ON BISECTING RANDOM GRAPHS

THANG NGUYEN BUI

February 1983

This research was supported in part by the Advanced Research Projects Agency under Contract N00014-80-C-0622.

© Massachusetts Institute of Technology 1983

Massachusetts Institute of Technology
Laboratory for Computer Science
Cambridge Massachusetts, 02139

This empty page was substituted for a blank page in the original document.

ON BISECTING RANDOM GRAPHS

by

Thang Nguyen Bui

Submitted to the Department of Electrical Engineering and Computer Science
on January 13, 1983 in partial fulfillment of the
requirements for the Degree of Master of Science in
Electrical Engineering

ABSTRACT

A bisection of a graph with an even number of vertices is a partition of the vertex set into two disjoint sets of equal size. Given a bisection, the number of edges having one end in each of the two subsets of the bisection is called the size of the bisection. The bisection size of a graph is the minimum size of all possible bisections of the graph. Given a graph with an even number of vertices and a positive integer, the graph bisection problem is the problem of determining if the bisection size of the graph is less than the given number. The graph bisection problem is known to be NP-hard.

In this thesis, we give probabilistic lower bounds and upper bounds for the bisection size of random graphs, graphs in which an edge appears between any two vertices with a certain fixed probability, say p, independent of all other edges. In particular, we show that, with probability 1, the bisection size of random graphs on $2n$ vertices is greater than or equal to $n^2p - O\left(n^{3/2}\sqrt{p(1-p)}\right)$ and is less than $n^2p - O\left(n\sqrt{p(1-p)}\right)$. Upper bound and lower bound on the bisection size are given in the case p is a function of n, specifically when $p = p(n) = \frac{c}{n}$. We also consider some heuristics for solving the graph bisection problem.

Thesis Supervisor : Ronald L. Rivest

Title : Associate Professor of Electrical Engineering and Computer Science

Keywords : Random graphs, bisection size, probabilistic bounds.

This empty page was substituted for a blank page in the original document.

ACKNOWLEDGMENTS

First, I would like to thank my advisor, Professor Ronald L. Rivest, for his guidance, helpful suggestions, and constant encouragement throughout the writing of this thesis. I also would like to thank Sandeep Bhatt, Professors Tom Leighton, Charles Leiserson, and Mike Sipser for listening to my problem with interest and offering helpful suggestions. I thank Professor George Lueker for providing me with a copy of his paper and some useful discussions, and Professor Joel Spencer for giving me some good insights to the method of second moment.

I am grateful to my parents and my brothers for their love, support, and encouragement.

This empty page was substituted for a blank page in the original document.

TABLE OF CONTENTS

Abstract 3

Acknowledgments 5

Table of Contents 7

Chapter 1. Introduction 9

Chapter 2. The Graph Bisection Problem 13

2.1. Some Graph Theoretic Definitions 13

2.2. NP-completeness of the Graph Bisection Problem 15

2.3. Known Lower Bounds on the Bisection Size of Graphs 17

2.4. Known Upper Bounds on the Bisection Size of Graphs 19

2.5. Bounds for Graphs with Special Structures 20

Chapter 3. Bounds for Bisection Size of Graphs in $\mathcal{G}(2n, p)$ 23

3.1. Preliminary Results 23

3.2. Lower Bounds on the Bisection Size 26

3.3. Upper Bounds on the Bisection Size 28

Chapter 4. Graph Bisection Heuristics 31

4.1. Descriptions of Some Heuristics 31

4.2. Some Empirical Results 37

Chapter 5. Conclusion 41

References 43

This empty page was substituted for a blank page in the original document.

Chapter 1

Introduction

In this thesis, we study the graph bisection problem. We shall consider only undirected, simple graphs having unit cost on all of its edges, and an even number of vertices. We consider the problem of partitioning the vertex set into two disjoint sets of equal size which minimize the number of cut edges (edges having one endpoint in each subset of the partition). This problem is called the graph bisection problem[1].

By relaxing some constraints in the graph bisection problem we obtain the graph partitioning problem. Specifically, let an undirected, simple graph with costs on its edges be given. The graph partitioning problem is the problem of partitioning the set of vertices into disjoint subsets, each having cardinality smaller than a given fixed number, so as to minimize the total cost of the edges having ends in different subsets of the partition. Even though the graph bisection problem seems simpler than the graph partitioning problem, it still retains the important feature of the latter. Furthermore, an algorithm for solving the graph bisection problem can be adapted as a heuristic to approximately solve the general graph partitioning problem [KL70].

Aside from being of theoretical interest, the graph partitioning problem serves as an abstraction for various practical problems. An example is the problem of placing components of a circuit on printed circuit boards. The objective here is to perform

[1]For simplicity we use this definition here. The graph bisection problem will be defined more carefully in the next chapter as a decision problem.

the placement so as to minimize the number of connections between components on different boards, for these connections are slower and more expensive than those connecting components on the same board. This problem can be represented as a graph whose vertices depicting the components and the edges representing the connections of the circuit. The maximum cardinality of each subset of the partition corresponds to the maximum number of components each board can have.

Another example arises in the management of a page or segmentation memory structure of a computer system. The problem here is to assign components (e.g. subroutines or procedures) of a large program into different fixed-size pages of the memory so that the number of references between components on different pages of the memory is minimized.

In recent years, the graph partitioning problem has been used as a model for an important problem in the Very Large Scale Integration (VLSI) design process. This problem arises in the placement phase of the design process, in which tens or hundreds of thousands of components must be placed on a wafer subject to certain constraints. With such a large number of components to be arranged, it is desirable to have this process automated. At the present, most models for the placement problem are of the form of a graph partitioning problem [Br77].

It is known that the graph partitioning problem (when phrased as a decision problem) is NP-complete [HR]. That means at the present no algorithm is known to solve the graph partitioning problem in time polynomial in the length of the input (assuming a reasonable encoding of the problem). Nonetheless, there are approximate algorithms, or heuristics to solve this problem. Hence, it is useful to know how good a solution a given heuristic can provide, without having to use empirical methods. It should be noted that if we consider the graph partitioning problem in which each subsets of the partition has size less than or equal to 2 then this is the problem of finding maximum matching, also if the size of each subset is restricted to be less than or equal to $n - 1$ where n is the number of vertices in the graph then this becomes the problem of finding a minimal cut set of the given graph. There exist polynomial-time algorithms for both of these problems.

It can be easily shown that the graph bisection problem (when phrased as a

decision problem) is also NP-complete. There are several heuristics often used for this problem, all of which have the form of a hill-climbing search or some variation of it. The analysis of the performance of a heuristic would be facilitated if we have some information about the objects the heuristic is applied on. In other words, we would like the input to the heuristic to be in some structured class of objects. To this end we choose random graphs as input to the heuristics. By a random graph we mean a graph in which an edge appears between any two vertices with a certain probability, possibly dependent on the number of vertices of the graph, but independent of all other edges. The performance of the heuristic on this class of graphs may give insight to the typical behaviour of the heuristic in practice.

To determine the performance of a heuristic, we must first know what kind of solution we expect to have, that is we must know the size of the optimal cut edge set of a random graph. Then by comparing this with the solution given by the heuristic we can determine how good the heuristic is.

The thesis is divided as follows. In Chapter 2 we review some standard graph theoretic notions, define the graph bisection problem, introduce models for random graphs, and present some known results. In Chapter 3 we give probabilistic lower bound and upper bound for the bisection size of random graphs. We review some known algorithms for solving the graph bisection problem and present some empirical data on the performance of some heuristics on random graphs in Chapter 4. Finally, Chapter 5 will provide a summary and some open questions.

This empty page was substituted for a blank page in the original document.

Chapter 2

The Graph Bisection Problem

In this chapter we review the known results about the bisection size of graphs. To facilitate this purpose and to lay the ground for later work we first present some standard graph notions. We also give the definitions for our models of random graphs. The graph bisection problem is also defined formally.

