V$, we define $E(U,W)$ to be the edges with exactly one endpoint in U and exactly one endpoint in W; further, we define $x(U,W) := \sum_{e\in E(U,W)} x_e$. If $U \subseteq V$ and $v \in V \setminus U$ then $x(v,U)$ denotes $\sum_{u\in U} x_{uv}$. A matrix with every entry equal to one is denoted $\mathbf{1}$. Note that $\mathbf{1} = ee^T$, where e denotes the vector of ones. All vectors will be column vectors. The transpose of a matrix M will be written M^T. If C denotes a cycle then $E(C)$ denotes the edges of the cycle and $x(E(C)) := \sum_{e\in E(C)} x_e$.

Our definition of x agrees with the work of Grötschel and Wakabayashi [17, 18] on the clustering problem. Note that it is the opposite of that in literature on the MAXCUT problem, where typically we take $x_e = 1$ if edge e appears in the cut.

2 Polyhedral theory for the equipartition polytope

Brunetta *et al.* [4] have developed a branch-and-cut algorithm for the equicut problem. Their work is based on that of Conforti *et al.* [8, 9], who developed a great deal of polyhedral theory for the equipartition problem.

They proved the following result regarding the dimension of the equicut partition:

Lemma 1 *([8], Lemma 3.5.) The dimension of the equicut polytope on $2S$ vertices is* $\binom{2S}{2} - S$.

Among the families of cutting planes that they describe for the equipartition problem on a graph with $2S$ vertices are the following two classes of facet defining inequalities:

- *Clique inequalities ([9], Theorem 6.1):* For every complete subgraph with q vertices, we have $x(E(K_q)) \geq \lfloor \frac{1}{2}q \rfloor^2$, provided $q \geq 3$ and odd.

- *Cycle inequalities ([9], Theorem 6.2):* For every cycle C of length $S+1$, we have

$$x(E(C)) \leq S - 1. \tag{1}$$

(It should be noted that Conforti *et al.* show that facet defining inequalities for the equipartition polytope on $2S-1$ vertices can be extended to facet defining inequalities for the equipartition polytope on $2S$ vertices in a natural way, and vice versa (see [8], Remark, page 59).)

The cycle inequalities can be used for the k-way equipartition problem, as we show in Corollary 1. However, the clique inequalities are no longer valid. This is a consequence of the following simple lemma.

Lemma 2 *Let $G' = (V', E')$ be a subgraph of G with $|V'| \leq k$. Let $a^T x \geq b$ be a valid inequality for the k-way equipartition problem with $a \geq 0$ and $a_e = 0$ if $e \notin E'$. We must then have $b = 0$, so the inequality is trivial.*

Proof: Feasible solutions to the k-way equipartition problem can be obtained where each vertex in V' is in a different division, and $x_e = 0$ for all $e \in E'$ for these feasible solutions. □

Many of the other families of cutting planes in [4, 8, 9] exploit the fact that the problem is an equipartition, and thus cannot be used directly for the k-way equipartition problem, again because of this lemma. We can convert them into useful inequalities by exploiting the fact that $x(E) = S(S-1)$ for the equipartition problem, so an inequality in the variables $x(E')$ is equivalent to an inequality in the variables $x(E \setminus E')$. Thus, the clique inequalities can be stated equivalently as

$$x(E \setminus E(K_q)) \leq S(S-1) - \lfloor \frac{1}{2} q \rfloor^2 \tag{2}$$

where q is odd and at least three.

3 Polyhedral theory for the clustering polytope

In the NP-hard clustering problem, we are given a set of p observations, each of which possesses k characteristics. The objective is to divide the observations into clusters where the observations within each cluster are similar to one another. For example, the observations could consist of different types of computers, and the characteristics could include the speed of the computer, the amount of RAM of the computer and the size of the hard disk of the computer. There are no *a priori* constraints on the number of clusters or on the number of elements in a cluster. The k-way equipartition problem is a version of the clustering problem where all the clusters are required to have the same prescribed size.

Grötschel and Wakabayashi [17, 18] described a simplex-based cutting plane algorithm for the clustering problem. The set of incidence vectors of feasible clusters with p observations are given by the solutions to the following set of constraints:

$$-x_{ij} + x_{il} + x_{jl} \leq 1 \quad \text{for } 1 \leq i < j < l \leq p \tag{3}$$

$$x_{ij} - x_{il} + x_{jl} \leq 1 \quad \text{for } 1 \leq i < j < l \leq p \tag{4}$$

$$x_{ij} + x_{il} - x_{jl} \leq 1 \quad \text{for } 1 \leq i < j < l \leq p \tag{5}$$

$$x_{ij} = 0 \text{ or } 1, \; 1 \leq i < j \leq p$$

where we interpret $x_{ij} = 1$ to mean that i and j are in the same cluster, and $x_{ij} = 0$ to mean that they are in different clusters. The constraints (3), (4), and (5) are called *triangle inequalities*. Constraint (3) corresponds to the logical condition that if i and j are in different clusters then l can not be in the same cluster as both i and j; constraints (4) and (5) have similar interpretations. All these inequalities define facets of the convex hull of the set of feasible solutions to the clustering problem. We used these inequalities as cutting planes in our algorithm for the k-way equipartition problem. The triangle inequalities are also presented in [4] and they are well known in the literature for the MAX-CUT problem; see, for example, Barahona and Mahjoub [2].

Other classes of facets for this problem are known, but a complete description of the convex hull is not currently known — see Grötschel and Wakabayashi [17, 18] for more details. We have the following theorem.

Theorem 1 *([18], Theorem 4.1.) For every nonempty disjoint subsets $U, W \subseteq V$, the 2-partition inequality*

$$x(U,W) - x(U) - x(W) \leq \min\{|U|, |W|\} \tag{6}$$

defines a facet of the clique partioning polytope, provided $|U| \neq |W|$.

Chopra and Rao [6] investigated a version of this problem with an upper bound p on the number of clusters, where the clusters can be any size. Let Q be any subset of the vertices of cardinality $p+1$. They show that the clique inequality

$$\sum_{E(Q)} x_e \geq 1 \tag{7}$$

is facet-defining. Further, they show that certain generalizations of this constraint are also facet defining.

4 Polyhedral theory for the k-way equipartition problem

We have used the inequalities described in §2 and §3. Further, we have developed some valid inequalities specifically for the k-way equipartition problem; we enumerate these inequalities in this section.

4.1 Dimension

The main result of this section is that the dimension of $Q(kS)$ is $d(kS)$ provided $S > 2$, where we define

$$d(q) := \binom{q}{2} - q. \tag{8}$$

It follows directly from Lemma 3.3 in [8] that this is an upper bound on the dimension of $Q(kS)$. In order to show the dimension is at least $d(kS)$, it suffices to exhibit $d(kS)+1$ linearly independent vectors in $Q(kS)$.

We first need to show several technical lemmas.

Lemma 3 *The $n \times n$ matrices $\mathbf{1} - I$ and $\mathbf{1} - 2I$ are of full rank, provided $n \geq 3$.*

Proof: Assume u satisfies $(\mathbf{1} - I)u = 0$. Subtracting row i from row j shows that $u_i = u_j =: \bar{u}$ for any $1 \leq i < j \leq n$. Any row then implies that $(n-1)\bar{u} = 0$, so $u = 0$ and the columns of $\mathbf{1} - I$ are linearly independent.

