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

John E. Mitchell
Department of Mathematical Sciences
Rensselaer Polytechnic Institute
Troy, NY 12180 USA
mitchj@rpi.edu
http://www.rpi.edu/~mitchj

ISMP2000,
Atlanta, GA,
August 2000

Abstract

The k-way equipartition problem is to divide the vertices of a graph into k equal sized sets. This problem has applications in diverse areas, including VLSI chip design and sports scheduling. We discuss the polyhedral structure of this problem and present some computational results of a branch-and-cut algorithm.

Contents

1 Introduction

Formulation:

- Given a **complete graph** $G = (V, E)$ with $|V| = kS$ vertices and with weights c_e on each edge e.
- **Partition the vertices into k sets each of S vertices** so as to minimize the total weight of edges having both endpoints in the same set.

Applications:

- **VLSI chip design**. Here, we want to minimize the number of vias.
- **Assignment of sports teams to divisions** to minimize travel distances — each team plays mainly against teams in its own division.
- **Minimizing frontwidth** in finite element calculations.
- Variants of **frequency assignment problems** in telecommunications.
- ...

2 Related problems

- **Clustering problems:**
  Divide the vertices into a (possibly unspecified) number of sets, which may be of **unequal sizes**.
- **Equipartition problem:**
  Divide the vertices into **two sets of equal size**.
- **Matching problem:**
  Divide the vertices into $n/2$ sets each with **two vertices**.

3 Integer programming formulation

Define a binary variable x_{ij} with

$$x_{ij} = \begin{cases} \mathbf{1} & \textbf{if } i \textbf{ and } j \textbf{ are in the same set} \\ 0 & \text{otherwise} \end{cases}$$

Our formulation is:

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

where $\delta(v)$ denotes the set of edges incident to vertex v.

We regard x_{ij} and x_{ji} as the same variable.

Let n denote the number of edges.

The k**-way equipartition problem polytope** is:

$$Q(kS) := \text{conv}\{x \in \{0,1\}^n : \sum_{e\in\delta(v)} x_e = S-1 \;\; \forall\, v \in K_{kS}, \; x \text{ is the incidence vector of a clustering}\}$$

where *conv* indicates *convex hull*.

The **initial relaxation** of the k-way equipartition problem uses the polytope

$$\bar{Q}(kS) := \{x \in [0,1]^n : \sum_{e\in\delta(v)} x_e = S-1 \;\; \forall\, v \in K_{kS}\},$$

Here, K_q denotes the complete graph on q vertices.

The problem is NP**-hard** provided $S \geq 3$ — see Garey and Johnson (1979).

4 The equipartition polytope

Brunetta, Conforti, and Rinaldi (1997) have developed a branch-and-cut algorithm for the equicut problem. Their work is based on that of Conforti, Rao, and Sassano (1990), who developed a great deal of polyhedral theory for the equipartition problem.

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

Facet defining inequalities:

- *Clique inequalities* (Conforti *et al.*, Theorem 6.1): For every complete subgraph with $q \geq 3$ vertices, q odd, we have

$$x(E(K_q)) \geq \lfloor \frac{1}{2} q \rfloor^2.$$

- *Cycle inequalities* (Conforti *et al.*, Theorem 6.2): For every cycle C of length $S+1$, we have

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

The **cycle inequalities** can be used for the k-way equipartition problem.

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 set, and $x_e = 0$ for all $e \in E'$ for these feasible solutions. □

We can convert them into valid 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

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

5 The clustering polytope

- **Clustering problem**: given a set of p observations, each of which possesses l characteristics; divide the observations into clusters where the observations within each cluster are similar to one another.
- **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 (1989,1990) described a **cutting plane algorithm** for the clustering problem.

5.1 Triangle inequalities

The set of **feasible solutions** are given by the following:

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

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

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

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

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 by Brunetta *et al.* and they are well known in the literature for the MAX-CUT problem; see, for example, Barahona and Mahjoub (1986).

5.2 Other classes of facets

Theorem 1 *(Grötschel and Wakabayashi (1989,1990))* *For every nonempty disjoint subsets $U, W \subseteq V$, the* **2-partition inequality**

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

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

A complete description of the convex hull is not currently known.