2.1. Some Graph Theoretic Definitions

An **(undirected) graph** $G = (V, E)$ consists of a set V of **vertices** and a set E of unordered pairs of vertices, called **edges** (we use the notation (a, b) to denote the unordered pair a and b). We consider only simple graphs, that is graphs satisfying the condition that, for all $v \in V, (v, v) \notin E$ and there is at most one edge between any two vertices in V. Let $|A|$ denote the cardinality of the set A. Let $G = (V, E)$ be a graph, for each $v \in V$, the **degree** of v, denoted by $\deg(v)$, is defined as

$$\deg(v) = |\{ w \in V \mid (v, w) \in E \}|. \tag{2-1}$$

Let $A \subseteq V$, the **subgraph of G induced by A** is the graph obtained from G by deleting all vertices in $V - A$ and all edges in E incident to those vertices in $V - A$. We call G a **complete** graph if $|E| = \binom{|V|}{2}$. A sequence of vertices $v_1, \ldots, v_m$ is called a **path** between v_1 and v_m in G if for all $i \in \{1, \ldots, m-1\}, (v_i, v_{i+1}) \in E$. If $v_1 = v_m$ then the path is called a **cycle**. G is said to be **connected** if there is a path between any two vertices in V. A **tree** is a connected graph having no cycle. If we label the vertices in V with the integers in $\{1, 2, \ldots, |V|\}$ so that each vertex has a distinct label, then the **adjacency matrix** $A(G)$ of G, a square matrix of order

$|V|$, whose element in the ith row and the jth column is defined as follows.

$$a_{ij} = \begin{cases} 1, & \text{if } (i,j) \in E; \\ 0, & \text{if } (i,j) \notin E. \end{cases} \tag{2-2}$$

Now suppose $|V| = 2n$ and let $A, B \subseteq V$. The pair (A, B) is called a **bisection** of G if $A \cap B = \emptyset, A \cup B = V$, and $|A| = |B| = n$. Let (A, B) be a bisection of G. For each $a \in A$ define

$$\text{in}_{(A,B)}(a) = \text{degree of } a \text{ in the subgraph of } G \text{ induced by } A. \tag{2-3}$$

$$\text{ex}_{(A,B)}(a) = \text{degree of } a \text{ in the subgraph of} G \text{ induced by } \{a\} \cup B. \tag{2-4}$$

Similar definitions apply for each $b \in B$. We shall omit the subscript (A, B) and just write $\text{in}(a)$ and $\text{ex}(a)$ when there is no confusion. We now define the size of (A, B) written $|(A, B)|$ as

$$|(A,B)| = \sum_{a \in A} \text{ex}(a) \tag{2-5}$$

$$= \sum_{b \in B} \text{ex}(b) \tag{2-6}$$

$$= |E| - \frac{1}{2}\Big(\sum_{a \in A} \text{in}(a) + \sum_{b \in B} \text{in}(b)\Big). \tag{2-7}$$

The **bisection size** of G is defined as

$$\text{C}(G) = \min\{\, |(A,B)| \mid (A,B) \text{ is a bisection of } G \,\}. \tag{2-8}$$

The graph bisection problem can now be stated.

Instance : $G = (V, E)$, $|V| = 2n$, $0 \leq k < |E|$.

Question : Is $\text{C}(G) \leq k$?

In this thesis we shall deal mainly with a special kind of graphs, namely the random graphs. Random graphs were first considered by Erdös [E59], [E61], and were studied in some details by Erdös and Rényi in [ER59], [ER60]. Random graphs have been used in nonconstructive proofs of combinatorial theorems, see, for example, the book by Erdös and Spencer [ES74]. Random graphs have also been studied for their own interest and many results concerning random graphs have been discovered [BES80, Bo81a, Bo81b, Bo82, BE76, GM75, Mat76, P76]. For an extensive bibliography on the literature of random graphs see [Ka82]. The two most

often used models of random graphs are closely related. We start with a fixed set of n distinct (labeled) vertices. In the first model, we choose each of the $\binom{n}{2}$ possible edges with a fixed probability $p, 0 < p < 1$, independent of the choices of all other edges. In the second model, we take all graphs on n vertices and m edges, where m may be dependent on n, and consider them as points of a probability space, having equal probability. In the first model we write $\mathcal{G}(n,p)$ to denote the probability space of all graphs with a fixed set of n labeled vertices and the probability of a graph with m edges is $p^m q^{\binom{n}{2}-m}$, where $q = 1 - p$. In the second model we use $\mathcal{G}(n,m)$ to denote the probability space containing all graphs on n vertices with m edges $0 \leq m \leq \binom{n}{2}$, and each graph in $\mathcal{G}(n,m)$ has equal probability. Another commonly used model of random graphs is the model $\mathcal{G}(n,p)$ with p being a function from $\mathbf{N}$ to $(0,1)$. It is useful to define the following concept which will be needed later. Let Q be a property of graphs in $\mathcal{G}(n,p)$ or $\mathcal{G}(n,m)$. We say **almost every (a.e.)** graph in $\mathcal{G}(n,p)$ has property Q if $\mathbf{Pr}\{\, G_{n,p} \in \mathcal{G}(n,p) \mid G_{n,p} \text{ has } Q \,\} \rightarrow 1$ as $n \rightarrow \infty$. Similar definition applies for graphs in $\mathcal{G}(n,m)$. Unless otherwise indicated, all logarithms in this thesis are natural logarithms.

2.2. NP-completeness of the Graph Bisection Problem

The NP-complete problems are the hardest problems in a class of problems, called NP, that can be solved in polynomial time by nondeterministic Turing machines. The corresponding class of probems which can be solved in polynomial time by deterministic Turing machines is called P. One of the most important open questions in theoretical computer science is whether P is equal to NP. This means that we do not now know any deterministic, polynomial time algorithms for solving an NP-complete problem. (Note that P $\subseteq$ NP and the general consensus is that P is not equal to NP.) For more precise definition of NP-completeness see [GJ79]. The graph bisection problem can be easily shown to be NP-complete by reducing it to the simple max cut problem, which is known to be NP-complete [GJS76]. The simple max cut problem is the following problem.

Instance : Graph $G = (V, E)$, positive integer K.

Question : Is there a partition of V into 2 disjoint sets such that the number of edges having one end in each of the two sets is at least K?

The proof of the following proposition is almost identical to the one given in [GJS76] for proving the NP-completeness of a problem slightly different from the graph bisection problem.

Proposition 2.1. *The graph bisection problem is NP-complete.*

Proof : We reduce the simple max cut problem to the graph bisection problem. Let $G = (V, E)$ and $K \in N$ be an instance of the simple max cut problem. Let $|V| = n$, and $U = \{u_1, \ldots, u_n\}$ be a set of new vertices. We construct an instance for the graph bisection problem as follows. Construct $G' = (V', E')$ such that

$$\begin{aligned} V' &= V \bigcup U, \\ E' &= \{(u, v) \mid u, v \in V', (u, v) \notin E\}, \\ K' &= n^2 - K. \end{aligned}$$

In a sense G' is the complement of G, with the extra vertices.

Claim : G has a max cut of at least size K if and only if G' has a bisection of size less than or equal to K'.

Suppose (A, B) is a partition of G such that $|\{(u, v) \in E \mid u \in A, v \in B\}| \geq K$. Since $K > 0$, $|A|, |B| > 0$. Let $j = n - |A|$. Let

$$\begin{aligned} A' &= A \bigcup \{u_1, \ldots, u_j\}, \\ B' &= V' - A'. \end{aligned}$$

Then (A', B') is a bisection of G'. Also, it is easily seen that

$$\begin{aligned} |(A', B')| &= n^2 - |\{(u, v) \notin E' \mid u \in A', v \in B'\}| \\ &= n^2 - |\{(u, v) \in E \mid u \in A, v \in B\}| \\ &\leq n^2 - K \\ &= K'. \end{aligned}$$

Conversely, if (A', B') is a bisection of G', such that $|(A', B')| \leq n^2 - K = K'$. then let $A = A' \bigcap V$ and $B = B' \bigcap V$, clearly (A, B) is a partition of G satisfying

$$\begin{aligned} |(A, B)| &= |\{(u, v) \in E \mid u \in A, v \in B\}| \\ &= n^2 - |\{(u, v) \notin E' \mid u \in A', v \in B'\}| \\ &\geq n^2 - (n^2 - K) \\ &= K. \end{aligned}$$

This proves the claim and the proposition. ∎

2.3. Known Lower Bounds on the Bisection Size of Graphs

As indicated in the previous section the graph bisection problem is *NP*-complete, this means that at the present an algorithm to determine the bisection size of a graph takes time exponential in the size of the graph (e.g. the number of the vertices). It is, therefore, useful to know a lower bound of the bisection size of a given graph so that one can determine how good is an approximate algorithm for finding the bisection size. The first few results presented below give the lower bounds for bisection size based on the knowledge of the adjacency matrix of the graph. Let $G = (V, E)$ be graph with $|V| = 2n$. Let $A(G)$ be the adjacency matrix of G and let U be an arbitrary diagonal matrix of order $2n$ such that

$$\sum_{i=1}^{2n} u_{ii} = -2\,|E|. \tag{2-9}$$

We denote by λ_1, λ_2 the two largest eigenvalues of the real symmetric matrix $(A + U)$.

Proposition 2.2. [DH73] *Let $G = (V, E)$ be a graph with $|V| = 2n$ and let $A, U, \lambda_1, \lambda_2$ be defined as above, then*

$$C(G) \geq -\frac{n}{2}(\lambda_1 + \lambda_2). \tag{2-10}$$

If we know the maximum degree of the vertices in G then we have the following stronger result than Proposition 2.2.

Proposition 2.3. [DH73] *Let $G = (V, E)$ be a graph with $|V| = 2n$ and let $A, U, \lambda_1, \lambda_2$ be defined as above, and for all $v \in V$, $deg(v) \leq d$ for some fixed constant d. Let δ_1, δ_2 be in the closed interval $[0, \pi/4]$, and let x be the simultaneous solution to the following equations*

$$x \sin 2\delta_1 = (1 - x) \sin 2\delta_2 \tag{2-11}$$

$$-\frac{1}{2}(\delta_1 + \delta_2) = x[1 - \sin 2\delta_1 + 2(d-1)(1 - \cos(\delta_1 + \delta_2))] \tag{2-12}$$

then

$$C(G) \geq xn. \tag{2-13}$$

Given n and m, if we define $\mathrm{C}_{n,m}$ to be the maximal bisection size of all graphs on n vertices and m edges, then we have the following result given by Goldberg and Gardner [GG82].