Now assume u satisfies $(\mathbf{1} - 2I)u = 0$. Subtracting row i from row j again shows that $u_i = u_j =: \bar{u}$ for any $1 \leq i < j \leq n$. Any row then implies that $(n-2)\bar{u} = 0$, so $u = 0$ and the columns of $\mathbf{1} - 2I$ are linearly independent. □

Define the $n^2 \times n^2$ matrix

$$\bar{M} := \begin{pmatrix} \mathbf{1} - I & I & \dots & I \\ I & \mathbf{1} - I & \dots & I \\ \vdots & \vdots & \ddots & \vdots \\ I & I & \dots & \mathbf{1} - I \end{pmatrix} \tag{9}$$

consisting of n^2 blocks, with each block being $n \times n$.

Lemma 4 *The matrix $\bar{M}$ has full rank provided $n \geq 3$.*

Proof: Let u^i, $i = 1, \dots, n$, be n-vectors. Let $u = ((u^1)^T, \dots, (u^n)^T)^T$. Assume $\bar{M}u = 0$. It suffices to show that $u = 0$. For $i = 1, \dots, n-1$, premultiplying the ith block of equations by $\mathbf{1} - I$ and subtracting the last block of equations gives

$$(\mathbf{1} - I)(\mathbf{1} - I)u^i + \sum_{j=1, j \neq i}^{n-1} (\mathbf{1} - I)u^j - \sum_{j=1}^{n-1} u^j = 0 \text{ for } i = 1, \dots, n-1.$$

This can be rewritten

$$\begin{aligned} 0 &= (\mathbf{1}^2 - 2\mathbf{1})u^i + \sum_{j=1, j \neq i}^{n-1} (\mathbf{1} - 2I)u^j \\ &= (\mathbf{1} - 2I)(\mathbf{1}u^i + \sum_{j=1, j \neq i}^{n-1} u^j). \end{aligned}$$

It follows from Lemma 3 that $\mathbf{1}u^i + \sum_{j=1, j \neq i}^{n-1} u^j = 0$. Subtracting this expression from the ith block of equations for $\bar{M}u = 0$ gives $-u^i + u^n = 0$, so $u^i =: \bar{u}$ for $i = 1, \dots, n$. The ith block of equations of $\bar{M}u = 0$ then gives

$$0 = \mathbf{1}\bar{u} + (n-2)\bar{u}.$$

Now, the eigenvalues of $\mathbf{1}$ are n with multiplicity one, and 0 with multiplicity $n-1$. Thus, we must have $\bar{u} = 0$ provided $n > 2$, so $\bar{M}$ is of full rank. □

We use induction on the value of k to prove the result on the dimension of $Q(kS)$. In particular, assuming the dimension is $d(kS)$ for kS vertices, we show it is $d((k+1)S)$ for $(k+1)S$ vertices. The base case of $k = 2$ follows from Lemma 1.

The origin is not in $Q(kS)$, so there are $d(kS) + 1$ linearly independent incidence vectors of k-way equipartitions in $Q(kS)$. Each of these can be extended to a $(k+1)$-way equipartition of $Q((k+1)S)$ by placing the additional S vertices in a single cluster.

The additional S vertices provide an additional $S^2k + S(S-1)/2$ edges. We generate a further $S^2k+S(S-1)/2-(S-1)$ incidence vectors of points in $Q((k+1)S)$ in two stages. First, we pick any k-way partition of the original kS vertices, and we designate one of the resulting clusters as special. For each vertex j in one of the non-special clusters and for each new vertex i, we generate a clustering by interchanging i and j. This gives $S^2(k-1)$ incidence vectors.

Next, we keep the non-special clusters as they are. The equipartition polytope consisting of the $2S$ vertices in the new cluster C^1 and the special cluster C^2 has dimension $d(2S) = 2S(2S-1)/2 - 2S$. Since it does not contain the origin, it must contain $d(2S) + 1$ linearly independent incidence vectors. We can write the first $S(2S-1)$ components of these $d(2S) + 1$ incidence vectors as follows:

$$\Phi := \begin{bmatrix} \Xi \\ \Lambda \\ \Gamma \end{bmatrix} \text{ corresponding to edges with } \begin{cases} \text{both endpoints in } C^1 \\ \text{one endpoint each in } C^1 \text{ and } C^2 \\ \text{both endpoints in } C^2 \end{cases}$$

Thus, Φ has $d(2S) + 1$ columns, Λ has S^2 rows, and Ξ and Γ each have $S(S-1)/2$ rows.

In order to prove that the dimension of the union of these three sets of incidence vectors in $Q((k+1)S)$ is large enough, we need to look first at submatrices of Φ.

Lemma 5 *Without loss of generality, we can assume Λ has full row rank.*

Proof: Note that Λ has S^2 rows. Construct the first S^2 incidence vectors by interchanging each pair of elements from C^1 and C^2. The first S^2 columns of Λ are then $\bar{M}$ with $n = S$. This can be seen by ordering the rows so that the ith block of rows correspond to the edges with one of the endpoints being the ith vertex in C^1; the jth block of columns correspond to the incidence vectors obtained by interchanging the jth element of C^1 with each element in C^2 in succession. The result then follows from Lemma 4. □

Lemma 6 *Without loss of generality, we can assume*

$$rank \begin{bmatrix} \Xi \\ \Lambda \end{bmatrix} - \; rank(\Lambda) \geq rank(\Phi) - \; rank \begin{bmatrix} \Xi \\ \Lambda \end{bmatrix}.$$

Proof: We show by contradiction that we cannot have both of the following relationships:

$$\begin{aligned} \text{rank} \begin{bmatrix} \Xi \\ \Lambda \end{bmatrix} - \text{rank}(\Lambda) &< \text{rank}(\Phi) - \text{rank} \begin{bmatrix} \Xi \\ \Lambda \end{bmatrix} && (10) \\ \text{rank} \begin{bmatrix} \Lambda \\ \Gamma \end{bmatrix} - \text{rank}(\Lambda) &< \text{rank}(\Phi) - \text{rank} \begin{bmatrix} \Lambda \\ \Gamma \end{bmatrix}. && (11) \end{aligned}$$

The following relationships hold for any matrices Ξ, Γ, Λ:

$$\begin{aligned} \text{rank} \begin{bmatrix} \Xi \\ \Lambda \\ \Gamma \end{bmatrix} - \text{rank} \begin{bmatrix} \Xi \\ \Lambda \end{bmatrix} &\leq \text{rank} \begin{bmatrix} \Lambda \\ \Gamma \end{bmatrix} - \text{rank}(\Lambda) && (12) \\ \text{rank} \begin{bmatrix} \Xi \\ \Lambda \\ \Gamma \end{bmatrix} - \text{rank} \begin{bmatrix} \Lambda \\ \Gamma \end{bmatrix} &\leq \text{rank} \begin{bmatrix} \Xi \\ \Lambda \end{bmatrix} - \text{rank}(\Lambda). && (13) \end{aligned}$$

Using (10) followed by (12) then (11) and (13) gives the contradiction

$$\text{rank} \begin{bmatrix} \Xi \\ \Lambda \end{bmatrix} - \text{rank}(\Lambda) < \text{rank} \begin{bmatrix} \Xi \\ \Lambda \end{bmatrix} - \text{rank}(\Lambda).$$

Thus, at least one of (10) and (11) is not satisfied. By the symmetry of the definition of Φ, we can choose Ξ and Γ so that the result follows. □

Lemma 7 *Under the assumptions of Lemmas 5 and 6, the rank of* $\begin{bmatrix} \Xi \\ \Lambda \end{bmatrix}$ *is at least* $\frac{3S^2-3S}{2} + 1$.

