Math Meth Oper Res (2002) ■:■–■

Mathematical Methods
of Operations Research

Multiset graph partitioning*

William W. Hager and Yaroslav Krylyuk

Department of Mathematics, University of Florida, Gainesville, FL 32611
(e-mail: hager@math.ufl.edu, http://www.math.ufl.edu/~hager)

Manuscript received: March 2001/Final version received: October 2001

Abstract. Local optimality conditions are given for a quadratic programming formulation of the multiset graph partitioning problem. These conditions are related to the structure of the graph and properties of the weights.

Key words: Graph partitioning, min-cut, max-cut, quadratic programming, optimality conditions

AMS(MOS) subject classifications: 90C35, 90C27, 90C20

1. Introduction

In multiset min-cut graph partitioning, the vertices of a graph are partitioned into sets of given sizes while minimizing the weighted sum of edges connecting vertices in different sets. Although we focus on min-cut in this paper, the max-cut problem can be treated in a similar fashion. Let $\mathbf{A}_0$ be an n by n symmetric weight matrix associated with an undirected graph with vertex set V:

$$V = \{1, 2, \dots, n\}.$$

By assumption the diagonal of $\mathbf{A}_0$ is zero. Let $\mathbf{m}$ be a vector of k positive integers that sum to n, where m_i is the number of vertices in the i-th set of the partition, $1 \leq i \leq k$. Let $\mathbf{A} = \mathbf{A}_0 + \mathbf{D}$ where $\mathbf{D}$ is any diagonal matrix whose elements satisfy the condition

$$d_{ii} + d_{jj} \geq 2a_{ij}$$

* This work was supported by the National Science Foundation.

for all $1 \le i, j \le n$. In [14] we show that graph partitioning is closely related to the following continuous quadratic programming problem:

$$\begin{aligned} &\text{maximize trace } \mathbf{X}^\top \mathbf{A}\mathbf{X} \\ &\text{subject to } \mathbf{X1} = \mathbf{1}, \quad \mathbf{X}^\top \mathbf{1} = \mathbf{m}, \quad \mathbf{X} \ge \mathbf{0}. \end{aligned} \tag{1}$$

Here $\mathbf{X}$ is an n by k matrix, $\mathbf{1}$ is a vector of the appropriate size whose elements are all one, and "$^\top$" denotes transpose.

Solutions of (1) and the optimal partition in the graph partitioning problem are related in the following way: If $\mathscr{S}_j$ is the set of vertices assigned to the j-th set in an optimal partition, then the matrix $\mathbf{X}$ defined by

$$x_{ij} = \begin{cases} 1 & \text{if } i \in S_j, \\ 0 & \text{if } i \notin S_j, \end{cases}$$

is a solution of (1). Conversely, there exists a solution of (1) whose matrix entries are all 0 or 1 (a 0/1 solution), and an optimal partition of the vertices of the graph is given by

$$\mathscr{S}_j = \{i : x_{ij} = 1\}.$$

Given any solution of (1) whose entries are not all 0/1, there is a simple procedure (given in Algorithm 3 below) for moving to a 0/1 solution.

In this paper, we give necessary and sufficient conditions characterizing local minimizers of (1), based on the general theory for quadratic programming [6, 9]. These conditions are related to the structure of the graph and the edge weights. When $\mathbf{X}$ is a 0/1 matrix, we reformulate the first-order optimality conditions in terms of a feasibility problem for a linear system of inequalities.

Graph partitioning problems arise in circuit board and micro-chip design and in other layout problems (see [21]) and in sparse matrix pivoting strategies [7, 13, 15]. In parallel computing, graph partitioning problems arise when tasks are partitioned among processors in order to minimize the communication between processors and to balance the processor loads. An application of graph partitioning to parallel molecular dynamics simulations is given in [26]. The maximum clique problem is another graph problem that has been given a quadratic programming formulation [12, 23]. Work related to other approaches to graph partitioning includes [1, 2, 3, 4, 8, 11, 16, 17, 18, 20, 19, 22, 24, 25, 27].

2. Local optimality conditions

Our local optimality conditions make use of the following terminology. First, the *support* of a matrix $\mathbf{Y}$ is the set of indices associated with nonzero elements:

$$\operatorname{supp} \mathbf{Y} = \{(i, j) : y_{ij} \neq 0\}.$$

Second, a *path matrix* is an n by k matrix whose entries are all contained in the set $\{0, +1, -1\}$ and whose nonzero entries are connected with two sequences