Proposition 2.4.[GG82] *Let n and m be given. Let s be the largest integer such that $m \geq s(2n-1) - \frac{1}{2}s(s-1)$, and let $r = m - 2(2n-1) + \frac{1}{2}s(s-1)$.*

$$\mathrm{C}_{2n,m} \geq \left[\frac{s}{2}\left(2n - \frac{s}{2}\right)\right] + \max\{0, r - \left[n - \frac{s}{2}\right] - 1\} \qquad (2\text{–}14)$$

where $[a]$ denotes the integer nearest to a.

Proof : We shall construct a graph $G = (V, E)$ with $|V| = 2n$, $|E| = m$ and such that $\mathrm{C}(G)$ equals the right hand side of (2–14). We define G as follows.

$$\begin{aligned} V &= \{x_1, \ldots, x_{2n}\}, \\ E &= \{(x_i, x_j) \mid i = 1, \ldots, s; j = 1, \ldots, n; i \neq j\} \\ &\quad \bigcup \{(x_{s+1}, x_j) \mid j = s+2, \ldots, s+r+1\}. \end{aligned}$$

Thus G has a clique of size s and each vertex of that clique is also connected to all other vertices in V. There is also one vertex outside of the clique, namely x_{s+1}, which is connected to r other vertices not inside the clique. Let (A, B) be an optimal bisection of G. Assume that a is the number of vertices in the clique that appear in A and hence there are $s - a$ vertices of the clique appearing in B. Without loss of generality assume that $a \leq \frac{s}{2}$ then $a \leq s - a$. It is clear that

$$\mathrm{C}(G) = |(A, B)| \leq (s-a)n + a(n - (s-a)) + \max\{0, r - (n-a) - 1\}$$

hence for the optimal bisection we must have $a = \frac{s}{2}$. That is

$$\mathrm{C}(G) = \left[s\left(n - \frac{s}{4}\right)\right] + \max\{0, r - \left[n - \frac{s}{2}\right] - 1\}$$

completing the proof of the proposition. ∎

For random graphs in $\mathcal{G}(2n, m)$ the following lower bound is given in [Mac78].

Proposition 2.5.[Mac78] *Let $s \geq 9$ be fixed. Let $G_{2n,m}$ be a random graph in $\mathcal{G}(2n, m)$, with $m = 2sn$. Then*

$$\Pr\left\{\mathrm{C}(G_{2n,m}) < \left(\frac{1}{2} - \sqrt{\frac{\log 2}{2s}}\right)m\right\} \rightarrow 0 \qquad (2\text{–}15)$$

as $n \rightarrow \infty$.

2.4. Known Upper Bounds on the Bisection Size of Graphs

By counting argument we can easily get the following upper bound on the bisection size of graphs in general.

Proposition 2.6.[GG82] *Let $G = (V, E)$ be a graph with $|V| = 2n, |E| = m$. Then*

$$C(G) \leq \frac{mn}{2n-1}. \tag{2-16}$$

Proof : Since the number of different bisections is $\frac{1}{2}\binom{2n}{n}$, we have

$$\frac{1}{2}\binom{2n}{n}C(G) \leq \sum_{(A,B)} |(A,B)| \tag{2-17}$$

where the sum is over all possible bisections of G. It is easily shown that each edge in G appears $\binom{2n-2}{n-1}$ times in the sum on the right hand side of (2–17). Hence the above inequality becomes

$$\frac{1}{2}\binom{2n}{n}C(G) \leq \binom{2n-2}{n-1}m.$$

Thus

$$C(G) \leq \frac{2\binom{2n-2}{n-1}m}{\binom{2n}{n}} = \frac{mn}{2n-1}. \quad \blacksquare$$

The above result holds for any graph.

In the case of random graphs in $\mathcal{G}(2n, m)$ we have the following upper bound given in [Mac78].

Proposition 2.7. *Choose $s > 0$. Let $G_{2n,m}$ be a random graph in $\mathcal{G}(2n, m)$ where $m = 2sn$. Then with probability 1, $C(G_{2n,m})$ is less than or equal to $m(\frac{1}{2} - H(s))$, as $n \to \infty$, where $H(s) \approx 0.238s^{-1/2}$ as $s \to \infty$. This approximation is also good for s as small as 1.*

This upper bound is established by constructive method. An algorithm is exhibited and its performance is used to derive the upper bound. In practice, this algorithm is found to be inferior to most other well known algorithms for finding bisection size. The problem is that these superior algorithms are too complex to enable an analytical analysis be made on their performances. Therefore, one would expect that even if the random variable denoting the bisection size of random

graphs in $\mathcal{G}(2n, m)$ does not converge, the actual upper bound will be lower than that given in Proposition 2.7. It should be noted that the upper bound given in Proposition 2.7 is only for random graphs in $\mathcal{G}(2n, m)$ with a linear number of edges.

2.5. Bounds for Graphs with Special Structures

In general if the graph possesses some special structure we can produce a tighter bounds on its bisection size. The following results are given in [Mac78] we present them here without proof.

Proposition 2.8. *Let $G = (V, E)$ be a graph $d = \max\{ deg(v) \mid v \in V \} \geq 3$, with $|E| = m$, then*

$$\mathrm{C}(G) \leq \frac{d-1}{2d} m + O(d^2). \tag{2–18}$$

Proposition 2.9. *Let $G = (V, E)$ be a graph with $|V| = 2n, |E| = m$, if $m < n$ then $\mathrm{C}(G) = 0$.*

Proposition 2.10. *Let $s \in (1/2, 3/2)$ be given. Then there exists an $\epsilon > 0$ such that for any n_0 there is $n > n_0$ and a graph $G = (V, E)$ with $|V| = n, |E| \leq sn$ such that $\max\{ deg(v) \mid v \in V \} \leq 3$ and $\mathrm{C}(G) \geq \epsilon n$.*

Proposition 2.11. *Given a tree G with n vertices and let d be the maximum degree of all the vertices. Then*

$$\mathrm{C}(G) \leq \begin{cases} \frac{1}{2} + 2\log_3 n, & \text{if } d = 3 \text{ or } 4; \\ \frac{1}{2} + \frac{1}{2}(d+1)\log_d n, & \text{if } d \geq 5. \end{cases} \tag{2–19}$$

Proposition 2.12. *Let G be a complete ternary tree having n vertices with $n = \frac{1}{2}(3^l - 1)$, for some l even. Then*

$$\mathrm{C}(G) \geq \lceil \log_3 n \rceil - \lceil \log_3 \log_3 n \rceil. \tag{2–20}$$

The above proposition also holds for odd l, but then the number of vertices in the tree is odd and it does not have a bisection as defined in the first section of this chapter.

In this chapter we have reviewed the known results concerning the bisection size of graphs. Except for the results given by MacGregor [Mac78] for random graphs in $\mathcal{G}(2n, m)$ all the other results are given only to first order term. In the next chapter we shall give bounds for random graphs to second order term. We also expand our model of random graphs to deal with random graphs having a nonlinear number of edges, in contrast to MacGregor's results.

This empty page was substituted for a blank page in the original document.

Chapter 3

Bounds for Bisection Size of Graphs in $\mathcal{G}(2n,p)$.

In this chapter we consider random graphs in $\mathcal{G}(2n,p)$. We defined $\mathcal{G}(2n,p)$ in Chapter 2 as a probability space consists of all graphs on $2n$ vertices, and the probability of a graph on $2n$ vertices and m edges is $p^m q^{\binom{2n}{2}-m}$, where $q = 1-p$. We show in this chapter that with probability 1 the bisection size of $G_{2n,p} \in \mathcal{G}(2n,p)$ is less than $n^2p - O(n)$ and is greater than $n^2p - O(n^{3/2})$. We then extend our model so that p can be a function of n (from $\mathbf{N}$ to $(0,1)$). In particular we consider the function $p = p(n) = \frac{c}{n}$. Under this model, we show that a.e. graph in $\mathcal{G}(2n,p(n))$ has bisection size greater than $cn - n\sqrt{2c\log 2}$ and less than $cn - 0.476c^{1/2}n$.

3.1. Preliminary Results

In this section we first present without proofs some elementary results in probability theory which will be needed later in the chapter. Proofs of these results can be found in most books on probability theory, e.g. [F74]. We next present some fundamental results about random graphs.

Chebyshev's Inequality *Let X be a random variable, for any $t > 0$, we have*

$$\mathbf{Pr}\{|X| \geq t\} \leq \frac{\mathbf{E}(X^2)}{t^2} \tag{3-1}$$

In particular, if $\mathbf{E}(X) = \mu$ then

$$\mathbf{Pr}\{|X-\mu| \geq t\} \leq \frac{\mathbf{Var}(X)}{t^2} \tag{3-2}$$

where $\mathbf{E}(X)$ and $\mathbf{Var}(X)$ are the expectation and variance of X, respectively.