Proof: From Lemma 6, we have:

$$\text{rank} \begin{bmatrix} \Xi \\ \Lambda \end{bmatrix} - \text{rank}(\Lambda) \geq \text{rank}(\Phi) - \text{rank} \begin{bmatrix} \Xi \\ \Lambda \end{bmatrix}.$$

Using the fact that

$$\text{rank}(\Phi) = 2S(2S-1)/2 - (2S-1)$$

gives us

$$\begin{aligned} \operatorname{rank}\begin{bmatrix} \Xi \\ \Lambda \end{bmatrix} &\geq \frac{1}{2}(\operatorname{rank}(\Phi) + \operatorname{rank}(\Lambda)) \\ &= \frac{1}{2}(2S(2S-1)/2 - (2S-1) + S^2) \\ &= \frac{1}{2}(3S^2 - 3S + 1). \end{aligned}$$

Since the rank must be integral, we can round up this last lower bound, giving the required result. $\square$

We now let Φ denote $\frac{1}{2}(3S^2 - 3S) + 1$ of these columns, ensuring that the resulting columns of $\begin{bmatrix} \Xi \\ \Lambda \end{bmatrix}$ are linearly independent.

The matrix of these three sets of incidence vectors can be written as follows:

$$\check{M} := \left[\begin{array}{c|ccc|c} \mathbf{1} & & P & & \Xi \\ \hline 0 & & 0 & & \Lambda \\ \hline 0 & \bar{M} & & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & & \bar{M} & 0 \\ \hline & & \mathbf{1} & & \Gamma \\ \cline{2-5} & & & & \\ & & Q & & \mathbf{1} \\ \hat{M} & & & & \\ \cline{2-5} & & & & \\ & & 0 & & 0 \\ & & & & \end{array}\right] \qquad \begin{array}{l} C^1 \\ C^1C^2 \\ C^1C^3 \\ \vdots \\ C^1C^{k+1} \\ C^2 \\ C^3 \\ \vdots \\ C^{k+1} \\ C^2C^3 \\ \vdots \\ C^kC^{k+1} \end{array}$$

Here, the single label C^i indicates the $S(S-1)/2$ rows corresponding to edges with both endpoints in C^i and the double label C^iC^j indicates the S^2 rows corresponding to edges with one endpoint in C^i and the other endpoint in C^j. The first block of columns corresponds to the $d(kS)+1$ incidence vectors arising from the k-way equipartition problem on the vertices in sets $C^2, \ldots, C^{k+1}$, so the columns of $\hat{M}$ are linearly independent. The middle set of $S^2(k-1)$ vectors correspond to the first additional set described above, and the last block of columns corresponds to the final set of incidence vectors defined above. We do not need to specify the matrices P and Q further.

Lemma 8 *The rank of the matrix $\check{M}$ is $d((k+1)S)+1$, provided $S \geq 3$.*

Proof: We note first that the number of columns of $\check{M}$ is

$$\begin{aligned} & d(kS) + 1 + S^2(k-1) + \frac{3S^2 - 3S}{2} + 1 \\ = & \ \frac{1}{2}(kS(kS-1)) - kS + 1 + S^2(k-1) + \frac{3S^2 - 3S}{2} + 1 \\ = & \ \frac{1}{2}((k^2 + 2k + 1)S^2 - (3k+3)S) + 2 \\ = & \ d((k+1)S) + 2. \end{aligned}$$

The result will follow if we can show that the dimension of the nullspace of $\check{M}$ is one.

Let $({u^1}^T, {u^2}^T, {u^3}^T)^T$ be in the nullspace of $\check{M}$. From Lemma 4, it follows that $u^2 = 0$. The first two blocks of rows then require that

$$(e^T u^1) \begin{bmatrix} e \\ 0 \end{bmatrix} + \begin{bmatrix} \Xi \\ \Lambda \end{bmatrix} u^3 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

It follows from Lemma 7 and the subsequent redefinition of Φ that there is a unique solution u^3, up to scalar multiplication.

The rows corresponding to the original edges give the equations

$$\hat{M} u^1 + \begin{bmatrix} \Gamma \\ \mathbf{1} \\ 0 \end{bmatrix} u^3 = 0.$$

From the induction hypothesis, the columns of $\hat{M}$ are linearly independent. Since there is a unique choice for u^3 (up to scalar multiplication), it follows that there is a unique choice for u^1 (again, up to the same scalar multiplication). Thus, the dimension of the nullspace of $\check{M}$ is one, and the result follows. □

We can now summarize by giving the dimension of $Q(kS)$.

Theorem 2 *The dimension of $Q(kS)$ is*

$$d(kS) = \binom{kS}{2} - kS,$$

provided $S > 2$ and $k \geq 2$.

Proof: The sequence of Lemmas 5–8 show that this result holds by induction on the value of k. □

4.2 Lifting inequalities

In this section, we show that certain facet defining inequalities for the equipartition problem can be converted into facet defining inequalities for the k-way equipartition problem. We assume that the vertices of our graph are $V := \cup_{i=1}^{k} C^i$ and that $|C^i| = S$ for $i = 1, \ldots, k$. We make three assumptions about an inequality $a^T x \leq b$:

1. The inequality defines a facet of the p-way equipartition polytope on the graph with vertices $\cup_{i=1}^{p} C^i$, for some $p < k$. For notational purposes, we assume that the incidence vector of the equipartition with the vertices in C^i in different divisions for $i = 1, \ldots, p$ satisfies the constraint at equality.

2. The inequality is valid for the k-way equipartition problem on the graph with vertices V, where the coefficient of any edge that does not have both endpoints in $\cup_{i=1}^{p} C^i$ is zero.

3. For each vertex $v \in \cup_{i=1}^{p} C^i$, there exists an incidence vector x^v of a p-way equipartition that satisfies the constraint at equality and that has $x_e^v = 0$ for every edge e incident to vertex v with nonzero coefficient a_e.

Of course, the equipartition problem corresponds to $p = 2$.

Under these assumptions, we construct three sets of vectors in $Q(kS)$ that satisfy the constraint at equality, and such that the dimension of the union of all these vectors is $d(kS) - 1$. The three sets are defined as follows:

1. Generate a linearly independent set of $d((k-p)S) + 1$ incidence vectors for the $(k-p)$-way equipartition problem on vertices $\cup_{i=p+1}^{k} C^i$. Extend these to incidence vectors on the graph with vertices V by placing C^i, $i = 1, \ldots, p$ in their own divisions.