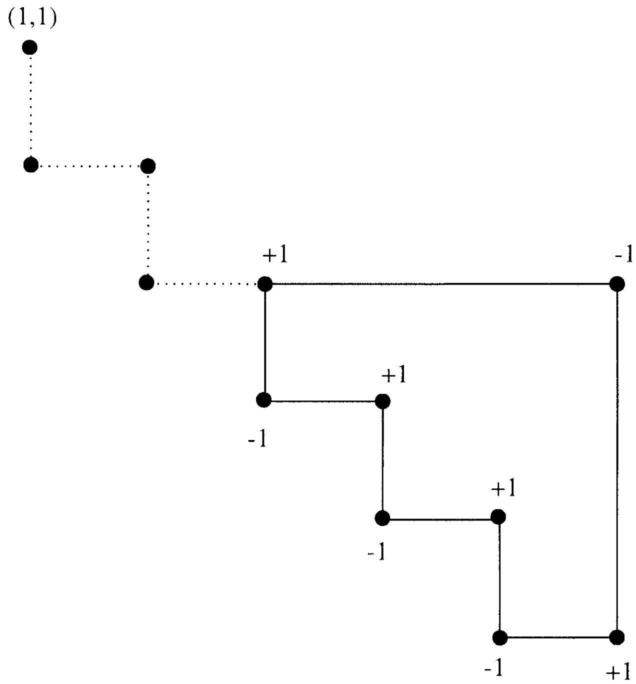

Fig. 1. A path matrix.

of integers in the following way. If $(r_1, r_2, \ldots, r_l)$ and $(c_1, c_2, \ldots, c_l)$ are sequences of integers with $1 \le r_i \le n$ and $1 \le c_i \le k$ for $1 \le i \le l$, where the elements in each sequence are distinct, the associated path matrix $\mathbf{Y}$ is entirely zero except for the (r_i, c_i) elements which are $+1$, and the (r_{i+1}, c_i) and (r_1, c_l) elements which are -1, for each i. The set of all matrices $\mathbf{Y}$ constructed in this way is denoted $\mathscr{P}$. An example of such a matrix is depicted in Figure 1. All the elements of the matrix are zero except for the $+1$ and -1 entries associated with the row/column pairs of corners on the solid path. We let "path($\mathbf{r}, \mathbf{c}$)" denote the path matrix in $\mathscr{P}$ associated with the vectors $\mathbf{r}$ and $\mathbf{c}$.

The first-order optimality system (Karusch-Kuhn-Tucker conditions) associated with a local minimizer of (1) is the following: There exist vectors $\boldsymbol{\lambda} \in \mathbf{R}^n$ and $\boldsymbol{\mu} \in \mathbf{R}^k$, and an n by k matrix $\boldsymbol{\omega}$ such that

$$2\mathbf{AX} + \boldsymbol{\lambda}\mathbf{1}^\top + \mathbf{1}\boldsymbol{\mu}^\top + \boldsymbol{\omega} = \mathbf{0}, \quad \boldsymbol{\omega} \ge \mathbf{0}, \quad \boldsymbol{\omega} \cdot \mathbf{X} = 0. \tag{2}$$

Throughout, "$\cdot$" denotes the usual dot product for matrices and vectors (sum of the products between respective components). Since the constraints of (1) are linear equalities and inequalities, the first-order optimality system holds at any local minimizer. The main result in our paper is the following:

Theorem 2.1. *Suppose that* $\mathbf{X}$ *is feasible in (1) and* $a_{ii} + a_{jj} \ge 2a_{ij}$ *for each i and j. A necessary and sufficient condition for* $\mathbf{X}$ *to be a local minimizer of (1) is that the following conditions hold:*

(P1) *There exist* $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$, *and* $\boldsymbol{\omega}$ *satisfying (2).*
(P2) Trace $\mathbf{Y}^\top \mathbf{A}\mathbf{Y} = 0$ *and* trace $\mathbf{Y}^\top \mathbf{A}\mathbf{Z} \le 0$ *for each* $\mathbf{Y}, \mathbf{Z} \in \mathscr{P}_{\mathbf{X}}(\Omega_0)$ *where*

$$\Omega_0 = \{(i,j) : \omega_{ij} = 0\}, \text{ and}$$

$$\mathscr{P}_{\mathbf{X}}(\Omega_0) = \{\mathbf{Y} \in \mathscr{P} : \operatorname{supp} \mathbf{Y} \subset \Omega_0,\, y_{ij} \ge 0 \text{ when } x_{ij} = 0 \text{ and } (i,j) \in \Omega_0\}.$$

In proving this result, we need to consider perturbations $\mathbf{Y}$ around a local minimizer $\mathbf{X}$ of (1) that satisfy the homogeneous constraints $\mathbf{Y1} = \mathbf{0} = \mathbf{Y}^\top \mathbf{1}$. We now show that any such matrix can be expressed as a positive linear combination of path matrices whose supports are contained in that of $\mathbf{Y}$. This result is based on the following construction of a single path matrix whose support is contained in that of $\mathbf{Y}$:

Algorithm 1 (Given $\mathbf{Y1} = \mathbf{0} = \mathbf{Y}^\top \mathbf{1}$, $\mathbf{Y} \ne \mathbf{0}$, find $\mathbf{P} \in \mathscr{P}$, supp $\mathbf{P} \subset$ supp $\mathbf{Y}$)