Chopra and Rao (1993) 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 **another clique inequality**

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

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

6 The k-way equipartition problem polytope

We have used the inequalities for the equipartition polytope and the clustering polytope as cutting planes, as appropriate. Further, we have developed some valid inequalities specifically for the k-way equipartition problem.

6.1 Dimension

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

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

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

6.2 Lifting inequalities

Certain facet defining inequalities for the equipartition problem can be **converted** into facet defining inequalities for the k-way equipartition problem.

Divide the vertices of K_{kS} into $\cup_{i=1}^{k} C^i$, with $|C^i| = S$ for $i = 1, \ldots, k$.

Assumptions about $a^T x \leq b$:

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 $p < k$.

2. The inequality is **valid** for the k-way equipartition problem on K_{kS}, if $a_e = 0$ for any edge e that does not have both endpoints in $\cup_{i=1}^{p} C^i$.

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 has $p = 2$.

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

Cycle inequalities are facet defining

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 sets, so there are at least two edges on the cycle whose endpoints are in different sets.
3. For any vertex on the cycle, the k-way equipartition where all the other vertices on the cycle are placed in a single set satisfies the third assumption. For any vertex not on the cycle, the third assumption is satisfied by any k-way equipartition.

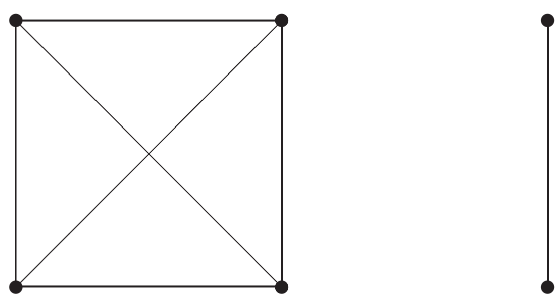
The result then follows as an immediate consequence of Theorem 3. □

6.3 Valid inequalities

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

$$\sum_{e \in E(U)} x_e \leq \binom{S}{2} + \binom{p}{2}. \tag{8}$$

Example: $S = 4$, $|U| = 6$, use at most seven edges:



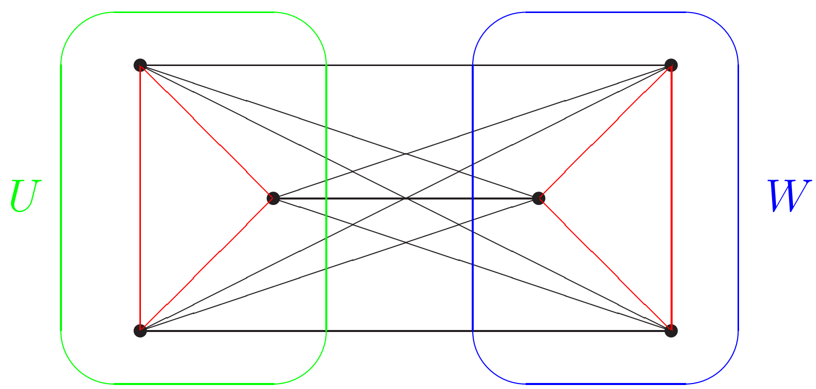
For the equipartition polytope, this constraint is **implied** by the degree constraints.

Useful for $k > 2$. For example, consider $k = 3$, $S = 4$. Breaking the vertices into two sets of six vertices, with edge weights $x_e = 0.6$, satisfies the degree constraints, triangle constraints, and cycle constraints, but violates (8).

Theorem 5 *Let U and W be two disjoint subsets of V with $|U| = |W| = S - 1$. The following is a valid inequality:*

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

Example: $S = 4$, $|U| = |W| = 3$, right hand side is 18, red edges have coefficient 3, black edges have coefficient 2:



Configurations with equality in constraint:

- U and W in separate sets.
- U plus one element from W in one set.
- W plus one element from U in one set.

7 Separation routines

Use the following routines in order. Each subsequent one is only invoked if the previous ones did not find enough violated constraints.

