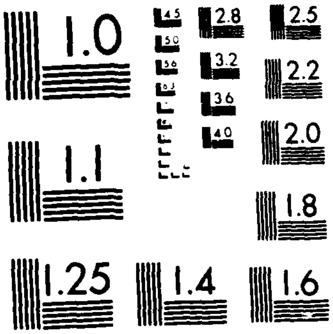
AD-A185 547 OPTIMIZATION BY SIMULATED ANNEALING: A TIME-COMPLEXITY 1/2
ANALYSIS(U) ILLINOIS UNIV AT URBANA DEPT OF ELECTRICAL
ENGINEERING G H SASAKI OCT 87 UILU-ENG-87-2261
UNCLASSIFIED N00014-82-K-0359 F/G 12/4 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

October 1987 UILU-ENG-87-2261

2

COORDINATED SCIENCE LABORATORY
College of Engineering

OPTIMIZATION BY SIMULATED ANNEALING: A TIME-COMPLEXITY ANALYSIS

Galen Hajime Sasaki

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

A185 547

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS None
2a. SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) UILU-ENG-87-2261	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 National Science Foundation and Office of Naval Research
6c. ADDRESS *(City, State, and ZIP Code)* 1101 W. Springfield Avenue Urbana, IL 61801		7b. ADDRESS *(City, State, and ZIP Code)* 1800 G. Street, N.W. 800 N. Quincy St. Washington, D.C. 20550 Arlington, VA 22217
8a. NAME OF FUNDING/SPONSORING ORGANIZATION National Science Foundation & Office of Naval Research	8b. OFFICE SYMBOL *(If applicable)*	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER NSF - ECS 83-52030 ONR - N00014-82-K-0359
8c. ADDRESS *(City, State, and ZIP Code)* 1800 G Street, N.W. 800 N. Quincy St. Washington, D.C. 20550 Arlington, VA 22217		10. SOURCE OF FUNDING NUMBERS

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

11 TITLE *(Include Security Classification)*

Optimization by Simulated Annealing: A Time-Complexity Analysis

12 PERSONAL AUTHOR(S)

Sasaki, Galen Hajime

13a. TYPE OF REPORT Technical	13b. TIME COVERED FROM ________ TO ________	14. DATE OF REPORT *(Year, Month, Day)* October 1987	15. PAGE COUNT 110

16. SUPPLEMENTARY NOTATION

17 COSATI CODES		
FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
simulated annealing, optimization, search, complexity, matchings, set partitioning

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

In this thesis, results of a study of the heuristic random search optimization method called simulated annealing are given. Most of the results are concerned with the average amount of time simulated annealing takes to find an acceptable solution.

We analyzed the average time complexity of simulated annealing for the matching problem. Although the matching problem has worst-case polynomial time complexity, we show that there is a sequence of graphs where the average time complexity of a "natural" version of simulated annealing is at least exponential. In contrast, we show that the "natural" version of simulated annealing has a worst-case polynomial average time complexity if it is only required to find "near" maximum matchings. An exponential lower bound on the minimum average time complexity over a wide class of simulated annealing algorithms when our attention is restricted to constant temperature schedules is also given.

(continued on back of page)

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

83 APR edition may be used until exhausted.
All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. Abstract (cont.)

The typical case for simulated annealing for the matching problem is also analyzed. Since we were not able to discover a method to exactly analyze the average time complexity of simulated annealing for the matching problem for "typical" graphs, we used approximations to estimate the average time complexity and then checked the accuracy of the approximation with data from computer simulations. Our results indicate that if we only consider graphs that have at least as many edges as they have nodes then the average time complexity of simulated annealing for a typical graph with n nodes is $O(n^4)$.

A technique for producing easy-to-analyze annealing processes, called the template method, is given. It is our hope that this method will produce interesting examples of simulated annealing that will help us to understand the heuristic. We provide two examples of using the template method to analyze the finite-time behavior of simulated annealing as a function of the temperature schedule. A generalization of simulated annealing, which we refer to as the threshold random search algorithm, is presented. We also give conditions under which no monotone decreasing temperature schedule is optimal.

Finally, we discuss the use of quadratic penalty methods in conjunction with simulated annealing to solve problems with equality constraints. An experimental evaluation is made between adaptive and static quadratic penalty methods, and it is shown that adaptive quadratic penalty methods can provide low-valued solutions over a wider range of penalty parameter values than static quadratic penalty methods.

OPTIMIZATION BY SIMULATED ANNEALING: A TIME-COMPLEXITY ANALYSIS

BY

GALEN HAJIME SASAKI

B.S., University of Hawaii, 1981
M.S., University of Illinois, 1984

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1987

Urbana, Illinois

OPTIMIZATION BY SIMULATED ANNEALING: A TIME-COMPLEXITY ANALYSIS

Galen Hajime Sasaki, Ph.D.
Department of Electrical Engineering
University of Illinois at Urbana-Champaign, 1987
Bruce Hajek, Advisor

In this thesis, results of a study of the heuristic random search optimization method called *simulated annealing* are given. Most of the results are concerned with the average amount of time simulated annealing takes to find an acceptable solution.

We analyzed the average time complexity of simulated annealing for the matching problem. Although the matching problem has worst-case polynomial time complexity, we show that there is a sequence of graphs where the average time complexity of a "natural" version of simulated annealing is at least exponential. In contrast, we show that the "natural" version of simulated annealing has a worst-case polynomial average time complexity if it is only required to find "near" maximum matchings. An exponential lower bound on the minimum average time complexity over a wide class of simulated annealing algorithms when our attention is restricted to constant temperature schedules is also given.

The typical case for simulated annealing for the matching problem is also analyzed. Since we were not able to discover a method to exactly analyze the average time complexity of simulated annealing for the matching problem for "typical" graphs, we used approximations to estimate the average time complexity and then checked the accuracy of the approximation with data from computer simulations. Our results indicate that if we only consider graphs that have at least as many edges as they have nodes then the average time complexity of simulated annealing for a typical graph with n nodes is $O(n^4)$.

A technique for producing easy-to-analyze annealing processes, called the *template* method, is given. It is our hope that this method will produce interesting examples of simulated annealing that will help us to understand the heuristic. We provide two examples of using the template method to analyze the finite-time behavior of simulated annealing as a function of the temperature schedule. A generalization of simulated annealing, which we refer to as the *threshold random search* algorithm, is presented. We also give conditions under which *no* monotone decreasing temperature schedule is optimal.

Finally, we discuss the use of quadratic penalty methods in conjunction with simulated annealing to solve problems with equality constraints. An experimental evaluation is made between adaptive and static quadratic penalty methods, and it is shown that adaptive quadratic penalty methods can provide low-valued solutions over a wider range of penalty parameter values than static quadratic penalty methods.

DEDICATION

To my mother and father.

ACKNOWLEDGEMENTS

I would like to thank Professor Bruce Hajek, my thesis advisor, for his guidance, encouragement, and enthusiasm. His excellence as an advisor has made my life as a graduate student very enjoyable.

I would also like to thank Professor T. Basar, Professor C. L. Liu, and Professor M. Loui for serving on my thesis committee. Their valuable comments have improved the content of this thesis.

My friends, R. Cruz, R. Barton, and A. Lam, have also provided useful discussions for which I am grateful.

Finally, I would like to thank my mother and father for their encouragement and infinite patience.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Simulated annealing is a heuristic random search technique, introduced independently by Kirkpatrick et al. [1] and Černy [2], for finding approximate solutions to combinatorial optimization problems. It is a variation of the local improvement technique in which an initial solution is repeatedly improved by perturbing it until it reaches a "local minimum," i.e., a solution where no improvement is possible by perturbing it. A drawback of the local improvement method is that the search may terminate in poor local minima. Simulated annealing tries to avoid getting stuck in poor local minima by randomly accepting some perturbations that worsen the solution as well as accepting all perturbations that improve it.

Simulated annealing has many of the attractions of the local improvement method, such as the relative ease of implementation on new problems and the modest amounts of memory usually required by these implementations. Since simulated annealing is a simple heuristic method, it has been applied to solve a variety of problems, such as generating error-correcting codes [3], restoring images automatically [4], and designing VLSI circuits automatically [1], and since simulated annealing typically requires only a modest amount of memory, it has usually been applied to solve problems with many variables. Empirical results show that simulated annealing will usually find a better solution than the local improvement method, but at a cost of a longer run time.

With these properties, simulated annealing has been and will be applied to find approximate solutions to many useful problems. Hence, it is important to analyze the performance of simulated annealing and identify important parameters that govern that performance. For the remainder of this introduction we will precisely describe a simulated annealing algorithm applied to solving a generic combinatorial optimization probiem, briefly review some of the directions of past theoretical research, and discuss the organization of this thesis.

Suppose we want to solve the generic combinatorial optimization problem

$$\min\{c(s): s \in S\},$$

where S is a finite set and c is a cost function c: $S \to R$. In addition, suppose we have a transition probability matrix R over S and a sequence $(T_k: k \geq 0)$ (called the *temperature schedule*) of positive extended real valued numbers. Typically, the temperature schedule is monotone decreasing to zero. A state s is referred to as a *neighbor* of s′ if $R_{s's} > 0$. A simulated annealing algorithm applied to this problem constructs a sequence $(X_k: k \geq 0)$ of states in the following way. An initial state X_0 is chosen. Given $X_k = s$, a potential next state Y_k is chosen with probability distribution

$$P[Y_k = s' \mid X_k = s] = R_{ss'}.$$

Then we set

$$X_{k+1} = \begin{cases} Y_k & \text{if } c(Y_k) \leq c(s) \\ Y_k & \text{with probability } p_k \text{ if } c(Y_k) > c(s) \\ X_k & \text{otherwise,} \end{cases}$$

where

$$p_k = \exp\left[\frac{-\max\{c(Y_k) - c(s), 0\}}{T_k}\right].$$

This specifies how the sequence $(X_k: k \geq 0)$ is chosen. The random process $(X_k: k \geq 0)$ produced by the algorithm is a discrete time Markov chain, and we will call it the *annealing process* on system (S, c, R) and with temperature schedule $(T_k: k \geq 0)$.

For some specified time K (which is possibly a random time), the algorithm returns X_K. If the amount of memory permits, the algorithm can be modified to return the lowest valued member of $\{X_0, X_1, \ldots, X_K\}$ rather than X_K. Note that if the temperature schedule is identically equal to zero then the simulated annealing algorithm is a local improvement algorithm.

We now briefly review some directions of past theoretical research on simulated annealing. One of the popular directions of theoretical research on simulated annealing is to determine conditions on the transition probability matrix R and the temperature schedule $(T_k\colon k \geq 0)$ so that

$$\lim_{k\to\infty} P[X_k \in S^*] = 1,$$

where S^* is the subset of states of S that have minimal cost. The results of this type lead to insights on the dynamics of the annealing process.

Another direction of theoretical research is to analyze the finite time behavior of the annealing process. We will give short descriptions of three results of such analysis. The first result, by Mitra et al. [5], is an upper bound on the distance between the state probability vector of X_k and a probability vector $(\pi_s\colon s \in S)$ such that $\sum_{s\in S^*} \pi_s = 1$. The following is a simple corollary to their result. Suppose the temperature schedule $(T_k\colon k \geq 0)$ is such that $T_k = \dfrac{\gamma}{\log(k + k_0 + 1)}$, $k_0 \geq 1$, and $\gamma \geq rL$, where r is the radius of the graph underlying the annealing process $(X_k\colon k \geq 0)$ and L is a Lipschitz-like constant of the cost function. Then, for a large number of iterations k,

$$\sum_{s\in S} |P[X_k = s] - \pi_s| \;=\; O(1/k^{\min\{a,b\}}),$$

where a and b respectively increase and decrease with increasing γ.

The second result, by Gelfand and Mitter [6], is a lower bound for $P[X_j \in S^*\colon \text{for some } j \leq k]$. If the temperature schedule $(T_k\colon k \geq 0)$ is such that $\exp(-1/T_k) = k^{-1/\Gamma}$ and Γ is large enough, then this lower bound converges exponentially fast to zero. However, if Γ is sufficiently small, the bound converges to a strictly positive number.

The third result, which is a corollary to a result by Lundy and Mees [7], is an upper bound on $E[\min\{k\colon X_k \in S^*\}]$, and is derived in the following way. Let $c^* = \min\{c(s)\colon \in S\}$. Suppose for some positive r there are positive scalars α and γ such that for all $j \geq 0$,

$$c(X_{j+r}) - c(X_j) < \gamma$$

and

$$E[c(X_{j+r}) - c(X_j) \,|\, c(X_j) > c^*] \le -\alpha.$$

A trivial extension of Wald's equation [8] yields

$$E[\min\{k: X_k \in S^*\}] < \frac{c(X_0) - c^* + r\gamma}{\alpha}.$$

However, Lundy and Mees do not indicate how to find values of α and r.

Besides trivial examples, most theoretical results on the finite-time behavior of the annealing process have been derived without much consideration of typical applications of simulated annealing. Therefore, the bounds of these results may be very loose if they are directly applied to a particular application. In Chapters 2 and 3, we analyze simulated annealing for a particular nontrivial problem, the *matching problem*. Upper and lower bounds on the average time it takes a simulated annealing algorithm to find a solution of the matching problem are presented in Chapter 2. These bounds are worst-case bounds over all instances of the problem of a specified size. In Chapter 3, we attempt to determine the average length of time a simulated annealing algorithm takes to find a solution to the matching problem for a "typical" instance. The results of Chapter 3 are based on approximations, and these approximations are checked for accuracy by comparing them to with data from computer simulations.

In Chapter 4, we present a collection of results. A simple technique to cook up easy-to-analyze annealing processes is given in Section 4.2. It is our hope that interesting annealing processes that will help us to better understand simulated annealing will be produced by this method. In Section 4.3, we present a random search heuristic, called the *threshold search algorithm*, that is a generalization of simulated annealing. In Section 4.4, conditions are given that insure that *no* monotone decreasing temperature schedules are optimal.

In Chapter 5, we consider using simulated annealing to solve problems that have equality constraints. A technique used to solve problems with equality constraints is the *quadratic penalty* method, which transforms an equality constrained problem into an unconstrained problem. Simulated annealing can then be used to solve the transformed problem. The *method of multipliers* is an adaptive quadratic penalty method, and through experiments we compare this method, and a variation of this method, with the quadratic penalty method.

Finally, we summarize the results of this thesis and provide possible directions for future research in Chapter 6.

CHAPTER 2

MATCHING PROBLEM: AVERAGE PERFORMANCE FOR WORST-CASE GRAPHS

2.1. Introduction

The introduction of this chapter is divided into four subsections. The motivation of the results of the chapter is provided in Subsection 2.1.1. Subsection 2.1.2 contains the basic simulated annealing algorithm for the matching problem that is analyzed in Subsection 2.1.3, Sections 2.2 and 2.3. In Subsection 2.1.3, a convergence in probability result of the basic simulated annealing algorithm in Subsection 2.1.2 is given. Finally, an organization of the rest of the chapter is presented in Subsection 2.1.4.

2.1.1. Motivation

In this chapter, we consider simulated annealing applied to maximum matching, a fundamental problem in combinatorial optimization. An instance of the maximum matching problem is a simple graph G = (V,E), where V denotes the set of nodes of G and E denotes the set of (undirected) edges of G. A matching M in G is a subset of E such that no two edges in M share a node. The maximum matching problem, for instance G, is to find a matching in G with maximum cardinality.

The maximum matching problem is easy in the sense that there is a known deterministic algorithm which solves the problem in $O(\sqrt{|V|}|E|)$ steps (see [9]), where $|V|$ is the cardinality of V. However, we do not consider maximum matching to be trivial, since the deterministic algorithm is somewhat subtle.

2.1.2. The basic simulated annealing algorithm for maximum matching

We will here describe what is perhaps the most obvious way to apply simulated annealing to search for the maximum matching of a graph $G = (V,E)$. Let $T_1, T_2, \ldots$ be a nonincreasing temperature schedule. We say that an edge e is *matchable* relative to a matching M if $e \notin M$ and if M+e is a matching (here M+e is our notation for $M \cup \{e\}$, which we use only if $e \notin M$). Let Q(M) denote the set of matchable edges relative to matching M.

To begin the algorithm, choose an arbitrary matching X_0 in G -- for example, X_0 could be the empty set $\emptyset$. Having selected $X_0, X_1, \ldots, X_k$, choose X_{k+1} as follows. Choose an edge e at random, all edges in E being equally likely.

If e is matchable relative to X_k, let $X_{k+1} = X_k + e$.

$$\text{If } e \in X_k, \text{ let } X_{k+1} = \begin{cases} X_k - e & \text{with probability } \exp(-1/T_k) \\ X_k & \text{with probability } 1 - \exp(-1/T_k). \end{cases}$$

Else, let $X_{k+1} = X_k$.

Note that $(X_k\colon k \geq 0)$ is an annealing process on system (S, c, R), where S is the set of all matchings, c(s) is the negative of the cardinality of s, and R is a transition probability matrix over S such that

$$R_{ij} = \begin{cases} \frac{1}{|E|} & i \neq j \text{ and } |i \oplus j| = 1 \\ 0 & |i \oplus j| > 1 \\ 1 - \sum_{k \neq i} R_{ik} & i = j. \end{cases}$$

2.1.3. Convergence in probability

We begin by giving some standard notation [10]. Given a matching M in G, a node v is *exposed* if no edge in M is incident to v. A *path* p in G is a sequence of nodes $p = [v_1, v_2, \ldots, v_k]$, where $k \geq 1$, the nodes $v_1, v_2, \ldots, v_k$ are distinct, and $[v_i, v_{i+1}] \in E$ for

$1 \leq i \leq k-1$. The length of such a path is k-1. The path is *augmenting* for M if its length is odd (so k is even), if v_1 and v_k are exposed, and if $[v_i,v_{i+1}] \in M$ for even values of i with $2 \leq i \leq k\text{-}2$. A well-known result of Berge and Norman and Rabin is that a matching M does not have maximum cardinality if and only if there exists an augmenting path for M [10, Theorem 10.1].

Let M_0 be a matching which does not have maximum cardinality, and let $[v_1, v_2, \ldots, v_k]$ be an augmenting path for M_0. Starting from M_0, it is possible for the basic simulated annealing algorithm to reach a higher cardinality matching by passing through the sequence of matchings $M_1, M_2, \ldots, M_{k-1}$ given by

$$\begin{array}{ll}
M_1 = M_0 - [v_2, v_3] & M_2 = M_1 + [v_1,v_2] \\
M_3 = M_2 - [v_4, v_5] & M_4 = M_3 + [v_3,v_4] \\
\vdots & \vdots \\
M_{k-3} = M_{k-4} - [v_{k-2},v_{k-1}] & M_{k-2} = M_{k-3} + [v_{k-3},v_{k-2}]
\end{array}$$

and finally

$$M_{k-1} = M_{k-2} + [v_k,v_{k-1}].$$

The matchings in the sequence have cardinality at least as large as the cardinality of M_0 minus one. In the terminology of [11], the depths of the local maxima for the matching problem are at most one. The following theorem is thus an immediate consequence of [11, Theorem 1]. A matching M is said to be maximal if no edge is matchable relative to M. Let S^* denote the set of matchings with maximum cardinality.

Theorem 2.1.1 Let $G = (V,E)$ be a graph with a nonempty set of edges E. If all maximal matchings of G are in S^* then

$$\lim_{k \to \infty} P[X_k \in S^*] = 1 \text{ if and only if } \lim_{k \to \infty} \exp(-1/T_k) = 0.$$

If some maximal matching is not in S^* then

$$\lim_{k\to+\infty} P[X_k \in S^*] = 1 \text{ if and only if } \lim_{k\to+\infty} \exp(-1/T_k) = 0 \text{ and } \sum_{k=0}^{\infty} \exp(-1/T_k) = +\infty.$$

Theorem 2.1.1 gives a large-time asymptotic result for each fixed instance G, and the conditions do not depend on the size of G. In contrast, our goal in this paper is to give asymptotic results as $|V|$ tends to infinity. Interesting, general work on the analysis of simulated annealing run for a finite number of iterations has appeared (see [5], [12], and [6], for example). However, the general theory does not determine, for example, whether simulated annealing exactly (or nearly) solves the maximum matching problem in an amount of time growing as a polynomial in $|V|$. Moreover, it is not clear yet that any general theory could answer such questions. In this chapter, we present results that we would like to see established more generally.

2.1.4. Organization of the chapter

In Section 2.2, we show that for a certain family of graphs the basic annealing algorithm, or any other algorithm in a fairly large related class, *cannot* find maximum cardinality matchings using average time that is upper bounded by a polynomial in $|V|$. In contrast, we show, in Section 2.3, that a degenerate form of the basic simulated annealing algorithm (obtained by letting T_k be a suitably chosen constant, independent of k) produces matchings with *nearly* maximum cardinality using average time that is upper bounded by a polynomial in $|V|$. Sections 2.2 and 2.3 can be read independently. In Section 2.4, we present a lower bound on the average time simulated annealing takes to find nearly maximum cardinality matchings when the temperature schedule is restricted to be of constant value.

2.2. The Impossibility of Maximum Matching in Polynomial Average Time Using Certain Simulated Annealing Type Algorithms

Certain local search algorithms for the maximum matching problem will be considered in this section. The algorithms will not be restricted much in an attempt to include several implementations of simulated annealing. Both the basic simulated annealing algorithm (given in Subsection

2.1.2), when $X_0 = \emptyset$, and a particular multistart-descent algorithm will be included. Nevertheless, it will be proved that the algorithms cannot reach a maximum matching in average time bounded by a polynomial in $|V|$, for a particular family of graphs.

First, we allow the "temperature" to depend on both time and the current and past states of the algorithm. Second, we assume that the type of each move can be specified from among the three possibilities whenever they exist: addition of an edge, deletion of an edge, no change. The key restriction we do impose is that given the type of a move, the location of the edge to be added or deleted is uniformly distributed over the possible locations.

We thus view the sequence $X_0, X_1, \ldots$ of states generated by the algorithm as a controlled Markov process. Suppose that "controls" a_t and d_t are given such that for each t,

C.1. $a_t, d_t, a_t + d_t \in [0,1]$ with probability one, and a_t and d_t are functions of $(X_0, \ldots, X_t)$.

C.2. $$P[X_{t+1} = M' \mid X_t = M, X_{t-1}, \ldots, X_0] = \begin{cases} 1 - a_t - d_t & \text{if } M = M' \\ d_t/|M| & \text{if } e \in M \text{ and } M' = M-e \\ a_t/|Q(M)| & \text{if } e \in Q(M) \text{ and } M' = M+e \\ 0 & \text{if } |M \oplus M'| \geq 2, \end{cases}$$

where $M \oplus M'$ denotes the symmetric difference of M and M' (recall that Q(M) is the set of edges matchable relative to M).

Clearly, if we choose the controls appropriately we can use this controlled Markov process to mimic the basic simulated annealing process of Subsection 2.1.2. We can also control the Markov process to mimic a multistart-descent algorithm (although only at half speed). To do this we assume that $X_0 = \emptyset$. We then let $a_t = 1$ for $0 \leq t < S_1$, where S_1 is the first time that a maximal matching is reached. Then we let $d_t = 1$ for $S_1 \leq t < 2S_1$, which guarantees that $X_t = \emptyset$ for

$t = 2S_1$. We then keep repeating this process.

The family of graphs we will focus on is $\{G_1, G_2, G_3, \ldots\}$, where $G_n = (V,E)$,

$$V = \{u_{ij}: 1 \leq i, j \leq n+1\} \cup \{v_{ij}: 1 \leq i, j \leq n+1\},$$

$$E = H \cup B, \quad H = \bigcup_{j=1}^{n+1} H_j, \quad B = \bigcup_{j=1}^{n} B_j,$$

$$H_j = \{[u_{ij}, v_{ij}]: 1 \leq i \leq n+1\} \qquad \text{for } j \text{ such that } 1 \leq j \leq n+1,$$

and

$$B_j = \{[v_{ij}, u_{k,j+1}]: 1 \leq i, k \leq n+1\} \qquad \text{for } j \text{ such that } 1 \leq j \leq n.$$

Graph G_n is a bipartite graph with $2(n+1)^2$ nodes and $(n+1)^3$ edges. For each j, the subgraph of G_n induced by the nodes of edges of B_j is a complete bipartite graph, and the subgraph of G_n induced by H_j consists of n+1 disjoint edges. The set of edges H is a matching, and it is maximum since it leaves no nodes exposed. In addition, there are no other maximum matchings since, by induction, any matching which has no nodes exposed must include the edges in $H_1, H_2, \ldots, H_{n+1}$. As an example, G_3 is sketched in Figure 2.2.1.

The main result of this section is the following theorem.

Theorem 2.2.1 There exist positive constants σ_1 and σ_2 such that the following is true. For any $n \geq 1$, let (X,a,d) be a controlled process for finding the maximum matching of G_n satisfying conditions C.1 and C.2. Define R^* by

$$R^* = \min\{k: X_k \text{ is a maximum matching}\}.$$

Then

$$E\{R^* \mid X_0 = \emptyset\} \geq \sigma_1 \exp(\sigma_2 n).$$

In the proof of Theorem 2.2.1, a function g(M) is used to measure the distance from a matching M to the unique maximum matching H. We will present the function g after defining some of its components. Let $V_0(M) = U_{n+2}(M) = 0$. For all j, such that $1 \leq j \leq n+1$, let $U_j(M)$ (resp.,

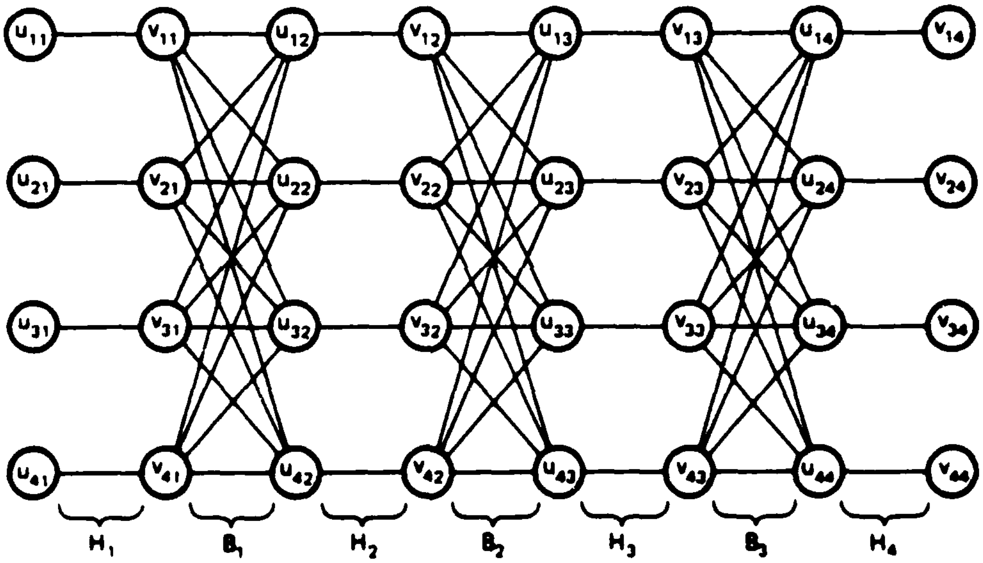


Figure 2.2.1. Sketch of G_3.