0. Find (i,j) such that $y_{ij} > 0$, set $l = 1$, $r_1 = i$, $c_1 = j$.
1. Holding j fixed, find a new i such that $y_{ij} < 0$.
2. If $i = r_p$ for some $1 \le p \le l$, goto step 7.
3. Increment l and set $r_l = i$.
4. Holding i fixed, find a new j such that $y_{ij} > 0$.
5. If $j = c_p$ for some $1 \le p < l$, set $r_p = i$, decrement l, and goto step 7.
6. Otherwise set $c_l = j$ and goto step 1.
7. Set $\mathbf{r} = (r_p, r_{p+1}, \ldots, r_l)$, $\mathbf{c} = (c_p, c_{p+1}, \ldots, c_l)$, and $\mathbf{P} = \text{path}(\mathbf{r}, \mathbf{c})$.

end Algorithm 1

Since $\mathbf{Y}$ is nonzero in Algorithm 1, and its row and column sums vanish, the positive element appearing in step 0 and step 3, and the negative element in step 1 all exist. Since the rows $r_1, r_2, \ldots$ and the columns $c_1, c_2, \ldots$ are chosen from finite sets, a row or a column eventually repeats; hence, Algorithm 1 will eventually branch to step 7 from either step 2 or step 5. Observe in the construction of Algorithm 1, $p_{ij} = 1$ only if $y_{ij} > 0$, and $p_{ij} = -1$ only if $y_{ij} < 0$. The construction given in Algorithm 1 is illustrated in Figure 1. In this case, the first positive element of $\mathbf{Y}$ is the $(1,1)$ element and termination is achieved when a row repeats.

Lemma 2.2. *Given an n by k matrix $\mathbf{Y}$ satisfying $\mathbf{Y1} = \mathbf{0} = \mathbf{Y}^\top \mathbf{1}$, $\mathbf{Y} \ne \mathbf{0}$, there exists an integer $p > 0$, scalars $\alpha_l > 0$ and matrices $\mathbf{P}_l \in \mathscr{P}$, for $1 \le l \le p$, such that*

$$\mathbf{Y} = \sum_{l=1}^{p} \alpha_l \mathbf{P}_l, \tag{3}$$

where for each l, supp $\mathbf{P}_l \subset$ supp $\mathbf{Y}$, *and*

$$(\mathbf{P}_l)_{ij} = +1 \quad \text{only if } y_{ij} > 0, \text{ and} \tag{4}$$

$$(\mathbf{P}_l)_{ij} = -1 \quad \text{only if } y_{ij} < 0. \tag{5}$$

Proof. The proof is by induction. We initialize $\mathbf{Y}_0 = \mathbf{Y}$, and suppose that for $l = 0, 1, \ldots, q-1$ we have positive scalars α_l and path matrices $\mathbf{P}_l$ whose support is contained in that of $\mathbf{Y}$, and which satisfy (4)–(5). Let $\mathbf{Y}_l$ be defined by the recurrence

$$\mathbf{Y}_{l+1} = \mathbf{Y}_l - \alpha_l \mathbf{P}_l, \quad 0 \le l < q.$$

We assume that these matrices have the following properties: (a) the nonzero elements of $\mathbf{Y}_{l+1}$ and the corresponding elements of $\mathbf{Y}_l$ have the same sign, and (b) the support of $\mathbf{Y}_{l+1}$ is strictly contained in the support of $\mathbf{Y}_l$ whenever $0 \le l < q$. Since $\mathbf{Y}_q^\top \mathbf{1} = \mathbf{0} = \mathbf{1}^\top \mathbf{Y}_q$, Algorithm 1 yields a path matrix $\mathbf{P}_q$ with support contain in that of $\mathbf{Y}_q$, and as observed after the statement of the algorithm, $(\mathbf{P}_q)_{ij} = +1$ only if $(\mathbf{Y}_q)_{ij} > 0$, and $(\mathbf{P}_q)_{ij} = -1$ only if $(\mathbf{Y}_q)_{ij} < 0$. Consequently, the elements of $\mathbf{Y}_q - \alpha \mathbf{P}_q$ all have the same sign for $\alpha > 0$ sufficiently small. Let α_q be the smallest α for which a nonzero element of $\mathbf{Y}_q$ is zero in $\mathbf{Y}_q - \alpha \mathbf{P}_q$. By the choice of α_q, (a) and (b) hold for $l = q$. This completes the induction step. Since the support of the $\mathbf{Y}_l$ is strictly decreasing, there exists a p such that $\mathbf{Y}_{p+1} = \mathbf{0}$, which yields (3) since

$$\mathbf{Y}_{p+1} = \mathbf{Y} - \sum_{l=0}^{p} \alpha_l \mathbf{P}_l. \qquad \square$$