- **Triangle inequalities**: Enumerate the inequalities, bucket sort by violation, and add a subset of the most violated constraints.
- **2-partition inequalities**: Use routines similar to those in Grötschel and Wakabayashi (1989,1990).
- **Cycle inequalities**: Look for cliques of size $S+1$, by looking at edges with large x_e value: slowly build up cliques, adding vertices that are joined to the clique by a large edge. If the clique has too much weight, check all the cycle inequalities using edges just from the clique.
- **Cliques of size $S+2$**: Look for these at the same time as looking for cycle inequalities.
- **Cycle inequalities (again)**: Look directly for cycle inequalities, by searching for paths that eventually join, in a depth first search manner.

8 The algorithm

1. **Initialize:** Read in data, set up initial relaxation with simple bounds and degree constraints. Set up tolerances.

2. Approximately **solve the current relaxation** using a primal-dual interior point method. The tolerance for this step is gradually tightened as the algorithm proceeds.

3. Use a **primal heuristic** to generate a feasible solution. Update the upper bound if successful. The primal heuristic is a greedy search followed by 2-change moves.

4. **Check for violated constraints**. If none found and the duality gap small enough, **STOP**. Otherwise, return to **Step 2**, after adding constraints.

9 Computational Results

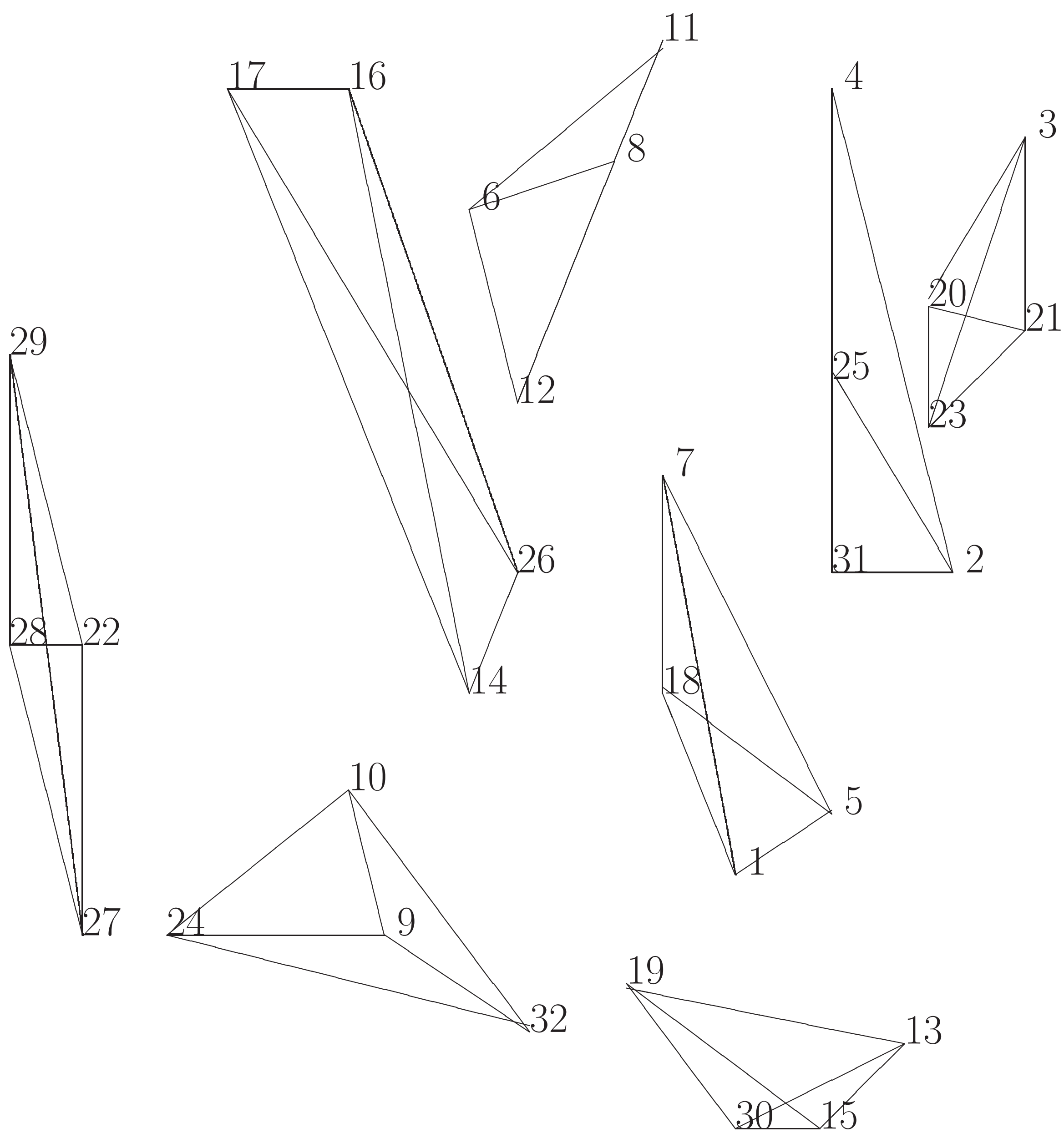
- Experiments on a **Sun Enterprise workstation**.
- Algorithm coded in **FORTRAN**.
- Runtimes reported in **seconds**.
- In all experiments, we took $S = 4$.
- For some problems, we finished the process off with **CPLEX6.1**, in order to obtain an optimal solution. CPLEX was run on a Sun SPARC 20/71 Workstation, it read in the cutting plane formulation from an MPS file, and runtimes are again reported in seconds.