$V_j(M)$) be the number of nodes in $\{u_{ij}: 1 \le i \le n+1\}$ (resp., $\{v_{ij}: 1 \le i \le n+1\}$) that are exposed relative to M. The function g is such that, for a matching M,

$$g(M) = c|B \cap M| + \sum_{j=1}^{n} \psi(V_j(M), U_{j+1}(M)),$$

where c = 18,

$$\psi(x,y) = 2 \min\{x,y\} + I_{\{x>0, y>0, x\neq y\}},$$

and $I_{\{\ \}}$ is the indicator function. Note that g(M) includes the term $|B \cap M|$ and a second term which is related to the set of edges in B that are matchable relative to M. Some trivial properties of g are that g is nonnegative, g(H) = 0, and $g(\emptyset) = 2n(n+1)$.

The next set of lemmas and definitions is used to show that $(g(X_k): k \ge 0)$ tends to drift away from zero (and hence $(X_k: k \ge 0)$ drifts away from H) when $(g(X_k): k \ge 0)$ is below a certain threshold (see Equation (2.2.3), p. 18). After the lemmas and definitions the proof of Theorem 2.2.1 is presented.

Lemma 2.2.1: Suppose $x,y \ge 0$. Then

(a) $$\psi(x+1, y) - \psi(x,y) \begin{cases} \in \{1,2,3\} & \text{if } y \ge \max\{x,1\} \\ = 0 & \text{otherwise,} \end{cases}$$

(b) $$\psi(x, y+1) - \psi(x,y) \begin{cases} \in \{1,2,3\} & \text{if } x \ge \max\{y,1\} \\ = 0; & \text{otherwise,} \end{cases}$$

(c) $\psi(x+1, y+1) - \psi(x,y) \in \{2,3\}$.

Proof: Easy by inspection of Table 2.2.1.

□

Table 2.2.1. Values of $\psi(x,y)$.

x \ y	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	2	3	3	3	3	3
2	0	3	4	5	5	5	5
3	0	3	5	6	7	7	7
4	0	3	5	7	8	9	9
5	0	3	5	7	9	10	11
6	0	3	5	7	9	11	12

Lemma 2.2.2: Let M be a matching of G_n.

(a) Suppose $e \in M \cap H_j$. Then $g(M\text{-}e) - g(M) > 0$ if and only if

$$V_{j-1}(M) \geq \max\{U_j(M),1\} \quad \text{or} \quad U_{j+1}(M) \geq \max\{V_j(M),1\}.$$

(b) $g(M-e) - g(M) \in \{0, 1, \ldots, 6\}$ for $e \in M \cap H$.

(c) $g(M-e) - g(M) \in \{-c+2, -c+3\}$ for $e \in M \cap B$.

Proof: It is easy to see that, for $e \in M \cap H_j$,

$$g(M-e) - g(M) = \psi(V_{j-1}(M), U_j(M) + 1) - \psi(V_{j-1}(M), U_j(M))$$
$$+ \psi(V_j(M) + 1, U_{j+1}(M)) - \psi(V_j(M), U_{j+1}(M)),$$

and, for $e \in M \cap B_j$

$$g(M-e) - g(M) = -c + \psi(V_j(M) + 1, U_{j+1}(M) + 1) - \psi(V_j(M), U_{j+1}(M)).$$

Note that Lemma 2.2.2 can be easily deduced from these equations and Lemma 2.2.1.

□

For each matching M of G_n, define

$A(M) = \{e$ is matchable relative to M and $g(M+e) \neq g(M)\}$,

$D(M) = \{e \in M: g(M\text{-}e) \neq g(M)\}$,

$A_+(M) = \{e \in A(M): g(M+e) > g(M)\}$, $A_-(M) = \{e \in A(M): g(M+e) < g(M)\}$,

$D_+(M) = \{e \in D(M): g(M\text{-}e) > g(M)\}$, $D_-(M) = \{e \in D(M): g(M\text{-}e) < g(M)\}$.

Lemma 2.2.3: Let M be a matching of G_n and let $0<\delta<1$. Then

(a) $D_+(M) \subset M \cap H$, $D_-(M) = M \cap B$, $A_-(M) \subset H\text{-}M$, $A_+(M) = B \cap Q(M)$,

(b) $|A_-(M)| \leq 2|A_+(M)|$,

(c) $|D_-(M)| \leq n\delta$ if $g(M) < nc\delta$,

(d) $|D_+(M)| \geq n(1 - \delta(\frac{c}{2} + 1))$ if $0 < g(M) < nc\delta$.

Proof: Part (a) is a consequence of parts (b) and (c) of Lemma 2.2.2 and the fact that $c > 3$.

We will now prove the following two facts, which imply part (b): every edge in $A_-(M)$ has a node in common with an edge in $A_+(M)$ and for every edge in $A_+(M)$ there are at most two edges in $A_-(M)$ that have nodes in common with it. Let $e \in A_-(M)$. Then $e \in H_j$ for some j by part (a), and moreover at least one of $V_{j-1}(M)$ or $U_{j+1}(M)$ is strictly positive by part (a) of Lemma 2.2.2. Thus, there is at least one edge e′ in $B_{j-1} \cup B_j$ which is matchable relative to M and has a node in common with e. Then e′ is in $A_+(M)$ and hence we can conclude that every edge in $A_-(M)$ has a node in common with an edge in $A_+(M)$. On the other hand, part (a) implies

$A_-(M) \subset H$, $A_+(M) \subset B$, and therefore for every edge in $A_+(M)$ there are at most two edges in $A_-(M)$ that have nodes in common with it. Part (b) is proved.

By part (a)

$$|D_-(M)| \leq |B \cap M| \leq g(M)/c < n\delta,$$

which proves part (c).

We now prove part (d), which will complete the proof of Lemma 2.2.3. Let M be a matching with $0 < g(M) < nc\delta$. The fact that $g(M) > 0$ implies that M is not equal to the unique maximum matching H, which in turn implies that there exists at least one exposed node. Since $g(M) < nc$, M contains fewer than n edges from $B_1 \cup B_2 \cup \cdots \cup B_n$. Hence $M \cap B_k = \varnothing$ for some k. Now, the set of nodes

$$Z = \{v_{ij}\text{: } 1 \leq i \leq n{+}1,\ 1 \leq j \leq k\} \cup \{u_{ij}\text{: } 1 \leq i \leq n{+}1,\ k{+}1 \leq j \leq n{+}1\}$$

contains exactly half of the nodes of the graph. Since $M \cap B_k = \varnothing$, each edge in M is incident to a node in Z and a node not in Z. Thus, Z contains half, and therefore at least one, of the exposed nodes, so at least one of the 2n numbers

$$V_1(M), \ldots, V_n(M), U_2(M), \ldots, U_{n+1}(M)$$

is nonzero. By the symmetry between the U_j's and V_j's, we can restrict attention to the case that for some j with $1 \leq j \leq n$, $U_{j+1}(M)$ is as least as large as any of the other 2n-1 numbers. Then $U_{j+1}(M) \geq \max\{V_j(M), 1\}$ so $M \cap H_j \subset D_+(M)$ by part (a) of Lemma 2.2.2. Hence

$$\begin{aligned}|D_+(M)| \geq |M \cap H_j| &= n{+}1 - V_j(M) - |M \cap B_j| \\ &= n{+}1 - \min\{V_j(M), U_{j+1}(M)\} - |M \cap B_j| \\ &\geq n - g(M)/2 - g(M)/c \geq n(1 - \delta(\tfrac{c}{2} + 1)),\end{aligned}$$

which proves part (d).

□

Lemma 2.2.4: Now set $\delta = 1/43$. If M is a matching of G_n such that $0 < g(M) < nc\delta$, then

$$|A(M)|^{-1} \sum_{e\in A(M)} [g(M+e) - g(M)] \geq 1 \text{ if } A(M) \neq \varnothing \tag{2.2.1}$$

and

$$|D(M)|^{-1} \sum_{e\in D(M)} [g(M-e) - g(M)] \geq \frac{1}{2} \qquad \text{if } D(M) \neq \varnothing. \tag{2.2.2}$$

Proof: By part (a) of Lemma 2.2.3 and parts (b) and (c) of Lemma 2.2.2, we have

$$g(M+e) - g(M) \geq \begin{cases} c-3 & \text{if } e \in A_+(M) \\ -6 & \text{if } e \in A_-(M), \end{cases}$$

which, together with part (b) of Lemma 2.2.3, yields

$$|A(M)|^{-1} \sum_{e\in A(M)} [g(M+e) - g(M)] \geq |A(M)|^{-1}[(c-3)|A_+(M)| - 6|A_-(M)|] \geq 1.$$

Similarly, by part (a) of Lemma 2.2.3 and parts (b) and (c) of Lemma 2.2.2, we have

$$g(M-e) - g(M) \begin{cases} \geq 1 & \text{if } e \in D_+(M) \\ \geq -c+2 & \text{if } e \in D_-(M), \end{cases}$$

which, together with parts (c) and (d) of Lemma 2.2.3, yields (2.2.2).

□

Proof of Theorem 2.2.1: Let $t(0) = 0$, and, for all $k > 0$, let $t(k) = \min\{t > t(k-1): g(X_t) \neq g(X_{t(k-1)})\}$. We can and do assume that $P[R^* < +\infty | X_0 = \varnothing] = 1$. It follows that, with probability one, $R^* \in \{t(1), t(2), \ldots\}$, which implies that

$$P[t(k+1) < \infty | R^* > t(k), X_0 = \varnothing] = 1.$$

Given a matching M, define s(M,1) and s(M,-1) to be the normalized sums appearing in Inequalities (2.2.1) and (2.2.2) respectively, whenever they are well defined. By Lemma 2.2.4, $s(M,\theta)$

$\geq 1/2$ if $0 < g(M) < nc\delta$ and if $s(M,\theta)$ is well defined.

Let $\Theta_k = 1$ if the jump at time $t(k+1)$ is caused by the addition of an edge, and let $\Theta_k = -1$ if the jump at time t_{k+1} is caused by the deletion of an edge.

Suppose M is a matching with $0 < g(M) < nc\delta$. Then

$$E[g(X_{t(k+1)}) - g(X_{t(k)}) \,|\, X_{t(k+1)-1} = M,\ \Theta_k = \theta,\ X_{t(k+1)-2}, \ldots, X_0] = s(M,\theta) \geq \frac{1}{2}.$$

Averaging over appropriate values of Θ_k and $(X_i\colon t(k) < i < t(k+1))$, it follows that

$$E[g(X_{t(k+1)}) - g(X_{t(k)}) - \frac{1}{2} \,|\, g(X_{t(k)}) < nc\delta,\ R^* > t(k),\ X_0, \ldots, X_{t(k)}] \geq 0. \qquad (2.2.3)$$

Also, by parts (b) and (c) of Lemma 2.2.2, the magnitudes of the increments of $g(X_t)$ are bounded by c - 2. Thus, Theorem 2.3 of [13] is in force if we define $(Y, \varepsilon_0, a^*, b^*)$ by

$$Y_k = -g(X_{t(k)}), \quad \varepsilon_0 = \frac{1}{2}, \quad a^* = -n\delta c, \quad \text{and } b^* = 0. \qquad (2.2.4)$$

Using the fact that $Y_0 = -g(\emptyset) = -2n(n+1) \leq a^*$, this produces constants $\eta > 0$, $\rho \in (0,1)$ and $D^* > 0$ such that

$$P[g(X_{t(k)}) = 0,\ R^* > t(k-1) \,|X_0 = \emptyset] \leq u, \qquad (2.2.5)$$

where $u = D^*\exp(-\eta nc\delta)/(1-\rho)$. The term $P[R^* = t(k)|X_0 = \emptyset]$ is less than or equal to the left side of Inequality (2.2.5), because if $X_{t(k)}$ is the maximum matching then $g(X_{t(k)})$ is equal to zero. Therefore, $P[R^* = t(k)|X_0 = \emptyset] \leq u$. Since $R^* \in \{t(1), t(2), \ldots\}$ and, for all k, $t(k) \geq k$, we have

$$P[R^* > k|X_0 = \emptyset] \geq P[R^* > t(k)|X_0 = \emptyset] = 1-\sum_{j=1}^{k} P[R^* = t(j)|X_0 = \emptyset] \geq \max\{0, 1\text{-}ku\}.$$

Hence,

$$E[R^*|X_0 = \emptyset] = \sum_{k=0}^{\infty} P[R^* > k|X_0 = \emptyset] \geq \sum_{k=0}^{\infty} \max\{0, 1\text{-}ku\} \geq \frac{1}{2u}.$$

Thus, taking $\sigma_1 = (1\text{-}\rho)/2D^*$ and $\sigma_2 = \eta c\delta$, Theorem 2.2.1 is proved.

□

Remark: Some extra work shows that Conditions D.1 and D.2 of [13] are satisfied for Y, a^* and b^* given in (2.2.4) and $\eta = .0033683$, $\rho = .9998$, $a^* = -n\delta c$, $b^*=0$ and $D^* = 1$. This shows that Theorem 2.2.1 above is true for $\sigma_1 = .0001$ and $\sigma_2 = .0014$.

2.3. Near Maximum Matching in Polynomial Time

Let d^* denote the maximum node degree of the graph G and let m^* denote the maximum of the cardinalities of matchings in G. The next theorem is the main result of this section.

Theorem 2.3.1: Let $\beta > 1$. Consider a run of the basic simulated annealing algorithm (of Subsection 2.1.2) with $T_k = T$ for all k, where $\exp(-1/T) = \lambda$, and λ is given by

$$\lambda = \frac{1}{3|V|\omega^*} \quad \text{and} \quad \omega^* = \beta|V|(2d^*)^{\beta-1}.$$

Let R denote the random time $R = \min\{k\colon |X_k| \geq m^* (1-\frac{1}{\beta})\}$. Then $ER \leq 24\beta^2|V|^5(2d^*)^{2\beta-2}$.

Remarks: (1) If β and d^* are bounded as $|V| \to \infty$, then $ER = O(|V|^5)$. In the proof below we see that three of these five factors of $|V|$ arise from our upper-bound D_1 on the mean time the algorithm takes to make a single move. A smaller average run time can be achieved by using an efficient implementation of an algorithm that simulates $X_{J(1)}, X_{J(2)}, \ldots$, where J(k) is the time process $(X_k\colon k \geq 0)$ makes its kth move (see [14]).

(2) Since $2d^* \leq 2|V|$, we have with no restriction on d^* that $ER \leq 6\beta^2 2^{2\beta}|V|^{3+2\beta}$. Also note that if $\beta > m*$ then X_R is a maximum matching.

(3) We will briefly comment on our choice of constant T (equivalently, on our choice of temperature). It must be large enough so that the process (X_k: $k \geq 0$) "jumps sufficiently often", which is reflected in the bounds given in Lemmas 2.3.1 and 2.3.2a below. On the other hand, T must be small enough so that there is a net drift towards larger matchings, enabling us to obtain the bound of Lemma 2.3.3 below.

We chose T_k to be independent of k, though we can see some motivation for letting it decrease as k increases. More precisely, it is clear that an improved algorithm can be obtained by letting T_k be a decreasing function of $|X_k|$. For example, it is shown in the proof of Claim 2.3.1 below that a matching M in G has an augmenting path of length at most $1 + 2|M|/(m^*-|M|)$. This bound increases sharply as $|M|$ approaches the final value of $|X_k|$, which is $m^*(1-1/\beta)$, and in the proof we replace $|M|$ in the bound by this final value. However, working with the $|M|$-dependent bound shows that a larger value of T can be used when $|X_k|$ is small so that the algorithm "jumps more often," while maintaining a sufficient drift towards larger matchings.

We chose T_k to be independent of k primarily for two reasons: (1) we wanted to demonstrate that T_k can be chosen independently of the algorithm state ("open-loop" in control-theoretic terms) and (2) we do not think the complexity bounds can be improved much by letting T_k be either a function of $|X_k|$ or a decreasing function of k, because our choice of T_k is tuned to the situation when $|X_k|$ is close to its target value $m^*(1-1/\beta)$, and this situation is the most time consuming for the algorithm anyway.

Finally, we think it is significant that we need to decrease T as the problem size increases. It suggests that if the sequence $T_1, T_2, \ldots$ is to be chosen independently of the graph, it should be decreasing.

Proof: Define a random process $(Y_k: k \geq 0)$ by

$$Y_i = X_{J(i)} \qquad i = 0,1,2,...$$

where $J(0) = 0$ and, for $i \geq 0$,

$$J(i+1) = \min\{k > J(i): X_k \neq X_{k-1}\},$$

and define

$$R(Y) = \min\{i: |Y_i| \geq m^* (1 - \frac{1}{\beta})\}.$$

Note that $|Y_{i+1}| - |Y_i| \in \{-1,1\}$ with probability one for each i.

Next, define a random process $(Z_k: k \geq 0)$ by

$$Z_i = Y_{S(i)} \qquad i = 0,1,...$$

where $S(0) = 0$, $S(1) = 1$ and, for $i \geq 1$,

$$S(i+1) = \min\{j: j > S(i), |Y_j| - |Y_{j-2}| \in \{-2, 2\}\},$$

and define

$$R(Z) = \min\{i: |Z_i| \geq m^*(1 - \frac{1}{\beta})\}.$$

Define constants D_1, D_2 and D_3 by

$$D_1 = 6|V|^2 \omega^*, \qquad D_2 = 2\omega^*, \qquad D_3 = 2|V|.$$

Lemma 2.3.1:

$$E[J(i+1) - J(i)|J(i), (X_0, X_1, \ldots, X_{J(i)})] \leq D_1.$$

Lemma 2.3.2a:

$$E[(S(i+1) - S(i))I_{\{i < R(Z)\}} |S(i), (Y_0, Y_1, \ldots, Y_{S(i)})] \leq D_2.$$

Lemma 2.3.3:

$$ER(Z) \leq D_3.$$

We will next prove Theorem 2.3.1, assuming the lemmas are true. We have

$$ER = EJ(R(Y)) = E\sum_{i=0}^{\infty} (J(i+1) - J(i))I_{\{i < R(Y)\}} = \sum_{i=0}^{\infty} E[J(i+1) - J(i) \mid i < R(Y)]P[i < R(Y)].$$

Now the outcome of the event $\{i < R(Y)\}$ is determined by $(J(i), (X_0, \ldots ,X_{J(i)}))$ so we can apply Lemma 2.3.1 to get

$$ER \leq \sum_{i=0}^{\infty} D_1 P[i < R(Y)] = D_1 ER(Y).$$

Similarly, the fact $R(Y) = S(R(Z))$ and Lemma 2.3.2a imply that $ER(Y) \leq D_2 ER(Z)$. We conclude that $ER \leq D_1 D_2 D_3$ from $ER \leq D_1 ER_Y$, Lemma 2.3.3, $ER(Y) \leq D_2 ER(Z)$, and Lemma 2.3.3. This will establish the theorem once we prove the three lemmas above.

□

Proof of Lemma 2.3.1: By the strong Markov property of $(X_k: k \geq 0)$,

$$E[J(i+1) - J(i) \mid J(i), (X_0, X_1, \ldots , X_J(i))] = E[J(i+1) - J(i) \mid Y_i].$$

Since $\lambda_k = \lambda$ for all k, the transition probabilities of X are time invariant, so

$$E[J(i+1) - J(i) \mid Y_i = M] = P[X_{k+1} \neq X_k \mid X_k = M]^{-1}.$$

Now, fix a matching M. One of two cases is true:

Case 1 Some edge in E is matchable relative to M. Then

$$P[X_{k+1} \neq X_k \mid X_k = M] \geq \frac{1}{|E|}.$$

Case 2: No two of the $|V| - 2|M|$ exposed nodes are connected by an edge in the graph. Then

$$|E| \leq \binom{|V|}{2} - \binom{|V| - 2|M|}{2} = |M|(2|V| - 2|M| - 1) \leq 2|M||V|,$$

so that

$$P[X_{k+1} \neq X_k | X_k = M] = \frac{\lambda|M|}{|E|} \geq \frac{\lambda}{2|V|}.$$

Hence, in either case, $E[J(i+1) - J(i)|Y_i = M] \leq \max\{|E|, 2|V|/\lambda\} = D_1$.

□

Proof of Lemma 2.3.2a: Lemma 2.3.2a is trivial for $i = 0$, so we fix i with $i \geq 1$. Let m be an integer with $1 \leq m < m^*(1-1/\beta)$. Define a set of matchings B by $B = \{M: |M| = m-1 \text{ or } |M| = m\}$ and let M_o be a fixed matching in B. Consider the event $F = \{Y_{S(i)} = M_o, Y_{S(i)-1} \in B \text{ and } R(Z) > i\}$. The outcome of F is determined by $(S(i), (Y_0, Y_1, \ldots, Y_{S(i)}))$, and the union of events of the form of F as M_o and m vary as above is equal to the event $\{R(Z) > i\}$. Hence, it suffices to prove that

$$E[S(i+1) - S(i)|S(i), (Y_0, Y_1, \ldots, Y_{S(i)}]I_F \leq D_2$$

for arbitrary fixed values of m and M_o as above. Without loss of generality, we assume that if $|M_o| = m-1$ then $Q(M_o) \neq \varnothing$; otherwise, $F = \varnothing$ (recall that $Q(M_o)$ is the set of matchable edges relative to M_o).

If the event F is true then $S(i+1) = \min\{j > S(i): Y_j \notin B\}$. Using this and the strong Markov property of $(Y_k: k \geq 0)$ we have

$$\begin{aligned} E[S(i+1) - S(i)|S(i),(Y_0, \ldots, Y_{S(i)})]I_F &= E[\min\{j > S(i): Y_j \notin B\} - S(i)|Y_{S(i)} = M_o]I_F \\ &= E[S|Y_0 = M_o]I_F, \end{aligned} \tag{2.3.1}$$

where S denotes the stopping time $S = \min\{j \geq 1: Y_j \notin B\}$.

Let $\bar{B}$ be the set of matchings $\bar{B} = \{M: |M| \geq m-1\}$. Note that $B \subset \bar{B}$. We let $(\bar{Y}_k: k \geq 0)$ denote a stationary-transition Markov chain with state space $\bar{B}$ and one-step transition probabilities determined by conditioning $(Y_k: k \geq 0)$ to stay in $\bar{B}$ for each consecutive jump:

$$P[\bar{Y}_{k+1} = M'|\bar{Y}_k = M] = P[Y_{k+1} = M'|Y_k = M]/\Delta(M)$$

for M, M′ in B, where $\Delta(M) = P\{Y_{k+1} \in \bar{B} | Y_k = M\}$.

Define a stopping time $\bar{S}_+$ by $\bar{S}_+ = \min\{k \geq 1: \bar{Y}_k \in \bar{B} - B\}$. Let $\bar{S}_-$ denote a random variable on the same (or possibly enlarged) probability space as $(\bar{Y}_0, \bar{Y}_1, \ldots)$ such that

$$P[\bar{S}_- > k | \bar{Y}_0, \bar{Y}_1, \ldots] = \prod_{j=0}^{k-1} \Delta(\bar{Y}_j).$$

Let $\bar{S} = \min\{\bar{S}_+, \bar{S}_-\}$. If we impose the conditions $\bar{Y}_0 = M_o$ and $Y_0 = M_o$ then $(\bar{S}, (\bar{Y}_k: 0 \leq k < \bar{S}))$ and $(S, (Y_k: 0 \leq k < S))$ have the same distribution. Since $\bar{S} \leq \bar{S}_+$, it follows that

$$E[S | Y_0 = M_o] \leq E[\bar{S}_+ | \bar{Y}_0 = M_o]. \tag{2.3.2}$$

Lemma 2.3.2a is implied by (2.3.1), (2.3.2) and Lemma 2.3.2b, which is stated and proved next.

□

Lemma 2.3.2b: Under the conditions given in the proof of Lemma 2.3.2a

$$E[\bar{S}_+ | \bar{Y}_0 = M_o] \leq 2\omega^*.$$

Proof of Lemma 2.3.2b: Either $|M_o| = m$ or $|M_o| = m-1$. We will prove that if $|M_o| = m$, then

$$E[\bar{S}_+ | \bar{Y}_0 = M_o] \leq 2\omega^* - 1. \tag{2.3.3}$$

This will imply the lemma in general. Hence, we assume that $|M_o| = m$ for the rest of the proof of Lemma 2.3.2b.

For any matching M, let f(M) denote the length of the shortest augmenting path for M. The function f(M) is well defined if M is not a maximum matching (in particular if $|M| = m$), and $f(M) \in \{1, 3, 5, \ldots\}$. Let $\bar{L}$ denote the maximum of f(M) over all M with $|M| = m$.

Claim 2.3.1 [15] $\bar{L} \le 2\beta-1$.

Proof Let M be a matching with $|M| = m$ and let M^* be a maximum cardinality matching in G. Let G′ denote the graph with a set of nodes V and a set of edges $M \oplus M^*$, where $M \oplus M^*$ denotes the symmetric difference of M and M^*. Each node in G′ has incident to it at most one edge from M and at most one edge from M^*. Thus, all maximal connected components of G′ are paths or cycles, and all cycles have even length. The cycles and even length paths each have an equal number of edges from M and M^*, while each odd length path has exactly one more edge of M^* than M and is an augmenting path for M. Thus, there are at least $m^* - m$ node-disjoint augmenting paths for M, which, altogether, have at most m edges of M. Thus, one of the augmenting paths has no more than $m/(m^*-m)$ edges of M and hence has length at most $1+2m/(m^*-m)$. Finally, $1+2m/(m^*-m) \le 2\beta-1$, since $m \le m^*(1-1/\beta)$, and the proof is complete.

□

Claim 2.3.2. Suppose M is a matching with $|M| = m$ and define p_0 and p_1 by

$$p_0 = \frac{1}{2m} \qquad \text{and} \qquad p_1 = \min\{\frac{d^*}{m}, 1 - p_0\}.$$

Then for all $k \ge 0$

(a) $P[f(\bar{Y}_{2k+2}) \le f(M) - 2 \,|\, \bar{Y}_{2k} = M] \ge p_0$ if $f(M) \ge 3$,

(b) $P[|\bar{Y}_{2k+1}| \ge m+1 \,|\, \bar{Y}_{2k} = M] \ge p_0$ if $f(M) = 1$,

(c) $P[f(\bar{Y}_{2k+2}) > f(M) + 2 \,|\, \bar{Y}_{2k} = M] = 0$,

(d) $P[f(\bar{Y}_{2k+2}) = f(M) + 2 \,|\, \bar{Y}_{2k} = M] \le p_1$.

Proof. We will first prove part (a) under the assumption that $f(M) \ge 5$. Choose an augmenting path p for M of length f(M) and label some of the nodes and edges of it as indicated in Figure 2.3.1. Since p is an augmenting path of the shortest length, no neighbor of u_1, except possibly

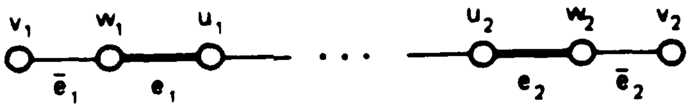


Figure 2.3.1. An augmenting path for M.

node v_1, can be an exposed node. Also, if u_1 and v_1 are neighbors, then w_1 and v_2 are not. Thus, there are at most two choices for an edge e′, namely e_1 and possibly either $[u_1,v_1]$ or $[w_1,v_2]$, such that $f(M - e_1 + e') \geq f(M)$. There is also at least one choice of e′, namely $e' = \bar{e}_1$, such that $f(M - e_1 + e') = f(M) - 2$. Thus,

$$P[f(\bar{Y}_{2k+2}) = f(M) - 2 \,|\, \bar{Y}_{2k+1} = M - e_1] \geq 1/3.$$

This is true with e_1 replaced by e_2 as well, so

$$P[f(\bar{Y}_{2k+2}) = f(M) - 2 \,|\, \bar{Y}_{2k} = M] \geq P[\bar{Y}_{2k+1} = M - e_1 \text{ or } \bar{Y}_{2k+1} = M - e_2 \,|\, \bar{Y}_{2k} = M]/3$$

$$= \frac{2}{3|M|} \geq \frac{1}{2|M|} = p_0.$$

This establishes part (a) if $f(M) \geq 5$.

