June 1990

UILU-ENG-90-2219
DAC-20

COORDINATED SCIENCE LABORATORY
College of Engineering

SUFFICIENT CONDITIONS, COST BOUNDS, AND APPROXIMATION ALGORITHMS FOR THE GRAPH BISECTIONING PROBLEM

Youssef Saab
Vasant Rao

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION: Unclassified

1b. RESTRICTIVE MARKINGS: None

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE

3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S): UILU-ENG-90-2219 (DAC-20)

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION: Coordinated Science Lab, University of Illinois

6b. OFFICE SYMBOL (If applicable): N/A

7a. NAME OF MONITORING ORGANIZATION: Semiconductor Research Corporation

6c. ADDRESS (City, State, and ZIP Code): 1101 W. Springfield Ave., Urbana, IL 61801

7b. ADDRESS (City, State, and ZIP Code): P.O. Box 12053, Research Triangle Park, NC 27709

8a. NAME OF FUNDING/SPONSORING ORGANIZATION: Semiconductor Research Corporation

8b. OFFICE SYMBOL (If applicable)

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: SRC 86-12-109

8c. ADDRESS (City, State, and ZIP Code): P.O. Box 12053, Research Triangle Park, NC 27709

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification): Sufficient Conditions, Cost Bounds, and Approximation Algorithms for the Graph Bisectioning Problem

12. PERSONAL AUTHOR(S): Saab, Youssef and Rao, Vasant

13a. TYPE OF REPORT: Technical

13b. TIME COVERED: FROM ______ TO ______

14. DATE OF REPORT (Year, Month, Day): 1990 June

15. PAGE COUNT: 43

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number): Graph bisectioning, lower bound, upper bound, conditions for optimality, heuristics.

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

In this report, the graph bisectioning problem is studied. Lower bounds and upper bounds on the cost of a bisection are derived. Also, conditions for optimality of a bisection are given.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION: Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL

22b. TELEPHONE (Include Area Code)

22c. OFFICE SYMBOL

SUFFICIENT CONDITIONS, COST BOUNDS, AND APPROXIMATION ALGORITHMS FOR THE GRAPH BISECTIONING PROBLEM

Youssef Saab and Vasant Rao

Coordinated Science Laboratory
and Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
1101 W. Springfield Avenue,
Urbana, Illinois 61801.

ABSTRACT

The graph bisectioning problem has several applications in VLSI layout such as floor planning and module placement [16-19]. In this paper we begin by presenting a necessary and sufficient condition for optimality of a given bisection. Two other sufficient conditions are then derived. Upper and lower bounds on the cost of an optimal bisection are obtained. A duality between the graph bisectioning problem and a nonlinear programming problem of maximizing a concave function is exhibited. We also show that for dense graphs, a bisection that approximates an optimal one can be easily found. Finally, we exhibit a class of graphs for which the ratio of the upper and lower bounds approaches 1 as the number of vertices in the graph increases.

† This work was supported by the Semiconductor Research Corporation under contract # SRC 86-12-109.

1. Preliminaries

A graph $G(V,E)$ consists of a set of vertices V, and a set of edges E, where an edge $e \in E$ is a pair of vertices (not necessarily distinct) in V. If an edge $e = \{u,v\}$, then we say that e is *incident* on u and v. We also say that u and v are the *ends* of e. If each edge in E is an ordered pair of vertices in V, the graph G is said to be *directed*; otherwise, G is said to be *undirected*. A *weighting function* on a set S is a mapping $f : S \to \mathbf{R}$ from this set to the reals. The *weight* of an element $s \in S$ is denoted by $f(s)$. A set S is said to be *weighted* if it has a weighting function defined on it. In this case, the *weight* of any subset $A \subseteq S$ is defined to be $f(A) = \sum_{s \in A} f(s)$. A graph $G(V,E)$ is said to be an *edge-weighted graph* if the set E is a weighted set. Similarly, a graph $G(V,E)$ is a *vertex-weighted graph* if the set V is a weighted set. A graph $G(V,E)$ is said to be *connected* if for any two vertices $u,v \in V$ there exist a non-null sequence $v_0 v_1 \cdots v_k$ of vertices in V such that $u = v_0$, $v = v_k$, and for $0 \leqslant i \leqslant k-1$ the pair $\{v_i, v_{i+1}\}$ is an edge in E (such a sequence is called a path).

In this paper, we will only consider **undirected edge-weighted graphs**. These graphs may or may not be connected. For more information about graphs, the reader is referred to [1].

The Connection matrix of a graph : Let $G(V,E)$ be an undirected, edge-weighted graph. Let $n = |V|$ be the cardinality of the vertex set V. The *connection matrix* $C(G)$ of G is

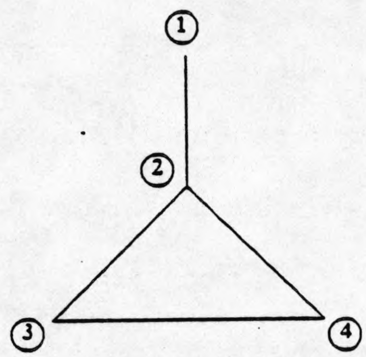

Figure 1 : A graph on 4 vertices

an $n \times n$ matrix whose $ij-th$ entry c_{ij} is the weight of the edge $\{i,j\}$ if $\{i,j\}$ is an edge, and is 0 otherwise. For example, consider the graph shown in Figure 1 with all edges having unit weight. Its connection matrix is

$$C = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}.$$

Note that the connection matrix $C(G)$ is symmetric because G is an undirected graph. If there is only one graph under consideration we will use the notation C instead of $C(G)$ to denote the connection matrix.

Partition of a set : We say that the two non-empty subsets S_1 and S_2 *partition* a set S if $S = S_1 \cup S_2$, and $S_1 \cap S_2 = \emptyset$. A partition of a set S into two subsets S_1 and S_2 is denoted the the *unordered* pair (S_1,S_2) (i.e., (S_1,S_2) and (S_2,S_1) represent the same partition). A partition of a graph $G(V,E)$ is a partition of its vertex set V. A partition (V_1,V_2) of a graph G is said to be a *bisection* if $|V_1| = |V_2|$, i.e., the two subsets have equal cardinality. An edge $e \in E$ is said to be *cut* by a partition (V_1,V_2) of G if its ends belong to two different subsets of the partition. The set of all edges cut in $G(V,E)$ by the partition (V_1,V_2) will be denoted by $E_c(V_1,V_2)$. Once again we will simply use E_c instead of $E_c(V_1,V_2)$ if there is only one partition under consideration.

Cost of a partition : The *cost* of a partition is simply the sum of the weights of all the edges cut by the partition. More formally, if (V_1,V_2) is a partition of a graph G, then it's cost is defined by

$$cost(V_1,V_2) = \sum_{\{i,j\} \in E_c} c_{ij} = \sum_{i \in V_1} \sum_{j \in V_2} c_{ij}, \tag{1.1}$$

where c_{ij} is the $ij-th$ entry of the connection matrix C of the graph G. A bisection is *optimal* if it has the lowest cost among all bisections of the graph.

The Graph Bisectioning Problem (GB) : Let $G(V,E)$ be an undirected, edge-weighted graph on an even number n of vertices. The graph bisectioning problem (GB) is to find an optimal bisection (V_1,V_2) of G.

The GB problem has been shown to be NP-Complete [2]. Therefore, finding an optimal solution is, in general, intractable; and many heuristics have been suggested for its solution [3-9]. In this paper, we will focus on some theoretical aspects of GB. In another paper [20] we will provide some new heuristics for GB. At this stage we wish to emphasize that finding a partition (not necessarily of equal cardinality) of the lowest cost among all partitions of a graph $G(V,E)$ is an easy problem with polynomial-time-complexity of $O(n^{1/2}m^2)$ [23] where $n=|V|$ and $m=|E|$. It is therefore constraining the partitions to be bisections that makes the problem intractable.

Representation, Notations, and Conventions : All the graphs that we consider are undirected, edge-weighted graphs. We will always represent a graph $G(V,E)$ by its connection matrix C. The cardinality of the vertex set V will always be denoted by n, and is assumed to be even. Without loss of generality the vertex set V can be assumed to be $V=\{1,2,\ldots,n\}$. The entries of the connection matrix C can be assumed to be non-negative. If this is not the case, a constant α greater or equal to the negative of the minimum entry in C can be added to all the entries of C. This will not change the optimal solution of the graph bisectioning problem since the cost of all bisections will be raised by the same amount, namely, $n^2\alpha/4$. Also, the diagonal entries of the matrix C can be assumed to be zeroes, since an edge connecting a vertex to itself does not affect the cost of a bisection.

The rest of this paper is divided into five sections. In Section 2, we derive conditions for optimality of a given bisection. In Section 3, we derive lower bounds on the cost of any bisection. We also exhibit a duality between GB and a nonlinear programming problem. In Section 4, we give an upper bound on the cost of an optimal bisection. In Section 5, we introduce the notion of ϵ-approximations and show that for dense graphs an entire class of polynomial-time algorithms are ϵ-approximation algorithms. We also exhibit a class of graphs for which the ratio of the upper and lower bound approaches one as the size of the graph increases.

2. Conditions for Optimality

In this section, a necessary and sufficient condition and two other sufficient conditions for the optimality of a given bisection are presented.