2. Start with the equipartition $C^1, \ldots, C^k$. For each vertex $i \in \cup_{l=1}^{p} C^l$ and each vertex $j \in \cup_{q=p+1}^{k} C^q$, generate a new equipartition by interchanging vertices i and j. If necessary, rearrange the vertices in $(\cup_{l=1}^{p} C^l) \setminus i$ so as to satisfy the constraint at equality; the final assumption above ensures that this can be done. This gives $p(k-p)S^2$ equipartitions.

3. Generate $d(pS)$ linearly independent incidence vectors for the equipartition problem on the graph with vertices $\cup_{i=1}^{p} C^i$, each of which satisfies the constraint at equality. Extend each of these out to the whole of V by placing each of $C^{p+1}, \ldots, C^k$ in its own division.

Lemma 9 *The procedure outlined above gives $d(kS)+1$ points in $Q(kS)$ that satisfy the constraint at equality.*

Proof: The total number p of points is

$$
\begin{aligned}
p &= d((k-p)S)+1+p(k-p)S^2+d(pS) \\
&= \frac{1}{2}(k-p)S((k-p)S-1)-(k-p)S+p(k-p)S^2+\frac{1}{2}pS(pS-1)-pS \\
&= \frac{k^2S^2}{2}-\frac{3kS}{2}+1 \\
&= \frac{1}{2}kS(kS-1)-kS+1 \\
&= d(kS)+1,
\end{aligned}
$$

as required. ☐

The matrix of these three sets of incidence vectors can be written as follows:

$$
\check{M} := \left[\begin{array}{c|ccc|c}
\mathbf{1} & & P & & \Xi \\
\hline
0 & & 0 & & \Lambda \\
\hline
0 & \bar{M} & & 0 & 0 \\
\vdots & & \ddots & & \vdots \\
0 & 0 & & \bar{M} & 0 \\
\hline
 & & Q & & \mathbf{1} \\
\hat{M} & & & & \\
\cline{2-5}
 & & 0 & & 0
\end{array}\right]
\begin{array}{l}
C^1 \\ \vdots \\ C^p \\ C^1C^2 \\ \vdots \\ C^{p-1}C^p \\ C^1C^{p+1} \\ \vdots \\ C^pC^k \\ C^{p+1} \\ \vdots \\ C^k \\ C^{p+1}C^{p+2} \\ \vdots \\ C^{k-1}C^k
\end{array}
$$

Here, the single label C^i indicates the $S(S-1)/2$ rows corresponding to edges with both endpoints in C^i and the double label C^iC^j indicates the S^2 rows corresponding to edges with one endpoint in C^i and the other endpoint in C^j. Note that $\check{M}$, $\hat{M}$, P, Q, Ξ, and Λ all have different meanings from in §4.1. The matrix $\bar{M}$ is defined as in equation (9). The first block of columns corresponds to the $d((k-p)S)+1$

incidence vectors arising from the $(k-p)$-way equipartition problem on the vertices in sets C^{p+1},...,C^k, so the columns of $\hat{M}$ are linearly independent. The middle set of $p(k-p)S^2$ vectors correspond to the second set of incidence vectors described above. The matrices Ξ and Λ are redefined so that the last block of columns corresponds to the final set of incidence vectors defined above. We do not need to specify the matrices P and Q further.

Lemma 10 *The rank of matrix $\check{M}$ is $d(kS)$.*

Proof: Let $(u^{1^T}, u^{2^T}, u^{3^T})^T$ be in the nullspace of $\check{M}$. From Lemma 4, it follows that $u^2 = 0$. The first two blocks of rows then require that

$$(e^T u^1) \begin{bmatrix} e \\ 0 \end{bmatrix} + \begin{bmatrix} \Xi \\ \Gamma \end{bmatrix} u^3 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

It follows from the linear independence of the columns of $[\Xi^T, \Gamma^T]^T$ that there is a unique solution u^3, up to scalar multiplication.

The rows corresponding to the edges with both endpoints in $\cup_{i=p+1}^k C^i$ give the equations

$$\hat{M} u^1 + \begin{bmatrix} \mathbf{1} \\ 0 \end{bmatrix} u^3 = 0.$$

The columns of $\hat{M}$ are linearly independent. Since there is a unique choice for u^3 (up to scalar multiplication), it follows that there is a unique choice for u^1 (again, up to the same scalar multiplication). Thus, the dimension of the nullspace of $\check{M}$ is one, and the result follows from Lemma 9. □

This leads directly to the following theorem:

Theorem 3 *Any inequality satisfying the three assumptions given above defines a facet of $Q(kS)$, provided $S \geq 3$.*

A corollary of this theorem is that the cycle inequalities defined in (1) can be extended to facet defining inequalities for the k-way equipartition problem.

Corollary 1 *Given a cycle C of length $S+1$, the inequality $x(E(C)) \leq S-1$ defines a facet of $Q(kS)$, provided $S \geq 3$.*

Proof: The assumptions of Theorem 3 hold:

1. The inequality defines a facet of the equipartition polytope on $2S$ vertices, as noted earlier.

2. The inequality is valid for the k-way equipartition problem since the vertices of the cycle must belong to at least two divisions, so there are at least two edges on the cycle whose endpoints are in different divisions.

3. For any vertex on the cycle, the k-way equipartition where all the other vertices on the cycle are placed in a single division satisfies the third assumption. For any vertex not on the cycle, the third assumption is satisfied by any k-way equipartition.

The result then follows as an immediate consequence of Theorem 3. □

For the clique inequalities given in (2), the third assumption will not be satisfied for any vertex not in the clique K_q. This is because every edge incident to each of these vertices appears with coefficient a_e equal to one in the constraint and every equipartition must use $p-1$ of these edges.

4.3 Valid inequalities

There are several families of inequalities that we have used in our cutting plane approach, beyond those indicated in §2 and §3. Some of these families are defined on the vertices of an equipartition polytope, although they are not facet defining for the equipartion problem. Other families use more than $2S$ vertices.

4.3.1 Inequalities from the equipartition polytope

There is a limit on the number of edges that can be used from any complete subgraph.

Theorem 4 *Let $U \subseteq V$, with $|U| = S + p$ and $2 \leq p \leq S-1$. The following is a valid inequality:*

$$\sum_{e \in E(U)} x_e \leq \binom{S}{2} + \binom{p}{2}. \tag{14}$$

Proof: The following configuration satisfies the constraint at equality:

- Let $\hat{U} \subseteq U$ with $|\hat{U}| = S$ be one division and place the remaining vertices from U in another division.

All other valid configurations satisfy the constraint strictly. □

When $k = 2$, we must have $x(u, U) \geq p$ for each of the $S - p$ vertices $u \notin U$. Summing the degree constraints for the $S + p$ vertices in U gives

$$\begin{aligned} 2 \sum_{e \in E(U)} x_e &= (S-1)(S+p) - \sum_{u \notin U} \sum_{v \in U} x_{uv} \\ &\leq S(S-1) + p(p-1) + p(S-p) - (S-p)p \\ &= 2 \binom{S}{2} + 2 \binom{p}{2}. \end{aligned} \tag{15}$$

Thus, for the equipartition problem, (14) is implied by the degree constraints. Nonetheless, this inequality is violated by some points in the LP relaxation of the k-way equipartition problem for $k > 2$. For example, if $k = 3$ and $S = 4$, divide the twelve vertices into two equal sets. Set $x_e = 0.6$ for each edge with both endpoints within one set, and take $x_e = 0$ otherwise. This point satisfies the degree constraints, the triangle constraints, and the cycle constraints but it violates (14).