We will now complete the proof of part (a) by considering the case $f(M) = 3$. Let $[v_1, w_1, w_2, v_2]$ be an augmenting path for M of length 3 and let $e = [w_1, w_2]$. Then e is in M, and nodes v_1 and v_2 are not neighbors. Now, if e' is an edge such that

$$f(M - e + e') \geq 3 \tag{2.3.4}$$

then e' must be incident to either v_1 or w_1 and to either v_2 or w_2.

Moreover, if $e' = [v_1, w_2]$ is such an edge, then v_2 and w_1 must not be neighbors. Thus, there are at most two choices of e' such that (2.3.4) is true, namely e and possibly one of $[v_1, w_2]$ or $[v_2, w_1]$. There are also at least two values of e' such that $f(M - e + e') = 1$, namely $[v_1, w_1]$ and $[v_2, w_2]$. Thus,

$$P[f(\bar{Y}_{2k+2}) = f(M) - 2 \,|\, \bar{Y}_{2k} = M] \geq \frac{2}{4} P[\bar{Y}_{2k+1} = M - e \,|\, \bar{Y}_{2k} = M] = p_0.$$

Part (a) is proved.

Turning to part (b), assume that $f(M) = 1$. Then $Q(M)$ is not empty. Hence,

$$P[\,|\bar{Y}_{2k+1}| \geq m+1 \,|\, \bar{Y}_{2k} = M] = P[\,|\bar{Y}_{2k+1}| = m+1 \,|\, \bar{Y}_{2k} = M] = |Q(M)|/(|Q(M)| + \lambda m)$$

$$\geq (1+m\lambda)^{-1} \geq (1+m)^{-1} \geq p_0,$$

so that part (b) is proved.

Parts (c) and (d) will now be proved. Choose an augmenting path p for M of length f(M). Let

$$\Gamma_+ = \{(e_1,e_2)\colon e_1 \in M, e_2 \text{ is matchable relative to } M - e_1, \text{ and } f(M - e_1 + e_2) \geq f(M) + 2\}.$$

Suppose $(e_1,e_2) \in \Gamma_+$. Then e_1 and e_2 are incident to a common node (otherwise e_1 is matchable relative to $M - e_1 + e_2$, e_2 is matchable relative to M, and hence $f(M - e_1 + e_2) = f(M) = 1$, a contradiction) and $e_1 \neq e_2$. Since p is not an augmenting path for $M - e_1 + e_2$, at least one of e_1 or e_2 is incident to a node of p. This means that either e_1 is an edge of p or e_2 is incident to one of the exposed nodes on the ends of p. Thus, we have narrowed down the possibilities to one of the four cases shown in Figure 2.3.2. We can rule out the first three of these cases, because in these cases there is an augmenting path for $M - e_1 + e_2$ with length at most the length of p. We have thus

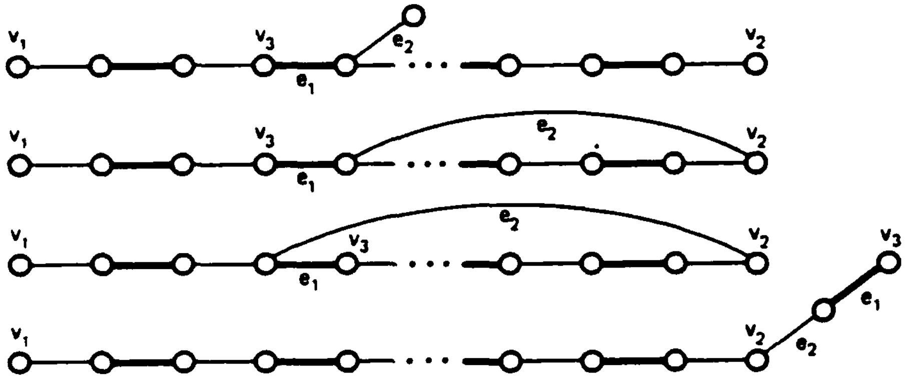


Figure 2.3.2. Four possibilities for (e_1,e_2) are pictured. Edges in the path p are drawn straight and horizontally. Edges in M are bold. Nodes v_1 and v_3 are the end nodes of an augmenting path for $M - e_1 + e_2$. Only the fourth possibility can really occur.

shown that if $(e_1,e_2) \in \Gamma_+$, then e_2 is incident to an exposed node of p, e_1 and e_2 are incident to a common node, and e_1 is not in the path p. It follows that $f(M - e_1 + e_2) = f(M) + 2$, for any (e_1,e_2) in Γ_+, which proves part (c).

Define

$$W = \{e_2 \colon (e_1,e_2) \in \Gamma_+ \text{ for some } e_1\}.$$

If $e \in W$ then there is exactly one edge, call it $w(e)$, such that $(w(e), e) \in \Gamma_+$. Each edge in W is incident to an exposed node of p so that $|W| \leq 2d^*$. Thus,

$$P[f(\bar{Y}_{2k+2}) = f(M)+2 \,|\, \bar{Y}_{2k} = M] = \sum_{(e_1,e_2)\in\Gamma_+} P[\bar{Y}_{2k+2} = M - e_1 + e_2,\ \bar{Y}_{2k+1} = M - e_1 \,|\, \bar{Y}_{2k} = M]$$

$$= \sum_{e \in W} P[\bar{Y}_{2k+2} = M - w(e) + e \,|\, \bar{Y}_{2k+1} = M - w(e)]P[\bar{Y}_{2k+1} = M - w(e) \,|\, \bar{Y}_{2k} = M]$$

$$\leq \sum_{e \in W} \frac{1}{|Q(M-w(e))|} \cdot \frac{1}{|M|} \leq |W| \frac{1}{2} \frac{1}{|M|} \leq \frac{d^*}{|M|}.$$

Together with parts (a), (b), and (c), this proves part (d) so that Claim 2.3.2 is completely proved.

□

We will now complete the proof of Lemma 2.3.2b using Claims 2.3.1 and 2.3.2. Define a process $(U_k\colon k \geq 0)$ by $U_k = \frac{1}{2}(1 + f(\bar{Y}_{2k}))I_{\{2k \leq \bar{S}_+\}}$. Note that U_k takes values in $\{0, 1,..., L\}$ where $L = (1+\bar{L})/2$. Claim 2.3.1 implies that $L \leq \beta$ and Claim 2.3.2 implies that

$$P[U_{k+1} = i \,|\, U_k = j, U_{k-1}, \ldots, U_0] \begin{cases} \geq p_0 & \text{if } i = j-1 \\ = 0 & \text{if } i \geq j+2 \\ \leq p_1 & \text{if } i = j+1 \end{cases} . \qquad (2.3.5)$$

Let $(W_k\colon k \geq 0)$ denote the Markov chain with one-step transition probabilities shown in Figure 2.3.3. From Equation (2.3.5) it follows that if $W_0 = U_0$, then the Markov chain $(W_k\colon k \geq 0)$ stochastically dominates the process $(U_k\colon k \geq 0)$. Hence,

$$E[\bar{S}_+ \,|\, \bar{Y}_0 = M_o] + 1 = 2E[\min\{j\colon U_j = 0\} \,|\, \bar{Y}_0 = M_o]$$

$$\leq 2E[\min\{j\colon W_j = 0\} \,|\, W_0 = f(M_o)]$$

$$\leq 2E[\min\{j\colon W_j = 0\} \,|\, W_0 = L]$$

$$= \frac{2}{p_0} \sum_{j=0}^{L-1} (L-j) \left[\frac{p_1}{p_0}\right]^j$$

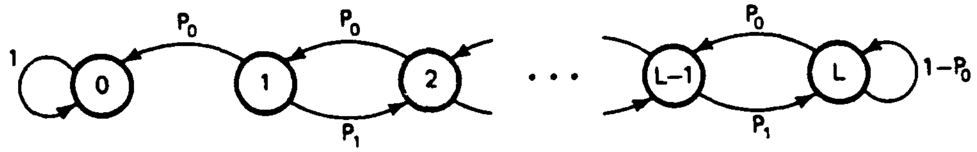


Figure 2.3.3. One-step transition probabilities for the Markov chain $(W_k: k \geq 0)$.

$$\leq \frac{2L}{p_0}\left[\frac{p_1}{p_0}\right]^{L-1} \leq \frac{2\beta}{p_0}\left[\frac{p_1}{p_0}\right]^{\beta-1} \leq 2\omega^*.$$

This establishes Inequality (2.3.3), so the proof of Lemma 2.3.2b, and hence also the proof of Lemma 2.3.2a, is complete.

□

Proof of Lemma 2.3.3: In the first part of the proof, we will refer to the setup in the proof of Lemma 2.3.2a. By the reasoning there, we see that for $i \geq 1$,

$$P[\ |Y_{S(i+1)}\ | > |Y_{S(i)}|\, |S(i),(Y_0, Y_1, \ldots, Y_{S(i)})]I_F = P[\,|Y_S| = m+1\,|Y_0 \fallingdotseq M_o]I_F$$

$$= P[\bar{S}_- > \bar{S}_+ \,|\bar{Y}_0 = M_o]I_F$$

$$= E[\prod_{j=0}^{\bar{S}_+-1} \Delta(\bar{Y}_j)\,|\bar{Y}_0 = M_o]I_{F.}$$

The term $\Delta(\bar{Y}_j)$ is equal to

$$P[Y_{k+1} \in \bar{B}\,|Y_k = \bar{Y}_j] = \begin{cases} \dfrac{|Q(\bar{Y}_j)|}{|Q(\bar{Y}_j)| + \lambda(m-1)} & \text{if } |\bar{Y}_j| = m-1 \\ 1 & \text{if } |\bar{Y}_j| \geq m. \end{cases}$$

As in the proof of Lemma 2.3.2a, we assume that if $|M_o| = m\text{-}1$, then $Q(M_o) \neq \varnothing$. Hence, if $\bar{Y}_0 = M_o$ and $|\bar{Y}_0| = m-1$, then $Q(\bar{Y}_0) \neq \varnothing$. Moreover, if $|\bar{Y}_j| = m-1$ for some $j \geq 1$, then $\bar{Y}_{j-1}$ is a matching containing $\bar{Y}_j$ and so $Q(\bar{Y}_j) \neq \varnothing$. Thus, given $\bar{Y}_0 = M_o$, $Q(\bar{Y}_j) \neq \varnothing$ whenever $|\bar{Y}_j| =$ m-1, for all $j \geq 0$. Therefore, given $\bar{Y}_0 = M_o$ we have $\Delta(\bar{Y}_j) \geq (1+(m-1)\lambda)^{-1} \geq (1+\frac{\lambda|V|}{2})^{-1}$. Also, note that $|\bar{Y}_j| \geq m$ for at least half of the values of j with $0 \leq j \leq \bar{S}_+-1$. Thus,

$$E[\prod_{j=0}^{\bar{S}_+-1} \Delta(\bar{Y}_j)\,|\bar{Y}_0 = M_o] \geq E[(1 + \frac{\lambda|V|}{2})^{-S_+/2}\ \ |\bar{Y}_0 = M_o] \geq (1 + \frac{\lambda|V|}{2})^{-\omega^*}$$

$$\geq \exp(-\lambda|V|\omega^*/2) = \exp(-1/6) \geq 5/6,$$

where for the second inequality we used Lemma 2.3.2b and Jensen's inequality, and for the last two inequalities we used the inequality $\exp(u) \geq 1+u$. Therefore, for $i \geq 1$,

$$P[\ |Z_{i+1}| > |Z_i|\ |Z_0, \ldots, Z_i]I_{\{R(Z) > i\}} \geq (5/6)I_{\{R(Z) > i\}}.$$

Now $|Z_{i+1}| - |Z_i| \in \{-2,-1,1,2\}$ so that

$$E[\ |Z_{i+1}| - |Z_i|\,|Z_0, \ldots, Z_i]I_{\{R(Z) > i\}} \geq (\frac{5}{6} - 2\ .\frac{1}{6})I_{\{R(Z) > i\}} = \frac{1}{2}I_{\{R(Z) > i\}},$$

which implies that the process

$$(|Z_i| - \frac{i-1}{2})I_{\{R(Z) > i\}} + (|Z_{R(Z)}| - \frac{R(Z)-1}{2})I_{\{R(Z) \le i\}}, \quad i = 1, 2,...$$

is a submartingale uniformly bounded from above. Thus, by a version of Doob's optional sampling theorem [16, Theorem 7.4.6.ii]

$$E\left[|Z_{R(Z)}| - \frac{R(Z)-1}{2}\right] \ge E\left[|Z_1| - \frac{1-1}{2}\right] \ge 0,$$

which yields

$$ER(Z) \le 2E|Z_{R(Z)}|+1 \le 2m^*+1 \le |V|+1 \le D_3.$$

Lemma 2.3.3, and hence Theorem 2.3.1, are completely proved.

□

2.4. Simulated Annealing when Temperature Schedules are Constant

In this section, we consider the limitations of restricting our attention to degenerate temperature schedules that have only a single value. The main result of this section is the theorem following the next set of definitions. Let $G'(n,d^*)$ be the set of all graphs with n nodes and maximum node degree d^*, $M'(G)$ the set of all matchings of graph G, and $m^*(G)$ the size of the largest matching of G. Let R(G) be the set of all symmetric transition probability matrices R over M'(G) with the following property: for all $i, j \in M'(G)$, there is a positive integer k and a sequence $i = i(1), i(2), \ldots, i(k) = j$ of matchings from M'(G) such that $R_{i(h)i(h+1)} > 0$ for all h such that $1 \le h \le k-1$. Let c be the cost function on M'(G) such that $c(M) = -|M|$. Suppose R is a transition probability matrix over M'(G). Let $(X_k^{(G,R,T)}: k \ge 0)$ be the annealing process on system (M'(G), c, R) with temperature schedule $(T_k : k \ge 0)$ such that $T_k = T$ for all $k \ge 0$.

Theorem 2.4.1: Suppose β and d^* are real numbers and n is a positive integer such that $\frac{n}{4} \ge \beta \ge \max\{8, \frac{2n}{d^*-1}\}$. Let $\Omega(\beta, n, d^*)$ equal

$$\max_{G\in G'(n,d^*)} \inf_{T\in(0,+\infty]} \min_{R\in R(G)} \max_{M\in M'(G)} E[\min\{k : |X_k^{(G,R,T)}| \geq (1-\beta^{-1})m^*(G)\} \mid X_0 = M].$$

Then

$$\Omega(\beta, n, d^*) \geq \frac{\lfloor \frac{n}{\beta} \rfloor^{\lfloor \frac{\beta}{4} \rfloor}}{\lfloor \frac{n}{\beta} \rfloor + 1} - 1.$$

Remarks: (1) From Theorem 2.4.1, if $\beta \geq \frac{n}{4}$ then

$$\Omega(\beta, n, d^*) \geq \Omega(\frac{n}{4}, n, d^*) \geq \frac{4^{\lfloor \frac{n}{16} \rfloor}}{5} - 1.$$

(2) Note that Theorem 2.3.1 implies that

$$\Omega(\beta, n, d^*) \leq 24\beta^2 n^5 (2d^*)^{2\beta-2}.$$

(3) Theorem 2.4.1 and part (2) of these remarks imply that if β is at least two and is constant on n then $\Omega(\beta, n, n)$ is upper and lower bounded by polynomial functions of n such that the functions have exponents that are linear functions of β.

We will prove Theorem 2.4.1 after presenting two lemmas and defining a graph $\Lambda(\lambda_H, \lambda_L)$, which is a generalization of the graph G_n of Section 2.2. Graph $\Lambda(\lambda_H, \lambda_L)$, shown in Figure 2.4.1, consists of $4\lambda_H\lambda_L$ nodes and has maximum node degree $2\lambda_H + 1$. Just as in graph G_n, the edges of $\Lambda(\lambda_H, \lambda_L)$ are partitioned into the subsets $H_1, H_2, \ldots, H_{\lambda_L}, B_1, B_2, \ldots, B_{\lambda_L-1}$. These subsets are indicated in Figure 2.4.1. For each i, such that $1 \leq i \leq \lambda_L$, H_i consists of $2\lambda_H$ edges that are disjoint, and, for each j, such that $1 \leq j \leq \lambda_L-1$, the subgraph induced by the nodes of the edges of B_j is a complete bipartite graph consisting of $4\lambda_H$ nodes. Also note that $H = \bigcup_{i=1}^{\lambda_L} H_i$ is a unique maximum matching of $\Lambda(\lambda_H, \lambda_L)$. If $\lambda_H = \lambda_L = n$ then $\Lambda(\lambda_H, \lambda_L)$ is isomorphic to G_{2n-1} of Section 2.2.

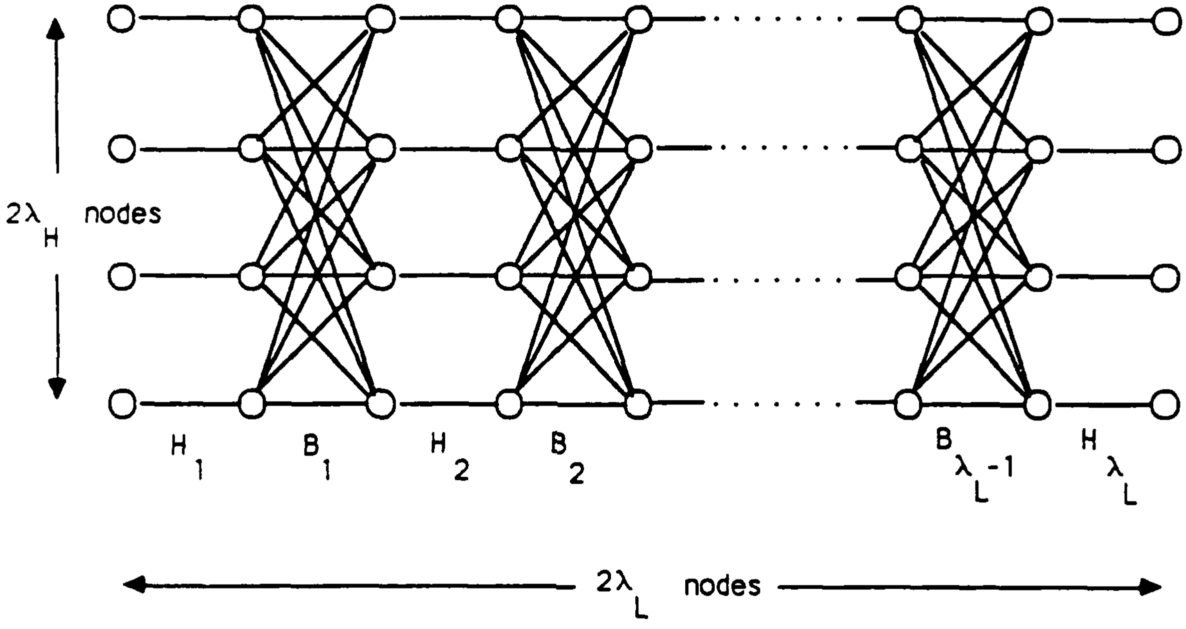


Figure 2.4.1. Sketch of graph $\Lambda(\lambda_H, \lambda_L)$.

Lemma 2.4.1: Let λ_H and n be positive integers such that $n \leq 2\lambda_H - 1$. Let S_n be the set of all matchings of $\Lambda(\lambda_H, \lambda_L)$ of size $|H| - n$. Then

$$\frac{|S_{n+1}|}{|S_n|} \geq \frac{(2\lambda_H - n)^{\lambda_L}}{n+1}.$$

Proof. Suppose M is a matching of S_n. Then $M \oplus H$ is a set of n node-disjoint paths, and each path is characterized by a sequence of edges $e_1, e_2, \ldots, e_{2k+1}$, where $e_1 \in H_j$, $e_2 \in B_j$, $e_3 \in H_{j+1}$, $e_v \in B_{j+1}$, $\ldots$, $e_{2k+1} \in H_{j+k}$ for some j and k. We will call such kinds of paths *stretched* paths. Since $(H \oplus M) \oplus H = M$, there is a one-to-one and onto correspondence between elements of S_n and elements of the set, T_n, of all sets of n node-disjoint stretched paths.

We will now proceed to show that

$$\frac{|T_{n+1}|}{|T_n|} \geq \frac{(2\lambda_H - n)^{\lambda_L}}{n+1}, \qquad (2.4.1)$$

which is sufficient to prove the lemma. Observe that from each element t of T_n we can generate elements of T_{n+1} in the following way. We find a stretched path p that starts in H_1, ends in H_{λ_L}, and is node-disjoint from any of the paths in t. Since $n \leq 2\lambda_H - 1$, there are at least $(2\lambda_H - n)^{\lambda_L}$ such paths p. Then we add the path p to set t to form an element of T_{n+1}. In this way, each element of T_n can generate *at least* $(2\lambda_H - n)^{\lambda_L}$ elements of T_{n+1}. But, each element of T_{n+1} can be generated in this way by at most n+1 elements of T_n. We can then conclude the Inequality (2.4.1).

□

Lemma 2.4.2: Let S be a finite set and c be a cost function on S. Let R be a symmetric transition probability matrix over S such that for all i, j ∈ S, there is positive integer k and a sequence $i = i(1), i(2), \ldots., i(k) = j$ of states in S such that $R_{i(h)i(h+1)} > 0$ for all h such that $0 \leq h \leq k\text{-}1$. Let γ be some constant,

$$\zeta_\gamma = \{s \in S: c(s) \leq \gamma \text{ and } R_{ss'} > 0 \text{ for some } s' \text{ such that } c(s') > \gamma\},$$

$$\gamma_o = \min\{c(s): s \in S - \zeta_\gamma\},$$

and

$$A_\gamma = \{s \in S - \zeta_\gamma: c(s) = \gamma_o\}.$$

Let $(\tilde{X}_k: k \geq 0)$ be the annealing process corresponding to (S, c, R) with a temperature schedule $(T_k: k \geq 0)$ such that $T_k = T$ for all $k \geq 0$. Then for $T > 0$ there is a state $s \in S$ such that

$$E[\min\{k > 0: c(\tilde{X}_k) \leq \gamma\} \,|\, \tilde{X}_0 = s] \;\geq\; |A_\gamma| / |\zeta_\gamma|.$$

Proof: If ζ_γ or A_γ are empty then the lemma is trivial to prove. Thus, we will assume that both ζ_γ and A_γ are not empty. Since the purpose of the proof is to lower bound $E[\min\{k > 0: c(\tilde{X}_k) \leq \gamma\} \,|\, \tilde{X}_0 = s]$, without loss of generality we will assume that all states in ζ_γ

have cost exactly equal to γ. We have assumed that $T > 0$ and R is a symmetric transition probability matrix over S such that for all $i, j \in S$, there is an integer k and a sequence $i = i(1), i(2), \ldots, i(k) = j$ of states in S such that $R_{i(h)i(h+1)}>0$ for all h such that $0 \leq h \leq k-1$, and, therefore, process $(\tilde{X}_k: k \geq 0)$ has the limiting distribution $(\pi_s: s \in S)$, called the Gibbs distribution, where $\pi_s = \exp(-c(s)/T)/\sum_{\sigma\in S}\exp(-c(\sigma)/T)$. Then

$$\sum_{s\in\zeta_\gamma}\frac{\pi_s}{\sum_{\sigma\in\zeta_\gamma}\pi_\sigma}E[\min\{k \geq 1: \tilde{X}_k \in \zeta_\gamma\} \mid \tilde{X}_0 = s] = (\sum_{s\in\zeta_\gamma}\pi_s)^{-1}.$$

Thus, there is a state $\hat{s} \in \zeta_\gamma$ such that

$$(\sum_{s\in\zeta_\gamma}\pi_s)^{-1} \leq E[\min\{k \geq 1: \tilde{X}_k \in \zeta_\gamma\} \mid \tilde{X}_0 = \hat{s}].$$

Since

$$E[\min\{k\geq 1: \tilde{X}_k \in \zeta_\gamma\} \mid \tilde{X}_0 = \hat{s}] = 1 + \sum_{s\in S-\zeta_\gamma} R_{\hat{s}s}\exp(-[c(s)-c(\hat{s})]/T)E[\min\{k \geq 1: \tilde{X}_k \in \zeta_\gamma\} \mid \tilde{X}_0 = s]$$

and $\sum_{s\in\zeta_\gamma} R_{\hat{s}s} \leq 1$, there must exist a state $s \in S-\zeta_\gamma$ such that

$$\exp(-[c(s)-c(\hat{s})]/T)E[\min\{k \geq 1: \tilde{X}_k \in \zeta_\gamma\} \mid \tilde{X}_0 = s] + 1 \geq (\sum_{\sigma\in\zeta_\gamma}\pi_\sigma)^{-1}.$$

The previous inequality implies

$$\exp(-[c(s)-c(\hat{s})]/T)E[\min\{k \geq 1: \tilde{X}_k \in \zeta_\gamma\} \mid \tilde{X}_0 = s] \geq (\sum_{\sigma\in\zeta_\gamma}\pi_\sigma)^{-1} - 1 = \frac{\sum_{\sigma\in S-\zeta_\gamma}\exp(-c(\sigma)/T)}{|\zeta_\gamma|\exp(-\gamma/T)},$$

which in turn implies

$$E[\min\{k \geq 1: \tilde{X}_k \in \zeta_\gamma\} \mid \tilde{X}_0 = s] \geq \frac{\sum_{\sigma\in S-\zeta_\gamma}\exp(-[c(\sigma) - \gamma]/T)}{|\zeta_\gamma|\exp(-[c(s) - c(\hat{s})]/T)}$$

$$\geq \frac{|A_\gamma|\exp(-[\gamma_o - \gamma]/T)}{|\zeta_\gamma|\exp(-[c(s) - c(\hat{s})]/T)}$$

$$\geq \exp(([c(s) - c(\hat{s})] - [\gamma_o - \gamma])/T)\,|A_\gamma|/|\zeta_\gamma|$$

$$\geq |A_\gamma|/|\zeta_\gamma|,$$

and we are done.