Definition 2.1 : A bisection (V_1,V_2) of a graph $G(V,E)$ is said to be *m-optimal* if for any subset $X \subset V_1$ of cardinality m and for any subset $Y \subset V_2$ of equal cardinality m, we have $\text{cost}(V_1+Y-X, V_2+X-Y) \geqslant \text{cost}(V_1,V_2)$.

Informally, a bisection (V_1,V_2) of V is *m-optimal* if the exchange of any subset $X \subset V_1$ of cardinality m with any subset $Y \subset V_2$ of equal cardinality m cannot lead to a better bisection.

Lemma 2.1 : Let (V_1,V_2) and (P_1,P_2) be any two distinct bisections. Then there is a constant m such that $0 < m \leqslant \lfloor n/4 \rfloor$, and two subsets $X \subset V_1$ and $Y \subset V_2$ both of cardinality m such that exchanging X and Y in (V_1,V_2) produces the bisection (P_1,P_2), i.e., $P_1 = V_1+Y-X$ and $P_2 = V_2+X-Y$.

Proof : Let $A = P_1 \cap V_2$, $B = P_2 \cap V_1$, $C = P_1 \cap V_1$, and $D = P_2 \cap V_2$ as shown in Figure 2. Since both (P_1,P_2) and (V_1,V_2) are bisections of the set V, the sets A, B, C, and D are mutually disjoint. We can also write $P_1 = A \cup C$, $P_2 = B \cup D$, $V_1 = B \cup C$, and $V_2 = A \cup D$. But each of P_1, P_2, V_1, and V_2 has cardinality $\frac{n}{2}$. Hence, it follows that $|A| = |B|$, and $|C| = |D|$.

	P_1	P_2
V_1	C	B
V_2	A	D

Figure 2 : Pictorial representation of two distinct bisections

If $|A| = |B| \leqslant \lfloor n/4 \rfloor$ then we take $X = B$ and $Y = A$. Otherwise, it must be the case that $|C| = |D| \leqslant \lfloor n/4 \rfloor$ because $|B| + |C| = |V_1| = \frac{n}{2}$, and we take $X = C$ and $Y = D$. In either case we see that $P_1 = V_1 + Y - X$ and $P_2 = V_2 + X - Y$. □

The above lemma leads us to establish a necessary and sufficient condition for a bisection to be optimal.

Theorem 2.1 : A bisection (V_1, V_2) of a graph $G(V, E)$ is optimal if and only if it is *m-optimal* for each m satisfying $0 < m \leqslant \lfloor n/4 \rfloor$.

Proof : If a bisection is optimal then obviously it has to be *m-optimal* for $0 < m \leqslant \lfloor n/4 \rfloor$. Conversely, assume that a bisection (V_1, V_2) is not optimal, and let (P_1, P_2) be an optimal bisection. By Lemma 2.1, there is a subset $X \subset V_1$ and a subset $Y \subset V_2$ with $|X| = |Y| = m \leqslant \lfloor n/4 \rfloor$ such that if we exchange X and Y in (V_1, V_2) we get the bisection (P_1, P_2) which is defined to be better that (V_1, V_2). Therefore. (V_1, V_2) cannot be *m-optimal* for all m such that $0 < m \leqslant \lfloor n/4 \rfloor$. □

Although Theorem 2.1 provides us with a necessary and sufficient condition for a bisection to be optimal, it is computationally intractable to check for m*-optimality* for all m such that $0<m \leqslant \lfloor n/4 \rfloor$. For example, the brute force algorithm to check for m*-optimality* of a given bisection (V_1,V_2) would require the generation of all subsets of V_1 of cardinality m and all subsets of V_2 of cardinality m. After that, a trial exchange of each pair of subsets must be done to check if any reduction in the cost of the bisection is possible. This requires computation time that is at least proportional to $\binom{n/2}{m}^2$. Therefore, to check for optimality, the brute force algorithm requires computation time that is at least proportional to $\sum_{m=1}^{\lfloor n/4 \rfloor} \binom{n/2}{m}^2$ which is about $\frac{1}{2}\binom{n}{n/2}$. This quantity grows exponentially with n. Therefore, sufficient conditions for optimality that are easier to check are desirable. We will now establish two such conditions.

Theorem 2.2 : Let C be a connection matrix of a graph $G(V,E)$ and let (V_1,V_2) be a given bisection. If

$$c_{ij} + c_{kl} \geqslant c_{ik} + c_{jl} \ \forall i,j \in V_1,\ i \neq j \text{ and } \forall k,l \in V_2,\ k \neq l \tag{2.1}$$

then (V_1,V_2) is an optimal bisection. Furthermore, if the inequality in (2.1) is always strict, then (V_1,V_2) will be the only optimal bisection.

Proof : Let (P_1,P_2) be an arbitrary bisection distinct from (V_1,V_2). By Lemma 2.1, there is a subset $X \subset V_1$ and a subset $Y \subset V_2$ with $|X| = |Y| = m \leqslant \lfloor n/4 \rfloor$ such that if we exchange X and Y in (V_1,V_2) we can get the bisection (P_1,P_2). Let $\hat{X} = V_1 - X$ and let $\hat{Y} = V_2 - Y$. The cost of the bisection (V_1,V_2) can be written as

$$cost(V_1,V_2) = \sum_{i \in V_1} \sum_{j \in V_2} c_{ij} = \sum_{i \in X} \sum_{l \in Y} c_{il} + \sum_{i \in X} \sum_{k \in \hat{Y}} c_{ik} + \sum_{j \in \hat{X}} \sum_{l \in Y} c_{jl} + \sum_{j \in \hat{X}} \sum_{k \in \hat{Y}} c_{jk}$$

and the cost of the bisection (P_1,P_2) can be written as

$$cost(P_1,P_2) = \sum_{i \in P_1} \sum_{j \in P_2} c_{ij} = \sum_{i \in X} \sum_{l \in Y} c_{il} + \sum_{i \in X} \sum_{j \in \hat{X}} c_{ij} + \sum_{k \in \hat{Y}} \sum_{j \in \hat{X}} c_{kj} + \sum_{k \in \hat{Y}} \sum_{l \in Y} c_{kl}$$

Hence,

$$cost(P_1,P_2) - cost(V_1,V_2) = \sum_{i \in X} \sum_{j \in \hat{X}} c_{ij} + \sum_{k \in \hat{Y}} \sum_{l \in Y} c_{kl} - \sum_{i \in X} \sum_{k \in \hat{Y}} c_{ik} - \sum_{j \in \hat{X}} \sum_{l \in Y} c_{jl}$$

But note that we can write

$$\sum_{i \in X} \sum_{j \in \hat{X}} c_{ij} = \frac{1}{(\frac{n}{2} - m)m} \sum_{i \in X} \sum_{j \in \hat{X}} \sum_{l \in Y} \sum_{k \in \hat{Y}} c_{ij}$$

$$\sum_{k \in \hat{Y}} \sum_{l \in Y} c_{kl} = \frac{1}{(\frac{n}{2} - m)m} \sum_{i \in X} \sum_{j \in \hat{X}} \sum_{l \in Y} \sum_{k \in \hat{Y}} c_{kl}$$

$$\sum_{i \in X} \sum_{k \in \hat{Y}} c_{ik} = \frac{1}{(\frac{n}{2} - m)m} \sum_{i \in X} \sum_{j \in \hat{X}} \sum_{l \in Y} \sum_{k \in \hat{Y}} c_{ik}$$

$$\sum_{j \in \hat{X}} \sum_{l \in Y} c_{jl} = \frac{1}{(\frac{n}{2} - m)m} \sum_{i \in X} \sum_{j \in \hat{X}} \sum_{l \in Y} \sum_{k \in \hat{Y}} c_{jl}$$

Therefore,

$$cost(P_1,P_2) - cost(V_1,V_2) = \frac{1}{(\frac{n}{2} - m)m} \sum_{i \in X} \sum_{j \in \hat{X}} \sum_{l \in Y} \sum_{k \in \hat{Y}} (c_{ij} + c_{kl} - c_{ik} - c_{jl})$$

But from Equation (2.1) we have $c_{ij} + c_{kl} \geqslant c_{ik} + c_{jl}$ for all $i,j \in V_1, i \neq j$ and all $k,l \in V_2, k \neq l$. Therefore we must have $c_{ij} + c_{kl} \geqslant c_{ik} + c_{jl}$ $\forall i \in X, j \in \hat{X}, k \in \hat{Y}$, and $l \in Y$ which means that

$$cost\,(P_1,P_2) \geqslant cost\,(V_1,V_2).$$

Since the choice of the bisection (P_1,P_2) was arbitrary, the bisection (V_1,V_2) is optimal. If strict inequality always holds in Equation (2.1) then we have

$$cost\,(P_1,P_2) > cost\,(V_1,V_2)$$

and so (V_1,V_2) is the unique optimal bisection. □

Figure 3 shows an edge-weighted graph on 4 vertices for which the bisection $(\{1,2\},\{3,4\})$ satisfies the hypothesis of Theorem 2.2. Therefore, it is an optimal bisection for this graph. However, theorem 2.2 does not guarantee the uniqueness of this bisection since strict inequality in (2.1) is violated ($c_{12} + c_{34} = c_{14} + c_{32} = 8$). It is easy to see that the bisection $(\{1,4\},\{3,2\})$ has the same cost as $(\{1,2\},\{3,4\})$, and hence is also optimal.