Proof of Theorem 2.1. If $\mathbf{X}$ is a local minimizer in (1), then (P1) holds since the constraints in (1) are linear equalities and inequalities. Let b be the bilinear form defined by

$$b(\mathbf{Y}, \mathbf{Z}) = \operatorname{trace} \mathbf{Y}^\top \mathbf{A} \mathbf{Z}.$$

According to [6, Thm. 1],

$$b(\mathbf{Y}, \mathbf{Y}) \le 0 \quad \text{for all } \mathbf{Y} \in \mathscr{F}_{\mathbf{X}}(\Omega_0), \tag{6}$$

where

$$\mathscr{F}_{\mathbf{X}}(\Omega_0) = \{\mathbf{Y} : \mathbf{Y}\mathbf{1} = \mathbf{0} = \mathbf{Y}^\top \mathbf{1},\ y_{ij} \ge 0 \text{ if } x_{ij} = 0,\ \operatorname{supp} \mathbf{Y} \subset \Omega_0\}.$$

The condition (6) is the copositivity condition studied in [5]. Since $\mathscr{P}_{\mathbf{X}}(\Omega_0) \subset \mathscr{F}_{\mathbf{X}}(\Omega_0)$, we have

$$b(\mathbf{Y}, \mathbf{Y}) \le 0 \quad \text{for any } \mathbf{Y} \in \mathscr{P}_{\mathbf{X}}(\Omega_0). \tag{7}$$

Since $b(\mathbf{Y}, \mathbf{Y})$ is unchanged after a permutation of rows and columns in $\mathbf{Y}$ and $\mathbf{A}$, we can assume, without loss of generality, that the nonzero elements of $\mathbf{Y}$ are the following:

$$y_{jj} = 1 \quad \text{for } 1 \le j \le i, \quad y_{j,j+1} = -1 \quad \text{for } 1 \le j < i, \quad y_{i1} = -1,$$

For this $\mathbf{Y}$, we have

$$b(\mathbf{Y}, \mathbf{Y}) = (a_{11} + a_{ii} - 2a_{i1}) + \sum_{j=1}^{i-1} (a_{jj} + a_{j+1,j+1} - 2a_{j,j+1}). \tag{8}$$

Since $a_{ii} + a_{jj} \geq 2a_{ij}$ for each i and j, (8) implies that $b(\mathbf{Y}, \mathbf{Y}) \geq 0$. Combining this with (7), we deduce that $b(\mathbf{Y}, \mathbf{Y}) = 0$ for each $\mathbf{Y} \in \mathscr{P}_{\mathbf{X}}(\Omega_0)$. If $\mathbf{Y}, \mathbf{Z} \in \mathscr{P}_{\mathbf{X}}(\Omega_0)$, then $\mathbf{Y} + \mathbf{Z} \in \mathscr{F}_{\mathbf{X}}(\Omega_0)$. Since $b(\mathbf{Y}, \mathbf{Y}) = 0 = b(\mathbf{Z}, \mathbf{Z}) = 0$, the relation $b(\mathbf{Y} + \mathbf{Z}, \mathbf{Y} + \mathbf{Z}) \leq 0$ from (6) implies that $b(\mathbf{Y}, \mathbf{Z}) \leq 0$. This completes the proof of (P2).

Conversely, suppose that (P1) and (P2) hold. By [6, Thm. 3] $\mathbf{X}$ is a local minimizer if (6) holds. Given any nonzero $\mathbf{Y} \in \mathscr{F}_{\mathbf{X}}(\Omega)$, the decomposition of $\mathbf{Y}$ into path matrices provided by Lemma 2.2 along with (P2) give (6). Hence, $\mathbf{X}$ is a local minimizer. □

Corollary 2.3. *Suppose that* $\mathbf{X}$ *is a local minimizer in (1), and* $a_{ii} + a_{jj} \geq 2a_{ij}$ *for each* i *and* j. *Let* $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$, *and* $\boldsymbol{\omega}$ *satisfy (2), and define* $\boldsymbol{\Omega}_0 = \{(i, j) : \omega_{ij} = 0\}$. *If* $\mathbf{Y} \in \mathscr{P}_{\mathbf{X}}(\Omega_0)$ *and for some column* l *of* $\mathbf{Y}$, *we have* $y_{il} \neq 0 \neq y_{jl}$, *then* $a_{ii} + a_{jj} = 2a_{ij}$.