□

Proof of Theorem 2.4.1: Let $\lambda_H = \lfloor\frac{n}{\beta}\rfloor$ and $\lambda_L = \lfloor\frac{\beta}{4}\rfloor$. Then $\Lambda(\lambda_H,\lambda_L)$ is a graph with at most n nodes and maximum node degree at most d^*. For positive integer k, let S_k be the set of all matchings of $\Lambda(\lambda_H,\lambda_L)$ of size $|H| - k$. From Lemma 2.4.1, $\frac{|S_{k+1}|}{|S_k|} \geq \frac{(2\lambda_H - k)^{\lambda_L}}{k+1}$ for all $k \leq \lfloor\frac{n}{\beta}\rfloor$. Thus, for all $k \leq \lfloor\frac{n}{\beta}\rfloor$, we have

$$\frac{|S_{k+1}|}{|S_k|} \geq \frac{(2\lambda_H - \lfloor\frac{n}{\beta}\rfloor)^{\lambda_L}}{\lfloor\frac{n}{\beta}\rfloor+1} = \frac{\lfloor\frac{n}{\beta}\rfloor^{\lambda_L}}{\lfloor\frac{n}{\beta}\rfloor+1} = \frac{\lfloor\frac{n}{\beta}\rfloor^{\lfloor\frac{\beta}{4}\rfloor}}{\lfloor\frac{n}{\beta}\rfloor+1}.$$

Let $Q = \frac{\lfloor\frac{n}{\beta}\rfloor^{\lfloor\frac{\beta}{4}\rfloor}}{\lfloor\frac{n}{\beta}\rfloor+1}$. Then $|S_k| \leq (1/Q)^{\lfloor\frac{n}{\beta}\rfloor+1-k}|S_{\lfloor\frac{n}{\beta}\rfloor+1}|$ for all $k \leq \lfloor\frac{n}{\beta}\rfloor$. Therefore,

$$\sum_{k=0}^{\lfloor\frac{n}{\beta}\rfloor}|S_k| \leq |S_{\lfloor\frac{n}{\beta}\rfloor+1}|\sum_{k=0}^{\lfloor\frac{n}{\beta}\rfloor}(1/Q)^{k+1} \leq |S_{\lfloor\frac{n}{\beta}\rfloor+1}|\sum_{k=0}^{\infty}(1/Q)^{k+1}. \qquad (2.4.2)$$

Since we have assumed that $8 \leq \beta \leq \frac{n}{4}$, Q is greater than one, and, thus, the right side of (2.4.2) is finite and is equal to $|S_{\lfloor\frac{n}{\beta}\rfloor+1}|(Q-1)^{-1}$. Hence, it follows that

$$\frac{|S_{\lfloor\frac{n}{\beta}\rfloor+1}|}{\sum_{j=0}^{\lfloor\frac{n}{\beta}\rfloor}|S_j|} \geq Q-1 = \frac{\lfloor\frac{n}{\beta}\rfloor^{\lfloor\frac{\beta}{4}\rfloor}}{\lfloor\frac{n}{\beta}\rfloor+1} - 1. \qquad (2.4.3)$$

Let $G = \Lambda(\lambda_H, \lambda_L)$. Lemma 2.4.2 and (2.4.3) imply that, for any $T > 0$ and any transition probability matrix $R \in R(G)$,

$$\max_{M \in M'(G)} E[\min\{k : |X_k^{(G,R,T)}| \geq |H| - \lfloor\frac{n}{\beta}\rfloor\} \mid X_0^{(G,R,T)} = M] \geq \frac{\lfloor\frac{n}{\beta}\rfloor^{\lfloor\frac{\beta}{4}\rfloor}}{\lfloor\frac{n}{\beta}\rfloor+1} - 1. \qquad (2.4.4)$$

Since $(1 - \beta^{-1})m^*(\Lambda(\lambda_H, \lambda_L)) = (1 - \beta^{-1})|H| \geq |H| - \lfloor\frac{n}{\beta}\rfloor$, Inequality (2.4.4) implies that, for any $T > 0$ and any transition probability matrix $R \in R(G)$,

$$\max_{M \in M'(G)} E[\min\{k : |X_k^{(G,R,T)}| \geq (1-\beta^{-1})m^*(G)\} \mid X_0^{(G,R,T)} = M] \geq \frac{\lfloor\frac{n}{\beta}\rfloor^{\lfloor\frac{\beta}{4}\rfloor}}{\lfloor\frac{n}{\beta}\rfloor+1} - 1,$$

which implies the theorem.

□

2.5. Speculations

We believe that Theorem 2.2.1 is true for constants σ_1 and σ_2 much larger than what we provided in the proof and that ER is significantly smaller than the upper bound given in Theorem 2.3.1. Moreover, we conjecture that for $0 < r < 1$, the average time needed for the controlled processes described in Section 2.2 to reach a matching having cardinality at least the maximum possible minus $|V|^r$ is not upper bounded by a polynomial in $|V|$ for some sequence of graphs. We believe that graph $\Lambda(\lambda_H, \lambda_L)$ of ˏection 2.4 and the techniques of Section 2.2 could be used to prove this conjecture.

The upper bound on ER given in Theorem 2.3.1 is valid for all graphs with a specified number of nodes and maximum node degree value. In the next chapter, we try to bound ER when we restrict our attention to graphs G that are "typical" in some sense.

Our methods of analyzing simulated annealing, like the deterministic methods known for solving the maximum matching problem, do not easily carry over to "industrial strength" variations of the problem or to other problems. More work will be needed to evaluate the average time complexity of simulated annealing and other search heuristics for a wide range of problems.

CHAPTER 3

MATCHING PROBLEM: AVERAGE PERFORMANCE FOR TYPICAL GRAPHS

3.1. Introduction

In Theorem 2.3.1, we presented an upper bound on the average amount of time the basic simulated annealing algorithm of Subsection 2.2.1 takes to find a matching with size at least a fraction $1 - \beta^{-1}$ of the maximum matching. If $\beta = |V|/2$ then the upper bound is

$$6|V|^7(2d^*)^{|V|-2}. \tag{3.1.1}$$

Expression (3.1.1) is an upper bound on the average amount of time the algorithm takes to find a maximum matching of a graph (V,E) with maximum node degree d^*. Therefore, if we exclude graphs with no edges then (3.1.1) is exponential in $|V|$. However, (3.1.1) is a bound applying to all graphs of a specified number of nodes and value of maximum node degree.

Our objective of this chapter is to find an upper bound on the average time complexity of simulated annealing for the matching problem for a "typical" graph. We will make this objective precise after the following definitions. Let m be an increasing positive integer valued function of n. Typically, $m(n) = \lfloor cn^\delta \rfloor$ where δ and c are constants. In this chapter, we will use m as a function only of n, so we write m for m(n). Let G(n,m) be the set of all graphs with node set $\{1, 2, \ldots, n\}$ and m edges. Suppose A_n is the subset of all graphs in G(n,m) with some property Q, and $|A_n|/|G(n,m)| \rightarrow 1$ as $n \rightarrow \infty$. Then we say that *almost every graph* has property Q. Our aim is to find a small function g such that for almost every graph the average time it takes the simulated annealing algorithm of Subsection 2.1.2 to find a maximum matching, if the graph has n nodes, is at most g(n). In this sense, g(n) upper bounds the average time complexity of simulated annealing for the matching problem for a "typical" graph with n nodes and m edges.

Since we do not know of an exact analytical method to find a small function g, we use heuristic approximations of the annealing process to estimate one. The approximations of the annealing process and an estimate for a small g are presented in Section 3.2. The estimate is compared with computer simulations in Section 3.3, and conclusions are given in Section 3.4.

3.2. Approximations and Estimates

We are interested in approximating two processes. The first is the annealing process $(X_k: k \geq 0)$ of the basic simulated annealing algorithm of Subsection 2.1.2 for a "typical" graph in $G(n,m)$ and $T_i = T$ for all $i \geq 0$. Note that since the sequence of temperature values is not decreasing, $(X_k: k \geq 0)$ is not, strictly speaking, an annealing process. We will approximate $(X_k: k \geq 0)$ by a process $(Y_k: k \geq 0)$ that also has a parameter T. We will then use $(Y_k: k \geq 0)$ to approximate

$$E[\min\{k: X_k \text{ is maximum}\} | X_0 = \emptyset], \tag{3.2.1}$$

when the graph is a typical element of $G(n,m)$, and to find a value of T so that (3.2.1) is small.

The second process we are interested in approximating is $(\tilde{X}_k : k \geq 0)$, which is the limit in distribution of the process $(X_{J(k)}: k \geq 0)$ as $T \to 0$, where $J(0) = 0$ and $J(k+1) = \min\{j \geq J(k): X_j \neq X_{J(k)}\}$ for all $k \geq 0$. The following is a procedure for simulating $(\tilde{X}_k: k \geq 0)$. Let $\tilde{X}_0$ equal a matching. Having selected $\tilde{X}_0, \tilde{X}_1, \ldots, \tilde{X}_k$ choose $\tilde{X}_{k+1}$ as follows. If $\tilde{X}_k$ is maximal then choose an edge e at random from $\tilde{X}_k$, all such edges being equally likely, and let $\tilde{X}_{k+1} = \tilde{X}_k - e$. If $\tilde{X}_k$ is not maximal then choose an edge e at random from the set of edges matchable relative to $\tilde{X}_k$, all such edges being equally likely, and let $\tilde{X}_{k+1} = \tilde{X}_k + e$. Note that $(\tilde{X}_k: k \geq 0)$ is not dependent on a temperature parameter, and, by the theorem of Berge and Norman and Rabin [10, Theorem 10.1], it will eventually visit a maximum matching.

We will approximate $(\tilde{X}_k: k \geq 0)$ by $(\tilde{Y}_k: k \geq 0)$, which we define as the limit in distribution as $T \to 0$ of $(X_{J'(k)}: k \geq 0)$, where $J'(0) = 0$ and $J'(k+1) = \min\{j \geq J'(k): Y_j \neq Y_{J'(k)}\}$ for all $k \geq 0$.

Then we will use $(\tilde{Y}_k: k \geq 0)$ to approximate

$$E[\min\{k \geq 0: \tilde{X}_k \text{ is maximum}\} | X_0 = \emptyset], \qquad (3.2.2)$$

when the graph is a "typical" element of G(n,m). We are interested in estimating (3.2.2) for two reasons. First, in Section 3.3, how close $(\tilde{Y}_k: k \geq 0)$ approximates $(\tilde{X}_k: k \geq 0)$ is determined by comparing the estimate (3.2.2) with data from computer simulations. This will be an additional check on how accurately $(Y_k: k \geq 0)$ approximates $(X_k: k \geq 0)$. Second, simulating $(\tilde{X}_k: k \geq 0)$ is an alternative to the simulated annealing algorithm for the matching problem, which at times may be preferable. Hence, we are also interested in its time complexity.

This section is organized as follows. Process $(Y_k: k \geq 0)$ is presented next. Using $(Y_k: k \geq 0)$, we give an estimate of a value of T that should make (3.2.1) small. Finally, we present an estimate for (3.2.1) (resp., (3.2.2)) using $(Y_k: k \geq 0)$ (resp., $(\tilde{Y}_k: k \geq 0)$).

The state space of $(Y_k: k \geq 0)$ is $\bigcup_{i=0}^{\lfloor \frac{n}{2} \rfloor} \{s(i,0), s(i,1), s(i,2), s(i,3)\}$ and each state corresponds to a *type* of matching rather than a particular matching. State s(i,j) corresponds to matchings of size i that have their matchable edges configured in the following way: for j equal to, respectively, 0, 1, 2, or 3, the set of matchable edges is, respectively, empty, a single edge, a path of length three, or two disjoint edges. The transition probability $P[Y_{k+1} = s(a_1,b_1) | Y_k = s(a_0,b_0)]$ approximates the "typical" one-step transition probability of $(X_k: k \geq 0)$ from an $s(a_0,b_0)$-type matching to an $s(a_1,b_1)$-type matching. Note that an s(i,j)-type matching does not exist for all i and j such that $0 \leq i \leq \lfloor \frac{n}{2} \rfloor$ and $0 \leq j \leq 3$. For example, all edges are matchable relative to a matching of size zero, but, there are at most three edges that are matchable relative to an s(0,0)-, s(0,1)-, s(0,2)-, or s(0,3)-type matching. These inconsistencies are due to the fact that process $(Y_k: k \geq 0)$ is based on some assumptions on G(n,m), T, and process $(X_k: k \geq 0)$. We ignore these inconsistencies, because process $(Y_k: k \geq 0)$ is relatively simple and it seems to do a fair job of approximating the behavior of $(X_k: k \geq 0)$.

We will now discuss three assumptions on G(n,m), T, and process $(X_k: k \geq 0)$ that process $(Y_k: k \geq 0)$ is based on. First, we assume that $\frac{n}{2}\exp(-1/T)$ is a small fraction. Under this assumption if X_0 is a matching that is not maximal then with high probability the next matching $(X_k: k \geq 0)$ visits will be a larger matching. We make this assumption because we are only interested in values of T that make (3.2.1) small, and if T is large enough so that the long-term drift of $(X_k: k \geq 0)$ is not towards larger matchings then (3.2.1) will be large.

Second, we assume that the average degree of the graph $(= \frac{2m}{n})$ is small relative to n. For the rest of this chapter we will use d for the average degree. We are most interested in the case when $\frac{d}{n}$ is small, because for a fixed value of n, the smaller d is the more time it takes $(X_k: k \geq 0)$ to reach a maximum matching.

Third, suppose that $(X_k: k \geq 0)$ typically occupies matchings of size i. Then we assume that $n - 2i$ is small. The value $n - 2i$ is an estimate of the number of exposed nodes relative to a matching of size i. The estimate will be accurate if $m \geq cn\log n$ and $c > \frac{1}{2}$, for then maximum matchings typically have cardinality near $\frac{n}{2}$ (see [17]). Our assumption that $n - 2i$ is small should be consistent with the assumption that $\exp(-1/T)\frac{n}{2}$ and $\frac{d}{n}$ are small, because then we would expect that for a large fraction of the time, before reaching a maximum matching, $(X_k: k \geq 0)$ will be in matchings that have size that are near maximum.

Before describing $(Y_k: k \geq 0)$ in more detail we will simplify the notation by letting $p(a_1,b_1;a_0,b_0) = P[Y_{k+1} = s(a_0,b_0) | Y_k = s(a_1,b_1)]$ and $q = n - 2i$. A schematic description of $(Y_k: k \geq 0)$ is given in Figure 3.2.1. Each box in the figure corresponds to the state labeled on its left or upper left corner. Inside each box is the configuration of matchable edges for that state,

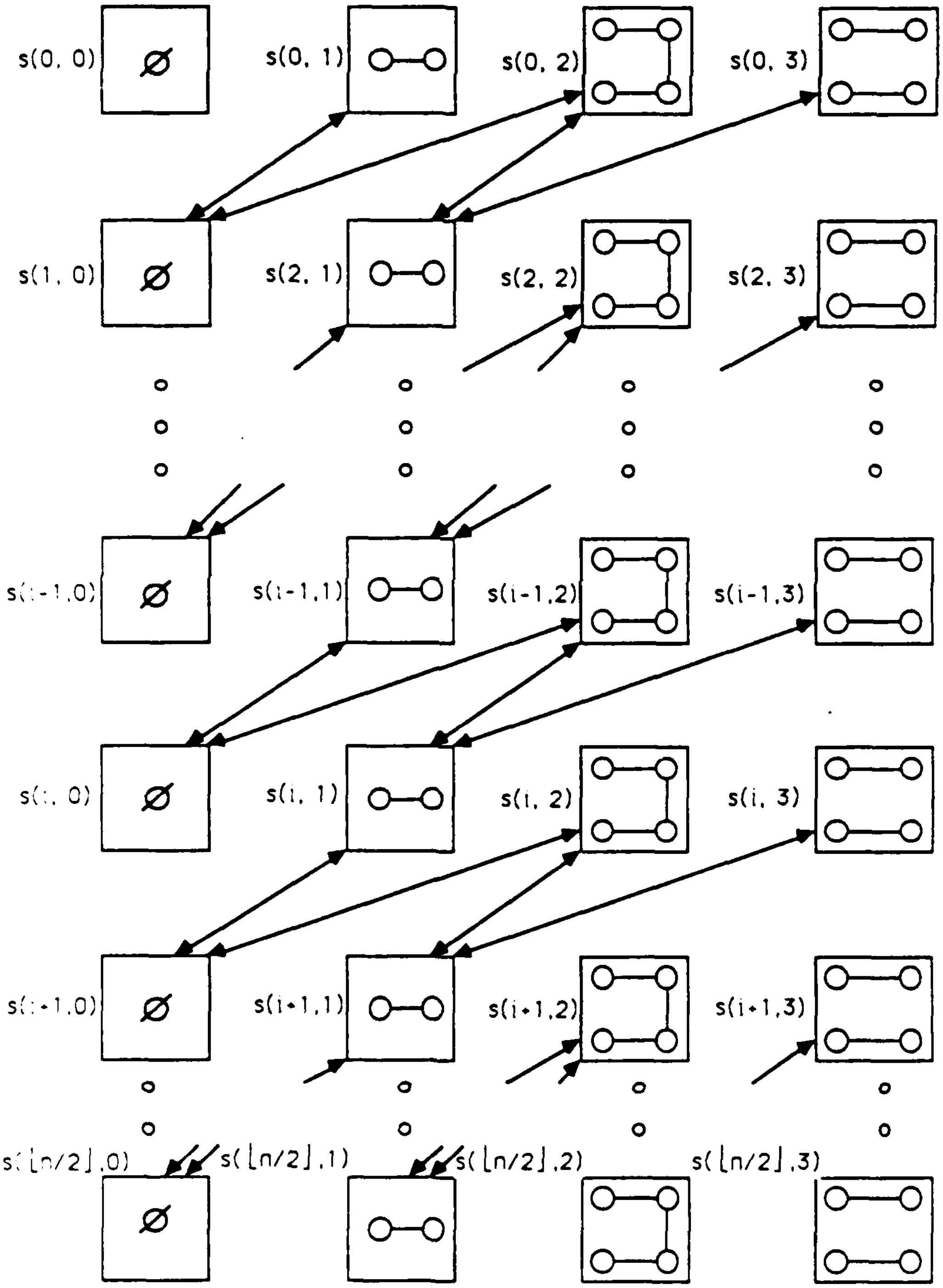


Figure 3.2.1. State diagram of the (Y_k: $k \geq 0$).

where $\varnothing$ means no matchable edges. An arrow from a box corresponding to state $s(a_0,b_0)$ to a box corresponding to $s(a_1,b_1)$ means the transition probability from $s(a_0,b_0)$ to $s(a_1,b_1)$ is positive. We do not indicate by arrows the positive transition probability of a state to itself. The transition probability values are presented in Figure 3.2.2. We will ignore states s(0,0), $s(\lfloor\frac{n}{2}\rfloor,2)$, and $s(\lfloor\frac{n}{2}\rfloor,2)$, because there are no transitions out of these states with strictly positive probability.

Next we will give heuristic reasons for our choice of the states and transition probabilities of $(Y_k: k \geq 0)$. Note that all transition probabilities of $(Y_k: k \geq 0)$ that correspond to increasing the size of the matching are chosen to be consistent with the configuration of matchable edges of the state of which it came from. For example, there are three matchable edges relative to an s(i,2)-type matching. Matching one of these edges gives you an s(i+1,0)-type matching and matching either one of the other two edges gives you an s(i+1,1)-type matching. Hence, $p(i,2;i+1,0) = m^{-1}$ and $p(i,2;i+1,1) = 2m^{-1}$. We assume $\exp(-1/T)\frac{n}{2}$ is small and, therefore, if $(X_k: k \geq 0)$ is in an s(i,2)- or s(i,3)-type matching then with high probability the next matching it visits will be a larger one. Thus, for all $i \leq \lfloor\frac{n}{2}\rfloor$, we ignore the possibility of a transition from either states s(i,2) or s(i,3) that corresponds to decreasing the matching size.

We will now discuss the transitions out of s(i,0), which is the state corresponding to maximal matchings of size i. In this discussion, we will suppose M is a "typical" maximal matching of size i. We assume that for each $e \in M$ the end nodes of e do not have a common neighbor, which is consistent with the assumption that $\frac{d}{n}$ is small. Since M is maximal, for all edges $e \in M$, all matchable edges relative to M-e share a node in common with e. Thus, we can partition M into two subsets A and B, where subset A contains all edges e such that each node of e has at least one neighbor that is exposed relative to M, and subset B contains all edges e such that at most one node of e has a neighbor that is exposed relative to M.

For all i such that $1 \le i \le \lfloor \frac{n}{2} \rfloor$,

$$p(i,0;i-1,2) = \min\{1,((n-2i)(d-1)/n)^2\}i\frac{\exp(-1/T)}{m},$$

$$p(i,0;i-1,1) = \max\{0,1-((n-2i)(d-1)/n)^2\}i\frac{\exp(-1/T)}{m},$$

$$p(i,1;i-1,2) = \min\{2(d-1),i\}\frac{\exp(-1/T)}{m},$$

and

$$p(i,1;i-1,3) = \max\{0,i-2(d-1)\}\frac{\exp(-1/T)}{m}.$$

For all i such that $0 \le i \le \lfloor \frac{n}{2} \rfloor - 1$,

$$p(i,2;i+1,0) = \frac{1}{m},$$

$$p(i,2;i+1,1) = \frac{2}{m},$$

and

$$p(i,3;i+1,1) = \frac{2}{m}.$$

All other transition probabilities are zero, with the exception of transition probabilities from a state to itself.

Figure 3.2.2. Transition probabilities of $(Y_k: k \ge 0)$.

Our estimate of the size of A and our characterization of the types of "typical" edges in sets A and B are based on the following argument. Let e = [i, j] be a "typical" edge of M. We assume that the degree of each node of e is d. We also assume that for each neighboring node v of i or j that is neither i nor j, the probability of v being exposed relative to M is $\frac{q}{n}$. Combining these assumptions on e and the assumption that q and $\frac{d}{n}$ are small we have the following approximations:

P[The nodes of e have no exposed neighbors]

$$\approx (1-\frac{q}{n})^{2(d-1)} \approx 1-\frac{2(d-1)}{n}q, \tag{3.2.3}$$

P[Each node of e has exactly one exposed neighbor]

$$\approx ((d-1)\frac{q}{n}(1-\frac{q}{n})^{d-2})^2 \approx ((d-1)\frac{q}{n})^2, \tag{3.2.4}$$

and

P[Each node of e has at least one exposed neighbor]

$$\approx (1-(1-\frac{q}{n})^{d-1})^2 \approx ((d-1)\frac{q}{n})^2. \tag{3.2.5}$$

Approximations (3.2.3) and (3.2.5) and the assumption that q and $\frac{d}{n}$ are small imply that most of the edges e of B are such that e is the only matchable edge relative to M – e. Hence, if e ∈ B then M – e is likely to be of type s(i-1,1). Approximations (3.2.4) and (3.2.5) imply that most of the edges e of A are such that the set of matchable edges relative to M-e is a path of length three. Therefore, if e ∈ A then M – e is likely to be of type s(i-1,2). Approximation (3.2.5) implies that $|A|$ is approximately $(q(d-1)/n)^2|M|$. Since B = M - A, we approximate $|B|$ by $[1-(q(d-1)/n)^2]|M|$. Based on these approximations, we let $p(i,0;i\text{-}1,2) = \min\{1,(q(d-1)/n)^2\}i\frac{\exp(-1/T)}{m}$ and $p(i,0;i\text{-}1,1) = \max\{0,1-(q(d-1)/n)^2\}i\frac{\exp(-1/T)}{m}$.

We will now discuss the transitions out of s(i,1), which is the state that corresponds to matchings of size i that have exactly one matchable edge relative to it. For $e = [u, v] \notin M$, H_e is the set of edges $[u', v']$ in M such that u' or v' is adjacent to one of u or v. Let U_e be the set of all nodes such that there is a path of length at most four from v to a node of e. The following assumption is consistent with the assumption that $\frac{d}{n}$ and q are small. We assume that the subgraph induced by U_e is a tree, and all nodes in U_e, except the nodes of e, are matched. Under these assumptions, an edge $e' \in M$ is such that $M\text{-}e'$ is an s(i-1,2)-type matching if and only if $e \in H_e$ (see Figure 3.2.3 for an example of H_e under these assumptions). We expect that if $e' \in H_e$ then the set edges matchable relative to $M - e'$ would be a path of length three. Since we assume q and $\frac{d}{n}$ are small, we also expect that most edges $e' \in M - H_e$ are such that the set of edges

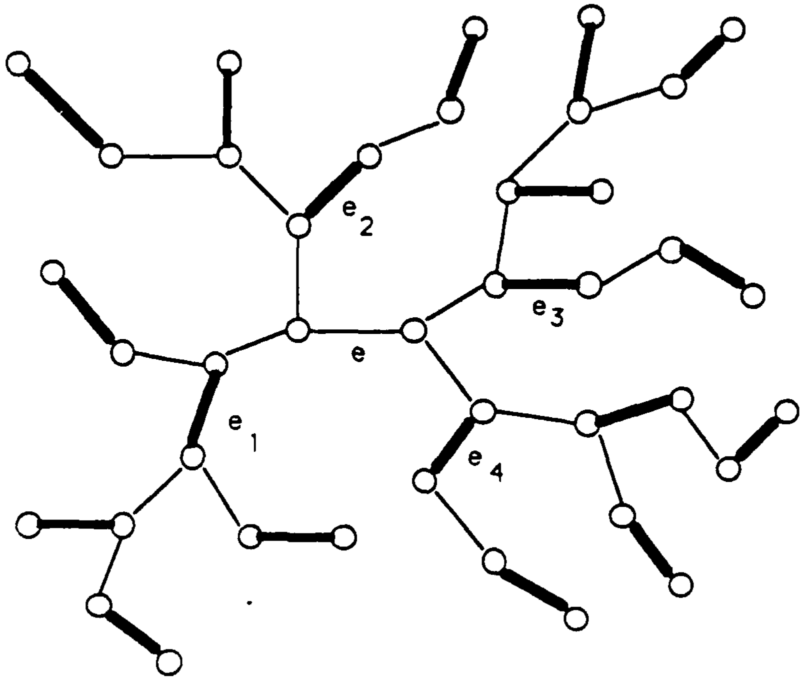


Figure 3.2.3. In this example, the matched edges are bold and $H_e = \{e_1, e_2, e_3, e_4\}$.

matchable relative to $M - e'$ are two disjoint edges. Therefore, we assume that if $e' = M - H_e$ then $M - e'$ is an s(i-1,3)-type matching. Since $|H_e|$ is the sum of the degrees of the nodes of e minus two, we approximate it with $2(d-1)$. Based on these approximations, we let $p(i,1;i\text{-}1,2) = \frac{\min\{2(d-1),i\}}{m}\exp(-1/T)$ and $p(i,1;i\text{-}1,3) = \frac{\max\{0,i-2(d-1)\}}{m}\exp(-1/T)$.

We now turn to determining a value of T that will make (3.2.1) small. Let

$$D(j) = \min\{k\colon Y_k = s(j+1,0) \text{ or } s(j-1,0)\},$$

$$A_{j,j+1} = P[Y_{D(j)} = s(j+1,0) \mid Y_0 = s(j,0)],$$

and

$$A_{j,j-1} = P[Y_{D(j)} = s(j-1,0) \mid Y_0 = s(j,0)] = 1 - A_{j,j+1}.$$

We will approximate (3.2.1) by

$$E[\min\{k \geq 0\colon Y_k = s(\lfloor\tfrac{n}{2}\rfloor,0)\} \mid Y_0 = s(0,1)], \tag{3.2.6}$$

which is equal to

$$\sum_{j=2}^{\lfloor\frac{n}{2}\rfloor-1} E[D(j) \mid Y_0 = s(j,0)]\ \frac{1}{A_{j,j+1}} \sum_{i=j}^{\lfloor\frac{n}{2}\rfloor-1} \prod_{k=j+1}^{i} (A_{k,k-1}/A_{k,k+1})$$
$$+ E[D(1) \mid Y_0 = s(1,0)] \sum_{j=1}^{\lfloor\frac{n}{2}\rfloor-1} \prod_{k=2}^{j} (A_{k,k-1}/A_{k,k+1}) + m, \tag{3.2.7}$$

where $\prod_{k=j+1}^{j} (\cdot) = 1$.