We will now discuss an alternate sufficient condition for optimality. Given an arbitrary bisection (V_1,V_2) of a graph $G(V,E)$, let us define for each vertex $i \in V$ the follow-

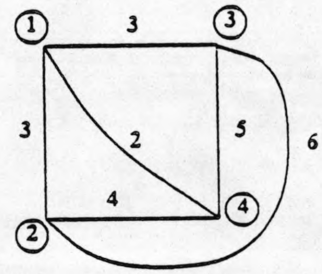

Figure 3 : A graph that satisfies the hypothesis of Theorem 2.2

ing:

$$E_i = \begin{cases} \sum_{j \in V_1} c_{ij} & \text{if } i \in V_2 \\ \sum_{j \in V_2} c_{ij} & \text{if } i \in V_1 \end{cases} \tag{2.2a}$$

$$I_i = \begin{cases} \sum_{j \in V_1} c_{ij} & \text{if } i \in V_1 \\ \sum_{j \in V_2} c_{ij} & \text{if } i \in V_2 \end{cases} \tag{2.2b}$$

$$D_i = I_i - E_i . \tag{2.2c}$$

It is clear from the above definitions that I_i measures how strongly vertex i is connected to its subset, and E_i measures how strongly vertex i is connected to the complement of its subset. One can think of I_i and E_i as the internal and external attraction of vertex i, respectively. Also, if $i \in V_1$ and $j \in V_2$, then $D_i + D_j + 2c_{ij}$ is the net increase in cost due to the exchange of i and j. Hence, if $D_i \geqslant 0 \; \forall i \in V$, then clearly (V_1,V_2) is *1−optimal* †. The next theorem gives us a lower bound on D_i that guarantees the *m−optimality* of the bisection (V_1,V_2) for all m between 1 and some positive integer $k \leqslant \lfloor n/4 \rfloor$.

Theorem 2.3 : Given an edge-weighted graph $G(V,E)$ with connection matrix C such that $c_{ij} \geqslant 0 \; \forall i,j \in V$ and $c_{ii} = 0 \; \forall i \in V$. Let (V_1,V_2) be a given bisection of the graph G. Let $M = \max_{1 \leqslant i \leqslant j \leqslant n} (c_{ij})$ denote the maximum edge-weight. Compute D_i for each vertex

† Provided $c_{ij} \geqslant 0$.

$i \in V$ from (2.2) and let $d = \min_{i \in V} D_i$. If $d \geqslant (k-1)M$ for some positive $k \leqslant \lfloor n/4 \rfloor$, then (V_1,V_2) is an m*-optimal* bisection for all m such that $1 \leqslant m \leqslant k$.

Proof : Let $X \subset V_1$ and $Y \subset V_2$ be any two nonempty subsets of equal cardinality $m \leqslant k$. Set $P_1 = V_1+Y-X$ and $P_2 = V_2+X-Y$, and let $\hat{X} = V_1 - X$ and $\hat{Y} = V_2 - Y$. Now as in the proof of Theorem 2.2 we have

$$cost(P_1,P_2) - cost(V_1,V_2) = \sum_{i \in X}\sum_{j \in \hat{X}} c_{ij} + \sum_{i \in \hat{Y}}\sum_{j \in Y} c_{ij} - \sum_{i \in X}\sum_{j \in \hat{Y}} c_{ij} - \sum_{i \in \hat{X}}\sum_{j \in Y} c_{ij}$$

Since $\hat{X} = V_1 - X$ we can write

$$\sum_{i \in X}\sum_{j \in \hat{X}} c_{ij} = \sum_{i \in X}\sum_{j \in V_1} c_{ij} - \sum_{i \in X}\sum_{j \in X} c_{ij} = \sum_{i \in X} I_i - \sum_{i \in X}\sum_{j \in X} c_{ij}$$

where we have used the definition of I_i from (2.2b). Similarly

$$\sum_{i \in Y}\sum_{j \in \hat{Y}} c_{ij} = \sum_{i \in Y} I_i - \sum_{i \in Y}\sum_{j \in Y} c_{ij}$$

$$\sum_{i \in X}\sum_{j \in \hat{Y}} c_{ij} = \sum_{i \in X} E_i - \sum_{i \in X}\sum_{j \in Y} c_{ij}$$

$$\sum_{i \in Y}\sum_{j \in \hat{X}} c_{ij} = \sum_{i \in Y} E_i - \sum_{i \in X}\sum_{j \in Y} c_{ij}$$

Combining the above equations we get

$$\begin{aligned} cost(P_1,P_2) - cost(V_1,V_2) &= \sum_{i \in X} I_i - \sum_{i \in X}\sum_{j \in X} c_{ij} + \sum_{i \in Y} I_i - \sum_{i \in Y}\sum_{j \in Y} c_{ij} \\ &\quad - \sum_{i \in X} E_i + \sum_{i \in X}\sum_{j \in Y} c_{ij} - \sum_{i \in Y} E_i + \sum_{i \in X}\sum_{j \in Y} c_{ij} \\ &= 2\sum_{i \in X}\sum_{j \in Y} c_{ij} - \sum_{i \in X}\sum_{j \in X} c_{ij} - \sum_{i \in Y}\sum_{j \in Y} c_{ij} + \sum_{i \in X} D_i + \sum_{i \in Y} D_i \end{aligned}$$

Since $c_{ij} \geqslant 0$ we have $2\sum_{i \in X}\sum_{j \in Y} c_{ij} \geqslant 0$. Also, we are given that $c_{ij} \leqslant M$, $m = |X| = |Y|$, and $c_{ii}=0$ for each $i \in V$. Therefore

$$-\sum_{i \in X} \sum_{j \in X} c_{ij} \geqslant -m(m-1)M$$

and

$$-\sum_{i \in Y} \sum_{j \in Y} c_{ij} \geqslant -m(m-1)M$$

Hence,

$$\begin{aligned} cost(P_1,P_2) - cost(V_1,V_2) &\geqslant -2m(m-1)M + \sum_{i \in X} D_i + \sum_{i \in Y} D_i \\ &= \sum_{i \in X} D_i - m(m-1)M + \sum_{i \in Y} D_i - m(m-1)M \\ &= \sum_{i \in X} \left[D_i - (m-1)M\right] + \sum_{i \in Y} \left[D_i - (m-1)M\right] \end{aligned}$$

which implies that if $D_i \geqslant (m-1)M \ \forall i \in V$ then $cost(P_1,P_2) \geqslant cost(V_1,V_2)$. But $m \leqslant k$; hence if

$$D_i \geqslant (k-1)M \ \forall i \in V \Longrightarrow cost(P_1,P_2) \geqslant cost(V_1,V_2)$$

for each m such that $1 \leqslant m \leqslant k$. But $X \subset V_1$ and $Y \subset V_2$ were arbitrary nonempty subsets of cardinality $m \leqslant k$. Therefore, (V_1,V_2) is m*-optimal* for each $1 \leqslant m \leqslant k$. □

An immediate consequence of Theorem 2.1 and Theorem 2.3 is

Corollary 2.1 : If $d \geqslant \frac{(n-4)}{4}M$ then (V_1,V_2) is an optimal bisection. Moreover, if $d > \frac{(n-4)}{4}M$ then (V_1,V_2) will be the unique optimal bisection.

Figure 4 shows a graph on six vertices. Consider the bisection $(\{1,2,3\},\{4,5,6\})$. The following table lists I_i, E_i, and D_i for each $i \in V$ computed using (2.2).

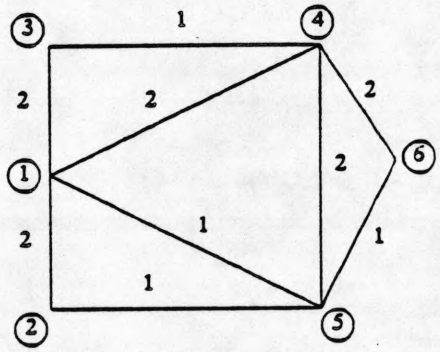

Figure 4 : A graph in which a bisection satisfies Corollary 2.1

i	I_i	E_i	D_i
1	4	3	1
2	2	1	1
3	2	1	1
4	4	3	1
5	3	2	1
6	3	0	3

For this example, $M=2$, $d=1$, and $\frac{(n-4)}{4}M = 1$. Therefore, the bisection $(\{1,2,3\},\{4,5,6\})$ satisfies the conditions of Corollary 2.1, and is therefore optimal.

Given a bisection (V_1,V_2) of a graph on an even number of vertices n, there are $\binom{n/2}{2}$ possible choices of $i,j \in V_1$ $i \neq j$ and $\binom{n/2}{2}$ possible choices of $k,l \in V_2$ $k \neq l$. Therefore, checking the validity of the condition (2.1) in Theorem 2.2 would require $O(n^4)$ time.

However, obtaining $d = \min_{1 \leq i \leq n} D_i$ from (2.2) would take $O(n^2)$ time, and therefore checking the validity of the condition of Corollary 2.1 would require only $O(n^2)$ time. Therefore, Corollary 2.1 gives us a sufficient condition that is computationally easier to check. However, this condition is too strong in the sense that very few graphs satisfy it.