Proof. Since $\mathbf{X}$ is a local minimizer, (P2) implies that $b(\mathbf{Y}, \mathbf{Y}) = 0$ for any $\mathbf{Y} \in \mathscr{P}_{\mathbf{X}}(\Omega_0)$. By (8) and the assumption $a_{ii} + a_{jj} \geq 2a_{ij}$, it follows that each of the terms in parentheses in (8) vanishes. The elements in each of these terms correspond to a $+1$ and -1 combination in some column of $\mathbf{Y}$. After inverting the permutation used in (8), we conclude that $a_{ii} + a_{jj} = 2a_{ij}$ if a column of $\mathbf{Y}$ contains a $+1$ and -1 combination in rows i and j. □

Corollary 2.4. *If* $a_{ii} + a_{jj} \geq 2a_{ij}$ *for each* i *and* j *and* $\mathbf{X}$ *is feasible in (1), then a necessary and sufficient condition for* $\mathbf{X}$ *to be a strict local minimizer is that (P1) holds and the set* $\mathscr{P}_{\mathbf{X}}(\Omega_0)$ *in (P2) is empty.*

Proof. If $\mathbf{X}$ is a strict local minimizer in (1), then (P1) and (P2) hold by Theorem 2.1. Suppose that $\mathscr{P}_{\mathbf{X}}(\Omega_0)$ is nonempty, let $\mathbf{Y} \in \mathscr{P}_{\mathbf{X}}(\Omega_0)$, and let L be the Lagrangian defined by

$$L(\mathbf{X}) = f(\mathbf{X}) + \boldsymbol{\lambda} \cdot (\mathbf{X1} - \mathbf{1}) + \boldsymbol{\mu} \cdot (\mathbf{X}^{\top}\mathbf{1} - \mathbf{m}) + \boldsymbol{\omega} \cdot \mathbf{X},$$

where $f(\mathbf{X}) = b(\mathbf{X}, \mathbf{X})$ is the cost function in (1). Expanding L in a Taylor series around $\mathbf{X}$ and utilizing (2), we deduce that

$$f(\mathbf{X} + \alpha\mathbf{Y}) = f(\mathbf{X}) - \alpha\boldsymbol{\omega} \cdot \mathbf{Y} + \alpha^2 f(\mathbf{Y})$$

whenever $\mathbf{Y1} = \mathbf{0} = \mathbf{Y}^{\top}\mathbf{1}$. Since $f(\mathbf{Y}) = 0$ and $\boldsymbol{\omega} \cdot \mathbf{Y} = 0$ for $\mathbf{Y} \in \mathscr{P}_{\mathbf{X}}(\Omega_0)$, it follows that $f(\mathbf{X} + \alpha\mathbf{Y}) = f(\mathbf{X})$ for all choices of α. Since $\mathbf{X} + \alpha\mathbf{Y}$ is feasible in (1) for $\alpha > 0$ sufficiently small, $\mathbf{X}$ is not a strict local minimum. This is a contradiction, so $\mathscr{P}_{\mathbf{X}}(\Omega_0)$ is empty.

Conversely, if (P1) holds and $\mathscr{P}_{\mathbf{X}}(\Omega_0)$ is empty, then $\mathbf{X}$ is a local minimizer of (1) by Theorem 2.1. By [6, Thm. 2] $\mathbf{X}$ is a strict local minimizer if $\mathscr{F}_{\mathbf{X}}(\Omega_0)$ is empty. By Lemma 2.2 elements of $\mathscr{F}_{\mathbf{X}}(\Omega_0)$ are positive combinations of elements of $\mathscr{P}_{\mathbf{X}}(\Omega_0)$. Since $\mathscr{P}_{\mathbf{X}}(\Omega_0)$ is empty, $\mathscr{F}_{\mathbf{X}}(\Omega_0)$ is empty and $\mathbf{X}$ is a strict local minimizer. □

Given any $\mathbf{X}$ feasible in (1), define

$$\Omega_+(\mathbf{X}) = \{(i, j) : 0 < x_{ij} < 1\},$$

the set of (free) indices associated with inactive constraints. A construction similar to that of Algorithm 1 can be used to generate a path matrix supported on $\Omega_+(\mathbf{X})$. In Algorithm 1, we use the fact that the row and column sums of $\mathbf{Y}$ vanish to deduce that in each row (column) where $\mathbf{Y}$ has a positive entry, there must be a negative entry. In the same way, if the row and column sums of $\mathbf{X}$ are integers, then in each row (column) where $\mathbf{X}$ has a fractional entry, there must be another fractional entry.

Algorithm 2 ($\mathbf{X1 = 1}$, $\mathbf{X}^\top\mathbf{1} = \mathbf{m}$, $\mathbf{X}$ not 0/1, find $\mathbf{P} \in \mathscr{P}$, supp $\mathbf{P} \subset \boldsymbol{\Omega}_+(\mathbf{X})$)

0. Find $(i, j) \in \Omega_+(\mathbf{X})$, set $l = 1$, $r_1 = i$, $c_1 = j$.
1. Holding j fixed, find a new i such that $(i, j) \in \Omega_+(\mathbf{X})$.
2. If $i = r_p$ for some $1 \leq p \leq l$, goto step 7.
3. Increment l and set $r_l = i$.
4. Holding i fixed, find a new j such that $(i, j) \in \Omega_+(\mathbf{X})$.
5. If $j = c_p$ for some $1 \leq p < l$, set $r_p = i$, decrement l, and goto step 7.
6. Otherwise set $c_l = j$ and goto step 1.
7. Set $\mathbf{r} = (r_p, r_{p+1}, \ldots, r_l)$, $\mathbf{c} = (c_p, c_{p+1}, \ldots, c_l)$, and $\mathbf{P} = \text{path}(\mathbf{r}, \mathbf{c})$.