To prevent (3.2.7) from being exponential in n, we want T to be such that $A_{j,j-1}/A_{j,j+1} \leq \frac{1}{2}$ for $j \leq \lfloor\frac{n}{2}\rfloor - 1$. Then (3.2.7) is at most

$$3 \sum_{j=2}^{\lfloor\frac{n}{2}\rfloor-1} E[D(j) \mid Y_0 = s(j,0)] + m. \tag{3.2.8}$$

It can be shown that if $2 \leq i \leq \lfloor\frac{n}{2}\rfloor - 1$ then $A_{i,i-1}/A_{i,i+1}$ equals

$$\max\{0,(\frac{n}{(d-1)q})^2-1\}\min\{i-1,2(d-1)\}\exp(-1/T)\frac{1}{2}\frac{3+\min\{i,2(d-1)\}\exp(-1/T)}{3+\min\{i-1,2(d-1)\}\exp(-1/T)}$$

$$\leq (\frac{n}{(d-1)(n-2i)})^2(d-1)\exp(-1/T)\frac{4}{3}$$

$$\leq \frac{n^2}{d-1}\exp(-1/T)\frac{1}{3}.$$

Thus, if T satisfies $\frac{n^2}{(d-1)}\exp(-1/T)\frac{2}{3}\leq 1$ then $A_{i,i-1}/A_{i,i+1} \leq \frac{1}{2}$ for $2 \leq i \leq \lfloor\frac{n}{2}\rfloor - 1$. Although we want T to satisfy $\frac{n^2}{(d-1)}\exp(-1/T)\frac{2}{3}\leq 1$ we do not want T to be too small, otherwise $E[D(i)|Y_0 = s(i,0)]$ will become very large. Hence, a value of T that makes (3.2.7) small is one where

$$\exp(-1/T) = c\frac{(d-1)}{n^2},$$

and c is a constant ≤ 1.5. If $(Y_k\colon k \geq 0)$ is an accurate approximation of $(X_k\colon k \geq 0)$ then such a value of T will also make (3.2.1) small.

We now turn to evaluating $E[D(i)|Y_0 = s(i,0)]$. The exact expression for $E[D(i)|Y_0 = s(i,0)]$ is complicated, so we will approximate it with

$$\frac{1.5m}{\exp(-1/T)\min\{1,(\frac{d-1}{n}(n-2i))^2\}i}. \tag{3.2.9}$$

This is an accurate approximation if T is very small. Using Approximation (3.2.9), we get the following approximation for (3.2.8):

$$4.5m\left[\exp(1/T)\sum_{i=2}^{\lfloor\frac{n}{2}\rfloor-1}\frac{1}{\min\{1,(\frac{d-1}{n}(n-2i))^2\}i}\right]+m$$

$$\leq (4.5m)\exp(1/T)\left[\sum_{i=2}^{\lfloor\frac{n}{2}\rfloor-1}\frac{1}{i} + \sum_{i=2}^{\lfloor\frac{n}{2}\rfloor-1}\frac{1}{i(\frac{d-1}{n}(n-2i))^2})\right]+m$$

$$\begin{aligned}
&\approx (4.5m)\exp(1/T)\left[\log(n) + \frac{n^2}{(d-1)^2}\left[\sum_{i=2}^{\lfloor\frac{n}{6}\rfloor}\frac{1}{i(n-2i)^2} + \sum_{i=\lfloor\frac{n}{6}\rfloor+1}^{\lfloor\frac{n}{2}\rfloor-1}\frac{1}{i(n-2i)^2}\right]\right] \\
&\le (4.5m)\exp(1/T)\left[\log(n) + \frac{n^2}{(d-1)^2}\left[\sum_{i=2}^{\lfloor\frac{n}{6}\rfloor}\frac{1}{2(n-4))^2} + \frac{1}{\lfloor\frac{n}{6}\rfloor+1}\sum_{i=\lfloor\frac{n}{6}\rfloor+1}^{\lfloor\frac{n}{2}\rfloor-1}\frac{1}{(n-2i)^2}\right]\right] \\
&\approx (4.5m)\exp(1/T)\left[\log(n) + \frac{n^2}{(d-1)^2}\left[\frac{n/6}{2n^2} + \frac{6}{n}\sum_{i=1}^{\infty}\frac{1}{4i^2}\right]\right] \\
&\le (4.5m)\exp(1/T)(\log(n) + \frac{5}{2}\frac{n}{(d-1)^2}) \\
&\approx \frac{9}{8}\exp(1/T)(2dn\log(n) + 5\frac{n^2}{d}).
\end{aligned} \qquad (3.2.10)$$

Finally, if $\exp(-1/T) \le 1.5\frac{(d-1)}{n^2}$ then Expression (3.2.10) is our estimate of (3.2.1). Note that $E[D(\lfloor\frac{n}{2}\rfloor-1)\,|\,Y_0 = s(\lfloor\frac{n}{2}\rfloor-1),0)] \ge \frac{1}{12}\exp(1/T)\frac{nm}{(d-1)^2} \approx \frac{1}{24}\exp(1/T)\frac{n^2}{d}$. Thus, if $(Y_k\colon k \ge 0)$ is a good approximation for $(X_k\colon k \ge 0)$ and $d = o((n/\log n)^{0.5})$ then $(X_k\colon k \ge 0)$ will be such that for a large portion of its time before it visits a maximum matching it will be in near maximum matchings. This is consistent with one of our assumptions used to define $(Y_k\colon k \ge 0)$.

If c is a constant such that $c \le 1.5$ and $\exp(-1/T) = c\frac{(d-1)}{n^2}$, then (3.2.10) will be approximately

$$\frac{9}{4c}n^3\log(n) + \frac{45}{8c}\frac{n^4}{d^2}. \qquad (3.2.11)$$

Therefore, if $(Y_k\colon k \ge 0)$ is a good approximation of $(X_k\colon k \ge 0)$, then (3.2.11) is an upper bound on (3.2.1) for almost every graph.

We now turn to approximating (3.2.2). Recall that $(\tilde{Y}_k\colon k \ge 0)$ is the limit in distribution of $(Y_{J(k)}\colon k \ge 0)$ as $T \to 0$, where $J(0) = 0$ and, for $k \ge 0$, $J(k+1) = \min\{j \ge J(k)\colon Y_j \ne Y_{J(k)}\}$. Let $t_0 = 0$ and $t_k = \min\{j \ge 0\colon \tilde{Y}_j = s(k,0)\}$. It is straightforward to show that

$E[t_{k+1}-t_k \,|\, \tilde{Y}_0 = s(0,1)] = 3\max\{(\frac{n}{(d-1)(n-2k)})^2,1\}+1$. Then

$$E[\min\{j \geq 0{:}\tilde{Y}_j = (\lfloor\frac{n}{2}\rfloor,0)\} \,|\, \tilde{Y}_0 = s(0,1)] \;=\; 3\sum_{i=1}^{\lfloor\frac{n}{2}\rfloor-1} [\max\{(\frac{n}{(d-1)(n-2i)})^2,1\}+1] + 1$$

$$\leq \frac{3}{2}(\frac{n^2}{(d-1)^2} + n). \qquad (3.2.12)$$

Finally, we will use (3.2.12) as our approximation for (3.2.2). If $(Y_k\text{: } k \geq 0)$ is a good approximation of $(X_k\text{: } k \geq 0)$ then (3.2.12) is an upper bound on (3.2.2) for almost every graph. Note that $E[t_{\lfloor\frac{n}{2}\rfloor} - t_{\lfloor\frac{n}{2}\rfloor-1} \,|\, \tilde{Y}_0 = s(0,1)] \geq \frac{3}{4}(\frac{n}{(d-1)})^2 + 1$. If our approximations are correct then $(\tilde{X}_k\text{: } k \geq 0)$ will spend a large portion of its time in near maximum matchings before finding a maximum matching.

3.3. Experimental Results

In this section, we will experimentally evaluate how well $(Y_k\text{: } k \geq 0)$ approximates $(X_k\text{: } k \geq 0)$ and how well $(\tilde{Y}_k\text{: } k \geq 0)$ approximates $(\tilde{X}_k\text{: } k \geq 0)$. First, we will focus on evaluating the accuracy of $(\tilde{Y}_k\text{: } k \geq 0)$. We check the accuracy of $(\tilde{Y}_k\text{: } k \geq 0)$ by using it to predict the sample means and sample standard deviations of $\bar{J} = \min\{j\text{: } \tilde{X}_j \text{ is maximum}\}$ and $\tilde{t} = \bar{t}_{m^*} - \bar{t}_{m^*-1}$, where m^* is the size of the largest matching and $\bar{t}_k = \min\{j\text{: } |\tilde{X}_j|=k\}$. Recall that G(n,m) is the set of all graphs with node set $\{1, 2, \ldots ,n\}$ and m edges. If $(\tilde{Y}_k\text{: } k \geq 0)$ is an accurate approximation of $(\tilde{X}_k\text{: } k \geq 0)$ then for most graphs in G(n,m) the mean of $\bar{J}$ will be (3.2.12) and the mean of $\tilde{t}$ will be approximately $E[t_{\lfloor\frac{n}{2}\rfloor} - t_{\lfloor\frac{n}{2}\rfloor-1} \,|\, Y_0 = s(0,1)]$, which is approximately $\frac{3}{4}(\frac{n}{(d-1)})^2$ if n is even.

We collected data from simulations of $(\tilde{X}_k\text{: } k \geq 0)$ as follows. For each $\delta \in \{0, 0.25, 0.5, 0.75, 1.0\}$, we let $m = \lfloor c_0 n^{1+\delta} \rfloor$, where $c_0 = 80(32)^{-(1+\delta)}$, and we considered five values of n: 32, 64, 128, 256, and 512. For each (δ,n) pair, we randomly and without bias selected one hundred graphs from G(n,m) with replacement. For each graph, we simulated $(\tilde{X}_k: k \geq 0)$ once. From these one hundred simulations we computed the sample mean $m(\tilde{J})$ and standard deviation $\sigma(\tilde{J})$ of $\tilde{J}$, and we computed the sample mean $m(\tilde{\tau})$ and standard deviation $\sigma(\tilde{\tau})$ of $\tilde{\tau}$. Note that $m(\tilde{J})$ (resp., $m(\tilde{\tau})$) is an estimate of the mean of (3.2.2) (resp., $E[\tilde{\tau} | \tilde{X}_0 = \emptyset]$) over all graphs of G(n,m). The sample means and standard deviations are given in the tables in Figure 3.3.1. These tables also contain the values of (3.2.12), which are listed under $a(\tilde{J})$, and the values of $\frac{3}{4}(\frac{n}{(d-1)})^2$, which are listed under $a(\tilde{\tau})$.

Note that the $m(\tilde{\tau})$ is at least a half of $m(\tilde{J})$, which is consistent with our prediction that $(\tilde{X}_k: k \geq 0)$ will spend a large portion of its time in near maximum matchings.

We estimate the rate at which $m(\tilde{\tau})$ and $m(\tilde{J})$ grow with n using the following procedure. First, for each δ, we find an argument $(\alpha(\tilde{\tau}),\beta(\tilde{\tau}))$ of

$$\min_{\alpha,\beta} \sum_{n=32,64,128,256,512} (\log\tilde{\tau}(n) - \log(e^{\alpha}n^{\beta}))^2,$$

where $\tilde{\tau}(n) = m(\tilde{\tau})$ for the graphs with n nodes. In this sense, for the value of δ, $e^{\alpha(\tilde{\tau})}n^{\beta(\tilde{\tau})}$ is the "best-fit" curve to values of $m(\tilde{\tau})$ as a function of n. Similarly, we find a pair $(\alpha(\tilde{J}),\beta(\tilde{J}))$ for values of $m(\tilde{J})$. The values of $(\alpha(\tilde{\tau}),\beta(\tilde{\tau}))$ and $(\alpha(\tilde{J}),\beta(\tilde{J}))$ are presented in Table 3.3.1. If $(\tilde{Y}_k: k \geq 0)$ is an accurate estimate of $(\tilde{X}_k: k \geq 0)$ we would expect $\beta(\tilde{\tau})$ to be approximately equal to $2(1-\delta)$ and $\beta(\tilde{J})$ to be approximately equal to $\max\{2-2\delta,1\}$. We also list the values of $\max\{2-2\delta,1\}$ (resp., $2(1-\delta)$) under $\beta(J^*)$ (resp., $\beta(\tau^*)$).

Table 3.3.1 shows that our predictions, (3.2.12) and $\frac{3}{4}(\frac{n}{(d-1)})^2$, tend to be more accurate for sparse graphs. This is not surprising, since our approximations were based on the assumption of a sparse graph. In addition, our predictions seem to be asymptotic upper bounds of the sample means.

$\delta = 0.00$							
n	m	m($\bar{J}$)	σ($\bar{J}$)	a($\bar{J}$)	m(τ)	σ(τ)	a(τ)
32	80	2.02e+02	2.93e+02	1.44e+02	1.61e+02	2.77e+02	4.80e+01
64	160	8.93e+02	6.64e+02	4.80e+02	6.67e+02	6.21e+02	1.92e+02
128	320	2.90e+03	2.41e+03	1.73e+03	2.01e+03	2.23e+03	7.68e+02
256	640	1.24e+04	1.06e+04	6.53e+03	8.37e+03	1.01e+04	3.07e+03
512	1280	4.00e+04	2.96e+04	2.53e+04	2.49e+04	2.83e+04	1.23e+04

$\delta = 0.25$							
n	m	m($\bar{J}$)	σ($\bar{J}$)	a($\bar{J}$)	m(τ)	σ(τ)	a(τ)
32	80	2.02e+02	2.93e+02	1.44e+02	1.61e+02	2.77e+02	4.80e+01
64	190	5.51e+02	4.65e+02	3.48e+02	4.07e+02	4.23e+02	1.26e+02
128	452	1.24e+03	8.17e+02	8.61e+02	8.04e+02	7.54e+02	3.34e+02
256	1076	3.19e+03	2.06e+03	2.18e+03	2.18e+03	2.04e+03	8.96e+02
512	2560	8.33e+03	5.51e+03	5.62e+03	5.52e+03	5.36e+03	2.43e+03

$\delta = 0.50$							
n	m	m($\bar{J}$)	σ($\bar{J}$)	a($\bar{J}$)	m(τ)	σ(τ)	a(τ)
32	80	2.02e+02	2.93e+02	1.44e+02	1.61e+02	2.77e+02	4.80e+01
64	226	3.24e+02	2.35e+02	2.63e+02	2.32e+02	2.17e+02	8.36e+01
128	640	5.98e+02	5.17e+02	4.95e+02	3.87e+02	5.04e+02	1.52e+02
256	1810	1.06e+03	8.48e+02	9.53e+02	6.68e+02	8.09e+02	2.85e+02
512	5120	1.77e+03	9.49e+02	1.86e+03	1.03e+03	9.52e+02	5.45e+02

$\delta = 0.75$							
n	m	m($\bar{J}$)	σ($\bar{J}$)	a($\bar{J}$)	m(τ)	σ(τ)	a(τ)
32	80	2.02e+02	2.93e+02	1.44e+02	1.61e+02	2.77e+02	4.80e+01
64	269	2.35e+02	1.68e+02	2.08e+02	1.63e+02	1.61e+02	5.60e+01
128	905	2.96e+02	1.68e+02	3.34e+02	1.81e+02	1.63e+02	7.12e+01
256	3044	3.80e+02	1.98e+02	5.73e+02	1.79e+02	1.92e+02	9.47e+01
512	10240	6.81e+02	3.22e+02	1.03e+03	3.15e+02	3.19e+02	1.29e+02

$\delta = 1.00$							
n	m	m($\bar{J}$)	σ($\bar{J}$)	a($\bar{J}$)	m(τ)	σ(τ)	a(τ)
32	80	2.02e+02	2.93e+02	1.44e+02	1.61e+02	2.77e+02	4.80e+01
64	320	1.35e+02	7.83e+01	1.72e+02	8.22e+01	7.23e+01	3.79e+01
128	1280	1.54e+02	6.56e+01	2.60e+02	6.83e+01	6.25e+01	3.40e+01
256	5120	2.36e+02	1.03e+02	4.49e+02	9.01e+01	9.78e+01	3.23e+01
512	20480	3.52e+02	7.09e+01	8.31e+02	7.50e+01	6.76e+01	3.15e+01

Figure 3.3.1. Tables of sample means m(τ) and m($\bar{J}$); sample standard deviation σ(τ) and σ($\bar{J}$); and our estimate a(τ) (resp., a($\bar{J}$)) of τ (resp., $\bar{J}$), based on the approximation ($\bar{Y}_k$: k ≥ 0) of ($\bar{X}_k$: k ≥ 0). The graphs considered have n nodes and $m = \lfloor c_0 n^{1+\delta} \rfloor$ edges.

Table 3.3.1. Function $e^{\alpha(\bar{\tau})}n^{\beta(\bar{\tau})}$ of n is the best-fit curve of $m(\bar{\tau})$ (resp., $m(\bar{J})$) as a function of n. The value of $\beta(\tau^*)$ (resp., $\beta(J^*)$) is the estimate of $\beta(\bar{\tau})$ (resp., $\beta(\bar{J})$) based on the approximation $(\tilde{Y}_k: k \geq 0)$ of $(X_k: k \geq 0)$.

δ	$\alpha(\bar{J})$	$\beta(\bar{J})$	$\beta(J^*)$	$\alpha(\bar{\tau})$	$\beta(\bar{\tau})$	$\beta(\tau^*)$
0.00	-1.22	1.91	2.00	-1.16	1.82	2.00
0.25	0.73	1.33	1.50	0.69	1.26	1.50
0.50	2.52	0.80	1.00	2.64	0.69	1.00
0.75	3.75	0.42	1.00	4.25	0.21	0.50
1.00	4.15	0.24	1.00	5.51	-0.21	0.00

In Figure 3.3.2, we have three graphs, corresponding to $\delta = 0, 0.5, 1.0$. In each graph, we plot $m(\bar{J})$, $e^{\alpha(\bar{J})}n^{\beta(\bar{J})}$, and (3.2.12) versus n corresponding to the particular value of δ for that graph. In Figure 3.3.3, we plot three graphs of $m(\bar{\tau})$, $e^{\alpha(\bar{\tau})}n^{\beta(\bar{\tau})}$, and $\frac{3}{4}(\frac{n}{(d-1)})^2$ versus n for δ equal to 0, 0.5, 1.0. In these graphs, the best-fit curves approximate the sample mean data fairly accurately, except for the curve approximating $m(\bar{J})$ when $\delta = 0$. These graphs also indicate that (3.2.12) (resp., $\frac{3}{4}(\frac{n}{(d-1)})^2$) is an asymptotic upper bound on $m(\bar{J})$ (resp., $m(\bar{\tau})$).

Next we will determine how accurately $(Y_k: k \geq 0)$ approximates $(X_k: k \geq 0)$ when $\exp(-1/T)$ is approximately $\frac{3}{4}\frac{(d-1)}{n^2}$. We check the accuracy of $(Y_k: k \geq 0)$ by using it to predict the sample means and the sample standard deviations of $\bar{I} = \min\{k: X_k \text{ is maximum}\}$. If $(Y_k: k \geq 0)$ is an accurate approximation of $(X_k: k \geq 0)$, then for most graphs in G(n,m) the mean of $\bar{I}$ will be bounded above by (3.2.10), which is approximately $3n^3\log n + 7.5\frac{n^4}{d^2}$.

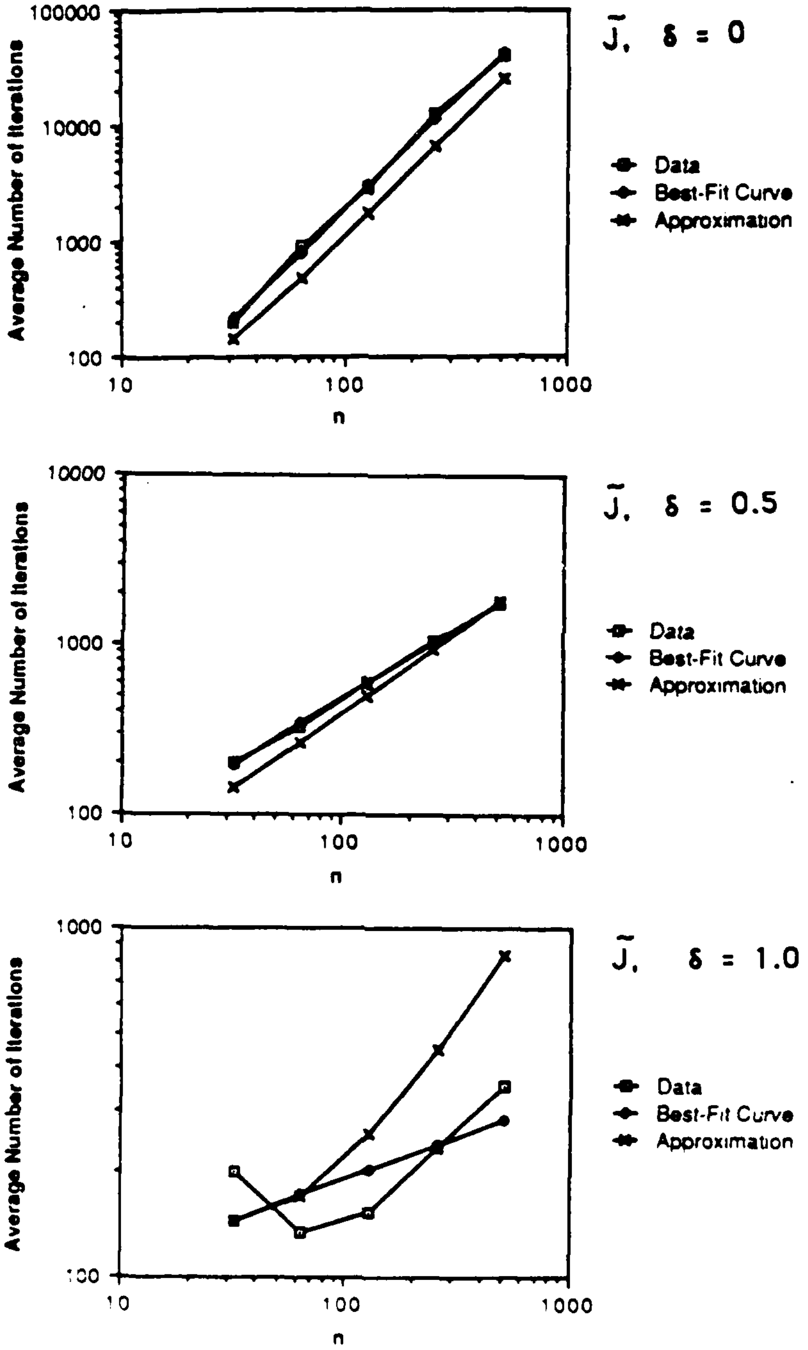


Figure 3.3.2. Graphs of $m(\tilde{J})$ (= "Data"), $e^{\alpha(\tilde{J})}n^{\beta(\tilde{J})}$ (= "Best-Fit Curve"), and (3.2.12) (= "Approximation") versus n, for $\tilde{\delta}$ = 0, 0.5, and 1.0.

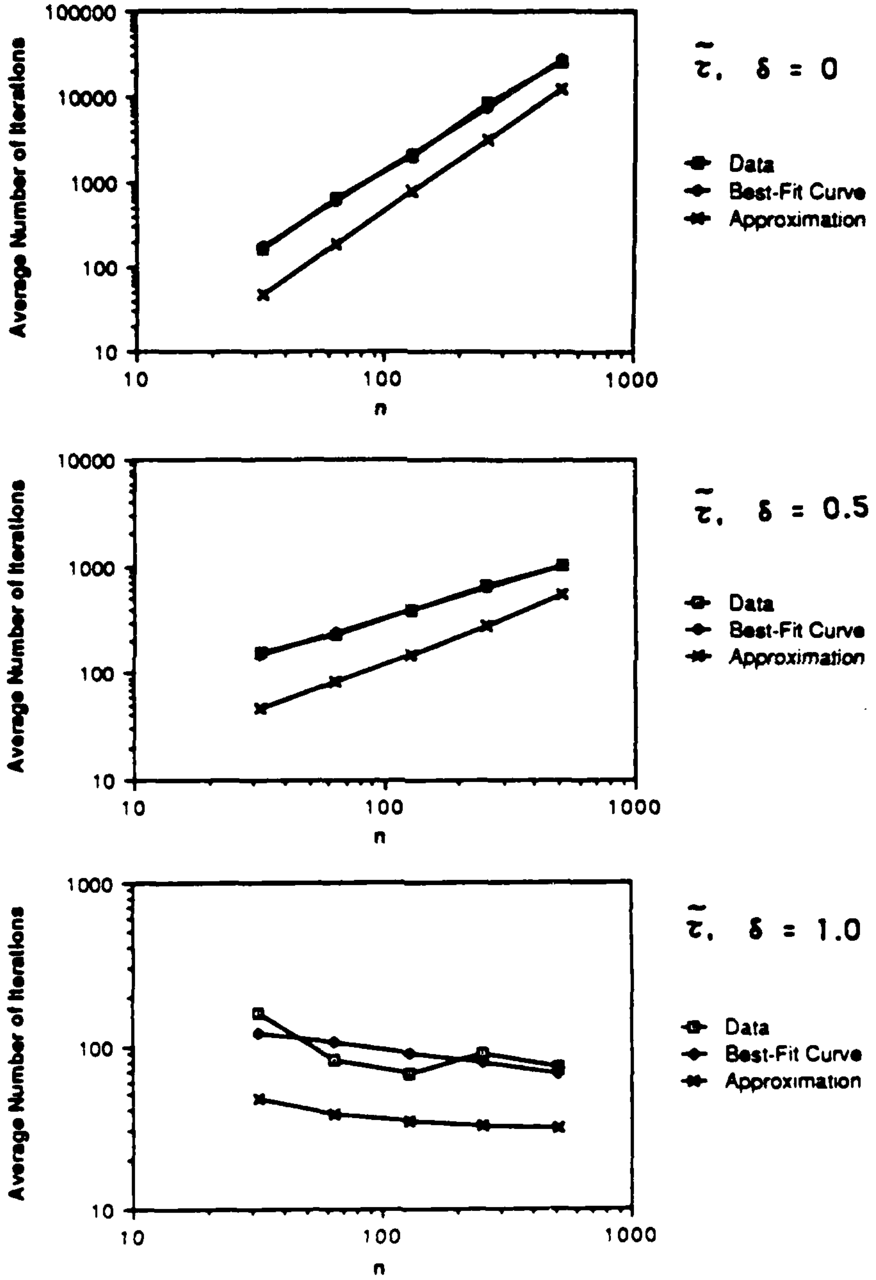


Figure 3.3.3. Graphs of m($\tilde{\tau}$) (= "Data"), $e^{\alpha(\tilde{\tau})}n^{\beta(\tilde{\tau})}$ (= "Best-Fit Curve"), and $\frac{3}{4}(\frac{n}{(d-1)})^2$ (= "Approximation") versus n, for δ = 0, 0.5, and 1.0.

For the values of n we considered, $\tilde{I}$ can be quite large, because the value of T we used was so small that for a large portion of the time $(X_k: k \geq 0)$ would be sitting in maximal matchings. To be more computationally efficient in estimating the mean of $\tilde{I}$ we do the following. First, set a register $\hat{I}$ to zero. Then simulate the process $(X_{J(k)}: k \geq 0)$, where $J(0) = 0$ and $J(k+1) = \min\{j \geq J(k): X_j \neq X_{J(k)}\}$. For every value of $k \geq 0$, we increment $\hat{I}$ by $\frac{m}{\gamma_k + |X_{J(k)}|}$, where γ_k is the number of edges that are matchable relative to $X_{J(k)}$. When $(X_{J(k)}: k \geq 0)$ reaches a maximum matching we stop the simulation and $\hat{I}$ is our estimate of $\tilde{I}$.