We wish to emphasize, once again, that the conditions of both Theorem 2.2 and Corollary 2.1 are only *sufficient* conditions for optimality. Therefore, there may exist (and indeed there are) optimal bisections that do not satisfy either or both conditions. To illustrate this point, we considered several (around 100) examples of graphs generated randomly of size ranging between 50 and 200 vertices. For each of these graphs, we generated a "near optimal" bisection (V_1, V_2) using one of the heuristics in [20]. We then tested these bisections in each case to determine to what extent the conditions of Theorem 2.2 and Theorem 2.3 were satisfied. Our experimental results showed the following:

(1) For dense graphs (more than 80% of the edges present), the condition of Theorem 2.2 was satisfied by almost all bisections. Therefore, we were able to conclude that most of these bisections are in fact optimal. This strengthened our intuition that it is relatively easy to obtain good bisections for dense graphs. Theorem 5.1 in Section 5 of this paper will make our intuition precise.

(2) For sparse graphs (less than 10% of the edges present), the condition (2.1) of Theorem 2.2 was satisfied by more than 80% of all quadruples (i, j, k, l) with $i, j \in V_1$, $i \neq j$ and $k, l \in V_2$, $k \neq l$ in almost all the cases considered.

(3) In very few cases of sparse graphs and in all cases of dense graphs, we were only able to show 1-optimality of the bisections using the condition of Theorem 2.3. For few

cases of dense graphs, we were able to guarantee 2-optimality of the bisection using this condition.

The above results show that the condition of Corollary 2.1, though easy to check, is almost never satisfied even by optimal bisections in practice for large graphs. However, the condition of Theorem 2.2, which is computationally more difficult to check, guarantees the optimality of a bisection generated by a heuristic in [20] in almost all dense graphs and a few sparse graphs considered.

3. Lower Bounds

Throughout this section we will use the following notation. Let

$$\lambda_1(M) \geqslant \lambda_2(M) \geqslant \cdots \geqslant \lambda_n(M) \tag{3.1a}$$

denote the eigenvalues of an $n \times n$ real symmetric matrix M arranged in descending order and let

$$Tr(M) = \sum_{i=1}^{n} m_{ii} \tag{3.1b}$$

denote the *trace* of M, which is the sum of all the diagonal entries. We now derive a lower bound on the cost of any bisection of a graph.

Theorem 3.1 : Let $G(V,E)$ be a graph on n vertices with connection matrix C. Let (V_1,V_2) be any bisection of G. Let $W = ½\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}$ denote the sum of all the edge-weights in the graph G. Then

$$cost(V_1,V_2) \geqslant W + \frac{1}{2}Tr(D) - \frac{n}{4}(\lambda_1(C+D) + \lambda_2(C+D)) \tag{3.2}$$

where D is any $n \times n$ real *diagonal* matrix, i.e., $d_{ij} = 0 \; \forall i \neq j$.

Proof : Let P be the $n \times n$ *indicator matrix* for the bisection (V_1,V_2) defined by

$$p_{ij} = \begin{cases} 1 & \textit{if vertices } i \textit{ and } j \textit{ are in the same subset} \\ 0 & \textit{otherwise} \end{cases}$$

The columns of P corresponding to vertices in the same subset of the bisection are identical. Thus, P has exactly two distinct columns. Furthermore, these two columns are orthogonal due to the fact that V_1 and V_2 are disjoint. Therefore, P has rank 2 and, hence, 0 is an eigenvalue of P of multiplicity $(n-2)$. In addition, the two distinct columns of P are orthogonal 0—1 vectors which are the eigenvectors of P, corresponding to the eigenvalue $\frac{n}{2}$. Therefore the eigenvalues of P are

$$\frac{n}{2} \geqslant \frac{n}{2} \geqslant 0 \geqslant \cdots \geqslant 0 \tag{3.3}$$

Let $W_c = cost(V_1,V_2)$ denote the cost of the bisection (V_1,V_2) and $W_{nc} = W - W_c$ denote the sum of the edge-weights *not* cut by the bisection. We first note that

$$2W_{nc} = 2W - 2W_c = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}\,p_{ij} \tag{3.4}$$

by using the definitions of W, W_c, and P. We also note that

$$\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}\,p_{ij} = Tr(D) \tag{3.5}$$

since D is a diagonal matrix and $p_{ii} = 1 \;\; \forall i$ by definition. Now consider

$$Tr((C+D)P^T) = \sum_{i=1}^{n}\left[\sum_{j=1}^{n}(c_{ij}+d_{ij})p_{ij}\right] = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}p_{ij} + \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}p_{ij} \quad (3.6)$$

Combining (3.4), (3.5), and (3.6) we get

$$Tr((C+D)P^T) = Tr(D) + 2W_{nc} = Tr(D) + 2(W - W_c) \quad (3.7)$$

Now by applying the main theorem of Hoffman and Wielandt [10] to real symmetric matrices $(C+D)$ and P we get

$$Tr((C+D)P^T) \leqslant \sum_{i=1}^{n}\lambda_i(C+D)\lambda_i(P) \quad (3.8)$$

which together with (3.7) gives

$$Tr(D) + 2(W - W_c) \leqslant \sum_{i=1}^{n}\lambda_i(C+D)\lambda_i(P)$$

But from (3.3) we have $\lambda_1(P) = \lambda_2(P) = n/2$ while $\lambda_i(P)=0 \; \forall i=3,4,\ldots,n$. Hence we conclude that

$$Tr(D) + 2(W - W_c) \leqslant \frac{n}{2}(\lambda_1(C+D) + \lambda_2(C+D))$$

which implies (3.2) by rearranging the above inequality and replacing W_c by $cost(V_1,V_2)$. □

The above theorem provides us with a lower bound for the cost of any bisection of a graph and this lower bound given by (3.2) is a function of the choice of the diagonal matrix D. If we take D to be the $n \times n$ zero matrix we get

$$cost(V_1,V_2) \geqslant W - \frac{n}{4}(\lambda_1(C) + \lambda_2(C)) \quad (3.9)$$

while if we take D to be any real $n \times n$ diagonal matrix with $Tr(D) = -2W$ we get the bound derived by Donath and Hoffman [11]

$$cost\,(V_1,V_2) \geqslant -\frac{n}{4}(\lambda_1(C+D)+\lambda_2(C+D)) \tag{3.10}$$

In most practical situations, the edges of a graph have non-negative edge weights, i.e, $c_{ij} \geqslant 0 \ \forall i,j \in V$. Therefore, the cost of any bisection of the graph is always non-negative. Hence, 0 is a trivial lower bound for the optimal cost in this case. However, there are graphs with non-negative edge weights for which the lower bound of (3.9) (zero diagonals) is strictly negative. For example, consider a graph which is a simple path on n vertices with unit edge weights. Hence, $W = n-1$. In this case, it can be shown that $\lambda_1(C) = 2\cos\left(\frac{\pi}{n+1}\right)$, and $\lambda_2(C) = 2\cos\left(\frac{2\pi}{n+1}\right)$ [12]. Therefore, the lower bound of (3.9) becomes

$$n-1-\frac{n}{2}cos\left(\frac{\pi}{n+1}\right)-\frac{n}{2}cos\left(\frac{2\pi}{n+1}\right) \tag{3.11}$$

which can be shown to be strictly negative for $n > 10$. Furthermore as $n \to \infty$ the lower bound of (3.11) approaches -1, while the optimal cost in this case is 1 for any n.

The above discussion motivates the need to seek a lower bound which is at least guaranteed to be non-negative for graphs with non-negative edge weights. The next theorem provides such a bound. We say that a real matrix is *non-negative* if each of its entries is non-negative.

Theorem 3.2 : Given a graph $G(V,E)$ on n vertices with non-negative edge weights and connection matrix C. Let $\rho_i = \sum_{j=1}^{n} c_{ij}$ denote the sum of the entries in the $i-th$ row of C. Let $\rho_{\max} = \max_{1 \leqslant i \leqslant n} (\rho_i)$ be the maximum row sum. Let R be the $n \times n$ diagonal matrix with entries defined as

$$r_{ii} = \rho_{\max} - \rho_i \ \forall i = 1,2, \ldots, n.$$

Then for any bisection (V_1,V_2) of G, we have

$$cost\,(V_1,V_2) \geqslant \frac{n}{4}(\rho_{\max} - \lambda_2(C + R)) \geqslant 0 \tag{3.12}$$

and the second inequality above is strict if G is a connected graph.

Proof: Since R is a diagonal matrix, we have from Theorem 3.1

$$cost\,(V_1,V_2) \geqslant W + \frac{1}{2}Tr\,(R) - \frac{n}{4}(\lambda_1(C+R)+\lambda_2(C+R)) \tag{3.13}$$

where W is the sum of all the edge-weights in the graph G. We first note that $W = \frac{1}{2}\sum_{i=1}^{n} \rho_i$ and $Tr\,(R) = \sum_{i=1}^{n}(\rho_{\max} - \rho_i)$. Therefore

$$W + \frac{1}{2}Tr\,(R) = \frac{n}{2}\rho_{\max}. \tag{3.14}$$

By definition, C and R are non-negative matrices and hence $C+R$ is also non-negative. Furthermore, $\rho_{\max}$ is an eigenvalue of the matrix $C + R$ with eigenvector $(1, 1, ..., 1)^T$. In fact, it can be shown that $\rho_{\max}$ is the largest eigenvalue of $C + R$, i.e., $\rho_{\max} = \lambda_1(C + R)$, by using the Gerschgorin circle theorem [13]. Hence $\lambda_2(C+R) \leqslant \rho_{\max}$. Combining these facts with (3.13) and (3.14) we get (3.12).