In our proofs we need to approximate expressions which contain the binomial random variable. For this purpose we use the following approximation given in [JK69] to approximate the binomial random variable by the normal random variable.

$$\mathbf{Pr}\{ S_n \leq k \} \approx \mathcal{N}\left(\frac{k + \frac{1}{2} - np}{\sqrt{npq}} \right), \tag{3-3}$$

where

$$\mathbf{Pr}\{ S_n \leq k \} = \sum_{i=0}^{k} \binom{n}{i} p^i q^{n-i} \tag{3-4}$$

and

$$\mathcal{N}(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{1}{2}y^2} dy \tag{3-5}$$

is the normal distribution function. The following approximation of the normal random variable given in [F74] will also be needed later.

$$\mathcal{N}(-x) = 1 - \mathcal{N}(x) \approx x^{-1}\eta(x) \tag{3-6}$$

as $x \to \infty$, where

$$\eta(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} \tag{3-7}$$

is the normal density function. In our computation we also use the following form of Stirling's formula for factorial

$$n! = n^n e^{-n} \sqrt{2\pi n}\,(1 + o(1)) \tag{3-8}$$

from which we can easily show that

$$\binom{2n}{n} \approx \frac{2^{2n}}{\sqrt{\pi n}}. \tag{3-9}$$

By a simple application of the Chebyshev's inequality we get the first part of the following result.

Proposition 3.1. *Given $\epsilon > 0$ and $p \in (0,1)$ fixed. A.e. graph in $\mathcal{G}(n,p)$ has at least $\frac{1}{2}(p-\epsilon)n^2$ edges and at most $\frac{1}{2}(p+\epsilon)n^2$ edges. If p is a function from $\mathbf{N}$ to $(0,1)$ such that $n^2 p(n) \to \infty$ and $(1-p(n))n^2 \to \infty$ then we again have : a.e. graph in $\mathcal{G}(n,p(n))$ has at least $\frac{1}{2}(p(n)-\epsilon)n^2$ edges and at most $\frac{1}{2}(p(n)+\epsilon)n^2$ edges, for any $\epsilon > 0$.*

The proof of the second part of the above proposition is given in the proof of Proposition 3.2.

As mentioned in the previous chapter, the models $\mathcal{G}(n,p)$ and $\mathcal{G}(n,m)$ are closely related. We show now under what conditions, properties of graphs in one model can be translated to properties of graphs in the other model. We first introduce the concept of convex property. Let Γ^* be a collection of sets, Γ^* is **convex** if $A_1 \subseteq A \subseteq A_2$ for some $A_1, A_2 \in \Gamma^*$ implies $A \in \Gamma^*$. We can then define a **convex property** of graph similarly.

Proposition 3.2.[Bo79] *Let $p = p(n)$ be a function from $\mathbf{N}$ to $(0,1)$ such that $n^2p(n)\to\infty$ and $(1-p(n))n^2\to\infty$ as $n\to\infty$, let Q be a graph property, and let $N = \binom{n}{2}$.*

(i) *Let $\epsilon > 0$ be fixed, and suppose that if $(1-\epsilon)Np(n) < m < (1+\epsilon)Np(n)$ then a.e. graph in $\mathcal{G}(n,m)$ has Q. Then a.e. graph in $\mathcal{G}(n,p(n))$ has Q.*

(ii) *If Q is a convex property and a.e. graph in $\mathcal{G}(n,p(n))$ has Q, then a.e. graph in $\mathcal{G}(n,\lfloor Np(n)\rfloor)$ has Q.*

Proof : Let Γ denote the set of graphs in $\mathcal{G}(n,p)$ and let Γ_m denote the set of graphs in $\mathcal{G}(n,m)$. Then $\Gamma = \bigcup_{m=o}^{N} \Gamma_m$, and

$$\mathbf{Pr}(\Gamma_m) = \binom{N}{m} p^m q^{N-m} \tag{3-10}$$

therefore,

$$\frac{\mathbf{Pr}(\Gamma_m)}{\mathbf{Pr}(\Gamma_{m+1})} = \frac{m+1}{N-m}\frac{q}{p} \tag{3-11}$$

and $\mathbf{Pr}(\Gamma_m)$ is maximal for some $m \in [Np-1, Np+1]$. Let $\epsilon \in (0,1)$ be given, since $n^2p\to\infty$ as $n\to\infty$, for sufficiently large n we have

$$\frac{\mathbf{Pr}(\Gamma_m)}{\mathbf{Pr}(\Gamma_{m+1})} < \begin{cases} 1-\epsilon & \text{if } m < (1-\epsilon)Np; \\ (1+\epsilon)^{-1} & \text{if } m > (1+\epsilon)Np. \end{cases} \tag{3-12}$$

Hence

$$\mathbf{Pr}\left(\bigcup_{m=(1+\epsilon)Np}^{(1-\epsilon)Np} \Gamma_m\right) \to 1 \qquad \text{as } n\to\infty. \tag{3-13}$$

This proves (i) and also the second part of Proposition 3.1. To prove (ii) we note that from (3-11) we get

$$\mathbf{Pr}\left(\bigcup_{m=0}^{Np} \Gamma_m\right) > \eta \tag{3-14}$$

for any $\eta \in (0, 1/2)$ and for sufficiently large n. From (3–13) and (3–14) we have : if $\Gamma^* \subseteq \Gamma$ is such that $\mathbf{Pr}(\Gamma^*) \to 1$ as $n \to \infty$ then for any $\epsilon > 0$, there exist m_1 and m_2 such that $(1-\epsilon)Np \leq m_1 \leq Np \leq m_2 \leq (1+\epsilon)Np$ and

$$\frac{|\Gamma_{m_i} \bigcap \Gamma^*|}{|\Gamma_{m_i}|} \quad \to \quad 1 \qquad \text{as } n \to \infty, \qquad i = 1, 2. \tag{3–15}$$

In particular if Γ^* is a convex set then we have

$$\frac{|\Gamma_m \bigcap \Gamma^*|}{|\Gamma_m|} \quad \to \quad 1 \qquad \text{as } n \to \infty, \tag{3–16}$$

if $m_1 \leq m \leq m_2$, say $m = \lfloor Np \rfloor$. This proves (ii) and completes the proof of the proposition. ∎

3.2. Lower Bounds on the Bisection Size

We first consider model $\mathcal{G}(2n, p)$ with p being a fixed constant in the interval $(0, 1)$, independent of n. For each $G_{2n,p} \in \mathcal{G}(2n, p)$ there are $\frac{1}{2}\binom{2n}{n}$ distinct bisections. We label the bisections $1, \ldots, \frac{1}{2}\binom{2n}{n}$. Let k be an integer in $[0, n^2]$. For each $i \in \{1, \ldots, \frac{1}{2}\binom{2n}{n}\}$ define the following random variable on $\mathcal{G}(2n, p)$

$$\mathbf{A}_i^{(k)}(G_{2n,p}) = \begin{cases} 1, & \text{if the } i\text{th bisection has size} \leq k; \\ 0, & \text{otherwise.} \end{cases} \tag{3–17}$$

We next define $\mathbf{B}_k(G_{2n,p})$ as a random variable on $\mathcal{G}(2n, p)$ denoting the number of bisections of $G_{2n,p}$ having size less than or equal to k. That is

$$\mathbf{B}_k(G_{2n,p}) = \sum_{i=1}^{\binom{2n}{n}/2} \mathbf{A}_i^{(k)}(G_{2n,p}). \tag{3–18}$$

It is clear that

$$\mathbf{E}(\mathbf{A}_i^{(k)}) = \sum_{j=0}^{k} \binom{n^2}{j} p^j q^{n^2-j}. \tag{3–19}$$

Now we are ready to give the lower bound for bisection size of graphs in $\mathcal{G}(2n, p)$.