9.1 Realignment in the NFL

- The **National Football League (NFL)** in the USA will shortly expand to 32 teams.
- At that point, it is likely that the teams will be **re-aligned** into eight divisions of four teams each.
- One possible objective is to **minimize the sum of intradivisional distances**, since each team will play a large proportion of its games against teams in its own division.
- This is a k-way equipartition problem, with $k = 8$ **and** $S = 4$.
- Cutting plane code used **57 seconds to get within 0.07%** of optimality. Added 323 constraints in 36 stages.
- For this problem, **four specific constraints** using edges from at least ten vertices were also added. Without these constraints, the algorithm got within 0.3% of optimality.
- CPLEX6.1 required 65 seconds and 2 nodes of the branch and bound tree to confirm optimality.

Optimal solution: (stylized map)

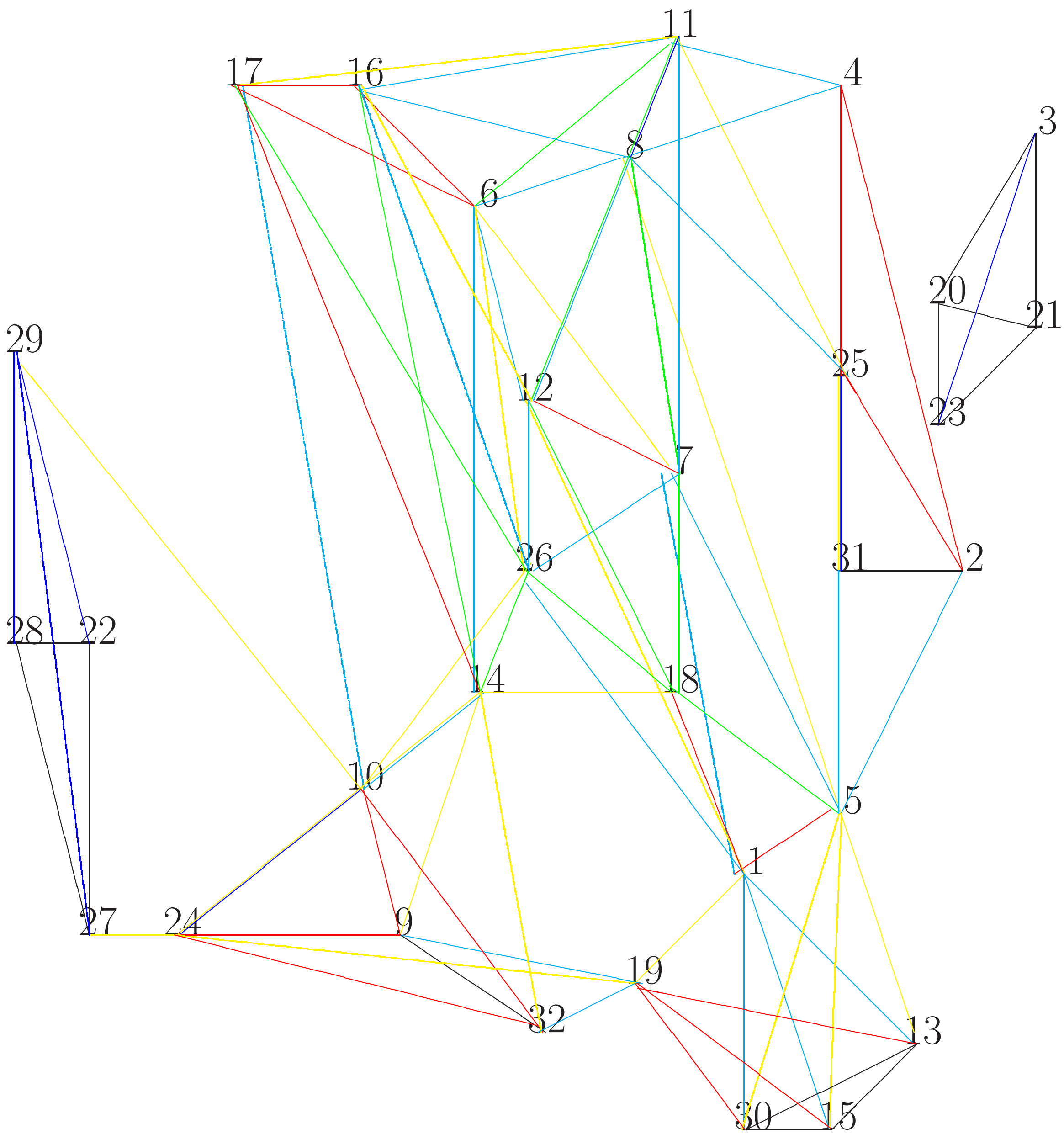
Optimal value: 27957 km
(confirmed by CPLEX)