The next theorem discusses two inequalities defined on cliques with $S+1$ vertices. These are stronger for the equipartition polytope than those given in (14), but they are implied by the cycle inequalities, so they are of limited use in solving the k-way equipartition problem.

Theorem 5 *Let $W \subseteq V$ with $|W| = S + 1$. The following is a valid inequality:*

$$\sum_{e \in E(W)} x_e \leq \binom{S}{2}. \tag{16}$$

Further, let $U \subseteq V$, with $|U| = S - 1$ and let v_1 and v_2 be two other vertices. The following is a valid inequality:

$$2 \sum_{e \in E(U)} x_e + (S-1) x_{v_1 v_2} + \sum_{i=1}^{2} \sum_{v \in U} x_{v_i v} \leq (S-1)^2. \tag{17}$$

Proof: The complete graph on W can be covered by S cycles of length $S + 1$, where each edge appears in exactly two of the cycles. Summing the cycle inequalities for these cycles gives:

$$2x(W) \leq S(S-1),$$

showing immediately the validity of (16).

Similarly, the complete graph on $\{U, v_1, v_2\}$ can be covered by $2(S - 1)$ cycles of length $S+1$, with edge (v_1, v_2) appearing in every cycle, each edge of the form (v_i, u)

($i = 1, 2,\ u \in U$) appearing in exactly four cycles, and each edge in $E(U)$ appearing in exactly two cycles. Summing the corresponding cycle inequalities gives (17). $\square$

In the next theorem, we give another valid inequality for the equipartion problem that extends to the k-way equipartition problem.

Theorem 6 *Let U and W be two disjoint subsets of V with $|U| = |W| = S - 1$. The following is a valid inequality:*

$$(S-1)\sum_{e\in E(U)} x_e + (S-1)\sum_{e\in E(W)} x_e + (S-2)\sum_{e\in E(U,W)} x_e \leq (S-2)(S-1)^2. \quad (18)$$

Proof: The following configurations satisfy the constraint at equality:

- Let U be part of one division and let W be part of another division.
- Let U together with one vertex from W be one division and let the rest of W be contained in another division.
- Let W together with one vertex from U be one division and let the rest of U be contained in another division.

All other valid configurations satisfy the constraint strictly. $\square$

Example: Consider a graph with $S = 4$, so $|U| = |W| = 3$. The right hand side of (18) is 18, and in Figure 1 the red edges have coefficient 3 while the black edges have coefficient 2.

Consider this theorem for the k-way equipartition problem with $k = 3$. Let $\bar{U} := V \setminus (U \cup W)$. This is not implied by the constraints considered earlier. For example, let $\delta = \frac{1}{S^2-4S+5}$, $\epsilon = \frac{2\delta}{S-1}$, $\nu = \frac{S-2}{(S+2)(S-1)}$, and $\gamma = \frac{4S}{(S+1)(S+2)}$ and then set x_e as:

$$x_{ij} = \begin{cases} 1-\epsilon & \text{if } i,j \in U \\ 1-\epsilon & \text{if } i,j \in W \\ \delta & \text{if } i \in U, j \in W \\ 1-\gamma & \text{if } i,j \in \bar{U} \\ \nu & \text{if } i \in \bar{U}, j \in U \cup W \\ 0 & \text{otherwise.} \end{cases}$$

This arrangement satisfies the degree constraints, the triangle inequalities, the cycle inequalities, and (14), but violates (18) by $(\delta - \epsilon)(S-1)^2(S-2)$.

We now consider an inequality that is equivalent to (14) if $S = 3$ and generalizes it for larger values of S. This inequality was suggested by PORTA [7] for the equipartition polytope with $S = 4$.

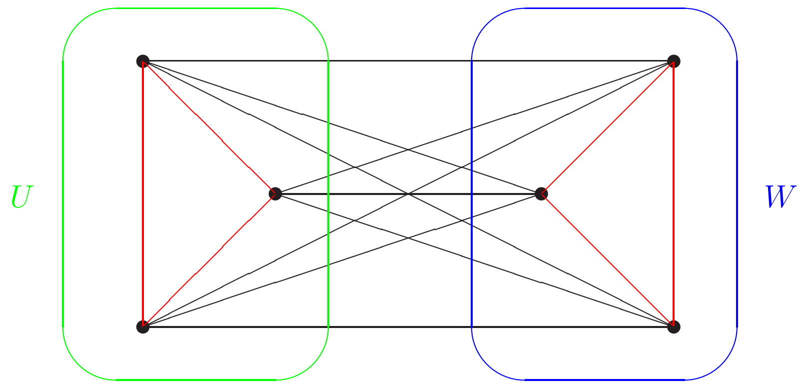

Figure 1: An illustration of Theorem 6

Theorem 7 *Let $U_1 \subseteq V$ and $U_2 \subseteq V$ be two disjoint sets with $|U_1| = S-3$ and $|U_2| = S+1$. Let v be a vertex from $V \setminus (U_1 \cup U_2)$. The following is a valid inequality:*

$$(S-2) \sum_{e \in E(v, U_1 \cup U_2)} x_e + \sum_{e \in E(U_2)} x_e \leq \frac{3S^2 - 9S + 8}{2}. \tag{19}$$

Proof: The following configurations satisfy the constraint at equality:

- Let any S vertices from U_2 be one division and let the remaining vertices be part of a second division.
- Let v together with any $S-1$ vertices from U_2 be one division, let the two remaining vertices in U_2 be part of a second division, and allocate the vertices in U_1 arbitrarily either into the second division or into any other division.

All other valid configurations satisfy the constraint strictly. □

Note that (19) is implied by the triangle inequalities and the degree constraints when $k = 2$. In particular, take (15) for the set $U = U_2 \cup v$, so $p = 2$. Let u be the unique element of $V \setminus (U_1 \cup U_2 \cup v)$. We have:

$$\begin{aligned} 2x(U_2) + 2x(v, U_2) &= (S-1)(S+2) - x(v, U_1) - x(U_1, U_2) - x(v, u) - x(u, U_2) \\ &= (S-1)(S+2) - ((S-3)(S-1) - x(u, U_1) - 2x(U_1)) \\ &\quad -(S-1-x(u, U_1)) \end{aligned}$$

$$
\begin{aligned}
&\qquad\text{from degree constraints for } u \text{ and } U_1 \\
&= \quad 4(S-1) + 2x(u, U_1) + 2x(U_1).
\end{aligned} \tag{20}
$$

Adding $(S-3)$ times the degree constraint for vertex v to one half of (20) gives:

$$
\begin{aligned}
&x(U_2) + (S-2)x(v, U_2) + (S-3)x(v, U_1) \\
&= \quad 2(S-1) + (S-1)(S-3) + x(u, U_1) + x(U_1) - (S-3)x(v, u) \\
&\leq \quad (S-1)^2 + \frac{(S-3)(S-4)}{2} + x(u, U_1) - (S-3)x(v, u) \\
&\qquad\qquad \text{since } |U_1| = S-3, \text{ giving an upper bound on } x(U_1) \\
&\leq \quad (S-1)^2 + \frac{(S-3)(S-4)}{2} + (S-3) - x(v, U_1) \\
&\qquad\qquad \text{from a triangle inequality on vertices } u, v, w \text{ for each } w \in U_1 \\
&= \quad \frac{3S^2 - 9S + 8}{2} - x(v, U_1),
\end{aligned}
$$

giving (19). As with constraint (14), this inequality is still useful in a cutting plane approach to the k-way equipartition problem when $k > 2$, even though it is redundant for $k = 2$.