If G is connected, then $C+R$ is an *irreducible* matrix. It therefore follows from the Perron-Frobenius theory for non-negative irreducible matrices [21] that $\rho_{\max} = \lambda_1(C+R) > \lambda_2(C+R)$. Hence, the bound in (3.12) is strictly positive in this case and truly reflects the fact that for a connected graph G with non-negative edge weights the cost of any bisection is strictly positive. □

We now go back to the lower bound as given by Theorem 3.1 and consider the problem of finding the diagonal matrix D that *maximizes* the lower bound. This would then result in the best possible lower bound of this kind. To this end we introduce the following definitions and notation.

Given a $n \times n$ real symmetric matrix C let $W = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}$ denote half the sum of all entries of C. Let $\mathbf{R}^n$ denote the n-dimensional Euclidean space. A *vector* $\mathbf{x} \in \mathbf{R}^n$ is an n-tuple of real numbers with x_i denoting the i-th component of the vector. Given any vector $\mathbf{x} \in \mathbf{R}^n$, let D_x denote the $n \times n$ diagonal matrix with the ii-th diagonal entry $= x_i$. Also define

$$\mu_1(\mathbf{x}) = \lambda_1(C + D_x) = \textit{largest eigenvalue of } C + D_x \tag{3.15a}$$

$$\mu_2(\mathbf{x}) = \lambda_2(C + D_x) = \textit{second largest eigenvalue of } C + D_x \tag{3.15b}$$

For a fixed matrix C let us define an objective function $f : \mathbf{R}^n \rightarrow \mathbf{R}$ as

$$f(\mathbf{x}) = W + \frac{1}{2}\sum_{i=1}^{n} x_i - \frac{n}{4}(\mu_1(\mathbf{x}) + \mu_2(\mathbf{x})) \tag{3.16}$$

Notice that $f(\mathbf{x})$ defined above is precisely the right hand side of inequality (3.2) and is therefore a lower bound for the cost of any bisection of a graph with connection matrix C. This suggests an interesting duality between the original graph bisectioning problem (GB) which is a combinatorial optimization problem and a nonlinear programming problem (NPP) in the following sense. Define Γ to be the set of all possible bisections of a graph on n vertices with n even. Then for any bisection $(V_1,V_2)\in\Gamma$ and any vector $x\in R^n$ we have

$$cost\,(V_1,V_2) \geqslant f\,(x) \tag{3.17}$$

where *cost* is defined by (1.1) and f defined by (3.16). Therefore,

$$\min_{(V_1,V_2)\in\Gamma} cost\,(V_1,V_2) \geqslant \max_{x\in R^n} f\,(x) \tag{3.18}$$

which exhibits the duality between the two optimization problems.

We now examine the behavior if the function f defined in (3.16) in an attempt to numerically solve the the nonlinear programming problem of maximizing $f\,(x)$ over $x\in R^n$. To this end we need some intermediate results.

Lemma 3.1 : If M is any $n\times n$ real symmetric matrix with eigenvalues $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n$ then

$$\lambda_1 + \lambda_2 = \max_{(x,y)\in\Omega} x^T M x + y^T M y \tag{3.19}$$

where $\Omega = \{(x,y): x\in R^n, y\in R^n, x^Tx=1, y^Ty=1, x^Ty=0\}$ denotes the set of all pairs of *orthonormal* vectors in R^n.

Proof : Let $u_1,u_2,\ldots,u_n$ denote the orthonormal set of eigenvectors of the matrix M such that $M u_i = \lambda_i u_i$ for each $i=1,2,\ldots,n$. Let (x,y) denote an arbitrary pair of orthonormal vectors in Ω. Write $x=\sum_{i=1}^{n}\alpha_i u_i$ and $y=\sum_{i=1}^{n}\beta_i u_i$, where, $\alpha_i = x^T u_i$ and $\beta_i = y^T u_i$. Clearly, $\sum_{i=1}^{n}\alpha_i^2 = 1$, $\sum_{i=1}^{n}\beta_i^2 = 1$, and $\sum_{i=1}^{n}\alpha_i\beta_i = 0$. Also, since x and y are orthonormal, we have $\alpha_i^2+\beta_i^2 \leqslant u_i^T u_i = 1$. Hence,

$$\mathbf{x}^T M \mathbf{x} + \mathbf{y}^T M \mathbf{y} = \sum_{i=1}^{n} \lambda_i (\alpha_i^2 + \beta_i^2)$$

$$\leqslant \lambda_1(\alpha_1^2+\beta_1^2) + \lambda_2 \sum_{i=2}^{n} (\alpha_i^2+\beta_i^2)$$

$$= \lambda_1(\alpha_1^2+\beta_1^2) + \lambda_2(2 - \alpha_1^2-\beta_1^2)$$

$$= \lambda_1(\alpha_1^2+\beta_1^2) + \lambda_2(1 - \alpha_1^2-\beta_1^2) + \lambda_2$$

$$\leqslant \lambda_1+\lambda_2.$$

We have thus shown that $\lambda_1+\lambda_2 \geqslant \mathbf{x}^T M \mathbf{x} + \mathbf{y}^T M \mathbf{y}$ for any pair of orthonormal vectors. But equality holds by setting $\mathbf{x} = \mathbf{u}_1$ and $\mathbf{y} = \mathbf{u}_2$. This establishes (3.19). □

Lemma 3.2 : Given any two $n \times n$ real symmetric matrices A and B. Then

$$\lambda_1(A+B)+\lambda_2(A+B) \leqslant \lambda_1(A)+\lambda_2(A)+\lambda_1(B)+\lambda_2(B). \tag{3.20}$$

Proof : Follows from a more general result II.4.4.14 on page 120 in [15] presented without proof. We will present here a simple proof of (3.20) having established Lemma 3.1 above. Note that

$$\lambda_1(A+B) + \lambda_2(A+B) = \max_{(\mathbf{x},\mathbf{y})\in\Omega} (\mathbf{x}^T(A+B)\mathbf{x} + \mathbf{y}^T(A+B)\mathbf{y})$$

$$= \max_{(\mathbf{x},\mathbf{y})\in\Omega} (\mathbf{x}^T A \mathbf{x} + \mathbf{y}^T A \mathbf{y} + \mathbf{x}^T B \mathbf{x} + \mathbf{y}^T B \mathbf{y})$$

$$\leqslant \max_{(\mathbf{x},\mathbf{y})\in\Omega} (\mathbf{x}^T A \mathbf{x} + \mathbf{y}^T A \mathbf{y}) + \max_{(\mathbf{x},\mathbf{y})\in\Omega} (\mathbf{x}^T B \mathbf{x} + \mathbf{y}^T B \mathbf{y})$$

$$= \lambda_1(A) + \lambda_2(A) + \lambda_1(B) + \lambda_2(B)$$

thus establishing (3.20). □

Theorem 3.3 : The function $f\ : \mathbf{R}^n \rightarrow \mathbf{R}$ defined in (3.16) is a *concave* function.

Proof : It clearly suffices to show that $\mu_1(\mathrm{x})+\mu_2(\mathrm{x})$ defined in (3.15) is a *convex* function. To prove this, we observe that given any $0 \leqslant \alpha \leqslant 1$ and any two vectors $\mathrm{x} \in \mathbf{R}^n$ and $\mathrm{y} \in \mathbf{R}^n$, we have

$$\begin{aligned}
&\mu_1(\alpha\mathrm{x}+(1-\alpha)\mathrm{y}) + \mu_2(\alpha\mathrm{x}+(1-\alpha)\mathrm{y}) \\
&= \lambda_1(\alpha(C+D_x) + (1-\alpha)(C+D_y)) + \lambda_2(\alpha(C+D_x) + (1-\alpha)(C+D_y)) \\
&\leqslant \lambda_1(\alpha(C+D_x)) + \lambda_1((1-\alpha)(C+D_y)) + \lambda_2(\alpha(C+D_x)) + \lambda_2((1-\alpha)(C+D_y)) \\
&= \alpha(\mu_1(\mathrm{x}) + \mu_2(\mathrm{x})) + (1-\alpha)(\mu_1(\mathrm{y}) + \mu_2(\mathrm{y}))
\end{aligned}$$

where the inequality above follows from applying Lemma 3.2 to the matrices $\alpha(C+D_x)$ and $(1-\alpha)(C+D_y)$. □

Since we have shown that f defined in (3.16) is concave and is bounded above by (3.17), there is an x^* that maximizes f, and furthermore, if f is differentiable at x^* then $\nabla f\,(x^*)=0$. The following results pertaining to the continuity and differentiability of f are derived using the theory of symmetric perturbations of symmetric operators [22]. We will simply state the relevant results here without proof.