Cutting plane solution:

Lower bound: 27938 km
Final gap: 0.07%

Key:
0.95–1.0
0.80–0.95
0.60–0.80
0.40–0.60
0.20–0.40
0.08–0.20



9.2 Random problems

- Generated **ten problems each** with 40, 60, 80, 100, 120, 140, 160 vertices.
- Vertices are randomly located in the **unit square**.
- Edge costs c_e are the integer part of the **Euclidean edge length** multiplied by 1000.
- **Average** values are reported in the table.

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
Stages	17.7	24.8	30.5	34.1	35.4	40.4	39.7
Iterations	188.1	320.2	461.1	571.3	678.7	759.8	792.2
Typical CPLEX run							
with cuts: time	333.4	646.1	2772.6	4009.1	> 10000		
with cuts: nodes	1300	228	1007	608	> 1500		
with triangles: time	> 57880						
with triangles: nodes	> 6						

10 Semidefinite Programming

Define the $n \times n$ symmetric matrix X as

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

Define the $n \times k$ matrix Y as

$$Y_{ip} := \begin{cases} 1 & \text{if vertex } i \text{ is in set } p \\ 0 & \text{otherwise.} \end{cases}$$

We then have $X = YY^T$, so **X is positive semidefinite**. (Construction due to Donath and Hoffman (1973).)

In computational experiments with this constraint, it only made a **marginal improvement in the lower bound** obtained by the cutting plane code, with a **considerable investment of computer time**.
For example, on the NFL realignment problem, the SDP constraint improved the lower bound from 27938km to 27939km (optimal value is 27957km). It required 438 seconds with package SDPT3-2.1.

Karisch and Rendl (1998) have investigated a semidefinite cutting plane algorithm using this formulation. They used triangle inequalities and constraints of the form (7) as cutting planes.

11 Conclusions

- We've found the **dimension** of the k-way equipartition problem, and given a **sufficient condition for lifting** facet defining inequalities from the equipartion problem to the k-way equipartition problem.
- **Other valid inequalities** have been shown.
- **Computational results** for problems with up to 160 vertices have been presented. The cutting plane code generally gets within 2% of proving optimality in a few minutes. For the smaller problems, CPLEX can be used to further reduce the gap to zero.
- The **semidefinite** constraint does not help much with the final formulation found by the cutting plane algorithm. It is more useful for formulations before many cutting planes have been added.

References

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

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

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

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

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

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

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

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

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

[10] 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.