We computed sample means of $\hat{I}$ in the same way we computed sample means of $\tilde{J}$ and $\tilde{\tau}$, i.e., for each value of n and δ, 100 graphs were randomly selected from G(n,m), and, for each graph, a simulation of $(X_{J(k)}: k \geq 0)$ was done and statistics were taken. The sample mean $m(\hat{I})$ and the sample standard deviation $\sigma(\hat{I})$ of $\hat{I}$ are given in the tables in Figure 3.3.4. Note that $m(\hat{I})$ is an estimate of the mean of (3.2.1) over all graphs of G(n,m). Also listed in the tables, under $a(\hat{I})$, is the value of $3n^3\log n + 7.5\frac{n^4}{d^2}$. We also compute the pair $(\alpha(\hat{I}),\beta(\hat{I}))$ for the best-fit curve $e^{\alpha(\hat{I})}n^{\beta(\hat{I})}$ of the values of $m(\hat{I})$ as a function of n. The $(\alpha(\hat{I}),\beta(\hat{I}))$ values are given in Table 3.3.2. If $(Y_k: k \geq 0)$ is an accurate approximation of $(X_k: k \geq 0)$ then $\beta(\hat{I})$ should be approximately $\max\{3,4-2\delta\}$. Included in Table 3.3.2 are the values of $\max\{3,4-2\delta\}$, which are listed under $\beta(I^*)$. Just as in the analysis of the rate of growth of $\tilde{J}$ and $\tilde{\tau}$, our predictions for $\hat{I}$ tend to be more accurate for sparse graphs.

Note that $\beta(\hat{I}) - \beta(\tilde{J})$ is approximately 2 for small values of δ ($\beta(\hat{I}) - \beta(\tilde{J})$ = 2.1, 1.9, 1.9, 1.7, and 1.4 for δ equal to 0, 0.25, 0.50, 0.75, and 1.0, respectively). To see why this is so, observe that our choice of T is small so that the process $(X_{J(k)}: k \geq 0)$ behaves like the process $(\tilde{X}_k: k \geq 0)$. Therefore, the sample mean of $\hat{I}$ should be approximately the sample mean of $\tilde{J}$ times the average value of $J(k+1) - J(k)$ over k such that $0 \leq k \leq k^*$ and $J(k^*) = \hat{I}$. Since T is small, this

$\delta = 0.00$				
n	m	$m(\tilde{I})$	$\sigma(\tilde{I})$	$a(\tilde{I})$
32	80	1.86e+05	2.17e+05	6.55e+05
64	160	2.78e+06	2.96e+06	8.30e+06
128	320	5.04e+07	4.02e+07	1.11e+08
256	640	6.34e+08	4.68e+08	1.57e+09
512	1280	1.20e+10	1.30e+10	2.31e+10

$\delta = 0.25$				
n	m	$m(\tilde{I})$	$\sigma(\tilde{I})$	$a(\tilde{I})$
32	80	1.86e+05	2.17e+05	6.55e+05
64	190	1.96e+06	1.82e+06	6.84e+06
128	452	1.77e+07	1.40e+07	7.09e+07
256	1076	1.67e+08	1.13e+08	7.35e+08
512	2560	1.71e+09	1.27e+09	7.67e+09

$\delta = 0.50$				
n	m	$m(\tilde{I})$	$\sigma(\tilde{I})$	$a(\tilde{I})$
32	80	1.86e+05	2.17e+05	6.55e+05
64	226	1.17e+06	9.31e+05	5.79e+06
128	640	7.70e+06	6.57e+06	5.07e+07
256	1810	4.25e+07	3.18e+07	4.40e+08
512	5120	3.33e+08	2.51e+08	3.80e+09

$\delta = 0.75$				
n	m	$m(\tilde{I})$	$\sigma(\tilde{I})$	$a(\tilde{I})$
32	80	1.86e+05	2.17e+05	6.55e+05
64	269	6.77e+05	5.30e+05	5.05e+06
128	905	2.80e+06	2.00e+06	4.06e+07
256	3044	1.32e+07	9.21e+06	3.36e+08
512	10240	7.27e+07	5.10e+07	2.83e+09

$\delta = 1.00$				
n	m	$m(\tilde{I})$	$\sigma(\tilde{I})$	$a(\tilde{I})$
32	80	1.86e+05	2.17e+05	6.55e+05
64	320	4.08e+05	3.24e+05	4.53e+06
128	1280	1.24e+06	1.00e+06	3.56e+07
256	5120	4.77e+06	3.72e+06	2.99e+08
512	20480	1.67e+07	1.16e+07	2.59e+09

Figure 3.3.4. Tables of sample mean $m(\tilde{I})$; sample standard deviation $\sigma(\tilde{I})$; and our estimate $a(\tilde{I})$ of $\tilde{I}$, based on approximation $(Y_k\colon k \geq 0)$ of $(X_k\colon k \geq 0)$. The graphs considered have n nodes and m $= \lfloor c_0 n^{1+\delta} \rfloor$ edges.

Table 3.3.2. Function $e^{\alpha(\hat{I})}n^{\beta(\hat{I})}$ of n is the best-fit curve of $m(\hat{I})$ as a function of n. The value of $\beta(I^*)$ is the estimate of $\beta(\hat{I})$ based on the approximation $(\tilde{Y}_k\colon k \geq 0)$ of $(X_k\colon k \geq 0)$.

δ	$\alpha(\hat{I})$	$\beta(\hat{I})$	$\beta(I^*)$
0.00	-1.67	3.98	4.00
0.25	0.81	3.27	3.50
0.50	2.83	2.68	3.00
0.75	4.55	2.17	2.50
1.00	6.20	1.65	2.00

average will be roughly the average amount of time $(X_k\colon k \geq 0)$ takes to leave a maximal matching that has size proportional to n. This average time is approximately $\frac{m}{n\exp(-1/T)} = \frac{2}{3}n^2$. Then $\beta(\hat{I}) - \beta(\hat{J})$ should equal to 2.

In Figure 3.3.5, we plot three graphs of $m(\hat{I})$, $e^{\alpha(\hat{I})}n^{\beta(\hat{I})}$, and $3n^3\log n + 7.5\frac{n^4}{d^2}$ versus n for δ equal to 0, 0.5, and 1.0. These graphs indicate that the best-fit curves are accurate approximations of the sample mean versus n. From these graphs it also seems that $3n^3\log n + 7.5\frac{n^4}{d^2}$ is an asymptotic upper bound for $m(\hat{I})$.

We conclude this section with some final remarks. Our experiments show that the process $(Y_k\colon k \geq 0)$ is a reasonable approximation of $(X_k\colon k \geq 0)$, and if the graphs are sparse then $(Y_k\colon k \geq 0)$ is an accurate approximation of $(X_k\colon k \geq 0)$. This is not surprising, since one of our underlying assumptions used to define $(Y_k\colon k \geq 0)$ is that the graph is sparse. Note that $(Y_k\colon k \geq 0)$ leads to estimates that seem to be asymptotic upper bounds on the average amount of

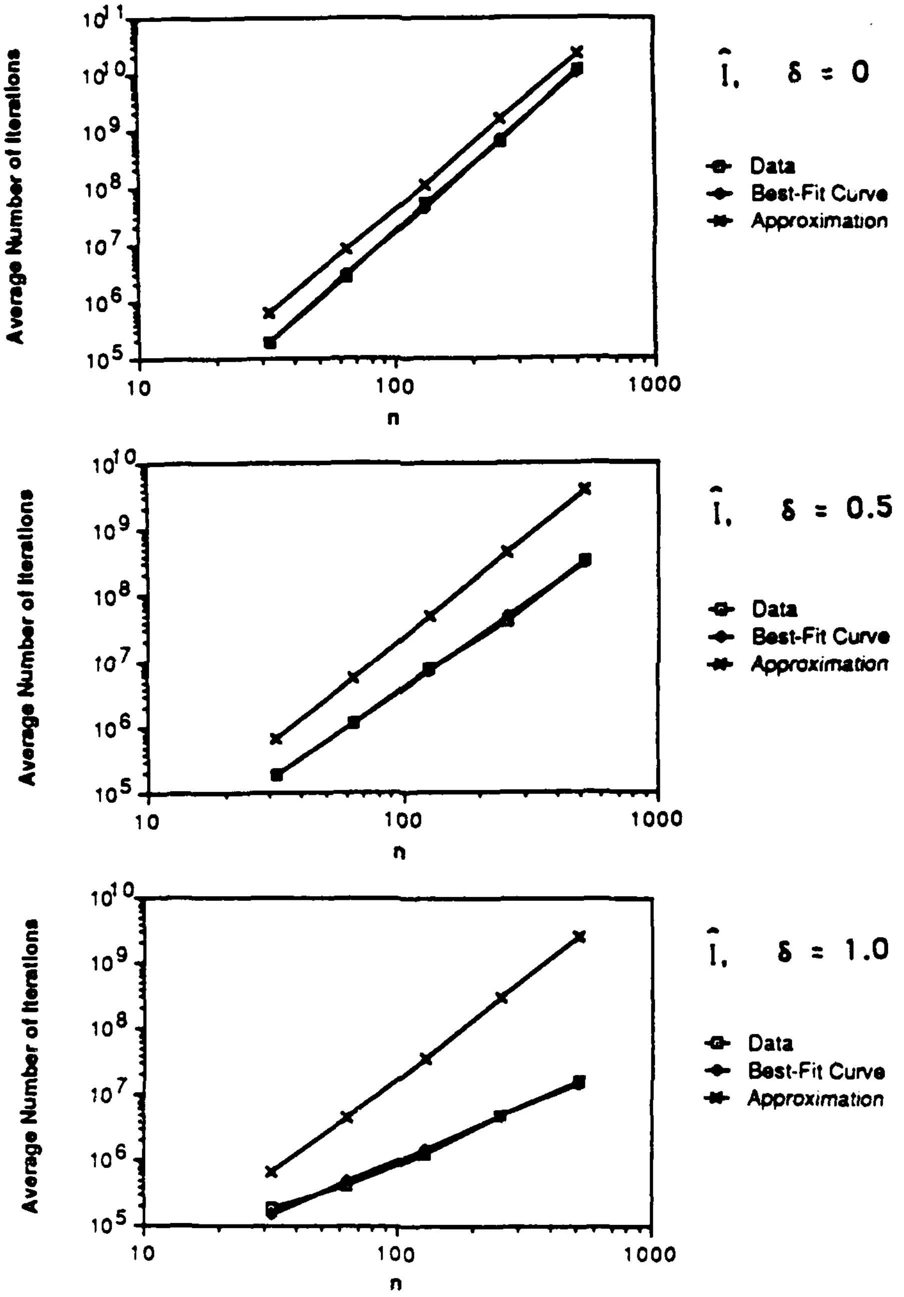


Figure 3.3.5. Graphs of m(Î) (= "Data"), $e^{\alpha(\hat{I})}n^{\beta(\hat{I})}$ (= "Best-Fit Curve"), and $3n^3\log n + 7.5\frac{n^4}{d^2}$ (= "Approximation") versus n, for δ = 0, 0.5, and 1.0.

time $(X_k: k \geq 0)$ and $(\bar{X}_k: k \geq 0)$ take to find a maximum matching. If the estimates are actual asymptotic upper bounds then for almost every graph (3.2.11) is an upper bound for (3.2.1) if $c \leq 1.5$. Note that if d is bounded below by one then (3.2.11) is $O(n^4)$. Also note that the sample mean of $\bar{t}$ is at least half the sample mean of $\bar{J}$. This is experimental evidence which helps to justify another of our assumptions used in defining $(Y_k: k \geq 0)$: the number of exposed nodes is small.

3.4. Conclusions

In the previous chapter, we presented results which showed that solving the matching problem by the basic simulated annealing algorithm of Section 2.1.2 takes average time that is exponential in the size of the instance. In addition, we also showed that to find a near maximum matching only takes average time that is polynomial in the size of the instance. In this chapter, we found an estimate (3.2.10) of a small upper bound on the average amount of time the basic simulated annealing algorithm takes to find a maximum matching for typical graphs with n nodes and m edges. If $m \geq n$ then (3.2.10) is $O(n^4)$. We checked this estimate with data from simulations and it seems that this estimate is an asymptotic upper bound.

An important reason why we were able to analyze the performance of simulated annealing for the matching problem on "typical" graphs was that we were able to approximate the annealing process $(X_k: k \geq 0)$ by a process $(Y_k: k \geq 0)$ that was quite homogeneous. In Section 4.2, we present a method of generating homogeneous, easy-to-analyze annealing processes.

CHAPTER 4

THE TEMPLATE METHOD, THE THRESHOLD RANDOM SEARCH ALGORITHM, AND THE NONMONOTONICITY OF OPTIMAL TEMPERATURE SCHEDULES

4.1. Introduction

In this chapter, a collection of miscellaneous results is presented. In Section 4.2, we give a simple technique, called the *template* method, that produces easy-to-analyze annealing processes. A random search algorithm, which we refer to as the *threshold random search* algorithm is presented in Section 4.3. The algorithm is a generalization of simulated annealing. In Section 4.4, sets of conditions are given under which *no* monotone decreasing temperature schedule is optimal.

4.2. The Template Method

In Chapters 2 and 3, bounds and estimates on the average amount of time simulated annealing takes to solve the matching problem were derived. However, we do not know of any other nontrivial combinatorial optimization problems amenable to such analysis. One of the reasons why simulated annealing applied to solving the matching problem could be analyzed is that there is a great deal of homogeneity in the annealing process.

In this section, a simple method we call the *template* method will be given that produces annealing processes that have a great deal of homogeneity and, as a result, are easy to analyze. The annealing processes produced by the template method are homogeneous so that the states can be classified into a relatively small number of types of states. In addition, for any state s and type α, the transition probability from s to some state of type α that has cost δ larger than the cost of s is dependent only on α, δ, the type of s, and the temperature schedule. Our hope is that the use of this method will produce interesting annealing processes that will help us to better understand the simulated annealing heuristic. We will begin by presenting this method and then give three examples of its use.

For the template method, we need a finite (preferably small) set S^{o} of state types, a set Δ^{c} of real numbers (corresponding to changes in cost of the states), and a transition probability matrix R^{o} from states in S^{o} to states in $S^{o}x\Delta^{c} = \{(s,\delta): s\in S^{o}, \delta\in\Delta^{c}\}$. We will also require the following condition on R^{o}. Let $Q^{(T)}$ be the probability transition matrix over S^{o} such that

$$Q^{(T)}_{s,s'} = \begin{cases} \sum_{\delta\in\Delta^{c}} R^{o}_{s,(s',\delta)}\exp(-\max\{0,\delta\}/T) & \text{if } s \neq s' \\ 1 - \sum_{t\neq s} Q_{s,t} & \text{if } s = s'. \end{cases}$$

We assume that for each $T > 0$, the Markov chain, which has states S^{o} and transition probability matrix $Q^{(T)}$, is irreducible and, therefore, ergodic. Then this Markov chain has a limiting distribution $\pi^{(T)}$ on S^{o}, which we can compute by solving the system of linear equations $\pi^{(T)}Q^{(T)} = \pi^{(T)}$. We call the triple (S^{o},Δ^{c},R^{o}) the *template system*, and we call

$$\sum_{s\in S^{o}} \pi_s^{(T)} \sum_{(s',\delta)\in S^{o}x\Delta^{c}} \delta R^{o}_{s,(s',\delta)}\exp(-\max\{0,\delta\}/T)$$

the *average drift* of the template system (S^{o},Δ^{c},R^{o}) for temperature T.

The following is one interpretation of the average drift of the template system. Let a set of states S^{∞}, a cost c^{∞} on S^{∞}, and a transition probability matrix R^{∞} over S^{∞} be defined as follows. The set S^{∞} is the set of all finite sequences of the form $[(s_0,\delta_0), (s_1,\delta_1), \ldots, (s_n,\delta_n)]$, where $n \geq 0$, and $s_i \in S^{o}$ and $\delta_i \in \Delta^{c}$ for all i such that $0 \leq i \leq n$. Let $\alpha \in S^{\infty}$, and let (s,δ) be the last pair in the sequence α. Then say α is a state of type s. The cost c^{∞} of a sequence $[(s_0,\delta_0), (s_1,\delta_1), \ldots, (s_n,\delta_n)]$ is zero if $n = 0$ and is $\sum_{j=1}^{n}\delta_j$ if $n \geq 1$. The matrix R^{∞} is such that

$$R^{\infty}_{\alpha,\alpha'} = \begin{cases} R^{o}_{i,(j,\delta)} & \text{if } \alpha \text{ is a state of type i and } \alpha'=\alpha \mid (j,\delta) \\ 0 & \text{otherwise,} \end{cases}$$

where $\alpha \mid (j,\delta)$ is the sequence formed by appending (j,δ) to the sequence α.

Let $(X_k: k \geq 0)$ be an annealing process on the system $(S^\infty, c^\infty, R^\infty)$ and with a temperature schedule that has all of its temperature values equal to T. Then it is easy to see that $\lim_{k\to\infty} E[(c^\infty(X_k) - c^\infty(X_0))/k]$ is equal to the average drift of the template system for temperature T.

A simple example of a template system (S^o, Δ^c, R^o) is when $S^o = \{A, B\}$, $\Delta^c = \{-1, 1\}$, and R^o is such that

$$R^o_{i,(j,\delta)} = \begin{cases} 1/2 & \text{if } i = A,\ j \in \{A, B\}, \text{ and } \delta = 1 \\ 1/2 & \text{if } i = B,\ j = A, \text{ and } \delta = 1 \\ 1/2 & \text{if } i = B,\ j = B, \text{ and } \delta = -1 \\ 0 & \text{otherwise } . \end{cases}$$

A state diagram of the system $(S^\infty, c^\infty, R^\infty)$ is partially presented in Figure 4.2.1. Each node corresponds to a state of S^∞ and inside each node is the type of the state. The cost c^∞ of the states is indicated on the left of the figure. An arrow from a state i to a state j indicates that $R^\infty_{ij} = \frac{1}{2}$.

Another example of a template system is given by $S^0 = \{0, 1, 2\}$, $\Delta^c = \{-2\text{x}10^6, -2\text{x}10^2, 10^2, 10^4, 10^6, 10^8\}$, and the matrix R^0, which is defined by

$$R_{i,(j,\delta)} = \begin{cases} 1/4 & \text{if } i=0,\ (j,\delta) \in \{(1,10^2),(0,10^4),(2,10^6),(0,10^8)\} \\ 1 & \text{if } i=1,\ j=0, \text{ and } \delta = -2\text{x}10^2 \\ 1 & \text{if } i=2,\ j=0, \text{ and } \delta = -2\text{x}10^6 \\ 0 & \text{otherwise} \end{cases}$$

The average drift for the template system (S^o, Δ^c, R^o) is

$$\frac{-10^2 e^{-10^2/T} + 10^4 e^{-10^4/T} - 10^6 e^{-10^6/T} + 10^8 e^{-10^8/T}}{4 + e^{-10^2/T} + e^{-10^6/T}} .$$

Note that this average drift is *not* unimodal in T. The implication is that we cannot guarantee that an iterative descent method will find a value of T that will minimize the average drift.

From this result we can also conclude that the average amount of time the following *finite* state annealing process takes to find an optimal solution is *not* unimodal as a function of

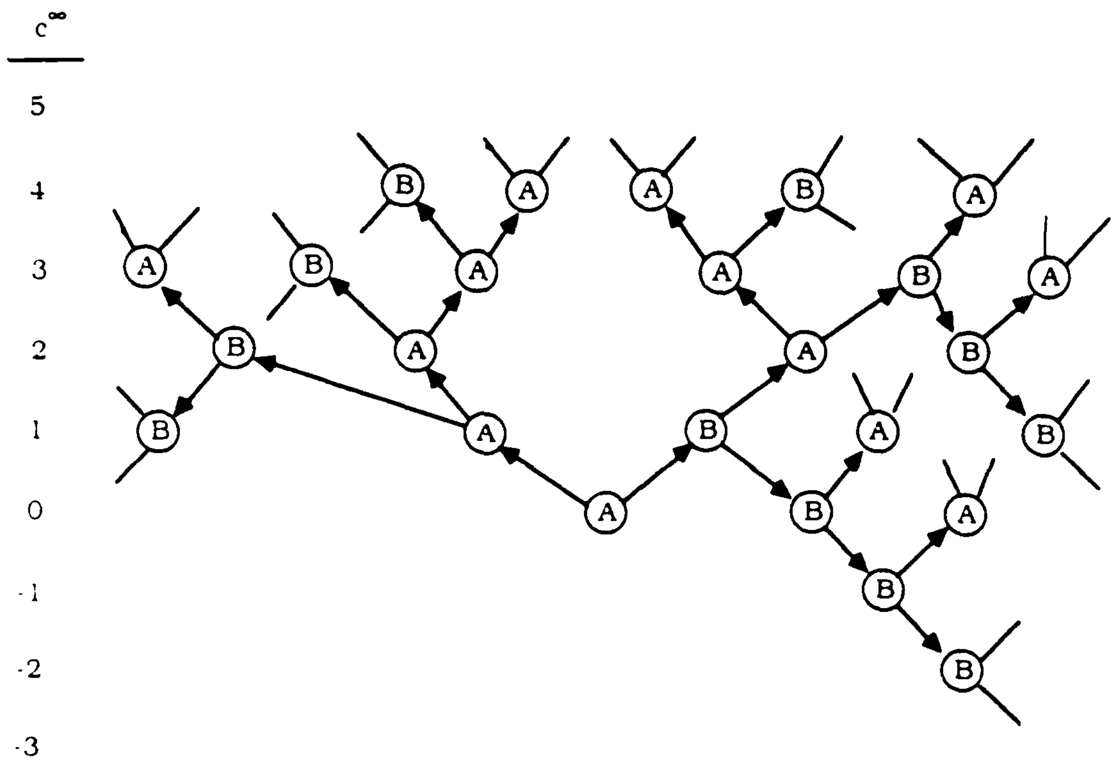


Figure 4.2.1. A partial state diagram of $(S^\infty, c^\infty, R^\infty)$ when $S^o = \{A, B\}$, $\Delta^c = \{-1, 1\}$, and R^o is such that

$$R^o_{i,(j,\delta)} = \begin{cases} 1/2 & \text{if } i = A,\ j \in \{A, B\}, \text{ and } \delta = 1 \\ 1/2 & \text{if } i = B,\ j = A, \text{ and } \delta = 1 \\ 1/2 & \text{if } i = B,\ j = B, \text{ and } \delta = -1 \\ 0 & \text{otherwise} \end{cases}$$

temperature. Let S be the set of all pairs of integers (i,j) such that $i \in \{0, 1, 2\}$ and $j \in \{0, 1, \ldots, 10^{100}\}$. Let $c((i,j)) = j$, and let R be a transition probability matrix R over S such that if $(i_1 j_1) \neq (i_2 j_2)$ then

$$R_{(i_1 j_1)(i_2 j_2)} = \begin{cases} 1/4 & \text{if } i_1 = 0,\ j_2 \leq 10^{100} \text{ and } (i_2,j_2) \in \{(0 j_1+10^4),\ (0 j_1+10^8)\} \\ 1/4 & \text{if } i_1 = 0,\ j_2 \leq 10^{100} \text{ and } (i_2,j_2) \in \{(1 j_1+10^2),\ (2 j_1+10^6)\} \\ 1 & \text{if } i_1 = 1,\ j_2 \geq 0, \text{ and } (i_2,j_2) = (0 j_1-2\text{x}10^2) \\ 1 & \text{if } i_1 = 2,\ j_2 \geq 0, \text{ and } (i_2,j_2) = (0 j_1-2\text{x}10^6) \\ 0 & \text{otherwise,} \end{cases}$$

and

$$R_{(i_1 j_1)(i_1 j_1)} = 1 - \sum_{(i_2 j_2) \neq (i_1 j_1)} R_{(i_1 j_1)(i_2 j_2)}.$$

Then the annealing process $(X_k: k \geq 0)$ on system (S, c, R) and with a temperature schedule that has all of its values equal to T is such that $E[\min\{k : c(X_k) \text{ is of minimum value}\} | X_0 = (0,10^{100})]$ is well approximated by 10^{100} divided by the magnitude of the average drift of the template system (S^o,Δ^c,R^o) with temperature T if this average drift is negative. Hence, $E[\min\{k: c(X_k) \text{ is of minimum value}\} | X_0 = (0,10^{100})]$ is not a unimodal function of T, as stated earlier.

Before presenting the third and final example of a template system we give some useful definitions. Suppose S is a set, c is a cost on S, and R is a transition probability matrix over S. We say that state i is *reachable* at height E from state j if there is a sequence of states j = i(0), i(1), . . . , i(p) = i such that $R_{i(k)i(k+1)} > 0$ for $0 \leq k < p$ and $c(i(k)) \leq E$ for $0 \leq k \leq p$. State s is said to be a *local minimum* if no state s′ with c(s′) < c(s) is reachable from s at height c(s). We define the *depth* of a local minimum s to be plus infinity if s is a global minimum. Otherwise, the depth of s is the smallest number E > 0, such that some state s′, such that c(s′) < c(s), can be reached from s at height c(s) + E.