Lemma 3.3 : For any $\mathrm{x} \in \mathbf{R}^n$, $\mu_1(\mathrm{x})$ defined in (3.15a) is a *simple* eigenvalue of the matrix $C+D_x$. Furthermore the function $\mu_1(\mathrm{x})$ is differentiable at each $\mathrm{x} \in \mathbf{R}^n$ and its partial derivative with respect to x_i is given by

$$\frac{\partial \mu_1}{\partial x_i} = u_{i,1}^2 \tag{3.20}$$

where $u_{i,1}$ is the i-th component of the normalized eigenvector corresponding to the largest eigenvalue $\lambda_1(C+D_x)$.

Lemma 3.4 : The function $\mu_2(x)$ defined in (3.15b) is a continuous function. At a given $x \in R^n$, if $\lambda_2(C+D_x)$ is a simple eigenvalue with normalized eigenvector $\mathbf{u}_2$, then $\mu_2(x)$ is differentiable with partial derivative

$$\frac{\partial \mu_2}{\partial x_i} = u_{i,2}^2 \tag{3.21a}$$

where $u_{i,2}$ is the i-th component of $\mathbf{u}_2$. If however, $\lambda_2(C+D_x)$ is a repeated eigenvalue with multiplicity $p>1$ and the corresponding orthonormal set of eigenvectors are $\mathbf{u}_2, \mathbf{u}_3, \ldots, \mathbf{u}_{p+1}$, then the right hand partial derivative of $\mu_2(x)$ with respect to x_i is

$$\frac{\partial \mu_2}{\partial x_i}^{+} = \sum_{j=2}^{p+1} u_{i,j}^2 \tag{3.21b}$$

while the left hand partial derivative is

$$\frac{\partial \mu_2}{\partial x_i}^{-} = 0. \tag{3.21c}$$

The above lemmas allow us to define partial derivatives of the function f defined in (3.16) as follows.

If $\mu_2(x)$ is a simple eigenvalue of the matrix $C+D_x$ then f is differentiable at x and

$$\frac{\partial f}{\partial x_i} = \frac{1}{2} - \frac{n}{4}\left(\frac{\partial \mu_1}{\partial x_i} + \frac{\partial \mu_2}{\partial x_i}\right) \qquad (3.22a)$$

where the partial derivatives of μ_1 and μ_2 are computed from (3.20) and (3.21a) respectively.

If, however, $\mu_2(x)$ is a repeated eigenvalue of the matrix $C + D_x$ then f has both right and left hand partial derivatives with respect to each x_i given by

$$\frac{\partial f}{\partial x_i}^{+} = \frac{1}{2} - \frac{n}{4}\left(\frac{\partial \mu_1}{\partial x_i} + \frac{\partial \mu_2}{\partial x_i}^{+}\right) \qquad (3.22b)$$

and

$$\frac{\partial f}{\partial x_i}^{-} = \frac{1}{2} - \frac{n}{4}\frac{\partial \mu_1}{\partial x_i} \qquad (3.22c)$$

Using the above derivative information we have implemented a computer program that attempts to seek the vector x that maximizes the function f by using the gradient search nonlinear programming technique. This maximum value of f is then the best possible lower bound for the cost of an optimal bisection to the original graph bisectioning problem that is possible from Theorem 3.1. We will not present the details of our algorithm here. We will, however, present some experimental results.

Table 3.1, below, contains a list of 11 graphs. In this table, n denotes the number of vertices in each graph, LB_1 is the lower bound computed from (3.9) (i.e., zero diagonal), LB_2 is the lower bound computed from (3.12) as given by Theorem 3.2 (i.e., choose diagonals to make row-sums equal). The fifth column lists the best lower bound LB_{best} obtained by maximizing the function f defined in (3.16) using a numerical non-linear programming algorithm. The optimum cost C_{opt} computed by a brute force method is shown in the last column for comparison.

The results of this table indicate that, first, there is a so-called *duality gap* between minimizing cost of a bisection and maximizing the function $\bar{f}$. Except for graph G1, the best lower bound was always strictly less than the optimum cost. Also, for graphs with a small number of vertices, the lower bound LB_1 is greater (and hence better) than LB_2. However, for larger graphs, the situation is reversed, i.e., LB_2 is better than LB_1. In fact, for the two graphs G10 and G11 on 20 vertices, the lower bound LB_1 was negative, while LB_2 was fairly close to the best bound LB_{best}. There is no definitive statement that can be made about which among LB_1 or LB_2 is better. We have some examples of graphs on 100 vertices for which $LB_1 > LB_2$, and others for which the opposite is true, i.e., $LB_2 > LB_1$. Furthermore, computing LB_{best} is computationally very expensive for large

Table 3.1 : A Comparison of Lower Bounds.

Graph	n	LB_1	LB_2	LB_{best}	C_{opt}
G1	4	22.24	20.00	24.00	24.00
G2	4	9.17	8.38	9.17	10.00
G3	6	8.90	4.76	10.12	12.00
G4	6	7.33	6.88	9.64	13.30
G5	6	12.11	8.61	13.68	20.86
G6	6	23.04	17.08	24.49	26.54
G7	8	11.22	9.65	11.24	13.00
G8	10	15.73	18.21	22.33	30.11
G9	16	17.23	22.22	24.66	28.00
G10	20	-7.26	5.22	7.75	14.00
G11	20	-4.00	1.10	1.56	9.00

graphs since it involves repeated eigenvalue/eigenvector decompositions of large matrices. Hence, from a practical standpoint, it is best to compute both LB_1 and LB_2 (each involves computing only the two largest eigenvalues of a $n \times n$ matrix) and report the larger of the two as a lower bound.

4. An Upper Bound

In this section we will derive an upper bound on the cost of a 1*−optimal* bisection for the case of graph bisectioning. This will clearly be an upper bound on the cost of an optimal bisection. We will then see that this bound is in fact achieved for a complete† graph with unit edge-weights.

Lemma 4.1 : Let (V_1,V_2) be any bisection of a graph with a $n \times n$ connection matrix C. For any $i \in V_1$ and for any $j \in V_2$, define

$$g_{ij} = D_i + D_j + 2c_{ij} \tag{4.1}$$

where D_i is defined in (2.2). Then

$$cost\,(V_1 + j - i\ ,\, V_2 + i - j\,) - cost\,(V_1\, ,\, V_2) = g_{ij}$$

Proof : Let i be any vertex in V_1, and let j be any vertex in V_2. Let $P_1 = V_1 + j - i$, and $P_2 = V_2 + i - j$. Also let $X = V_1 - i$ and $Y = V_2 - j$. Then the cost of the bisection (V_1,V_2) can be written as

† A graph with unit edge-weights is complete if it has an edge between every pair of vertices

$$cost(V_1,V_2) = \sum_{k \in V_1} \sum_{l \in V_2} c_{kl} = c_{ij} + (E_i - c_{ij}) + (E_j - c_{ij}) + \sum_{l \in X} \sum_{k \in Y} c_{kl}$$

and the cost of the bisection (P_1,P_2) can be written as

$$cost(P_1,P_2) = \sum_{k \in P_1} \sum_{l \in P_2} c_{kl} = c_{ij} + I_i + I_j + \sum_{l \in X} \sum_{k \in Y} c_{kl}$$

where E_i and I_i are defined in (2.2). From the above two equations we get

$$cost(P_1,P_2) - cost(V_1,V_2) = c_{ij} - E_i + c_{ij} - E_j + I_i + I_j = D_i + D_j + 2c_{ij} = g_{ij}$$

which establishes the proof. □

One can think of g_{ij} as the increase in cost that results from exchanging vertex i with vertex j. It follows immediately from Lemma 4.1 that if a bisection (V_1,V_2) is 1-*optimal* then

$$g_{ij} \geqslant 0 \;\; \forall i \in V_1, \forall j \in V_2. \tag{4.2}$$

This inequality will be used in the next theorem.

Theorem 4.1: Let $C = (c_{ij})$ be the $n \times n$ connection matrix of a graph $G(V,E)$. Let $W = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}$ denote the sum of weights of all edges in E. If bisection (V_1,V_2) is 1-*optimal*, then

$$cost(V_1,V_2) \leqslant \frac{nW}{2(n-1)}. \tag{4.3}$$

Proof: Since (V_1,V_2) is 1-*optimal* we must have, from (4.2),

$$\sum_{i \in V_1} \sum_{j \in V_2} g_{ij} \geqslant 0 \tag{4.4}$$

where g_{ij} is defined in (4.1). But by definition, and using (2.2), we have

$$g_{ij} = D_i + D_j + 2c_{ij} = I_i - E_i + I_j - E_j + 2c_{ij}$$

Substituting the above equation in (4.3) we get

$$\sum_{i \in V_1} \sum_{j \in V_2} \left(I_i - E_i + I_j - E_j + 2c_{ij} \right) \geq 0 \tag{4.5}$$