4.3.2 Inequalities using more than $2S$ vertices

There are valid inequalities for the k-way equipartition problem that are not implied by the inequalities presented earlier and which use edges incident to more that $2S$ vertices. We present some families of such inequalities of this form in this section.

The first valid inequality is a generalization of the cycle inequality (1).

Theorem 8 *Let C be a cycle with $pS + 1$ vertices for some integer $p \geq 1$. The following is a valid inequality:*

$$x(E(C)) \leq pS - p. \tag{21}$$

Proof: The vertices of the cycle must be assigned to at least $p+1$ divisions. Thus, at least $p+1$ of the $pS+1$ edges in the cycle have endpoints in different divisions. □

The next theorem generalizes the result of Theorem 4.

Theorem 9 *Let $U \subseteq V$ with $|U| = pS + q$, with $2 \leq p < k$ and $1 \leq q < S$. The following is a valid inequality:*

$$
\sum_{e \in E(U)} x_e \leq \begin{cases} p\binom{S}{2} + \binom{q}{2} & \text{if } q \geq 2 \\ p\binom{S}{2} & \text{if } q = 1. \end{cases} \tag{22}
$$

Proof: The configuration of the vertices in U that uses most edges is to form p divisions of S vertices and place the remaining q vertices in the same division. The number of edges used by this configuration is equal to the right hand side given in the theorem. □

The following theorem is defined on a subset of the edges of a set of vertices of size $2S + 2$. The edges are chosen in such a way that at most $S(S - 1)$ of them can be used in any valid solution.

Theorem 10 *Let $U \subseteq V$ with $|U| = 2S + 2$. Let $\bar{E} \subseteq E(U)$ and let $\bar{G} := (U, \bar{E})$ with vertices U and edges $\bar{E}$. If $\bar{E}$ is such that for any two vertex disjoint copies of K_S in $\bar{G}$ the remaining two vertices are not adjacent in $\bar{G}$, then the following is a valid inequality:*

$$\sum_{e \in \bar{E}} x_e \leq S(S - 1). \tag{23}$$

Proof: The only way that more edges can be used from $E(U)$ is to construct two divisions from U and then place the remaining two vertices in the same division. From the definition of $\bar{E}$, not all the edges used by this construction appear in $\bar{E}$. □

There are many different ways to form $\bar{E}$ to meet the conditions of this theorem. For example, the edges can be chosen so that there are not two disjoint copies of K_S within $\bar{E}$. Another possibility is to add two vertices to K_{2S}, with each of the additional two vertices having degree $S - 2$ in $\bar{G}$. We discuss further possibilities in §7.

The next theorem is also defined on a subset of the edges of a set of $2S+2$ vertices. The edge weights vary depending on the particular edge. They are chosen to allow multiple configurations to satisfy the constraint at equality.

Theorem 11 *Let $U_1 \subseteq V$ and $U_2 \subseteq V$ be two disjoint sets with $|U_1| = S$ and $|U_2| = S + 1$. Let v be a vertex from $V \setminus (U_1 \cup U_2)$. The following is a valid inequality:*

$$S \sum_{e \in E(U_1)} x_e + (S - 1) \sum_{e \in E(U_2)} x_e + (S - 1) \sum_{e \in E(U_1, U_2)} x_e + (S - 1) \sum_{e \in E(v, U_1)} x_e \leq \frac{S(S - 1)(2S - 1)}{2}. \tag{24}$$

Proof: The following configurations satisfy the constraint at equality:

- Let U_1 be one division and take any S vertices in U_2 to be a second division.

- Let v together with any $S-1$ vertices in U_1 be one division. Create one division of S vertices from the remaining $S+2$ vertices in $U_1 \cup U_2$, and place the last two vertices in the same division as each other.

All other valid configurations satisfy the constraint strictly. $\square$

5 Semidefinite Programming

We can use semidefinite programming to impose an additional constraint on the variables. Define the $n \times n$ symmetric matrix X as

$$X_{ij} := \begin{cases} 1 & \text{if } i = j \\ x_{ij} & \text{otherwise.} \end{cases}$$

It was shown by Donath and Hoffman [11] that the matrix X is positive semidefinite. In particular, define the $n \times k$ matrix Y as

$$Y_{ij} := \begin{cases} 1 & \text{if vertex } i \text{ is in division } j \\ 0 & \text{otherwise.} \end{cases}$$

We then have $X = YY^T$. Karisch and Rendl [19] have investigated a semidefinite cutting plane algorithm using this formulation. They used triangle inequalities and constraints of the form (7) as cutting planes.

Lisser and Rendl [22] have applied a similar semidefinite programming approach to network design problems. They initialize with the constraint that X be positive semidefinite, and they add nonnegativity requirements on the elements of X as cutting planes. They also investigate a polyhedral approach similar to ours, initializing as we do and adding triangle inequalities as cutting planes. Their results indicate that the semidefinite programming approach seems preferable for problems with large values of S and small values of k, while the polyhedral approach is better for problems with small values of S and large values of k.

6 Branch-and-cut

Our branch-and-cut algorithm for the k-way equipartition problem is as follows. We call one pass through Steps 2–6 an *outer iteration*.

1. Initialize: The feasible region for the initial linear programming relaxation is $\bar{Q}(kS)$. The initial incumbent integer feasible solution is a random assignment of vertices to divisions.

2. Approximately solve the current LP relaxation using an interior point algorithm.

3. If the gap between the value of the LP relaxation and the value of the incumbent integer solution is sufficiently small, STOP with optimality.

4. If the duality gap for the current LP relaxation is smaller than 10^{-8} or if 41 outer iterations have been performed, call the branch-and-cut solver in CPLEX [10] to attempt to verify that the incumbent integer solution is optimal and then STOP.

5. Use a variant of the Kernighan-Lin heuristic [21] to round the fractional solution to the LP relaxation into a good integer feasible solution. Replace the current incumbent solution with this solution if it is an improvement.

6. Use the separation routine (defined below) to find violated cutting planes and return to Step 2.

An interior point method was used in Step 2 to approximately solve the LP relaxations, with the required tolerance on the relative duality gap gradually tightened at each outer iteration. The initial tolerance was 0.3. It was multiplied by a factor between 0.2 and 1 depending on the outcome of the separation routines — the tolerance is decreased more quickly if the maximum triangle inequality violation is small. For more details on interior point cutting plane algorithms of this type, see [23].