Proposition 3.3. *Let $f(n)$ be a function such that $f(n) = o(1)$ and $f(n) = \Omega(\frac{1}{n})$. Then a.e. graph in $\mathcal{G}(2n, p)$, where p is a fixed constant, has bisection size greater than or equal to*

$$k(n) = n^2 p - n\sqrt{4npq \log 2 - 2pq \log n - 2pq \log f(n) + O(1)}. \tag{3–20}$$

Proof : Fix n, let $k = k(n)$ be given as in the statement of the proposition. Let $G_{2n,p} \in \mathcal{G}(2n,p)$ be given. We have

$$\begin{aligned}\Pr(\mathrm{C}(G_{2n,p}) \leq k) &= \Pr(\text{some bisections of } G_{2n,p} \text{ has size } \leq k) \\ &= \Pr(\mathrm{B}_k \neq 0) \\ &\leq \mathrm{E}(\mathrm{B}_k).\end{aligned}$$

Now let $M = \frac{1}{2}\binom{2n}{n}$, then

$$\begin{aligned}\mathrm{E}(\mathrm{B}_k) &= \sum_{i=1}^{M} \mathrm{E}(\mathrm{A}_i^{(k)}) \\ &= \sum_{i=1}^{M}\sum_{j=0}^{k}\binom{n^2}{j}p^j q^{n^2-j} \\ &= M\sum_{j=o}^{k}\binom{n^2}{j}p^j q^{n^2-j} \\ &\approx \frac{1}{2}\frac{2^{2n}}{\sqrt{\pi n}}\left(\frac{1}{\sqrt{2\pi}}\frac{\sqrt{n^2pq}}{n^2p-k}\exp\left(-\frac{1}{2pq}\left(np-\frac{k}{n}\right)^2\right)\right) \\ &\approx f(n) + o(1) \\ &= o(1)\end{aligned}$$

by a straightforward application of (3–3) and Stirling's formula, and by our choice of k. Thus

$$\lim_{n\to\infty} \Pr(\mathrm{C}(G_{2n,p}) \geq k) = 1$$

this completes the proof of the proposition. ∎

In Proposition 3.3 p is a constant, and hence by Proposition 3.1 the number of edges of random graphs in $\mathcal{G}(2n,p)$, with probability 1, lies in the interval $[(p-\epsilon)2n^2, (p+\epsilon)2n^2]$ for any $\epsilon > 0$. If we now instead consider p as a function of n, in particular $p = \frac{c}{n}$ then again by Proposition 3.1 the number of edges of a.e. graph in $\mathcal{G}(2n, p(n) = \frac{c}{n})$ lies in the interval $[(2c-\epsilon)n, (2c+\epsilon)n]$ for any $\epsilon > 0$. In other words, graphs in $\mathcal{G}(2n, p(n) = \frac{c}{n})$, with probability 1, have a linear number of edges. By using Propositions 3.1, 3.2, and 2.5 we get the following.

Proposition 3.4. *A.e. graph in $\mathcal{G}(2n,p)$, where $p = p(n) = \frac{c}{n}$ for some constant $c > 9$, has bisection size greater than or equal to*

$$k(n) = cn - n\sqrt{2c\log 2}\,(1 + o(1)) \qquad (3\text{–}21)$$

For the general case of p a function of n we conjecture that the following holds.

Conjecture. *Let $p(n)$ be a function from $\mathbf{N}$ to $(0,1)$ such that $n^2p(n)\to\infty$ as $n\to\infty$. A.e. graph in $\mathcal{G}(2n,p(n))$ has bisection size greater than or equal to*

$$k(n) = n^2p(n) - 2n^{3/2}\sqrt{p(n)q(n)\log 2}\,(1+o(1)) \tag{3-22}$$

3.3. Upper Bounds on the Bisection Size

As in the previous section, we first consider the model $\mathcal{G}(2n,p)$ with p being a fixed constant in $(0,1)$. Define the random variables $\mathbf{A}_i^{(k)}$ and $\mathbf{B}_k$ as in equations (3–18) and (3–19). We first need the following lemma which was given by Matula [Mat76] as a stronger version of the so called second moment method used frequently in [ES74].

Lemma 3.5. *Let X be a nonnegative, integer valued random variable with mean $\mathrm{E}(X)$, standard deviation $\sigma < \infty$, then*

$$\mathbf{Pr}\{X \neq 0\} \geq \frac{\mathbf{E}^2(X)}{\mathbf{E}(X^2)} \tag{3-23}$$

Proof : As $2ij \leq i^2 + j^2$, we have

$$\begin{aligned}
\mathbf{E}^2(X) &= \left(\sum_{i=1}^{\infty} i\mathbf{Pr}(X=i)\right)^2 \\
&= \sum_{j=1}^{\infty}\sum_{i=1}^{\infty} ij\mathbf{Pr}(X=i)\mathbf{Pr}(X=j) \\
&\leq \frac{1}{2}\sum_{j=1}^{\infty}\sum_{i=1}^{\infty}(i^2+j^2)\mathbf{Pr}(X=i)\mathbf{Pr}(X=j) \\
&= \left(\sum_{j=1}^{\infty} j^2\mathbf{Pr}(X=j)\right)\sum_{i=1}^{\infty}\mathbf{Pr}(X=i) \\
&= \mathbf{E}(X^2)\mathbf{Pr}(X\neq 0)
\end{aligned}$$

Thus the lemma is proved. ∎

Proposition 3.6. *Let $p \in (0,1)$ be a fixed constant. A.e. graph $\mathcal{G}(2n,p)$ has bisection size less than*

$$k(n) = n^2p - \alpha n \tag{3-24}$$

for some $\alpha < \frac{1}{2\pi}\sqrt{pq}$.

Proof : Fix n, let $k = k(n)$ be given as above. It is clear from the definition of $\mathbf{B}_k$ that

$$\mathrm{E}(\mathbf{B}_k) = M \sum_{j=0}^{k} \binom{n^2}{j} p^j q^{n^2-j} \tag{3-25}$$

where $M = \frac{1}{2}\binom{2n}{n}$. The second moment of $\mathbf{B}_k$ is

$$\begin{aligned}
\mathrm{E}(\mathbf{B}_k^2) &= \mathrm{E}\left[\left(\sum_{i=1}^{M} \mathbf{A}_i^{(k)}\right)^2\right] \\
&= \mathrm{E}\left[\sum_{i=1}^{M} \left(\mathbf{A}_i^{(k)}\right)^2\right] + \mathrm{E}\left[\sum_{i=1}^{M} \sum_{\substack{j=1 \\ j\neq i}}^{M} \mathbf{A}_i^{(k)} \mathbf{A}_j^{(k)}\right] \\
&\leq \mathrm{E}(\mathbf{B}_k) + \frac{1}{2}\mathrm{E}\left[\sum_{i=1}^{M} \sum_{\substack{j=1 \\ j\neq i}}^{M} \left(\mathbf{A}_i^{(k)}\right)^2 \left(\mathbf{A}_j^{(k)}\right)^2\right] \\
&\leq \mathrm{E}(\mathbf{B}_k) + (M-1)\mathrm{E}(\mathbf{B}_k) \\
&= M\mathrm{E}(\mathbf{B}_k)
\end{aligned} \tag{3-26}$$

as $\left(\mathbf{A}_i^{(k)}\right)^2 = \mathbf{A}_i^{(k)}$ and $2\mathbf{A}_i^{(k)}\mathbf{A}_j^{(k)} \leq \left(\mathbf{A}_i^{(k)}\right)^2 + \left(\mathbf{A}_j^{(k)}\right)^2$. By Lemma 3.5 we have

$$\begin{aligned}
\mathrm{Pr}(\mathbf{B}_k \neq 0) &\geq \frac{\mathrm{E}(\mathbf{B}_k)^2}{\mathrm{E}(\mathbf{B}_k^2)} \\
&\geq \frac{\mathrm{E}(\mathbf{B}_k)^2}{M\mathrm{E}(\mathbf{B}_k)} \\
&= \sum_{j=0}^{k} \binom{n^2}{j} p^j q^{n^2-j} \\
&\approx \frac{1}{\sqrt{2\pi}} \frac{\sqrt{n^2pq}}{n^2p-k} \exp\left(-\frac{1}{2pq}\left(np - \frac{k}{n}\right)^2\right)
\end{aligned}$$

for our choice of k, the right hand side of the above expression goes to 1 as $n \to \infty$. This proves the proposition. ∎

Consider next random graphs in $\mathcal{G}(2n, p(n) = \frac{c}{n})$ for some $c > 1$. We can use Proposition 3.2 to translate the upper bound given by Proposition 2.7 for graphs in $\mathcal{G}(2n, m = 2cn)$, to an upper bound for graphs in $\mathcal{G}(2n, p(n) = \frac{c}{n})$.

Proposition 3.7. *Let $p(n) = \frac{c}{n}$ for some constant $c > 1$. A.e. graph in $\mathcal{G}(2n, p(n))$ has bisection size less than*

$$k(n) = cn - 2H(c)cn. \tag{3-27}$$

where $H(c) \approx 0.238c^{-1/2}$.

We conjecture that the following holds for a.e.graph in $\mathcal{G}(n, p(n))$

Conjecture. *Let $p(n)$ be a function from $\mathbf{N}$ to $(0,1)$ such that $n^2p(n) \to \infty$ as $n \to \infty$. A.e. graph in $\mathcal{G}(2n, p(n))$ has bisection size less than*

$$k(n) = n^2p(n) - \alpha n\sqrt{p(n)q(n)} \tag{3-28}$$

for $\alpha \leq \frac{1}{2\pi}$.

From Proposition 3.1 we see that, with probability 1, graphs in $\mathcal{G}(2n, p)$ have $2n^2p(1 + o(1))$ edges. Our results in this chapter show that about half of these edges will appear as cut edges in the optimal bisections. This conclusion can be drawn from Chapter 2, what is new here is the presence of the second order terms in the bounds. In fact we need these second order terms if we wish to use these bounds to judge the performance of a heuristic on random graphs. This is the case as, given a random graph, we can easily show that with high probability, a random bisection will have size equal to about half of the number of edges in the random graph. Thus any improvement by a heuristic over a random bisection will reflect only in the terms of order $o(n^2)$.