end Algorithm 2

Given any $\mathbf{X}$ feasible in (1), we now show how to obtain a 0/1 matrix $\mathbf{Y}$ that is feasible in (1) with $\text{trace}\,\mathbf{X}^\top\mathbf{AX} \leq \text{trace}\,\mathbf{Y}^\top\mathbf{AY}$. This construction involves adding successive multiples of a path matrix to $\mathbf{X}$. We now give one step in this construction: If $\mathbf{X}$ is feasible in (1) with $\mathbf{X}$ not 0/1, Algorithm 2 yields $\mathbf{P} \in \mathscr{P}$ with $\text{supp}\,\mathbf{P} \subset \Omega_+(\mathbf{X})$. Observe that

$$\begin{aligned}&\text{trace}(\mathbf{X} + \alpha\mathbf{P})^\top\mathbf{A}(\mathbf{X} + \alpha\mathbf{P})\\ &\qquad = \text{trace}\,\mathbf{X}^\top\mathbf{AX} + 2\alpha\,\text{trace}\,\mathbf{P}^\top\mathbf{AX} + \alpha^2\,\text{trace}\,\mathbf{P}^\top\mathbf{AP}.\end{aligned}$$

It follows from (8) that

$$\text{trace}\,\mathbf{P}^\top\mathbf{AP} \geq 0 \tag{9}$$

whenever $a_{ii} + a_{jj} \geq 2a_{ij}$ for each i and j. If $\text{trace}\,\mathbf{P}^\top\mathbf{AX} \geq 0$, then increase α until the first positive component of $\mathbf{X} + \alpha\mathbf{P}$ becomes zero. If $\text{trace}\,\mathbf{P}^\top\mathbf{AX} \leq 0$, then decrease α until the first positive component of $\mathbf{X} + \alpha\mathbf{P}$ becomes zero. Let α_1 denote the value of α where this component becomes zero. Due to our choice for the sign of α_1, we have

$$\alpha_1\,\text{trace}\,\mathbf{P}^\top\mathbf{AX} \geq 0.$$

Combining this with (9) gives

$$\text{trace}(\mathbf{X} + \alpha_1\mathbf{P})^\top\mathbf{A}(\mathbf{X} + \alpha_1\mathbf{P}) \geq \text{trace}\,\mathbf{X}^\top\mathbf{AX}.$$

Also, by the choice of α_1, the support of $\mathbf{X} + \alpha_1\mathbf{P}$ is strictly contained in the support of $\mathbf{X}$. Continuing this process, with $\mathbf{X}$ replaced by $\mathbf{X} + \alpha_1\mathbf{P}$, we generate a sequence of feasible points for (1), with increasing value and with decreasing support. Eventually, we obtain a 0/1 matrix whose value in (1) is at least as big as that of the starting matrix $\mathbf{X}$. We summarize this construction in the following algorithm:

Algorithm 3 (Move from X to Y with ascent, Y feasible in (1), Y 0/1)
0. Initialize $\mathbf{X}_0 = \mathbf{X}$, $k = 0$.
1. Choose $\mathbf{P} \in \mathscr{P}$, $\operatorname{supp} \mathbf{P} \subset \Omega_+(\mathbf{X}_k)$, using Algorithm 2.
2. Set $s = 1$ if $\operatorname{trace} \mathbf{P}^\top \mathbf{A}\mathbf{X}_k \geq 0$, and $s = -1$ otherwise.
3. Set $\alpha_1 = \max\{\alpha : \mathbf{X}_k + \alpha s \mathbf{P} \geq \mathbf{0}\}$.
4. Set $\mathbf{X}_{k+1} = \mathbf{X}_k + \alpha_1 s \mathbf{P}$, and increment k.
5. If $\mathbf{X}_k$ is 0/1, set $\mathbf{Y} = \mathbf{X}_k$ and stop; otherwise goto step 1.

end Algorithm 3

We now give an equivalent formulation of the first-order conditions (2) for 0/1 matrices in terms of a feasibility problem for a system of inequalities. Given a 0/1 matrix $\mathbf{X}$ that is feasible in (1), define the set

$$\chi_j = \{i : x_{ij} = 1\}.$$

Since the column sums of $\mathbf{X}$ total n, we have

$$\bigcup_{j=1}^{k} \chi_j = \{1, 2, \ldots, n\} \quad \text{and} \quad \chi_i \cap \chi_j = \emptyset \quad \text{for } i \neq j.$$