Let $y = cost\,(V_1,V_2) = \sum_{i \in V_1} \sum_{j \in V_2} c_{ij}$ denote the cost of the bisection (V_1,V_2) for notational convenience. Using (2.2) we have

$$\sum_{i \in V_1} \sum_{j \in V_2} E_i = \frac{n}{2} \sum_{i \in V_1} E_i = \frac{n}{2} \sum_{i \in V_1} \sum_{j \in V_2} c_{ij} = \frac{n}{2} y$$

$$\sum_{i \in V_1} \sum_{j \in V_2} E_j = \frac{n}{2} \sum_{j \in V_2} E_j = \frac{n}{2} \sum_{i \in V_1} \sum_{j \in V_2} c_{ij} = \frac{n}{2} y$$

$$\sum_{i \in V_1} \sum_{j \in V_2} I_i = \frac{n}{2} \sum_{i \in V_1} I_i$$

$$\sum_{i \in V_1} \sum_{j \in V_2} I_j = \frac{n}{2} \sum_{j \in V_2} I_j$$

Substituting these in (4.5), and simplifying, we get

$$(2 - n)y + \frac{n}{2} \left(\sum_{i \in V_1} I_i + \sum_{j \in V_2} I_j \right) \geq 0 \tag{4.6}$$

But using the definitions of W and I_i we can write

$$2W = \sum_{i \in V_1} \sum_{j \in V_1} c_{ij} + 2 \sum_{i \in V_1} \sum_{j \in V_2} c_{ij} + \sum_{i \in V_2} \sum_{j \in V_2} c_{ij} = \sum_{i \in V_1} I_i + 2y + \sum_{j \in V_2} I_j$$

which on re-arranging gives

$$\sum_{i \in V_1} I_i + \sum_{j \in V_2} I_j = 2W - 2y \tag{4.7}$$

Using (4.7) in (4.6) we get

$$(2-n)y + n(W-y) \geqslant 0$$

which can be re-written as

$$y \leqslant \frac{nW}{2(n-1)}$$

thus establishing (4.3). □

To see that $\frac{nW}{2(n-1)}$ is a tight upper bound, consider a complete graph on n vertices with unit edge weights. Clearly, $W = \frac{n(n-1)}{2}$, and consequently the upper bound that we get is $\frac{n^2}{4}$, which is indeed the cost of any possible bisection.

5. Approximation Algorithms for GB

A goal in the design of heuristics for NP-Complete problems is to guarantee that the solutions obtained by such algorithms are fairly close optimal solutions. This leads us to the following notion:

ϵ-approximation : Let y^* be the cost of an optimal solution to some instance of a combinatorial problem. Let $\hat{y}$ be the cost of a solution obtained by an approximation algorithm (i.e., a heuristic) for that instance. For a given $\epsilon > 0$, we say that $\hat{y}$ is an $\epsilon -$ *approximation* , if

$$\frac{|\hat{y} - y^*|}{y^*} \leqslant \epsilon .$$

Furthermore, if the above inequality is true for every problem instance, then we say that the heuristic is an ϵ*-approximation algorithm.*

In this section we will show that any algorithm that always finds 1—*optimal* bisection for the graph bisectioning problem is an ϵ-approximation algorithm if the graph is *dense* enough. To this end, we need to introduce the notion of the density of an edge-weighted graph.

Definition 5.1 : Let $G(V,E)$ be an edge-weighted graph on $n \geq 2$ vertices with connection matrix C. Suppose that all the edge-weights in the graph are *non-negative* (i.e., $c_{ij} \geq 0$) and that the graph has no self-loops (i.e., $c_{ii}=0$). Let $W = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}$ denote the sum of all the edge-weights, and let $M = \max_{1 \leq i \leq j \leq n} (c_{ij})$ denote the maximum edge-weight in the graph†. The *density* of the graph G is then defined to be

$$\gamma = \frac{2W}{n(n-1)M} \tag{5.1}$$

It must be noted that $0 < \gamma \leq 1$ for any non-empty graph. If $\gamma = 1$, then we say that the graph is complete or full.

Theorem 5.1 : Let (P_1,P_2) be an *optimal bisection* of a graph $G(V,E)$ on n vertices with connection matrix C. Let $y^* = cost(P_1,P_2)$ denote the minimum cost. If $n \geq 2$ and if there exists some $\epsilon > 0$ such that the density of the graph $\gamma \geq \frac{1+\epsilon}{1+2\epsilon}$, then

† Note that $M > 0$ unless the graph is empty.

$$\frac{|y - y^*|}{y^*} \leqslant \epsilon \tag{5.2}$$

where y is the cost of *any* 1-optimal bisection.

Proof: We first note that $n \geqslant 2$, $\epsilon > 0$, and (5.1) allow us to write

$$2W = n(n-1)M\gamma > n(n-2)M\gamma \geqslant n(n-2)M\frac{1+\epsilon}{1+2\epsilon} \geqslant \frac{n(n-2)}{2}M,$$

and hence

$$4W - n(n-2)M > 0 \tag{5.3}$$

We also observe that by definition

$$2W = \sum_{i \in P_1}\sum_{j \in P_1} c_{ij} + 2\sum_{i \in P_1}\sum_{j \in P_2} c_{ij} + \sum_{i \in P_2}\sum_{j \in P_2} c_{ij}$$

But

$$\sum_{i \in P_1}\sum_{j \in P_1} c_{ij} \leqslant \frac{n}{2}(\frac{n}{2} - 1)M$$

$$\sum_{i \in P_1}\sum_{j \in P_2} c_{ij} = y^*$$

and

$$\sum_{i \in P_2}\sum_{j \in P_2} c_{ij} \leqslant \frac{n}{2}(\frac{n}{2} - 1)M$$

Therefore, $2W \leqslant \frac{n(n-2)}{2}M + 2y^*$ which together with (5.3) gives

$$0 < \frac{4W - n(n-2)M}{4} \leqslant y^* \tag{5.4}$$

But, $y^* \leqslant y$ by definition, and $y \leqslant \frac{nW}{2(n-1)}$ by Theorem 4.1, since y is the cost of some

1-optimal bisection. Therefore, we have

$$0 < \frac{4W - n(n-2)M}{4} \leqslant y^{*} \leqslant y \leqslant \frac{nW}{2(n-1)}$$

which leads to

$$\frac{|y - y^{*}|}{y^{*}} \leqslant \frac{2nW}{(n-1)(4W - n(n-2)M)} - 1 \tag{5.5}$$

We would now like to bound the right hand side of (5.5) by ϵ. To this end we use (5.1) and the hypothesis of the theorem to get

$$2W = n(n-1)M\gamma \geqslant n(n-1)M\frac{1+\epsilon}{1+2\epsilon} \tag{5.6}$$

But $n \geqslant 2$ and $\epsilon > 0$ implies

$$\frac{(n-1)}{1+2\epsilon} \geqslant \frac{(n-2)}{\frac{(n-2)}{(n-1)}+2\epsilon}$$

which on substituting in (5.6) yields

$$2W \geqslant \frac{n(n-2)(1+\epsilon)M}{2\epsilon+\frac{(n-2)}{(n-1)}} \tag{5.7}$$

$$\Longrightarrow 2W\left(2\epsilon + \frac{(n-2)}{(n-1)}\right) \geqslant n(n-2)(1+\epsilon)M$$

$$\Longrightarrow 2W(2(n-1)\epsilon + (n-2)) \geqslant n(n-1)(n-2)(1+\epsilon)M$$

$$\Longrightarrow 2W\ (2(n-1)(1+\epsilon) - n) \geqslant n(n-1)(n-2)(1+\epsilon)M$$

$$\Longrightarrow (1+\epsilon)(n-1)(4W - n(n-2)M) \geqslant 2nW$$

Using (5.3), this last inequality implies

$$\frac{2nW}{(n-1)(4W - n(n-2)M)} - 1 \leqslant \epsilon \tag{5.8}$$

From (5.5) and (5.8) we get $\dfrac{|y - y^*|}{y^*} \leqslant \epsilon$. □

Theorem 5.1 essentially states that any algorithm that always finds 1*—optimal* bisections is a good algorithm if the input graph is dense enough. The approximation gets better as the density increases.

We now discuss a special case of graphs with *unit edge-weights*. Consider a graph $G(V,E)$ on n vertices that has unit edge weights. In this case $M=1$ and the total edge-weight $W=|E|$, the number of edges in the graph. Also, $\dfrac{n(n-1)}{2}$ is the number of edges in a complete graph on n vertices, and the density $\gamma = \dfrac{2|E|}{n(n-1)}$ is simply the ratio of the number of edges in the graph to the number of edges in a complete graph. The following corollary is a direct consequence of Theorem 5.1 for graphs with unit edge-weights.

Corollary 5.1 : Consider a graph $G(V,E)$ on n vertices with unit edge weights. If

$$|E| \geqslant \frac{n(n-1)}{2} \frac{1+\epsilon}{1+2\epsilon}$$

for some $\epsilon > 0$, then

$$\frac{|y - y^*|}{y^*} \leqslant \epsilon,$$

where $\bar{y}$ is the cost of *any* 1-optimal bisection and y^* is the optimal cost.

For example, if a graph has more than 90% of the edges of a complete graph, then according to the above corollary, the cost of any 1-optimal bisection will be within 12.5%

of the optimal cost. Similarly, in order to guarantee that the cost of any 1-optimal bisection be within 20% of the optimal cost, the graph should contain at least 86% of the maximum possible edges. However, for the cost of any 1-optimal bisection to be guaranteed to be at most twice the optimal cost, the graph should contain at least 66.6% of the maximum possible edges. The Kernighan-Lin algorithm [3] is an example of an algorithm that guarantees the 1*–optimality* of the bisection it finds.