In the next chapter we shall present some known approximate algorithms for solving the graph bisection problem, and some empirical data of their performances on random graphs.

Chapter 4

Graph Bisection Heuristics

In this chapter we describe various heuristics that are used for solving the graph bisection problem. All of these heuristics have one basic working principle. They all start with a randomly chosen bisection, and improve upon that bisection. The differences among these heuristics are in how the improvement is done. The simplest of these is an algorithm which chooses the vertices to be exchanged by ordering the vertices in each part of the bisection based on the property of the initial bisection, and then choosing the best initial segments from each part of the bisection to be exchanged. The next approximation algorithm that we describe is given by Kernighan and Lin in [KL70], this algorithm also orders the vertices in each part of the original bisection and chooses the best initial segments to be exchanged. However, the ordering is done dynamically. That is, the order of one vertex depends both on the original bisection and the vertices appearing before it in the ordering, whereas the order of one vertex in the previous alogrithm depends only on the original bisection. An interesting algorithm proposed by Kirkpatrick, Gelatt, and Vecchi [KGV82] uses ideas in statistical mechanics to help solve the graph bisection problem. In the last section of this chapter we shall present some empirical results.

4.1. Descriptions of Some Heuristics

There is one obvious way to solve the graph bisection problem, namely, enumerating all the possible bisections and choose one that has a minimum size.

However, to enumerate all the bisections of a graph on $2n$ vertices, it takes time proportional to $\binom{2n}{n}$ which is exponential in n, hence this method is very inefficient for large n.

Another method is to generate the bisections randomly, and keep the one that has the smallest size among the ones that have been produced so far. This is faster than the exhaustive procedure described above, nonetheless, by experience [KL] reported that the number of optimal bisections or near optimal bisection is very small, hence the probability of getting an optimal solution by this method is quite small. In fact, [KL] has experimented with a class of 0-1 matrices of size 32×32 (these can be considered as the adjacency matrices) and found that the number of optimal bisections are typically 3 or 5 out of a total of $\binom{32}{16}$ bisections, which means that the probability of success on any trial is less than 10^{-7}.

By considering the given graph as a network and the cost of the edges in the given graph as the maximum flow capacity of the edges, then it seems that the graph bisection problem can be solved by using the well known Ford-Fulkerson max-flow, min-cut algorithm. This algorithm will give not only the maximum flow between any two points but also the minimum cut that separates those two points. Unfortunately, the algorithm does not have control over the size of the subsets of the partition, that is it will give the min-cut but the resulting two subsets of the partition need not have equal size as required by the graph bisection problem. Furthermore, there is no obvious way to modify the algorithm so that we have control over the size of the subsets of the bisection.

We now describe a simple heuristic called the **block** heuristic. A similar heuristic is used in the proof of Proposition 2.7 in [Mac78]. Let $G = (V, E)$ be a graph with $|V| = 2n$, the algorithm starts with a random bisection, say (A, B) of G. We define the gain of each vertex in the graph with respect to that bisection as follows. Let $a \in A$, the gain of A, denoted by g_a, is the difference in the number of edges connecting a to vertices in B, and the number of edges connecting a to vertices in A. We extend this definition to pair of vertices one in A and one in B. More formally, define

$$g_{a,b} = |(A, B)| - |(A', B')| \qquad (4\text{–}1)$$

where

$$A' = (A - \{a\}) \bigcup \{b\} \quad \text{and} \quad B' = (B - \{b\}) \bigcup \{a\}. \tag{4-2}$$

In other words, $g_{a,b}$ is the reduction in the size of the bisection when a and b are interchanged. Clearly,

$$g_{a,b} = g_a + g_b - 2\delta(a,b) \tag{4-3}$$

where

$$\delta(a,b) = \begin{cases} 1, & \text{if } (a,b) \in E; \\ 0, & \text{if } (a,b) \notin E. \end{cases} \tag{4-4}$$

This algorithm first calculates g_v for all $v \in V$. It then chooses $a_1 \in A, b_1 \in B$, such that g_{a_1,b_1} is maximum. It then computes the size of the new bisection resulted when a_1 and b_1 are interchanged, and sets a_1, b_1 aside. Next it chooses the best pair a_2, b_2 in $A - \{a_1\}$ and $B - \{b_1\}$ such that g_{a_2,b_2} is maximum, then sets a_2, b_2 aside. The process is repeated until there is only one pair of vertices left. The algorithm then returns the bisection with the smallest size found. Note that when choosing a_2 and b_2 the gains of a_2 and b_2 have not been updated to account for the fact that a_1 and b_1 have been interchanged. That is, at all stages we use the gains computed at the beginning of the algorithm. The algorithm can be described more formally as follows.

begin

1. Compute g_a, g_b for each $a \in A, b \in B$.
2. Let $Q_A = \emptyset, Q_B = \emptyset$.
3. **for** $i = 2$ **to** n **do**

   begin

4. Choose $a_i \in A - Q_A$ and $b_i \in B - Q_B$ such that g_{a_i,b_i} is maximum over all choices of a and b.
5. Set $Q_A = Q_A \cup \{a_i\}, Q_B = Q_B \cup \{b_i\}$

   end

6. Choose $k \in \{1, \ldots, n\}$ to maximize $\sum_{i=1}^{k} g_{a_i,b_i}$.
7. Interchange the subsets $\{a_1, \ldots, a_k\}$ and $\{b_1, \ldots, b_k\}$ to get the new bisection.

end

Figure 4.1. The Block Algorithm

Kernighan and Lin in [KL70] give a heuristic for solving the graph bisection problem which seems to work well in practice. Let a graph $G = (V, E), V = 2n$ be given. The main idea here is the same as before, that is to start with an arbitrary bisection, say (A, B), and improve upon it. The improvement is accomplished by interchanging subsets $X \subseteq A, Y \subseteq B$, and $|X| = |Y| < n$ so that the size of the bisection is decreased. If we consider all possible subsets of A and B, we shall be able to pick out the subsets, whose interchange will give us the optimal bisection. However, this will be an exponentially long procedure, which is undesirable. The Kernighan-Lin heuristic finds these subsets approximately by choosing elements of X and Y sequentially. This choosing process is done as follows. For each element $a \in A, b \in B$, let $g_{a,b}$ be defined as before. The algorithm first computes $g_{a,b}$ for all $a \in A, b \in B$. It then chooses $a_1 \in A, b_1 \in B$ such that

$$g_{a_1,b_1} = \max\{ g_{a,b} \mid a \in A, b \in B \}$$

The algorithm now updates the gains of all vertices in V, except a_1 and b_1, with respect to the new bisection $((A - \{a_1\}) \cup \{b_1\}, (B - \{b_1\}) \cup \{a_1\})$. The algorithm next repeats the process for this new bisection and chooses a new pair of vertices to be exchanged, except that a_1 and b_1 will not be considered anymore in choosing the next pair that will give the maximum reduction. That is, once a vertex is chosen to be exchanged it will no longer be considered in later stages. The process is repeated untill all vertices have been considered. We now have a list of pairs $(a_1, b_1), \ldots, (a_n, b_n)$. Clearly if all these pairs are interchanged the total reduction is zero. The algorithm now chooses a $k < n$ such that the interchange of the subsets $\{a_1, \ldots, a_k\}$ and $\{b_1, \ldots, b_k\}$ will give a maximum reduction over all choices of $k < n$. This whole process is called a pass of the algorithm. The algorithm can have several passes. Each pass, except the first one, starts with the bisection given as the result of the previous pass. The algorithm can have a fixed number of passes or it can run until no more improvement is possible. Another alternative is to have an entire new arbitrary bisection as the input to each pass, and keep the smallest bisection produced so far.

We now describe the algorithm formally. Let $G = (V, E)$ be a graph with $V = 2n$. Let (A, B) be a bisection of G. For each $a \in A$ $b \in B$ define $g_{a,b}$ as

before. The heuristic is shown in Figure 4.2. Steps 7 and 9 of the algorithm can be easily checked that the values of the g_a and g_b are correctly updated with respect to the new bisection, that is after the sets $\{a_1, \ldots, a_i\}$ and $\{b_1, \ldots, b_i\}$ have been interchanged. It is also easily shown that the running time of the algorithm is $O(n^2 \log n)$.

begin

1. Compute g_a, g_b for each $a \in A, b \in B$.
2. $Q_A = \emptyset, Q_B = \emptyset$.
3. **for** $i = 2$ **to** n **do**

   begin

4. Choose $a_i \in A - Q_A$ and $b_i \in B - Q_B$ such that g_{a_i,b_i} is maximum over all choices of a and b.
5. Set $Q_A = Q_A \cup \{a_i\}, Q_B = Q_B \cup \{b_i\}$
6. **for each** $a \in A - Q_A$ **do**
7. $g_a = g_a + 2\delta(a, a_i) - 2\delta(a, b_i)$
8. **for each** $b \in B - Q_B$ **do**
9. $g_b = g_b + 2\delta(b, b_i) - 2\delta(b, a_i)$

   end

10. Choose $k \in \{1, \ldots, n\}$ to maximize $\sum_{i=1}^{k} g_{a_i,b_i}$.
11. Interchange the subsets $\{a_1, \ldots, a_k\}$ and $\{b_1, \ldots, b_k\}$ to get the new bisection.

end

Figure 4.2. One pass of the Kernighan-Lin bisection heuristic

Other graph bisection heuristics which are variations of the Kernighan-Lin's heuristic have been considered by Macgregor [Mac78], he also considered hybrid of these heuristics. It should be noted that a slight variation of the Kernighan-Lin heuristic has been implemented by Fiduccia and Mattheyses [FM82] to run in linear time by using some clever data structures.