For our third example, let n, λ, ϕ, and D be positive integers, where $\lambda < n + 1$ and $\phi < 2 + n$. Let $S^o = \{0,1,\ldots,D\}$, $\Delta^c = \{-1,1\}$, and R^o be such that

$$R^0_{i,(j,\delta)} = \begin{cases} \lambda/n & \text{if } i < D \text{ and } (j,\delta) = (i+1,1) \\ 1/n & \text{if } i > 0 \text{ and } (j,\delta) = (i-1,-1) \\ 1-\lambda/n & \text{if } i = 0 \text{ and } (j,\delta) = (i,0) \\ 1-(\lambda+1)/n & \text{if } 0 < i < D, \text{ and } (j,\delta) = (i,0) \\ 1/n & \text{if } i = D \text{ and } (j,\delta) = (D,-1) \\ \phi/n & \text{if } i = D \text{ and } (j,\delta) = (D,1) \\ 1-(2+\phi)/n & \text{if } i = D \text{ and } (j,\delta) = (D,0) \\ 0 & \text{otherwise.} \end{cases}$$

The average drift of the template system for the temperature T is

$$\frac{(\exp(-1/T)\lambda)^D}{\sum_{j=0}^{D}(\exp(-1/T)\lambda)^j}\frac{1}{n}[-1 + \exp(-1/T)\phi]. \qquad (4.2.1)$$

The motivation for our third example is that the average drift is approximately inversely proportional to the average amount of time a *finite* state annealing process $(X_k: k \geq 0)$, on the following system (S,c,R), takes to find a minimum valued solution. As we shall see, the average amount of time $(X_k: k \geq 0)$ takes to find a minimum valued solution is dependent on the density of states, density of states around local minima, and the depth of local minima of the annealing process. Suppose $D << 10^{100}$, and let S be the set of all quadruples of integers (i,j,d,k), where $0 \leq i \leq 10^{100}$, $0 \leq j < \phi^i$, $0 \leq d \leq D$, and $0 \leq k < \lambda^d$. Let c be a cost function on S such that $c((i,j,d,k)) = i + d$. Let G be any graph with node set S and edge set E that has properties to be specified later. Let R be a symmetric transition probability matrix on S such that

$$R_{ss'} \begin{cases} = 1/n & \text{if } s \neq s' \text{ and } [s,s'] \in E \\ 0 & \text{if } s \neq s' \text{ and } [s,s'] \notin E \\ = 1 - \sum_{i \neq s} R_{si} & \text{if } s = s' \end{cases}$$

We will now present the properties of G. Set S can be partitioned into subsets $T(i,j)$ of nodes for i and j such that $0 \leq i \leq 10^{100}$ and $0 \leq j < \phi^i$, where $T(i,j) = \{(i,j,d,k): 0 \leq d \leq D \text{ and } 0 \leq k < \lambda^d\}$. The subgraph of G induced by subset $T(i,j)$ is a completely-balanced, λary tree of height D and has root $(i,j,0,0)$, (see [18] for the definition of a tree, a root of a tree, a leaf of a tree, and the distance between nodes on a tree). Also, for each i and j, all nodes (i,j,d,k) are at distance d away from root $(i,j,0,0)$. Note that $c((i,j,d,k)) = d + c((i,j,0,0))$. In Figure 4.2.2, an example of a subgraph induced by $T(i,j)$ is given as well as the cost of the nodes. For all i and j, the only nodes of $T(i,j)$ that have neighboring nodes not in $T(i,j)$ are the leaves of the subgraph induced by $T(i,j)$. Note that each node (i,j,D,k) is a leaf of the tree induced by subset $T(i,j)$. If $i < 10^{100}$ (respectively, $i = 10^{100}$) then the nodes of the set $\{(i+1,\hat{j},D,k): \lambda(j-1) \leq \hat{j} < \lambda j\}$ (respectively, $\emptyset$) are the only nodes that are neighbors of leaf (i,j,D,k) such that each node is in a subset of $T(\hat{i},\hat{j})$ for some $\hat{i} \geq i$ and $(i,j) \neq (\hat{i},\hat{j})$. Hence, the edges of E are such that the leaves of the tree induced by $T(i,j)$ are paired with the leaves of the tree induced by $T(i+1,\hat{j})$ where $\hat{j}$ is such that $\lambda(j-1) \leq \hat{j} < \lambda j$. Note that the graph

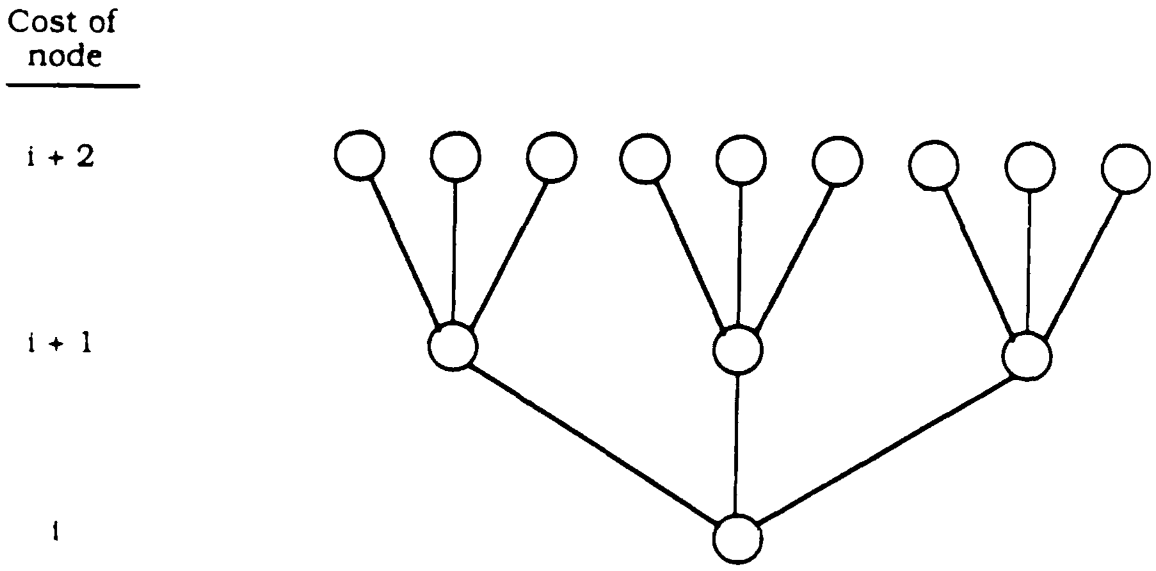


Figure 4.2.2. The subgraph induced by $T(i,j)$ when $D = 2$ and $\lambda = 3$.

induced by contracting subset T(i,j) into a node for each i, j is a completely balanced ϕary tree. The edges between the induced trees T(i,j) are illustrated in Figure 4.2.3.

Let $(X_k, k \geq 0)$ be the annealing process on system (S,c,R) with temperature schedule $(T_k: k \geq 0)$ such that $T_k = T$. This implies that if (4.2.1) is negative then the average amount of time it takes the annealing process to find a minimum valued solution given $X_0 = (10^{100},0,0,0)$ is approximately 10^{100} divided by the magnitude of (4.2.1), as we stated earlier. Note that ϕ is the rate at which the density of states of $(X_k: k \geq 0)$ is increasing with cost, D is the depth of all the local minima, and λ is the rate at which the density of states close to a local minima is increasing with cost.

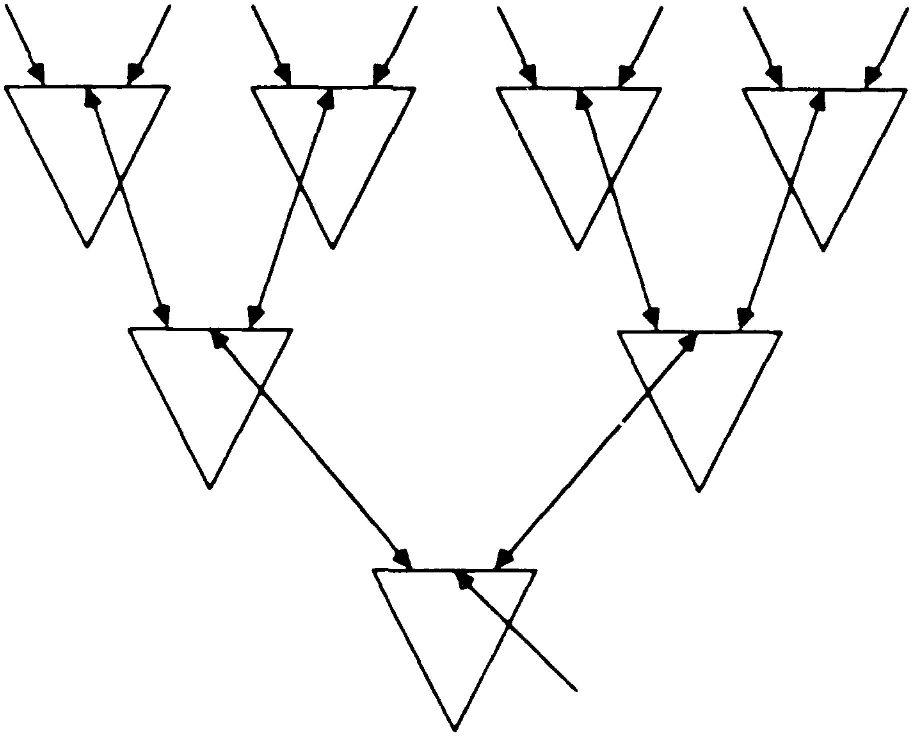


Figure 4.2.3. Part of the graph G when $\phi = 2$. The triangles are trees induced by subsets T(i,j). Arrows between two trees mean that the leaves of the two trees are paired.

To keep (4.2.1) small one can set $\exp(-1/T) = \frac{1}{2\phi}$. Then (4.2.1) becomes

$$\frac{-1}{2n}\frac{(\frac{\lambda}{2\phi})^D}{\sum_{j=0}^{D}(\frac{\lambda}{2\phi})^j}. \tag{4.2.2}$$

If $\phi << \lambda$ then (4.2.2) is approximately $\frac{-1}{2n}$. If $\phi >> \lambda$ then (4.2.2) is approximately $-(\lambda/2\phi)^D\frac{1}{2n}$. Hence, if $\phi >> \lambda$, the value of D strongly influences the average amount of time $(X_k: k \geq 0)$ takes to find a minimum valued state. If $\phi >> \lambda$ then the average drift goes to zero exponentially as D increases and, therefore, the average amount of time $(X_k: k \geq 0)$ takes to find a minimum valued state grows exponentially as D increases.

We believe that for typical annealing processes, states around local minima that have cost much higher than the cost of the global minima corresponds to the situation where $\lambda > \phi$. If this is true then the depth of local minima will not be so important to the drift of the annealing process until the process is in a near optimal state.

4.3. The Threshold Random Search Algorithm

The *threshold random search* algorithm is used to solve combinatorial optimization problems:

$$\min\{c(s): s \in S\},$$

where S is a finite set of states and c is a cost function of S. Just as with simulated annealing, the threshold random search algorithm requires a transition probability matrix R over S. In addition, a sequence of positive random variables $(t_k: k\geq 0)$, we call the *threshold schedule*, is needed. The threshold random search algorithm generates a sequence of states $(X_k: k\geq 0)$ as follows. An initial state X_0 is chosen from S. Given that $X_k = s$, a potential new state Y_k is chosen from S with probability distribution $P[Y_k = s' \mid X_k = s] = R_{ss'}$. Then we set

$$X_{k+1} = \begin{cases} Y_k & \text{if } c(Y_k) - c(X_k) \le t_k \\ X_k & \text{otherwise.} \end{cases}$$

If $t_k = -T_k \log(U_k)$, where $(T_k\colon k \ge 0)$ is a temperature schedule and $(U_k\colon k \ge 0)$ is a sequence of independent random variables distributed uniformly over the interval (0,1] then

$$P[X_{k+1} = j \mid X_k = i] = \begin{cases} R_{ij} \exp(-\max\{0, c(j) - c(i)\}/T_k) & \text{if } i \ne j \\ 1 - \sum_{h \ne i} P[X_{k+1} = h \mid X_k = i] & \text{otherwise.} \end{cases}$$

Thus, for this threshold schedule the threshold random search algorithm is equivalent to simulated annealing.

Let $(X_k^{(t)} : k \ge 0)$ be the process generated by the threshold random search algorithm with threshold schedule t. It can be shown that there exists a threshold schedule that minimizes $E[\min\{k. c(X_k^{(t)}) \le \gamma\}]$. More generally, as we shall see in the next corollary, it can be shown that there exists a threshold schedule that minimizes F, where $F(t) = E[\sum_{i=0}^{\infty} f^{(i)}(X_0^{(t)}, \ldots, X_i^{(t)})]$, for some nonnegative functions $f^{(i)}$. Note that if $f^{(i)}(x_0, \cdots, x_i) = I_{\{c(X_j) > \gamma \text{ for all } j \text{ such that } 0 \le j \le i\}}$ then $F(t) = E[\min\{k: c(X_k^{(t)}) \le \gamma\}]$.

The corollary is implied by the next proposition and the following simple observation: for every threshold schedule t, there is a sample path t′ of t such that $F(t) \ge F(t')$. Therefore, when searching for threshold schedules that minimize F we can restrict our attention to deterministic ones. Without loss of generality we will only consider deterministic threshold schedules $(t_k\colon k \ge 0)$, where $t_k \in \Delta$, for all $k \ge 0$, and $\Delta = \{c(j) - c(i)\colon i,j \in S,\ R_{ij} > 0, \text{ and } c(j) > c(i)\} \cup \{0\}$. Let A be the set of such kinds of deterministic threshold schedules.

Proposition 4.3.1 The functional F is minimized on A.

Proof: It is straightforward to show that A is compact with respect to the distance metric

$$\delta((t_k\colon k\geq 0),(s_k\colon k\geq 0)) = \sum_{k\geq 0} 2^{-k}|t_k - s_k|.$$

We will now show that F is lower semicontinuous on A, which will complete the proof. Since Δ is finite and $E[f^{(k)}(X_0^{(t)}, X_1^{(t)}, \ldots, X_k^{(t)})]$ depends only on the first k + 1 elements of the sequence t, $E[f^{(k)}(X_0^{(t)},X_1^{(t)},\ldots,X_k^{(t)})]$ is continuous on A. Thus, F is lower semicontinuous because it is the sum of a collection of nonnegative continuous functions.

□

Corollary 4.3.1: There is a deterministic threshold schedule that minimizes F.

Remark: A simulated annealing algorithm will typically be outperformed by a threshold random search algorithm with some deterministic threshold schedule. However, this is only of theoretical interest, at this point, since the problem of finding optimal threshold schedules is difficult.

4.4. The Nonmonotonicity of Optimal Temperature Schedules

Most analytical studies of simulated annealing consider only monotone decreasing temperature schedules. This is not surprising, since the purpose of the heuristic is to simulate an "annealing" process. In this section, we present sets of conditions under which *no* monotone decreasing temperature schedule is optimal. We will focus on the basic simulated annealing algorithm in Subsection 2.1.2, which is used to solve the matching problem for a graph G. Let (S, c, R) be as in Subsection 2.1.2, and let $(X^{(T)}\colon k\geq 0)$ be the annealing process with temperature schedule T, where $X_0^{(T)} = \varnothing$. The following proposition contains the main results of this section.

Proposition 4.4.1: Suppose there is a maximal but not maximum matching M of G such that $|M| \leq m$. Then

(a) there is a temperature schedule T that minimizes

$$E[\min\{k: |X_k^{(T)}|>m\}], \tag{4.4.1}$$

and for every J ≥ 0 there is temperature schedule T′ that minimizes

$$P[|X_J^{(T')}|>m]; \tag{4.4.2}$$

(b) there is a monotone decreasing temperature schedule T that minimizes (4.4.1) if and only if the infinity-valued temperature schedule (i.e., $T_k = \infty$ for all k) minimizes (4.4.1); and (c) there is a $\hat{J}$ such that if $J \geq \hat{J}$ then there is *no* monotone decreasing temperature schedule that minimizes (4.4.2).

Remark: If G is a single path consisting of 2n+1 nodes, m = 0.9n and n is sufficiently large, then the infinity-valued temperature schedule does not minimize (4.4.1). To see this, observe that by Theorem 2.3.1 there is a temperature schedule such that (4.4.1) is $O(n^5)$. Now suppose that T is the infinity-valued temperature schedule. If a matching M of G is such that $|M| > 0.75n$, then the number of matchable edges relative to M is at most 0.5n. The implication is that the ratio of |M| over the number of matchable edges relative to M is at least 0.6 and, hence, since T is the infinity-valued temperature schedule, $P[|X_{\min\{k>i: X_k \neq X_i\}}|<|M| \mid X_i=M] \geq 0.6$. Then it is straightforward to show that (4.4.1) grows exponentially with n. Therefore, for this graph G, none of the optimal temperature schedules are monotone decreasing if n is large enough.

To prove the proposition we will use the next lemma, which may be interesting in its own right. In the lemma we refer to a process $(\tilde{X}_k^{(T)}: k \geq 0)$, which is a more general form of the process $(X_k^{(T)}: k \geq 0)$. We also refer to a functional F(T) of temperature schedule T, where F(T) = $E[\sum_{j=0}^{\infty} f^{(j)}(\tilde{X}_0^{(T)}, \ldots, \tilde{X}_j^{(T)})]$ and $f^{(j)}$ is nonnegative. If $(\tilde{X}_k^{(T)}: k \geq 0)$, is equal to $(X_k^{(T)}: k \geq 0)$ then both (4.4.1) and (4.4.2) have the form of F(T).

Lemma 4.4.1: Let $\tilde{S}$ be a set of states, let $\tilde{c}$ be a cost on that set, and let $\tilde{R}$ be a transition probability matrix over $\tilde{S}$ such that if $\tilde{R}_{ij} > 0$ and $\tilde{c}(j) > \tilde{c}(i)$ then $\tilde{c}(j) - \tilde{c}(i) = 1$. Let $(\tilde{X}_k^{(T)} : k \geq 0)$

be the annealing process on $(\tilde{S}, \tilde{c}, \tilde{R})$, with temperature schedule T. Let $F(T) = E[\sum_{j=0}^{\infty} f^{(j)}(\tilde{X}_0^{(T)}, \ldots, \tilde{X}_j^{(T)})]$, where $f^{(j)}$ is nonnegative. Then there is a temperature schedule that minimizes F and is monotone decreasing if and only if there is a temperature schedule $(T_k: k \geq 0)$ that minimizes F, is monotone decreasing, and $T_k \in \{0, \infty\}$ for all $k \geq 0$.

Remark: If T is a zero-infinity valued temperature schedule, the annealing process $(\tilde{X}_k^{(T)} : k \geq 0)$ then there is a random threshold search algorithm with a deterministic threshold schedule that can produce an equivalent process.

Proof of Lemma 4.4.1: The "if" part of the lemma is immediate. We now turn to prove the "only if" part. Let $T = (T_k: k \geq 0)$ be an optimal temperature schedule that is monotone decreasing. For each $j \geq -1$, we will define a temperature schedule $T^{(j)} = (T_k^{(j)}: k \geq 0)$ inductively as follows. Let $T^{(-1)} = T$. For each $j \geq 0$,

$$T_k^{(j)} = \begin{cases} 0 & \text{if } k = j \text{ and } T_k^{(j-1)} < \infty \\ T_k^{(j-1)} & \text{otherwise,} \end{cases}$$

for all $k \geq 0$.

We will now show that $F(T^{(j)}) = F(T^{(j-1)})$, for $j \geq 0$, by the following inductive argument. Suppose $T^{(j-1)}$ minimizes F and $T_j^{(j-1)} < \infty$. Note that

$$F(T^{(j-1)}) = \sum_{\alpha,\beta \in S} E[\sum_{i=0}^{\infty} f^{(i)}(\tilde{X}_0^{(T^{(j-1)})}, \ldots, \tilde{X}_i^{(T^{(j-1)})}) | \tilde{X}_j^{(T^{(j-1)})} = \alpha, \tilde{X}_{j+1}^{(T^{(j-1)})} = \beta]$$

$$\times P[\tilde{X}_{j+1}^{(T^{(j-1)})} = \beta | \tilde{X}_j^{(T^{(j-1)})} = \alpha] P[\tilde{X}_j^{(T^{(j-1)})} = \alpha],$$

and the only terms in the expression that depend on $T_j^{(j-1)}$ are $P[\tilde{X}_{j+1}^{(T^{(j-1)})} = \beta \mid \tilde{X}_j^{(T^{(j-1)})} = \alpha]$ for $\alpha, \beta \in S$. Since $\tilde{R}_{ij} > 0$ and $c(j) > c(i)$ imply $c(j) - c(i) = 1$, $P[\tilde{X}_{j+1}^{(T^{(j-1)})} = \beta \mid \tilde{X}_j^{(T^{(j-1)})} = \alpha]$ equals one or $\exp(-1/T_j^{(j-1)})$. Hence, $F(T^{(j-1)})$ has the form $A' + B'\exp(-1/T_j^{(j-1)})$, where A' an1 B' are terms not dependent on $T_j^{(j-1)}$. Then it must be that $B' \geq 0$, because $T^{(j-1)}$ minimizes F and $T_j^{(j-1)} < \infty$.

Therefore, $F(T^{(j)}) = A' \leq A' + B'\exp(-1/T_j^{(j-1)}) = F(T^{(j-1)})$ for all $j \geq 0$. By the optimality of T, $F(T^{(j)}) = F(T)$ for all $j \geq 0$.

Let $T^{(\infty)} = (T_k^{(\infty)}: k \geq 0)$ be such that $T_k^{(\infty)} = \infty$, if $T_k = \infty$, and $T_k^{(\infty)} = 0$, otherwise. Using an argument similar to the one used in the proof of Proposition 4.3.1, one can show that F is lower semicontinuous on the set of all temperature schedules under the distance metric $\delta(t,s) = \sum_{i=0}^{\infty} 2^{-i} |\exp(-1/t_i) - \exp(-1/s_i)|$. Since $T^{(\infty)}$ is the pointwise limit of $T^{(j)}$ and $F(T) = F(T^{(j)})$ for all positive j, $F(T^{(\infty)}) = F(T)$. Thus, $T^{(\infty)}$ is optimal, zero-infinity valued, and monotone decreasing, and we are done.

□

Proof of Proposition 4.4.1. Part (a) follows from our argument in the proof of Lemma 4.4.1 that F is lower semicontinuous on the set of all temperature schedules, and the fact that the set of all temperature schedules is compact under the distance metric δ.

We now turn to prove part (b). The fact that R is symmetric and, for all $i, j \in S$, there is a sequence of matchings, $i = i(1), i(1), \ldots, i(k) = j$, such that $R_{i(h)i(h+1)} > 0$ for all h such that $1 \leq h < k$, implies that the infinity-valued temperature schedule leads to a finite value for F. Write $M = \{e_1, e_2, \ldots, e_n\}$, and let $M_0 = \varnothing$, and $M_i = \{e_1, e_2, \ldots, e_i\}$ for all $i>0$. Since $X_0^{(T)} = 0$, for all i such that $1 < i \leq n$, $R_{M_{i-1}M_i} > 0$, $R_{M_iM_{i-1}} > 0$, and $|M_{i-1}| > |M_i|$, then, for any temperature schedule T, $P[X_k^{(T)} \in \{M_0, \ldots, M_n\}] > 0$ for all $k \geq 0$. Hence, F is infinite for all monotone decreasing zero-infinity valued temperature schedules, with the exception of the infinity-valued temperature schedule. We can then conclude that, the only monotone decreasing zero-infinity valued temperature schedule that can possibly be optimal is the infinity-valued one. The previous conclusion and Lemma 4.4.1 imply there is a temperature schedule T that minimizes (4.4.1) and is monotone decreasing if and only if the infinity-valued temperature schedule minimizes (4.4.1). Thus, part (b) is proved.

We will now turn to prove part (c). Since R is symmetric and for all i, j ∈ S, there is a sequence of matchings, $i = i(1), i(1), \ldots, i(k) = j$, such that $R_{i(h)i(h+1)} > 0$ for all h such that $1 \leq h < k$, we know that

$$\lim_{k\to\infty} P[X_k^{(T)} \in \{M_0, \ldots, M_n\}]$$

exists and is strictly positive if T is the infinity-valued temperature schedule. Then there is a J such that

$$\inf_{j\geq J} P[X_j^{(T)} \in \{M_0, \ldots, M_n\}] > 0. \tag{4.4.3}$$

Inequality (4.4.3), the fact that M is maximal, and the fact that $R_{M_{i-1},M_i}>0$ and $|M_{i-1}| > |M_i|$, for all i such that $1 < i \leq n$, imply that there exists an $\varepsilon > 0$ such that $\inf_j P[|X_j^{(T)}| \leq m] > \varepsilon$ for any monotone decreasing zero-infinity valued temperature schedule T. However, Geman and Geman [4], Hajek [13], among others, have shown that there is a temperature schedule T such that

$$\lim_{k\to\infty} P[|X_k^{(T)}| \leq m] = 0.$$

Hence, there is a $\hat{J}$ such that if $J \geq \hat{J}$ then no monotone decreasing zero-infinity valued temperature schedule minimizes (4.4.2). Then Lemma 4.4.1 implies that there are no monotone decreasing temperature schedules that are optimal. Part (c) is now proved and we are completely done with the proof of the proposition.

□

CHAPTER 5

OPTIMIZATION WITH EQUALITY CONSTRAINTS

5.1. Optimization with Equality Constraints

In this chapter, we consider solving the following equality constrained problem (ECP) by simulated annealing.

$$\text{(ECP)} \qquad \text{minimize } f(x)$$
$$\text{subject to } x \in X,\ h(x) = 0,$$

where X is the set of solutions, f: $X \rightarrow R$, and $h(x) = (h_1(x), h_2(x), \ldots, h_k(x))^T$ for some $k \geq 1$. We can also include inequality constraints in this form, since $g(x) \leq 0$ can be written as $\max\{0,g(x)\} = 0$. A solution $x \in X$ is called *feasible* if $h(x) = 0$.

As an example, the following optimization version of the Graph Partitioning Problem [19] can be written as an equality constrained problem. An instance of this problem is a graph (V,E), a positive integer K, and a set of positive integers $\sigma_1,\sigma_2, \ldots, \sigma_K$ such that $\sum_{i=1}^{K} \sigma_i = |V|$. We call $(V_1,V_2, \ldots, V_K)$ a *partition* of V if each V_i is a subset of V, the V_i are disjoint, and their union is V. The *capacity* of the partition $(V_1, \ldots, V_K)$ is the number of edges [u,v] such that if $u \in V_i$ and $v \in V_j$ then $i \neq j$. The problem is to find a partition $(V_1, \ldots, V_K)$ with minimum capacity and such that $|V_i| - \sigma_i = 0$ for $i = 1, 2, \ldots, K$.

Another optimization problem that can be written as an equality constrained problem is the optimization version of the Minimum Cut Linear Problem [19]. An instance for this problem is a graph (V,E). The problem is to find a mapping π: $V \rightarrow \{1,2, \ldots, |V|\}$ that will minimize,

$$\max_{1 \le i < |V|} \ |\{(u,v) \in E: \pi(u) \le i < \pi(v)\}|$$

subject to $|\{v \in V: \pi(v) = j\}| - 1 = 0$ for $j = 1, 2, \ldots, |V|$.

Penalty methods are techniques used to solve equality constrained problems. The basic idea of the methods is to substitute some or all of the equality constraints by adding to the cost function penalty terms that give a high cost to infeasible points. In this section, we will focus on the *quadratic penalty* method, which we will describe next, and variations of it. For any scalar c, let us define the *augmented Lagrangian* function

$$L_c(x,\lambda) = f(x) + \lambda^T h(x) + \frac{c}{2}|h(x)|^2.$$

We refer to c as the *penalty parameter* and to λ as the *multiplier* vector (or simply multiplier). The quadratic penalty method consists of solving a sequence of problems of the form

$$\min\{L_{c_k}(x,\lambda_k): x \in X\},$$

where $(\lambda_k: k \ge 0)$ is a bounded sequence and $(c_k: k \ge 0)$ is a penalty parameter sequence such that $0 < c_k < c_{k+1}$ for all $k \ge 0$, and $c_k \to \infty$. For many applications, $\lambda_k = 0$ for all $k \ge 0$. In this section, we consider using a simulated annealing algorithm to minimize the augmented Lagrangian function for each c_k. Since the running time of simulated annealing is typically very long, we will only consider $L_{c_k}(\cdot,\lambda_k)$ for $k = 0$.

Aragon et al. [20] demonstrated that the quality of solutions produced by the quadratic penalty method in conjunction with simulated annealing may be sensitive to the value of the penalty parameter. In their experiments, very large values of c_0 resulted in poor solutions, probably, because the annealing process was greatly restricted to what state it could move to. For very small values of c_0, the final solution of the simulated annealing algorithm was far from being feasible, and their greedy fix-up algorithm was not effective enough to produce good solutions. To find a good parameter value experimentally may be impractical, since simulated annealing is typically very time consuming. Hence, it is worthwhile to find penalty methods that perform well over a wide range of parameter values. We will introduce adaptive penalty methods, which may be penalty methods of this type.

For the rest of this section, we will describe two adaptive penalty methods and the following simple *hardlimiter* penalty method. In Section 5.2, we compare the penalty methods by computer experiments. For the hardlimiter penalty method, we find an $x \in X$ that will minimize $f(x) + p(x,\gamma)$, where

$$p(x,\gamma) = \begin{cases} 0 & \text{if } |x| \leq \gamma \\ \infty & \text{otherwise,} \end{cases}$$

and γ is a nonnegative scalar parameter. Note that if $\gamma = 0$ then minimizing $f(x) + p(x,\gamma)$ is equivalent to solving the equality constrained problem (ECP).

The two adaptive penalty methods we consider are similar to the quadratic penalty method, because they both involve minimizing an augmented Lagrangian function, but in the adaptive methods the multiplier is adaptively and periodically adjusted.

The first adaptive penalty method is inspired by the method of multipliers (see [21]), which is used in nonlinear programming, and the second method is a slight modification of the first. The procedure of the method of multipliers is as follows. Let $(c_j: j \geq 0)$ be a positive monotone increasing sequence. For each j, solve

$$\begin{aligned} &\text{minimize } L_{c_j}(x,\lambda_j) \\ &\text{subject to } x \in X, \end{aligned}$$

and if x_j is an optimal solution then

$$\lambda_{j+1} = \lambda_j + c_j h(x_j).$$

Note that in practice, in place of x_j, we use the solution $\tilde{x}_j$ found by the simulated annealing algorithm, which is generally not optimal.