In all the experiments considered later, the data was integer. Thus, the algorithm terminates in Step 3 if the gap is smaller than one.

In Step 4, we provide the set of constraints from the final LP relaxation, together with the integrality restrictions, to the branch-and-cut solver. Since this set of constraints does not contain all of the triangle inequalities, the solution to this integer program may not be feasible in the k-way equipartition problem. If the integer solution returned by the branch-and-cut solver has the same value as the solution found in Step 5 then this confirms that the latter is the optimal integral solution.

The separation routine consists of the following parts:

6–i The algorithm first searches for **triangle inequalities** (3), (4), and (5), using complete enumeration. Inequalities are bucket sorted by the size of the violation. Starting with inequalities in the most violated bucket, a subset are added, ensuring that no two of these added inequalities share an edge. The violation of the last constraint added is restricted to be no smaller than a multiple of the violation of the first constraint added.

6–ii If no more than twenty triangle inequalities have been added or if the maximum violation of a triangle inequality is smaller than 0.3, a routine similar to that described in Grötschel and Wakabayashi [17, 18] is used to find violated **2-partition inequalities** (6), with $|U| = 1$ or 2 and $|W| \geq 3$.

6–iii If the maximum violation of a triangle inequality is smaller than 0.3, the algorithm searches for violated **cycle inequalities** (1) by building up cliques of vertices. This uses a breadth first search approach, based on adding vertices v to the test set U if x_{uv} is sufficiently large for one or more vertices in U. Initially, the algorithm only considers edges e with $x_e \geq 0.75$; it then considers edges with $x_e \geq 0.6$, then $x_e \geq 0.45$, and finally $x_e \geq 0.3$. Cliques of size $S+1$ with weight $x(U) > \frac{S(S-1)}{2}$ must contain violated cycle inequalities, so if we find a violated clique inequality of this form, we add the corresponding cycle inequalities.

6–iv **Clique inequalities** of the form (14) with $p = 1$ and $q = 2$ and of the form (18) are checked using the same breadth first search routine as in Step 6–iii.

6–v If no more than five cliques leading to violated inequalities have been found, the algorithm searches for **cycle inequalities** (1) directly, using a depth first search approach.

If one step does not produce enough violated cutting planes or if the cutting planes are not sufficiently violated, the separation routine moves to the next step. Otherwise, it returns to the main algorithm. In all the computational experiments described in §7, we had $S = 4$; different values of S may require different values for some of the parameters discussed in this section.

The cutting plane algorithm was written in Fortran 77 and implemented on a Sun Enterprise workstation. CPLEX was run on a Sun 20/71 workstation. It took an mps file as input, containing a description of the final LP relaxation as well as integrality restrictions.

7 Computational results

Test problems were drawn from three different sources. In all of these problems, the cluster size was set equal to four.

7.1 Realignment in the NFL

The first class of problems we examined were based on realignment in the National Football League. For this problem, $S = 4$ and $k = 8$. The minimum value for the total intradivisional travel distance is 27957 kilometers. The heuristic in the cutting plane algorithm found this optimal solution. The cutting plane algorithm found a lower bound of 27938 kilometers and CPLEX confirmed that the algorithm had found the optimal solution. The cutting plane algorithm used 57 seconds and CPLEX required a further 57 seconds. For the realignment problems, seven specific inequalities of the forms (19), (22), (23), and (24), were found by visual inspection of the fractional solution. For these problems, these constraints were added after Step 6–v on the first outer iteration when the cutting plane routines would otherwise have added no constraints.

The breakdown of these seven inequalities is as follows:

- One constraint of the form (19).
- Two constraints of the form (22) with $p = 2$ and $q = 2$.
- Three constraints of the form (23). These were defined in two different ways:
  - For one of the constraints, start with K_8. Delete one edge. Choose two additional vertices. Connect them to each end of the missing edge from K_8. Connect each of the two additional vertices to one of the original eight vertices.
  - For the other two constraints, start with one copy of K_4 and two copies of K_3. Add six edges between the two copies of K_3, ensuring that the maximum clique within these six vertices still has cardinality three. Connect one of the vertices in one of the copies of K_3 and two of the vertices in the other copy of K_3 to every vertex in the copy of K_4.
- One constraint of the form (24).

For this problem, we also added a semidefiniteness constraint to the final relaxation and resolved using a semidefinite programming algorithm. However, this only improved the lower bound by one, so we did not try using the semidefiniteness constraint for any of the other problems.

We also solved some variants of this problem; for details see [24]. Some of these variants benefited from the addition of one additional constraint of the form (14) with $p = 3$ and one additional constraint of the form (22) with $p = 2$ and $q = 1$. The cutting plane algorithm was able to solve some of these variants without branching.

7.2 Random geometric problems

We generated vertices in the unit square. The distance between two vertices was then defined as the integral part of 1000 times the Euclidean distance between the vertices. We generated problems with between 40 and 160 vertices. Ten problems of each size were generated.

Table 1 shows that the cutting plane algorithm was typically able to solve these problems to within about 2% of optimality. The rows in the table give the number of problems solved exactly with the cutting plane algorithm, the average final gap for the cutting plane code, the average runtime in seconds for the cutting plane code, the average number of cuts added, the average number of outer iterations, and the average number of interior point iterations.

The last two rows of the table show the performance of CPLEX on one instance of each size. In each case, the final relaxation formed by the cutting plane algorithm was submitted to the branch and bound routine in CPLEX; the runtime and size of the tree generated are reported. Note that feeding all the triangle inequalities to CPLEX is impractical; CPLEX was unable to solve a 40 vertex instance in 16 hours when given only the triangle inequalities.

7.3 Random network design problems

Lisser and Rendl [22] have described an application of the k-way equipartition problem in network design problems. Given estimates for the communication between each pair of vertices, it is desired to cluster the vertices into equal size clusters so as to maximize the sum of communication within clusters. The vertices in a cluster will then be connected using a Sonet or SDH ring. They discuss results for some proprietary France Telecom problems, and they also give computational results for some randomly generated problems. The problems have between 100 and 500 vertices. We experimented with dividing them into clusters of size 4, and the results are contained in Table 2.

Vertices	40	60	80	100	120	140	160
Solved exactly	4	0	0	0	0	0	0
Gap	2.3%	1.9%	1.7%	1.7%	2.3%	2.4%	2.0%
Time	20.1	54.4	127.4	221.9	504.2	708.6	975.9
Cuts added	265.2	457.9	568.7	715.2	886.7	1042.5	1202.4
Outer iterations	17.7	24.8	30.5	34.1	35.4	40.4	39.7
Inner iterations	188.1	320.2	461.1	571.3	678.7	759.8	792.2
Typical CPLEX run							
time	333.4	646.1	2772.6	4009.1	> 10000		
nodes	1300	228	1007	608	> 1500		

Table 1: Results on random geometric problems

Size	100	200	300	400	500
LP bound	184.850	763.423	1756.202	3125.942	4915.604
IP bound	185.752	765.719	1760.241	3131.535	4922.800
gap	0.49%	0.30%	0.23%	0.18%	0.15%
intra gap	7.1%	8.8%	10.2%	10.6%	10.8%
Rows	2850	4632	4468	5260	5333
Outer iterations	33	36	32	30	28
Inner iterations	257	411	384	405	382

Table 2: Results on random network design problems

In order to agree with [22], we have reported the bounds in terms of the sum of the edge lengths of the *inter*-divisional distances. The percentage differences are considerably smaller when given in this manner. We give the percentage gap in the sum of the total intradivisional distance for comparison. The code was allowed to run for 50000 seconds on each of these problems. We report the number of interior point iterations, the number of outer iterations, and the number of constraints in the final relaxation found in this time.