Finally, we present an algorithm proposed by Kirkpatrick, *et al.*, [KGV82], which makes an interesting connection between the annealing process and the

iterative improvement process of the graph bisection heuristics. Consider a system consisting of a large number of atoms, such as a sample of liquid or solid matter. The aggregate behaviour of the system can be observed by considering the average behaviour taken over an ensemble of identical systems. We associate with each configuration of the system in the ensemble a Boltzmann's probability, $\exp(-E(\{r_i\})/k_BT)$, where $E(\{r_i\})$ is the energy of the configuration defined by the atomic positions $\{r_i\}$, k_B is the Boltzman's constant, and T is the temperature. One wishes to know what happens to the system in the limit of low temperature, for instance, whether atoms remain fluid or solidify. It is known that ground states and configurations having energy close to them are very rare, nonetheless, they dominate the behaviour of the system at low temperature because as T is lowered the Boltzmann distribution collapsed into the lowest energy state or states. To find the low temperature states of a system it is necessary to use an annealing process. That is to first melt the substance, then lower the temperature slowly, and spend a long time at the temperatures near the freezing point. Otherwise, the resulting configuration will be metastable.

There is a simple algorithm given by Metropolis, *et al.* [M53] which simulates a collection of atoms in equilibrium at a given temperature. In each step of the algorithm, an atom is given a small random displacement, and the corresponding change in energy, ΔE, of the system is computed. If $\Delta E \leq 0$, then the displacment is accepted, and the configuration with the just displaced atom is used as the starting configuration for the next step. When $\Delta E > 0$, the configuration is accepted with probability $\mathbf{Pr}(\Delta E) = \exp(-\Delta E/k_BT)$. A random number is chosen uniformly in the interval $(0, 1)$, and compared with $\mathbf{Pr}(\Delta E)$. If it is less than $\mathbf{Pr}(\Delta E)$ then the new configuration is accepted, if not, the original configuration is retained and we repeat the process.

It is observed in [KGV82] that the iterative improvement process in a combinatorial optimization problem such as the graph bisection problem is similar to the microscopic rearrangement processes modelled by statistical mechanics, where an appropriate cost function for the graph bisection problem will play the role of energy. Using this analogy we note that in the process of finding the solution if we only accept rearrangements that reduce the cost function, then this is like rapid

quenching from high temperature to $T = 0$, thus the resulting solutions will often be local optima and metastable. By utilizing the Metropolis' algorithm described above, in which rearrangements that increase the cost function are sometimes accepted, we can expect to get better solutions as indicated by the observation made in actual physical processes.This algorithm has not been extensively tested but the results found so far seem very favourable.

4.2. Some Empirical Results

In this section we give the results produced by the block and Kernighan-Lin algorithms when random graphs in $\mathcal{G}(n, p = 1/2)$ are used as inputs. We ran both algorithms on graphs in $\mathcal{G}(n, p = 1/2)$ as n varies from 20 to 800 in increments of 20 and from 800 to 1000 in increments of 50. For each value of n 10 random graphs are generated, the results produced by each algorithm on these random graphs are averaged and used as the result of that algorithm for that value of n.

To speed up the running time some modifications are made in both algorithms in implementing them. In both algorithms the most time consuming step is in choosing the best pair to be interchanged. Assume that the current bisection is (A, B), instead of considering each pair $(a, b), a \in A, b \in B$ to find one with the largest $g_{a,b}$ (defined in the previous section), we simply find the pair (a, b) such that $g_a + g_b$ is largest, i.e., we can choose a and b independent of each other. In the block algorithm we further reduced the running time by considering blocks to be exchanged only of length less than $n/2$ for graphs with $2n$ vertices. That is, in step 3 of the block algorithm, the **for** loop runs from 2 to $\lceil n/2 \rceil$ instead of n. This is due to the observation in some experimentation that the length of the block to be exchanged in the block algorithm is almost always less than one fourth the number of vertices in the graph. The change in the performance as a result of these modifications is not significant, as observed and also reported in [Mac78].

For large value of n the Kernighan-Lin algorithm seems to perform quite better than the block algorithm but the running time is also longer. The bisection sizes produced by the Kernighan-Lin algorithm apparently lie well in halfway between the upper bounds and the lower bounds. The programs are written in MACLISP, and run on a DEC20. On an input of 1000 vertices the Kernighan-Lin algorithm

takes approximately 283 seconds and the block algorithm takes approximately 240 seconds.

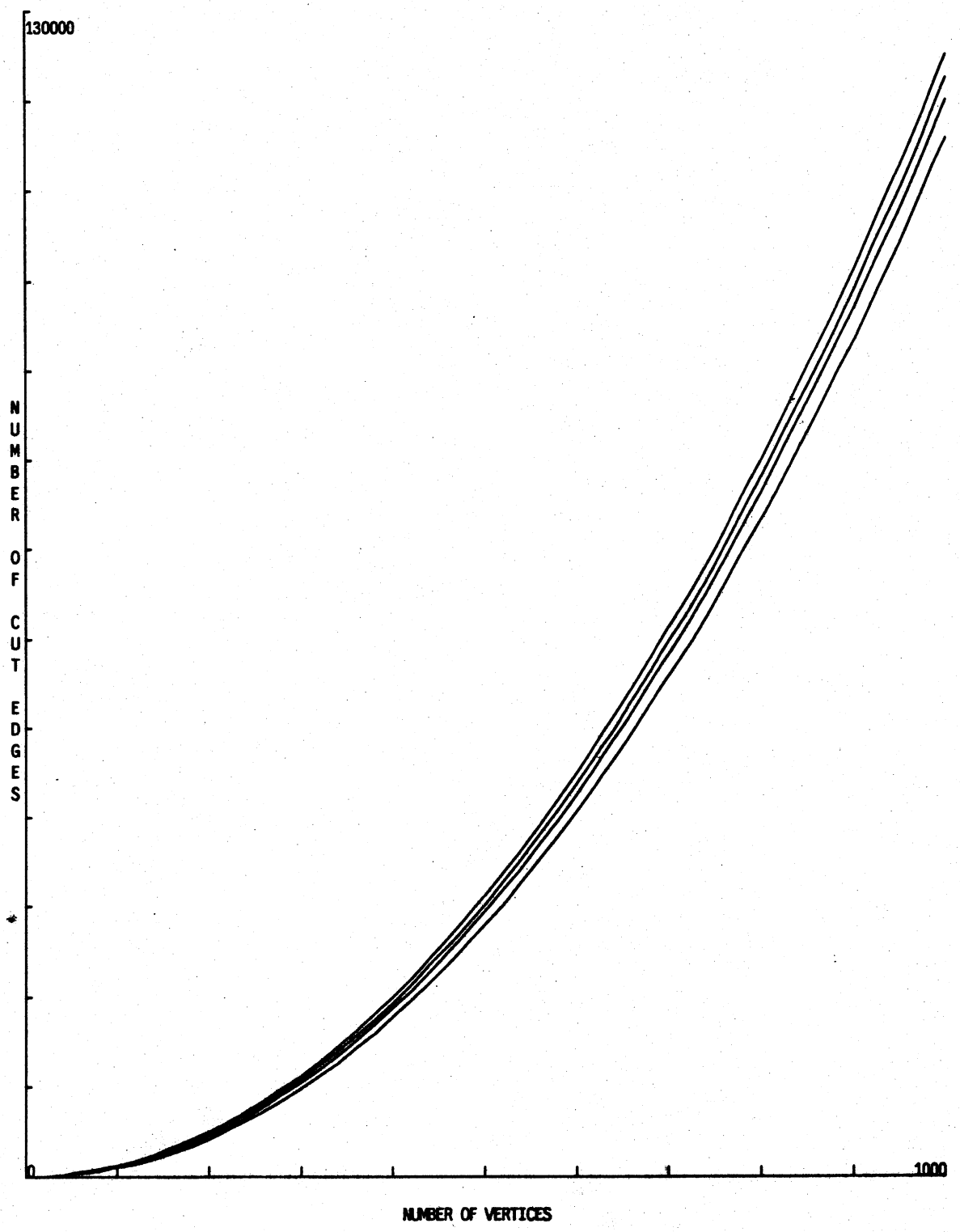

Figure 4.3. Graph of Vertex Size vs. Bisection Sizes

In Figure 4.3 the top curve is the upper bound given in Chapter 3, the curve second from the top is the bisection size produced by the block algorithm, the next curve represents the bisection sizes returned by the Kernighan-Lin algorithm, and the bottom curve is the lower bound given in Chapter 3. The graph was prepared using the MACSYMA program developed by the Mathlab group at MIT[2] [MAC77].