Theorem 2.5. *Given a* 0/1 *matrix* $\mathbf{X}$ *that is feasible in (1), let* $\mathbf{C}$ *be the* k *by* k *matrix with elements*

$$c_{ij} = \max_{l \in \chi_j} 2(\mathbf{A}(\mathbf{x}_i - \mathbf{x}_j))_l,$$

where $\mathbf{x}_i$ *is the* i*-th column of* $\mathbf{X}$*. There exists a solution* $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$, *and* $\boldsymbol{\omega}$ *of (2) if and only if there exists a solution* $\boldsymbol{\mu} \in \mathbf{R}^k$ *to the following* k *by* k *system of linear inequalities:*

$$\mu_j - \mu_i \geq c_{ij}, \quad 1 \leq i \leq k,\ 1 \leq j \leq k. \tag{10}$$

A solution $\boldsymbol{\lambda}$ *and* $\boldsymbol{\omega}$ *of (2) corresponding to a solution* $\boldsymbol{\mu}$ *of (10) is*

$$\lambda_l = -2(\mathbf{A}\mathbf{x}_j)_l - \mu_j \quad \textit{for all } l \in \chi_j,\ j = 1, 2, \ldots, k, \tag{11}$$

and

$$\boldsymbol{\omega} = -(2\mathbf{A}\mathbf{X} + \boldsymbol{\lambda}\mathbf{1}^\top + \mathbf{1}\boldsymbol{\mu}^\top). \tag{12}$$

Proof. Suppose $\boldsymbol{\mu}$ satisfies (10) and $\boldsymbol{\lambda}$ and $\boldsymbol{\omega}$ are given by (11) and (12). If $\boldsymbol{\omega}_j$ denotes the j-th column of $\boldsymbol{\omega}$, then

$$\boldsymbol{\omega}_j = -(2\mathbf{A}\mathbf{x}_j + \boldsymbol{\lambda} + \mathbf{1}\mu_j). \tag{13}$$

After substituting for $\boldsymbol{\lambda}$ using (11), we obtain $\omega_{lj} = 0$ for $l \in \mathscr{X}_j$. Hence, the complementary slackness condition and the inequality $\boldsymbol{\omega} \geq \mathbf{0}$ in (2) are satisfied for each j and for each $l \in \chi_j$. For $i \neq j$ and $l \in \chi_j$, (13) gives

$$\boldsymbol{\omega}_{li} = -(2\mathbf{A}\mathbf{x}_i + \boldsymbol{\lambda})_l - \mathbf{1}\mu_i. \tag{14}$$

Table 1. An example graph

Node	Connect to	Node	Connect to
1	7 12 13 14 15 16 17	9	11 15 19
2	12 17 18 20	11	14 17 18 20
3	5 11 13 14 18 19 20	12	14
4	6 9	13	18 20
5	7 9 10 12 16 19	14	16 18 20
6	16 18 20	16	18
7	8 9 11 16	17	18
8	15 18	18	20

Table 2. A partitioning and its refinement

Set	Chaco			Refinement		
1	1	12	14	1	12	14
2	7	11	16	7	9	11
3	5	9	10	5	10	16
4	4	6	20	4	6	20
5	2	13	17	2	17	18
6	8	15	18	8	13	15
7	3	19		3	19	

Substituting (11) in (14) yields

$$\omega_{li} = \mu_j - \mu_i - 2(\mathbf{A}(\mathbf{x}_i - \mathbf{x}_j))_l$$

for all $l \in \chi_j$. If (10) holds, then $\omega_{li} \geq 0$. Since $x_{li} = 0$, the last two conditions in (2) are satisfied.

Conversely, suppose that (2) holds. Obviously, (12) holds trivially. By the complementary slackness condition in (2), $\omega_{lj} = 0$ for all $l \in \chi_j$ since $x_{lj} = 1$. Hence, the first equation in (2), restricted to column j and rows $l \in \chi_j$ gives (11). Now consider the i-th column in (2), restricted to rows $l \in \chi_j$. After substituting for λ_l using (11), we have

$$2(\mathbf{A}\mathbf{x}_i - \mathbf{A}\mathbf{x}_j)_l + \mu_i - \mu_j + \omega_{li} = 0.$$

Since $\omega_{li} \geq 0$, we conclude that $\mu_j - \mu_i \geq 2(\mathbf{A}\mathbf{x}_i - \mathbf{A}\mathbf{x}_j)_l$ for each $l \in \chi_j$; this implies that $\mu_j - \mu_i \geq c_{ij}$. □

As a small illustration of Theorem 2.5, we consider the graph (see [10, Table 3], [25]) described in Table 1.