It should be noted that the algorithm gets close to the optimal solution quickly. For example on the 300 vertex problem, it has found the integer solution with value 1760.241 and has a lower bound of 1754.176, that is a gap of 0.35%, in 135 seconds. Results for the other problems are similar.

Because our cutting plane routines are more extensive, these bounds are slightly better than those obtained by Lisser and Rendl [22]. As they note, the polyhedral approach is superior to a purely semidefinite approach for problems with many divisions, as in these instances.

8 Conclusions

The use of a polyhedral approach allows the construction of better bounds on optimal values for instances of the k-way equipartition problem drawn from different settings. This can be used in conjunction with a branch-and-bound or branch-and-cut approach to find optimal solutions, at least for problems of a reasonable size.

The network design problems of §7.3 appear harder than the Euclidean problems discussed in §7.1 and §7.2. They require considerably more time and the final duality gaps (in terms of the intradivisional distances) are far larger. One of the reasons for the increased computational time is that the matrix AD^2A^T formed by the interior point method is denser, because the columns of A are denser, with more constraints needed to force certain difficult components of x to take the correct values. (Here, A denotes the constraint matrix and D is a diagonal matrix.) We observed a similar phenomenon for matching problems in [26], where instances with edge lengths corresponding to Euclidean distances required less computational work than instances with a random distribution of edge lengths. It would be of interest to try a simplex cutting plane approach for these problems, either on its own, or in combination with an interior point cutting plane method, as was done in [25] for the linear ordering problem.

References

[1] S. Areibi and A. Vannelli. Advanced search techniques for circuit partitioning. In P. Pardalos and H. Wolkowicz, editors, *Quadratic Assignment and Related Problems*, volume 16 of *DIMACS: Series in Discrete Mathematics and Theoretical Computer Science*, pages 77–96. AMS, 1994.

[2] F. Barahona and A. R. Mahjoub. On the cut polytope. *Mathematical Programming*, 36:157–173, 1986.

[3] S. J. Benson, Y. Ye, and X. Zhang. Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM Journal on Optimization*, 10(2):443–461, 2000.

[4] L. Brunetta, M. Conforti, and G. Rinaldi. A branch-and-cut algorithm for the equicut problem. *Mathematical Programming*, 78:243–263, 1997.

[5] S. Chopra. The graph partitioning polytope on series-parallel and 4-wheel free graphs. *SIAM Journal of Discrete Mathematics*, 7:16–31, 1994.

[6] S. Chopra and M. R. Rao. The partition problem. *Mathematical Programming*, 59:87–115, 1993.

[7] T. Christof and G. Reinelt. Combinatorial optimization and small polytopes. *Top (Spanish Statistical and Operations Research Society)*, 4:1–64, 1996.

[8] M. Conforti, M. R. Rao, and A. Sassano. The equipartition polytope I: Formulations, dimension and basic facets. *Mathematical Programming*, 49:49–70, 1990.

[9] M. Conforti, M. R. Rao, and A. Sassano. The equipartition polytope II: Valid inequalities and facets. *Mathematical Programming*, 49:71–90, 1990.

[10] CPLEX Optimization Inc. CPLEX Linear Optimizer and Mixed Integer Optimizer. Suite 279, 930 Tahoe Blvd. Bldg 802, Incline Village, NV 89541.

[11] W. E. Donath and A. J. Hoffman. Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, 17:420–425, 1973.

[12] J. Falkner, F. Rendl, and H. Wolkowicz. A computational study of graph partitioning. *Mathematical Programming*, 66:211–239, 1994.

[13] C. E. Ferreira, A. Martin, C. C. de Souza, R. Weismantel, and L. A. Wolsey. The node capacitated graph partitioning problem: formulations and valid inequalities. *Mathematical Programming*, 74:247–267, 1996.

[14] C. E. Ferreira, A. Martin, C. C. de Souza, R. Weismantel, and L. A. Wolsey. The node capacitated graph partitioning problem: a computational study. *Mathematical Programming*, 81:229–256, 1998.

[15] A. Frieze and M. Jerrum. Improved approximation algorithms for max k-cut and max bisection. In *Proceedings of the Fourth IPCO Conference*, pages 1–13, 1995.

[16] M. R. Garey and D. S. Johnson. *Computers and intractibility: a guide to the theory of NP-completeness*. Freeman, 1979.

[17] M. Grötschel and Y. Wakabayashi. A cutting plane algorithm for a clustering problem. *Mathematical Programming*, 45:59–96, 1989.

[18] M. Grötschel and Y. Wakabayashi. Facets of the clique partitioning polytope. *Mathematical Programming*, 47:367–387, 1990.

[19] S. E. Karisch and F. Rendl. Semidefinite programming and graph equipartition. In P.M. Pardalos and H. Wolkowicz, editors, *Topics in Semidefinite and Interior-Point Methods*, volume 18 of *Fields Institute Communications*, pages 77–95. AMS, 1998.

[20] G. Karypis and V. Kumar. Multilevel k-way hypergraph partitioning. Technical Report 98–036, Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455, 1998.

[21] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, 49:291–307, 1970.

[22] A. Lisser and F. Rendl. Telecommunication clustering using linear and semidefinite programming. Technical report, Institüt fı̈ Mathematik, Universität Klagenfurt, A – 9020 Klagenfurt, Austria, November 2000.

[23] J. E. Mitchell. Computational experience with an interior point cutting plane algorithm. *SIAM Journal on Optimization*, 10(4):1212–1227, 2000.

[24] J. E. Mitchell. Realignment in the NFL. Technical report, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, November 2000.

[25] J. E. Mitchell and B. Borchers. Solving linear ordering problems with a combined interior point/simplex cutting plane algorithm. In H. L. Frenk *et al.*, editor, *High Performance Optimization*, chapter 14, pages 349–366. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000.

[26] J. E. Mitchell and M. J. Todd. Solving combinatorial optimization problems using Karmarkar's algorithm. *Mathematical Programming*, 56:245–284, 1992.

[27] C. C. de Souza, R. Keunings, L. A. Wolsey, and O. Zone. A new approach to minimising the frontwidth in finite element calculations. *Computer Methods in Applied Mechanics and Engineering*, 111:323–334, 1994.

[28] H. Wolkowicz and Q. Zhao. Semidefinite programming relaxations for the graph partitioning problem. *Discrete Applied Mathematics*, 96/97:461–479, 1999.

[29] Y. Ye. A .699-approximation algorithm for max-bisection. Technical report, Department of Management Sciences, The University of Iowa, Iowa City, Iowa 52242, March 1999.