[2]Work of the Mathlab group is currently supported in part, by the United States Energy Research and Development Administration under Contract Number E(11-1)-3070 and by the National Aeronautics and Space Administration under Grant NSG 1323.

This empty page was substituted for a blank page in the original document.

Chapter 5

Conclusion

In this thesis we have shown that almost every graph in $\mathcal{G}(2n,p)$ has bisection size greater than $n^2p - O\left(n^{3/2}\sqrt{pq}\right)$ and less than $n^2p - O\left(n\sqrt{pq}\right)$. We also show that a.e. graph in $\mathcal{G}(2n,p(n)=\frac{c}{n})$ with $c > 9$ has bisection size greater than $cn - n\sqrt{2c\log 2}$ and less than $cn - 0.476c^{1/2}n$. These, together with other known results on bisection size, seem to indicate that, except for specially structured graphs, we always have to remove about half the number of edges of a graph to bisect it. Empirical results also give evidence that there is no heuristic that performs well on all kind of graphs [Mac78]. This suggests that if one wishes to obtain optimal or near optimal bisection of a graph, one needs to know more than just the number of vertices and edges. Knowing about the structure of the graph alone is not sufficient, however, one also needs special algorithms to exploit that structure.

Efforts in proving how good is the performance of a graph bisection heuristic on random graphs have been in general unsuccessful. Even though most graph bisection heuristics are deterministic, except that choosing the starting bisection is done randomly, they are still very difficult to analyze. The difficulty lies in the fact that the updating process and the interchanging of vertices change the distribution of the edges in a random graph. It is useful to have provably good graph bisection heuristics, for besides from being of interest in itself, it also has been shown by Leighton [L82] that by using a provably good graph bisection heuristic, we can get provably good algorithm for the crossing number problem, and from which we

can get provably good algorithm for the graph layout problem. Given a graph, the assingment of the nodes and edges of the graph to the points and tracks (vertical and horizontal) of a rectangular grid is called an **embedding** of the graph. The graph layout problem is the problem of finding an embedding of the graph such that the following quantities of the grid are minimum : area, number of edge crossings, number of total edge length, and the maximum edge length. The crossing number of the graph is the minimum number of edge crossings over all possible embeddings of the graph. The crossing number problem is then, of course, the problem of determining the crossing number of a given graph. Both the layout problem and the crossing number problem are known to be NP-complete. The layout problem is one of the vital parts in the VLSI design process, especially of chips which can do a large amount of computation reliably and efficiently.

Besides from the conjectures in Chapter 3, our main open question is to determine analytically how well a certain graph bisection heuristic does. In Proposition 3.4 we show the lower bound on the bisection size for random graphs having a linear number of edges, in particular this result only holds for the number of edges being greater than nine times the number of vertices. Recent experimental results by Goldberg and Gardner [GG82a] indicate that random graphs with the number of edges equal the number of vertices have bisection size between 25% and 30% of the number of edges. Thus it is of interest to know what is the lower bound for the bisection size of random graphs when the number of edges is less than nine times the number of vertices.

To more closely model the graph partitioning problem arising in actual application a more complicated model is needed. One possible model is the set of random hypergraphs. It would then be useful to have results for random hypergraphs similar to the known results for the random graphs. Also in actual application we usually need to partition the graph into more than two parts, and at the moment there is no good uniform method for extending the known graph bisection heuristics to solve the general graph partitioning problem. These are a few possible directions for further study concerning this subject.

References

[BES80] Babai, L., P. Erdös, and S. M. Selkow, "Random Graph Isomorphism," SIAM J. Comput., Vol. 9, No. 3, Aug. 1980, pp. 628–635.

[Bo79] Bollobás, B., Graph Theory : An Introductory Course, Springer Verlag, New York, (1979).

[Bo81a] Bollobás, B., "Degree Sequences of Random Graphs," Discrete Mathematics, 33(1981), pp. 1–19.

[Bo81b] Bollobás, B., "Random Graphs," Proceedings of the 8th British Combinatorial Conference, Lecture Notes Series, No.52, Cambridge University Press, 1981.

[Bo82] Bollobás, B., "Vertices of Given Degree in a Random Graph," J. of Graph Theory, Vol. 6, 1982, pp. 147–155.

[BE76] Bollobás, B., and P. Erdös, "Cliques in Random Graphs," Math. Proc. Camb. Phil. Soc., 80(1976), pp. 419–427.

[Br77] Breuer, M. A., "Min-cut Placement," J. Design Aut. and Fault Tol. Comp., Vol. 1, No. 4, Oct. 1977, pp. 343–362.

[DH73] Donath, W. E., and A. J. Hoffman, "Lower Bounds for the Partitioning of Graphs," IBM J. Res. Develop., Sept. 1973, pp. 420–425.

[E59] Erdös, P., "Graph Theory and Probability," Canadian J. Math., 11(1959), pp.34–38.

[E61] Erdös, P., "Graph Theory and Probability," Canadian J. Math., 13(1961), pp.346–352.

[ER59] Erdös, P., and A. Rényi, "On Random Graphs, I," Publ. Math. Debrecen, 6(1959), pp. 290–291.

[ER60] Erdös, P., and A. Rényi, "On the Evolution of Random Graphs," Publications of the Mathematical Institute of the Hungarian Academy of Sciences, 5(1960), pp. 17–60.

[ES74] Erdös, P., and J. Spencer, Probabilistic Methods in Combinatorics, Academic Press, New York and London, 1974.

[F74] Feller, W., An Introduction to Probability Theory and its Applications, Vol. 1, 3rd Ed., Wiley, New York, 1974.

[FM82] Fiduccia, C. M., and R. M. Mattheyses, "A Linear-Time Heuristic for Improving Network Partitions," submitted to the Design Automation Confereence, (January 1982).

[GJ79] Garey, M. R., and D. S. Johnson, Computers and Intractibility : A Guide to the Theory of NP-Completeness, Freeman, San Francisco, 1979.

[GJS76] Garey, M. R., D. S. Johnson, and L. Stockmeyer, "Some Simplified NP-complete Graph Problems," Theoretical Computer Science, 1(1976), pp.237–267.

[GG82] Goldberg, M. K. and R. Gardner, "On the Minimal Cut Problem," Proceeding of the Silver Jubilee Conference on Combinatorics, Waterloo, 1982.

[GG82a] Goldberg, M. K. and R. Gardner, Oral Communication.

[GM75] Grimmett, G. R., and C. J. McDiarmid, "On Colouring Random Graphs," Math. Proc. Camb. Phil. Soc., 77 (1975), pp. 313–324.

[HR] Hyafil, L., and R. L. Rivest, "Graph Partitioning and Constructing Optimal Decision Trees are Polynomial Complete Problems," Report No. 33, IRIA-Laboria, Rocquencourt, France.

[JK69] Johson, N. L. and S. Kotz, Discrete Distributions, John Wiley & Sons, 1969.

[Ka82] Karoński, M., "A Review of Random Graphs," J. of Graph Theory, Vol. 6, 1982, pp. 349–389.

[KL70] Kernighan, B. W., and S. Lin, "An Efficient Heuristic Procedure for Partitioning Graphs," The Bell System Tech. J., Feb. 1970, pp. 291–307.

[KGV82] Kirkpatrick, S., C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by Simulated Annealing," IBM Research Report RC 9355, April 1982, Yorktown Heights, New York.

[L82] Leighton, F. T., "A Layout Strategy for VLSI Which Is Provably Good," Proceedings of the 14th Anual ACM Symposium on Theory of Computing, May 1982, pp. 85–98.

[Mac78] Macgregor, R. M., "On Partitioning a Graph : A Theoretical and Empirical Study," Ph.D. Thesis, Stanford University, 1978, also appeared as Technical Report UCB/ERL M78/14.

[MAC77] MACSYMA Reference Manual, The Mathlab Group, Laboratory for Computer Science, MIT, 1977.

[Mat76] Matula, D. W., "The Largest Clique in a Random Graph," Technical Report CS 7608, Dept. of Computer Science, Southern Methodist University, Dallas, Texas.

[M53] Metropolis, N., A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, "Equation of State Calculations by Fast Computing Machines," J. Chemical Physics, Vol. 21, No. 6, June 1953, pp. 1087–1092.

[P76] Pósa, L., "Hamiltonian Circuits in Random Graphs," Discrete Mathematics, 14(1976), pp. 359–364.

This blank page was inserted to preserve pagination.

Scanning Agent Identification Target

Scanning of this document was supported in part by the **Corporation for National Research Initiatives,** using funds from the **Advanced Research Projects Agency** of the **United states Government** under Grant: **MDA972-92-J1029.**

The scanning agent for this project was the **Document Services** department of the **M.I.T Libraries.** Technical support for this project was also provided by the **M.I.T. Laboratory for Computer Sciences.**

Scanned
Date: 7/21/1995
M.I.T. Libraries
Document Services