Using the Chaco partitioning code [18], the vertices of the graph were partitioned into 6 sets of size 3 and 1 set of size 2, while minimizing the number of cut edges (edges connecting vertices in different sets in the partition). There were 38 cut edges for the Chaco partitioning shown in Table 2. Using an LP code, we found that the system (10) was infeasible, so this partitioning did not correspond to a local minimizer in (1). Using the gradient projection algorithm and the starting guess corresponding to the Chaco partitioning, we converged to the refinement shown in Table 2, for which the number of cut edges was 36. Notice that the refinement interchanged vertices 9 and 16 in sets 2 and 3, and vertices 13 and 18 in sets 5 and 6.

References

[1] Alpert CJ, Kahng AB (1995) Recent directions in netlist partition: A survey, Integration, the VLSI Journal, 19:1–81
[2] Barnes ER (1982) An algorithm for partitioning the nodes of a graph, SIAM Journal on Algebraic and Discrete Mathematics, 3:541–550
[3] Barnes ER, Hoffman AJ (1984) Partitioning, spectra, and linear programming, in Progress in Combinatorial Optimization, W. E. Pulleyblank, ed., Academic Press, New York, 1984
[4] Barnes ER, Vannelli A, Walker JQ (1988) A new heuristic for partitioning the nodes of a graph, SIAM Journal on Discrete Mathematics, 1:299–305
[5] Bomze IM (1996) Block pivoting and shortcut strategies for detecting copositivity, Linear Algebra Appl., 248:161–184
[6] Borwein JM (1982) Necessary and sufficient conditions for quadratic minimality, Numer. Funct. Anal. Optimization, 5:127–140
[7] Bui T, Jones C (1993) A heuristic for reducing fill in sparse matrix factorization, 6-th SIAM Conf. Parallel Processing for Scientific Computing, pp. 445–452
[8] Cheng C-K, Wei Y-CA (1991) An improved two-way partitioning algorithm with stable performance, IEEE Transactions on Computer Aided Design, 10:1502–1511
[9] Contesse LB (1980) Une caractérisation complète des minima locaux en programmation quadratique, Numer. Math., 34:315–332
[10] Donath WE, Hoffman AJ (1973) Lower bounds for the partitioning of graphs, IBM J. of Research and Development, 17:420–425
[11] Falkner J, Rendl F, Wolkowicz H (1994) A computational study of graph partitioning, Mathematical Programming, 66:211–240
[12] Gibbons LE, Hearn DW, Pardalos PM, Ramana MV (1997) Continuous characterizations of the maximum clique problem, Mathematics of Operations Research, 22:754–768
[13] Gilbert JR, Miller GL, Teng S-H (1998) Geometric mesh partitioning: implementation and experiments, SIAM J. Sci. Comput., 19:2091–2110
[14] Hager WW, Krylyuk Y (1999) Graph partitioning and continuous quadratic programming, SIAM J. Discrete Math., 12:500–523
[15] Heath MT, Raghavan P (1995) A Cartesian parallel nested dissection algorithm, SIAM J. Matrix Anal. Appl., 16:235–253
[16] Hendrickson B, Leland R (1995) An improved spectral graph partitioning algorithm for mapping parallel computations, SIAM J. Sci. Comput., 16:452–469
[17] Hendrickson B, Leland R (1993) A Multilevel Algorithm for Partitioning Graphs, Technical Report SAND93-1301, Sandia National Laboratories, Albuquerque, NM
[18] Hendrickson B, Rothberg E (1999) Improving the runtime and quality of nested dissection ordering, SIAM J. Scientific Computing, 20:468–489
[19] Karypis G, Kumar V (1998) A fast and high quality multilevel scheme for partitioning irregular graphs, SIAM J. Sci. Comput., 20:359–392
[20] Kernighan BW, Lin S (1970) An efficient heuristic procedure for partitioning graphs, Bell System Technical Journal, 49:291–307
[21] Lengauer T (1990) Combinatorial Algorithms for Integrated Circuit Layout, John Wiley, Chichester
[22] Miller GL, Teng S-H, Thurston W, Vavasis SA (1993) Automatic mesh partitioning, in Sparse Matrix Computations: Graph Theory Issues and Algorithms, J. Gilbert and J. Liu eds., IMA Vol. Math. Appl., Springer-Verlag
[23] Motzkin TS, Strauss EG (1965) Maxima for graphs and a new proof of a theorem of Turan, Canad. J. Math., 17:533–540
[24] Pothen A, Simon HD, Liou K-P (1990) Partitioning sparse matrices with eigenvectors of graphs, SIAM J. Matrix Anal. Appl., 11:430–452
[25] Rendl F, Wolkowicz H (1995) A projection technique for partitioning the nodes of a graph, Ann. Oper. Res., 58:172–191
[26] Teng S-H (1998) Provably good partitioning and load balancing algorithms for parallel adaptive N-body simulation, SIAM J. Sci. Comput., 19:635–656
[27] Wolkowicz H, Zhao Q (1999) Semidefinite programming relaxations for the graph partitioning problem, Discrete Appl. Math. 96/97:461–479