We will now define a class of graphs with unit edge weights for which the ratio of the upper bound $UB = \dfrac{nW}{2(n-1)}$ and the lower bound $LB_1 = W - \dfrac{n}{4}(\lambda_1(C) + \lambda_2(C))$ for GB approaches 1 as the number of vertices approaches ∞. Therefore, for an instance of GB belonging to this class, an algorithm that guarantees 1*–optimality* will find bisections whose costs get closer to an optimal cost as the number of vertices in the graph gets larger.

Definition 5.2 : Given undirected graphs $G_1(V_1,E_1)$ and $G_2(V_2,E_2)$, the *join of* G_1 and G_2 is the graph whose vertex set is $V = V_1 \cup V_2$ and whose edge set is $E = E_1 \cup E_2 \cup E_3$, where $E_3 = \{ \{i,j\} : i \in V_1, j \in V_2\}$. We denote the join of G_1 and G_2 by $G_1 + G_2$.

Informally, $G_1 + G_2$ is the graph obtained by joining every vertex of G_1 to every vertex of G_2 by an edge. In the remainder of this section, it is assumed that the graphs under consideration have unit edge weights. In this case the connection matrix is simply the adjacency matrix of the graph. A k*–regular* graph is a graph in which every vertex has exactly k edges incident on it. Let $\phi(G,\lambda)$ denote the characteristic polynomial of the connection (adjacency) matrix C of G, i.e., $\phi(G,\lambda) = \det(\lambda I - C)$.

Lemma 5.1 : Let G be an undirected graph having unit edge-weights with connection (adjacency) matrix C.

(1) If G is connected, then the largest eigenvalue of C has multiplicity 1.

(2) If G is k-regular, then the largest eigenvalue of C is k.

Lemma 5.2: If G_1 and G_2 are two undirected and k-regular graphs on n vertices each, then $\phi(G_1 + G_2, \lambda) = \dfrac{\phi(G_1,\lambda)}{(\lambda - k)} \dfrac{\phi(G_2,\lambda)}{(\lambda - k)} (\lambda - k - n)(\lambda - k + n)$.

Lemmas 5.1 and 5.2 are easy consequences of theorems proved in [15] and [12], respectively.

Lemma 5.3 : If G_1 and G_2 are two connected, undirected, k-regular graphs on $n > 2$ vertices each, then the largest eigenvalue of the connection matrix of $G_1 + G_2$ is $k + n$ and the second largest eigenvalue is strictly less than k.

Proof: By Lemma 5.2, the eigenvalues of the connection matrix of $G_1 + G_2$ are the roots of $\dfrac{\phi(G_1,\lambda)}{(\lambda - k)} \dfrac{\phi(G_2,\lambda)}{(\lambda - k)} (\lambda - k - n)(\lambda - k + n)$. But the roots of $\dfrac{\phi(G_1,\lambda)}{(\lambda - k)}$ and $\dfrac{\phi(G_2,\lambda)}{(\lambda - k)}$ are strictly less than k by Lemma 5.1, and the other roots are $k + n$ and $k - n$. □

Theorem 5.2 : Let G_1 and G_2 be any two connected, undirected, and k-regular graphs on $n > 2$ vertices each and $G = G_1 + G_2$ be the join of G_1 and G_2†. Then

$$\lim_{n \to \infty} \frac{UB}{LB_1} = 1.$$

where UB is the upper bound on the cost of an optimal bisection given by (4.3) and LB_1

† Note that $2n$ is the number of vertices in $G_1 + G_2$

is the lower bound given by (3.9).

Proof: Note that G is $(k+n)$–regular graph on $2n$ vertices having unit edge-weights. Therefore, $W = n(n+k)$, and consequently

$$UB = \frac{n^2(n+k)}{(2n-1)} \tag{5.9}$$

To compute the lower bound, by Lemma 5.3, we have $\lambda_1(C) = k+n$ and $\lambda_2(C) < k$. It follows from (3.9) that

$$LB_1 = n(n+k) - \frac{n}{2}(n+k-\lambda_2(C)) > \frac{n^2}{2} > 0 \tag{5.10}$$

Hence, we have

$$1 \leqslant \frac{UB}{LB_1} < \frac{n+k}{n-0.5}$$

from which it is clear that

$$\lim_{n \to \infty} \frac{UB}{LB_1} = 1$$

which establishes the required result. □

6. Conclusions

In this paper we presented several conditions for optimality of a given bisection of an edge-weighted undirected graph. Based on empirical results we concluded that one of the sufficient conditions developed was satisfied by almost all dense graphs and a few sparse graphs. Another sufficient condition that is computationally easier to check was, however, never satisfied for large graphs. In Section 3, we derived lower bounds for the cost of any bisection of a graph. This lower bound was shown to be a concave function of a diagonal matrix, thus establishing a duality between the graph bisectioning problem and a non-linear programming problem. Furthermore, we were able to construct a diagonal matrix that always produces a non-negative lower bound for graphs with non-negative edge-weights. In Section 4, we derived an upper bound on the cost of any 1-optimal bisection. In Section 5, we introduced the notion of an ϵ-approximation algorithm and showed that for dense graphs, a bisection that approximates an optimal one can be easily found by using any heuristic, such as the well-known Kernighan-Lin, that guarantees the 1-optimality of its bisections. Finally, we exhibited a class of graphs for which the ratio of the upper and lower bounds approaches 1 as the number of vertices in the graph increases.

REFERENCES

[1] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*, North-Holland, New York, 1976.

[2] Y. Perl and M. Snir, "Circuit Partitioning with Size and Connection Constraints," *Networks*, vol. 13, pp. 365-375, 1983.

[3] B.W. Kernighan and S. Lin, "An Efficient Heuristic Procedure for Partitioning Graphs," *Bell System Technical Journal*. vol. 49, pp. 291-307, February 1970,

[4] Earl E. Barnes, " An Algorithm for Partitioning the Nodes of a Graph," *IBM Technical Report RC8690*, 1981.

[5] Roy L. Russo, Peter H. Oden, and Peter K. Wolff, Sr, "A Heuristic Procedure for the Partitioning and Mapping of Computer Logic Graphs," *IEEE Transactions on Computers*, vol. C-20, no. 12, pp. 1455-1462, December 1971.

[6] T. Bui, S. Chaudhuri, T. Leighton, and M. Sipser, " Graph Bisection Algorithm with Good Average Case Behavior," *Proceedings of the 25th IEEE Symposium on Foundations of Computing*, pp. 181-192, 1984.

[7] C.M. Fiduccia and R.M. Mattheyses, "A Linear-Time Heuristics for Improving Network Partitions," *Proceedings of the 19th Design Automation Conference*, pp. 175-181, January 1982.

[8] J. Frankle and R.M. Karp, "A Graph Partitioning Algorithm Based on Spectral Analysis," *Proceedings of SRC Tropical Research Conference*, University of California, Berkeley, February 1985.

[9] K.S. Arun and V.B. Rao, "A New Principle-Components Algorithm for Graph Partitioning with Application in VLSI Placement Problems," *Proceedings of the 20th Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, California, November 1986.

[10] A. J. Hoffman and H. W. Wielandt, "The Variation of the Spectrum of a Normal Matrix," *Duke Mathematics Journal*, vol. 20, pp. 37-39, 1953.

[11] W. E. Donath and A. J. Hoffman, "Lower Bounds for the Partitioning of Graphs," *IBM Journal of Research and Development*, pp. 420-425, September 1973.

[12] A. J. Schwenk, "Computing the Characteristic Polynomial of a Graph," in R. A. Bari and F. Harary(Ed.) *Graphs and Combinatorics, Lecture Notes in Mathematics 406*, New York, pp. 247-261, 1974.

[13] G. Strang, *Linear Algebra and Its Applications*, Academic Press, New York, pp. 304, 1980.

[14] J. Savage, *The Complexity of Computing*, John Wiley and Sons, New York, 1976.

[15] M. Marcus and H. Minc, *A Survey of Matrix Theory and Matrix Inequalities*, Allyn and Bacon, Boston, 1964.

[16] A.E. Dunlop and B.W. Kernighan, "A Procedure for Placement of Standard-Cell VLSI Circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. CAD-4, no. 1, January 1985.

[17] David P. La Potin and Stephen W. Director, "Mason: A Global Floorplanning Approach for VLSI Design," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. CAD-5, no. 4, pp. 477-489, October 1986.

[18] M.A. Breuer, "Min-Cut Placement," *Journal of Design Automation and Fault-Tolerant Computing*, vol. 1, no. 4, pp. 343-362, October 1977.

[19] S. N. Bhatt and F. T. Leighton, "A Framework For Solving VLSI Graph Problems," *Journal of Computer and System Sciences*, vol. 28, pp. 300-343, 1984.

[20] Youssef Saab and Vasant Rao, "Fast Effective Heuristics For the Graph Bisectioning problem," *manuscript under preparation*.

[21] E. Seneta, *Non-Negative Matrices*, George Allen & Unwin Ltd, London, pp. 1, 1973.

[22] T. Kato, *A Short Introduction to Perturbation Theory for Linear Operators*, Computer Science Press, Inc, Rockville, Maryland, 1979.

[23] S. Even, *Graph Algorithms*, Springer-Verlag, New York, 1982.