In nonlinear programming applications the method of multipliers has advantages over the quadratic penalty method. For example, suppose that for the equality constrained problem (ECP) $X = R^n$, $f: R^n \rightarrow R$, $h: R^n \rightarrow R^m$, and $f, g \in C^2$. In addition, suppose the following assumption holds.

Assumption (S) [21]: There is a solution x^* that is a strict local minimum and a regular point of ECP, and $f, h \in C^2$ on some open sphere centered at x^*. Furthermore, x^* together with its associated Lagrange multiplier vector λ^* satisfies

$$z^T \nabla^2_{xx}(f(x) + \lambda^* h(x))z > 0,$$

for all $z \neq 0$ with $\nabla h(x^*)^T z = 0$.

From Proposition 2.4 of [21], for any value of λ_0, there exists a $\bar{c} > 0$ such that if $c_j \geq \bar{c}$ then $x_j \rightarrow x^*$ and $\lambda_j \rightarrow \lambda^*$, where x^* is a locally optimal solution of equality constrained problem and λ^* is its corresponding Lagrange multiplier. However, if we fix $\lambda_j = 0$ for all j (corresponding to the quadratic penalty function method) then to insure that x_j converges to a locally optimal solution of the equality constrained problem we must have $c_j \rightarrow \infty$. This suggests that the method of multipliers may be less sensitive to the values of c_j. However, c_j must still be sufficiently large. For example, suppose $f(x) = -x^2$, $h(x) = x$, and $X = [-1,1]$. If $c < 1$ then the points in X that minimize $L_c(x,\lambda)$, for any value of λ are in $\{-1,1\}$, and both -1 and 1 are infeasible solutions.

For the method of multipliers, it may be necessary to minimize a number of augmented Lagrangian functions. Since we intend to do the minimization by simulated annealing and simulated annealing, typically, takes a long time, this penalty method may be impractical. This motivates our next penalty method, which we call the *dynamic method of multipliers*. In this penalty method, simulated annealing is used only once to find the minimum of $L_c(x,\lambda)$, and the multiplier λ is updated every Z iterations during this run. The update rule for λ is $\lambda_{new} = \lambda_{old} + c\bar{h}$, where $\bar{h}$ is the average value of h(x) observed since the last time λ was updated. Note that λ is updated so that h(x) will drift in the direction of zero. Hence, even if $L_c(.,\lambda)$ is concave we have some hope that the final solution will be close to being feasible.

We list three disadvantages of this method: (1) there is the additional complexity of choosing parameter Z; (2) if Z is small then updating λ could be very time consuming; and (3) computing $\bar{h}$ may be time consuming if the number of equality constraints is large. We will set Z equal to the maximum number of neighbors a state in the annealing process has, which should be a large number. This is a somewhat arbitrary choice for Z and is not a general recommendation. Note that we can set Z to be sufficiently large so that the dynamic method of multipliers reduces to the quadratic penalty method. Since our value of Z will typically be a large value, the dynamic method of multipliers will be, in a sense, a perturbation of the quadratic penalty method.

The third disadvantage can be eliminated in many cases if $\bar{h}_i$ is updated only when $h_i(x)$ changes value or when λ_i needs to be updated.

5.2. Experimental Results

In order to compare the different penalty methods of the previous section we applied them to solve the optimization form of the Graph Partitioning Problem and simulated annealing was used to perform the optimization. Then computer experiments were done to compare the performance of the methods. We will first discuss the implementation of each penalty method. Then we will present and discuss the experimental results.

The basic form of our implementation of simulated annealing for the penalty methods is shown in Figure 5.2.1 (p. 84). The algorithm in this figure, as well as all other algorithms presented in this section, is written in pidgin Algol. We refer the reader to [10] for more details of pidgin Algol.

The input to the algorithm is a graph (V,E), an integer K, and positive integers $\sigma_1, \sigma_2, \ldots, \sigma_k$ such that $\sum_{i=1}^{K} \sigma_i = |V|$. For this instance of the problem, the set of states is the set of partitions of V. If the quadratic penalty method, the method of multipliers, or the dynamic method of multipliers is used then the cost of partition $(V_1, \ldots, V_K)$ is its capacity plus

$$\sum_{i=1}^{K}\lambda_i(\sigma_i - |V_i|) + \frac{c}{2}\sum_{i=1}^{K}(\sigma_i - |V_i|)^2.$$

If the hardlimiter penalty method is used then the cost of a partition $(V_1, \ldots, V_K)$ is its capacity plus $\sum_{i=1}^{K} p(\sigma_i - |V_i|, \gamma)$. Two partitions $(V_1, \ldots, V_K)$ and $(\tilde{V}_1, \ldots, \tilde{V}_K)$ are neighbors if $\sum_{i=1}^{K} \left| |V_i| - |\tilde{V}_i| \right| = 2$.

The main section of this algorithm is the *while* loop, which simulates the annealing process, and the fundamental procedure within the *while* loop is move( ). Procedure move( ) chooses a random neighbor of the current partition, each neighbor being equally likely to be chosen. If the cost of the neighbor is at most the cost of the current partition then the neighbor is accepted as the new current partition. Otherwise, it accepts this neighbor as the new current partition with probability exp(- Δ/T), and with probability 1 - exp(- Δ/T) it leaves the current partition as is, where Δ equals the cost of the neighbor minus the cost of the current partition. The temperature parameter T is decreased, by multiplying it by TFACTOR, and it is decreased after every MCLENGTH calls to move( ). For our experiments, the value of MCLENGTH was chosen to be the number of neighbors a partition has (=$(K-1)|V|$). This value of MCLENGTH was used in [22].

After the algorithm decreases T, it checks to see if it should stop simulating the annealing process. Variable *numaccept* stores the number of times neighbors were accepted as new current partitions since the last time T was decreased. If numaccept equals zero then the annealing process

BASIC SIMULATED ANNEALING ALGORITHM FOR GRAPH PARTITIONING

Input: Graph (V,E), positive integer K, positive integers $\sigma_1, \sigma_2, ..., \sigma_K$.
Output: A feasible partition of V with small capacity.

```
begin
    $(V_1, ..., V_K)$ :=init_state ( ), T := init_temperature ( );
    numaccept := 1, numtemp := 0;
    while (numaccept > 0) and (numtemp < R) do
        begin
          numaccept := 0, numtemp := numtemp +1;
          for i := 1, 2 , ..., MCLENGTH do move ( );
          T := TFACTOR * T;
        end
    if for some i $|V_i \neq \sigma_i$ then $(V_1, ..., V_K)$ := greedy_fix_up $(V_1, ..., V_K)$;
end

procedure move ( )
begin
    $(\tilde{V}_1, ..., \tilde{V}_K)$ := random_neighbor $(V_1, ..., V_K)$;
    $\Delta$ := cost $(\tilde{V}, ..., \tilde{V}_K)$ - cost $(\tilde{V}_1, ..., \tilde{V}_K)$;
    if $\Delta \leq 0$ then $(V_1, ..., V_K) := (\tilde{V}_1, ..., \tilde{V}_K)$, numaccept := numaccept +1;
    else if randunit ( ) $\leq$ exp $(-\Delta/T)$
    then $(V_1, ..., V_K) := (\tilde{V}_1, ..., \tilde{V}_K)$, numaccept := numaccept +1;
end

function randunit ( )
begin return (a random number uniformly distributed on the interval [0, 1])
        end
```

Figure 5.2.1. The basic implementation of the penalty method for graph partitioning, where the minimization is done by simulated annealing.

is assumed to be "frozen" and the simulation is stopped. Variable *numtemp* stores the number of times T is decreased. If numtemp exceeds a number R then the simulation is also terminated.

Initialization of the algorithm consists of initializing numtemp and numaccept, and calling *init_state()* and *init_temperature()*. Function *init_state()* returns a feasible partition of V where the annealing process can start from. Function *init_temperature* returns an initial value for T. The returned value is computed by randomly picking one hundred neighbors of an arbitrary feasible partition $(V_1, \ldots, V_K)$, and, for all neighbors that have bigger cost than $(V_1, \ldots, V_K)$, the sample average A of the cost of a neighbor of $(V_1, \ldots, V_K)$ minus the cost of $(V_1, \ldots, V_K)$ is computed. The temperature $\bar{T}$ that is returned is such that $\exp(-A/\bar{T}) = 0.4$. This method of initializing the temperature value was also used by Aragon et al. [20] and Kirkpatrick et al. [1].

After the simulation of the annealing process the final partition $(V_1, \cdots, V_K)$ computed may be infeasible. The function *greedy_fix_up* $(V_1, \cdots, V_K)$ is called and it returns a feasible partition. It does this by sequentially transferring nodes from subsets V_i to subsets V_j, such that $|V_i| > \sigma_i$, and $|V_j| < \sigma_j$, so that each transfer minimizes the increase in the capacity of the resulting partition.

We will now discuss the different implementations of the penalty function methods. For the hardlimiter (HL) and quadratic (Q) penalty methods, the implementation used is the one in Figure 5.2.1. The implementation of the method of multipliers (MM) is shown in Figure 5.2.2. In this implementation, the penalty parameters c_j are constant and equal to c and the augmented Lagrangian function is minimized J+1 times by simulated annealing. The first minimization is done by a long simulated annealing run and the next J minimizations are done by runs. In between minimizations the multiplier vector λ is updated.

The implementation of the dynamic method of multipliers (DMM) is shown in Figure 5.2.3. It is the same algorithm in Figure 5.2.1, excluding the additional lines enclosed in the two rectangles. These additional lines and the two additional variables $\Delta h = (\Delta h_i : i = 1, 2, \ldots, K)$ and

METHOD OF MULTIPLIERS

```
begin
    $(V_1, ..., V_K)$ :=init_state ( ), T := init_temperature ( );
    for i := 1, 2, , ...., K, do $\lambda_i$ := 0;
    numaccept := 1, numtemp := 0;
    while numaccept > 0) and (numtemp < R) do
        begin
            numaccept := 0, numtemp := numtemp +1;
            for j := 1, 2 , ...., MCLENGTH do move ( );
            T := TFACTOR * T;
        end
    $\hat{T}$ := $(TFACTOR)^{-numtemp/J}$ * T;
    for i := 1, 2 , ...., J do
        begin
            T := $\hat{T}$;
            for j := 1, 2 , ...., K do $\lambda_j + \lambda_j + c*(\sigma_j - |V_j|)$;
            numaccept := 1, numtemp := 0;
            while (numaccept > 0) and (numtemp < R/J ) do
                begin
                    numaccept := 0, numtemp := numtemp +1;
                    for t := 1, 2 , ...., MCLENGTH do move ( );
                    T := TFACTOR * T;
                end
        end
    if for some i , $|V_i| \neq \sigma_i$ then $(V_1, ..., V_K)$ := greedy_fix_up $(V_1, ..., V_K)$;
end
```

Figure 5.2.2. The implementation of the method of multipliers.

Zcount are used in maintaining λ. Note that $\Delta h_i/Z$ is the sample mean of $\sigma_i - |V_i|$ since the last time λ_i was updated.

For the algorithm in Figure 5.2.3, Δh_i is updated after each call to move( ). As remarked in the previous section, this can be very time consuming if K is large. A more efficient method is to update Δh_i only when $|V_i|$ changes value. To accomplish this we have a global counter globcount which is incremented immediately after every call to move( ), and, for each i, such that $1 \leq i \leq K$ we have additional variables lastdiff_i and lastcount_i: lastdiff_i equals $\sigma_i - |V_i|$ and lastcount_i equals the value of globcount when Δh_i was updated last. Every time $|V_i|$ changes value or λ_i is to be updated we increment Δh_i by $\sigma_i - |V_i| + (\text{globcount} - \text{lastcount}_i - 1)\text{lastdiff}_i$, set lastdiff_i to $\sigma_i - |V_i|$, and set lastcount_i to globcount.

Our experiments consisted of three cases: ($n = 240$, $m = 300$, $K = 2$, $\sigma_1 = \sigma_2 = 200$), ($n = 240$, $m = 300$, $K = 2$, $\sigma_i = 160$, $\sigma_2 = 80$), and ($n = 160$, $m = 200$, $K = 4$, $\sigma_1 = \sigma_2 = \sigma_3 = \sigma_4 = 40$). For each case, ten graphs were randomly and independently generated where each graph in G(n,m) was equally likely to be generated. Recall G(n,m) is the set of graphs with node set {1, 2,..., n} and m edges. For each case and parameter value considered, each algorithm was executed once for each of the ten random graphs. During these ten runs the average number of times T was decremented (AVG#TDEC) and the average capacity of the final partition (AVGCAP) were recorded. The number of times T was decremented is related to the running time of the algorithm, since the number of times T was decremented multiplied by MCLENGTH is equal to the length of the annealing process simulated.

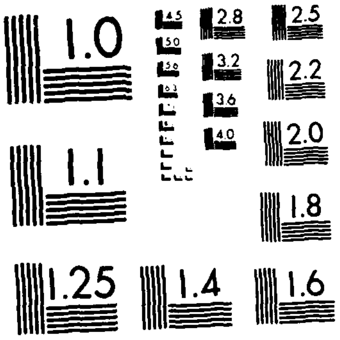
For all three cases, the average degree of the graph (equal to $2|E|/|V|$) is 2.5 and we compared the following penalty methods:

hardlimiter penalty method when $\gamma = 1$;

quadratic penalty method when $c = 10^{-4+i/2}$ for $i = 0, 1, \ldots, 10$;

AD-A185 547 OPTIMIZATION BY SIMULATED ANNEALING: A TIME-COMPLEXITY 2/2
ANALYSIS(U) ILLINOIS UNIV AT URBANA DEPT OF ELECTRICAL
ENGINEERING G H SASAKI OCT 87 UILU-ENG-87-2261
UNCLASSIFIED N00014-82-K-0359 F/G 12/4 NL

END
DATE
FILMED
[illegible]



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

DYNAMIC METHOD OF MULTIPLIERS

```
begin
    $(V_1, ...., V_K)$ := init_state ( ), T := init_temperature ( );
for i := 1, 2 , ...., K do $\lambda_i$ := 0, $\Delta h_i$ := 0 ;
Zcount := Z;
    numaccept := 1, numtemp := 0;
    while (numaccept > 0) and (numtemp < R) do
        begin
            numaccept := 0, numtemp := numtemp +1;
            for i := 1, 2 , ...., MCLENGTH do
                begin
                    move ( )
            for j := 1, 2 , ...., do $\Delta h_j := \Delta h_j + \sigma_j - |V_j|$;
            Zcount := Zcount -1
            if Zcount = 0 then
            begin
                for j := 1 , ...., K do $\lambda_j := \lambda_j + c\Delta h_j/Z$, $\Delta h_j$ := 0 ;
                Zcount := Z;
            end
                end
            T := TFACTOR * T;
        end
    if for some i $|V_i| \neq \sigma_i$ then $(V_i, ...., V_K)$ := greedy_fix_up $(V_1, ...., V_K)$;
end
```

Figure 5.2.3. The implementation of the dynamic method of multipliers.

method of multipliers when J = 4 and $c = 10^{-4+i/2}$ for i = 0, 1, . . . , 10;

dynamic method of multipliers when $Z = (K-1)|V|$ and $c = 10^{-4+i/2}$ for i = 0, 1, . . . , 10.

For the hardlimiter penalty method, the quadratic penalty method, and the dynamic method of multipliers, we let TFACTOR equal 0.9. However, we let TFACTOR equal 0.81 for the method of multipliers, since it consists of multiple simulated annealing runs. The result was that the sum of the lengths of the annealing processes simulated were roughly the same for all the penalty methods. We will now discuss some of the details of each case.

Case 1 ($n = 240$, $m = 300$, $K = 2$, $\sigma_1 = \sigma_2 = 120$): In this case, the resulting partition is required to be balanced. For all penalty methods, we ran all the algorithms with R equal to 250. We also ran the algorithm for the hardlimiter penalty method for R equal to 50, so that the length of the annealing process simulated would be roughly the same as for the other penalty methods. The data are given in tables in Figure 5.2.4 and are plotted in graphs in Figure 5.2.5.

Case 2 ($n = 240$, $m = 300$, $K = 2$, $\sigma_1 = 160$, $\sigma_2 = 80$): In this case, the partition is required to be unbalanced. For all penalty methods, we ran all the algorithms with R equal to 250. We also ran the algorithm for the hardlimiter penalty method with R equal to 40, so that the length of the annealing processes simulated would be roughly the same as for the other penalty methods. The data are given in tables in Figure 5.2.6 and plotted in graphs in Figure 5.2.7.

Case 3 ($n = 160$, $m = 200$, $K = 4$, $\sigma_1 = \sigma_2 = \sigma_3 = \sigma_4 = 40$):

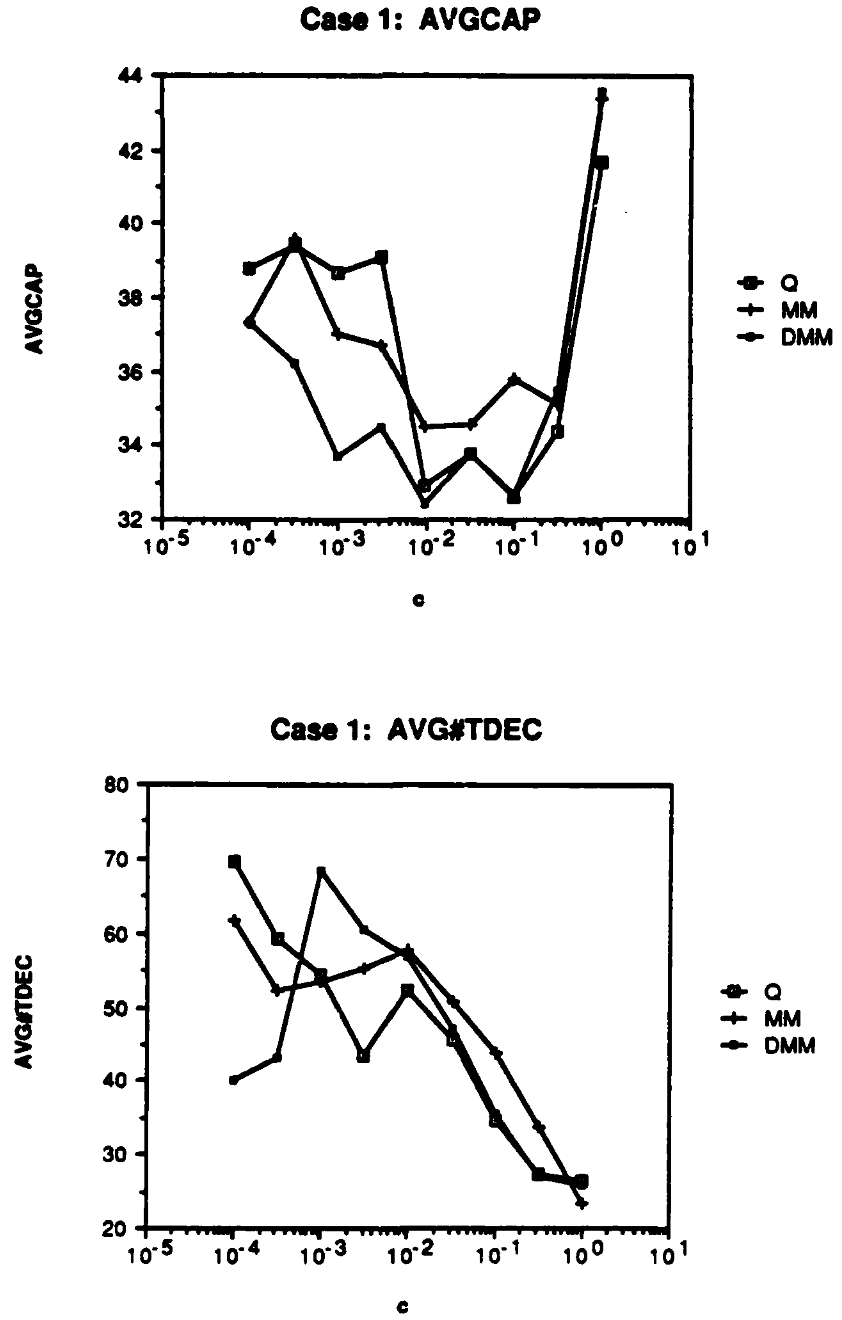
In this case, the partitions should be balanced, but we have four sets to the partition rather than two as in Case 1. For all penalty methods, we ran all the algorithms with R equal to 75. The data are given in tables in Figure 5.2.8 and plotted in graphs in Figure 5.2.9.

For all three cases, the quality of solutions for the quadratic penalty method, the dynamic method of multipliers, and the method of multipliers were dependent on the value of c. Very small values of c lead to poor solutions, because the solutions found by simulated annealing were far

		AVGCAP			AVG#TDEC		
R	$\log_{10}c$	Q	MM	DMM	Q	MM	DMM
250	-4.0	38.8	37.3	37.3	69.5	61.6	40.2
250	-3.5	39.4	39.6	36.2	59.2	52.3	43.2
250	-3.0	38.7	37.0	33.7	54.3	53.6	68.5
250	-2.5	39.1	36.7	34.5	43.6	55.4	60.5
250	-2.0	32.9	34.5	32.4	52.3	57.7	56.9
250	-1.5	33.8	34.6	33.8	45.7	50.9	47.0
250	-1.0	32.6	35.8	32.7	34.7	43.7	35.4
250	-0.5	34.4	35.1	35.5	27.3	33.6	27.1
250	0.0	41.7	43.4	43.6	26.3	23.5	25.7
250	0.5	83.8	80.3	83.3	19.5	27.1	20.4
250	1.0	127.5	125.9	125.9	18.2	24.7	18.8

HL		
R	AVGCAP	AVG#TDEC
250	32.2	235.3
50	33.5	50.0

Figure 5.2.4. Results from computer experiments for Case 1 ($n = 240$, $m = 300$, $K = 2$, $\sigma_1 = \sigma_2 = 120$). AVGCAP = the average capacity of the final partition and AVG#TDEC = the average number of times T was decremented. Q = quadratic penalty method; MM = method of multipliers; DMM = dynamic method of multipliers; and HL = hardlimiter penalty method.


Case 1: AVGCAP

Case 1: AVG#TDEC

Figure 5.2.5. Graphs of results from computer experiments for Case 1 (n = 240, m = 300, K = 2, $\sigma_1 = \sigma_2 = 120$). AVGCAP = the average capacity of the final partition and AVG#TDEC = the average number of times T was decremented. Q = quadratic penalty method; MM = method of multipliers; DMM = dynamic method of multipliers; and HL = hardlimiter penalty method.

		AVGCAP			AVG#TDEC		
R	$\log_{10}c$	Q	MM	DMM	Q	MM	DMM
250	-4.0	31.6	29.3	30.4	61.2	52.0	37.8
250	-3.5	31.1	28.9	30.3	58.7	51.5	42.8
250	-3.0	29.9	29.3	24.8	42.9	47.7	34.8
250	-2.5	27.4	27.6	23.7	38.8	44.3	37.3
250	-2.0	27.6	26.5	24.7	29.6	39.6	34.6
250	-1.5	23.3	27.1	23.8	35.1	34.2	38.4
250	-1.0	24.5	26.8	24.2	33.7	38.8	30.9
250	-0.5	24.7	28.8	25.7	29.2	34.9	29.6
250	0.0	31.9	32.3	33.9	30.7	32.0	26.9
250	0.5	68.7	66.0	69.3	21.9	25.3	20.1
250	1.0	111.7	109.9	112.4	19.1	22.6	19.0

HL		
R	AVGCAP	AVG#TDEC
250	25.1	83.4
40	25.8	39.3

Figure 5.2.6. Results from computer experiments for Case 2 (n = 240, m = 300, K = 2, σ_1 = 160, σ_2 = 80). AVGCAP = the average capacity of the final partition and AVG#TDEC = the average number of times T was decremented. Q = quadratic penalty method; MM = method of multipliers; DMM = dynamic method of multipliers; and HL = hardlimiter penalty method.

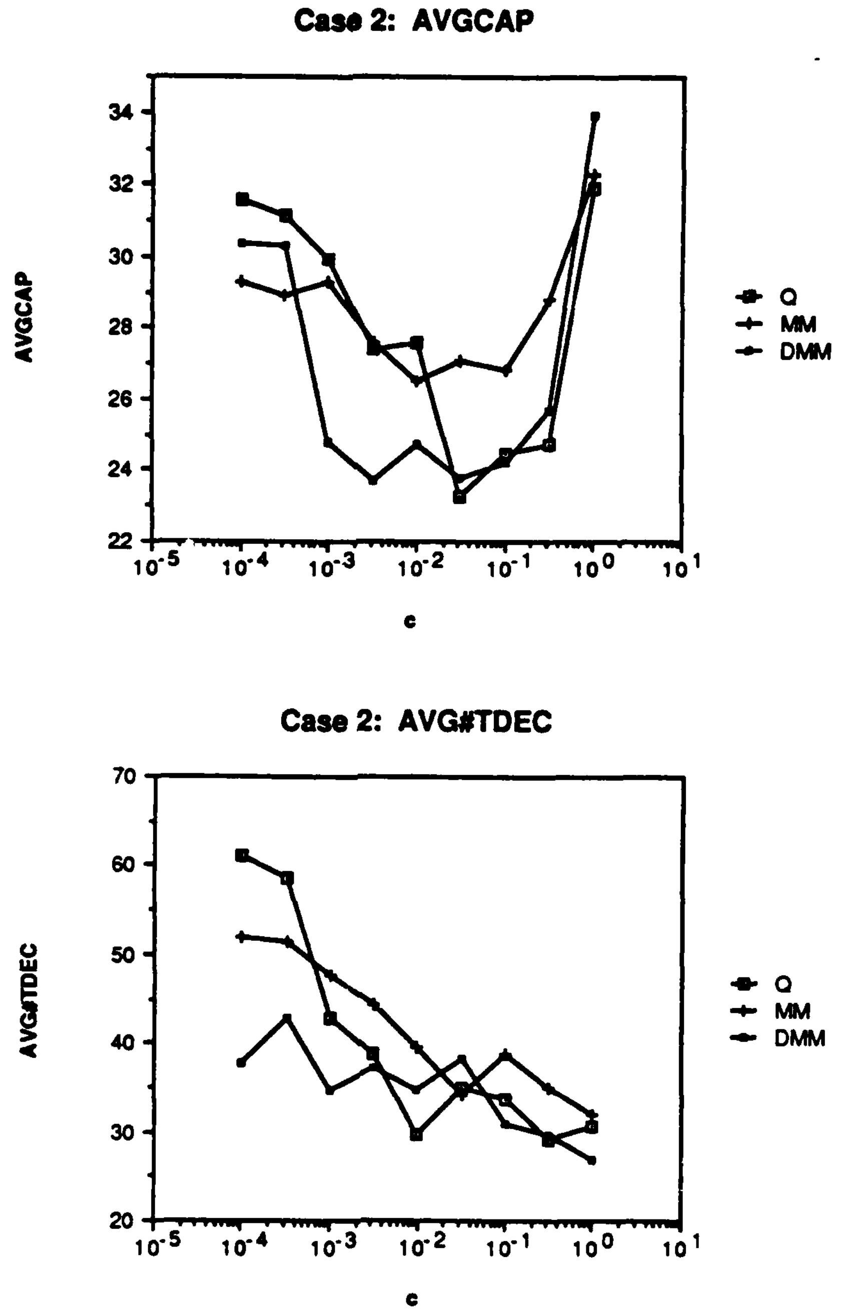
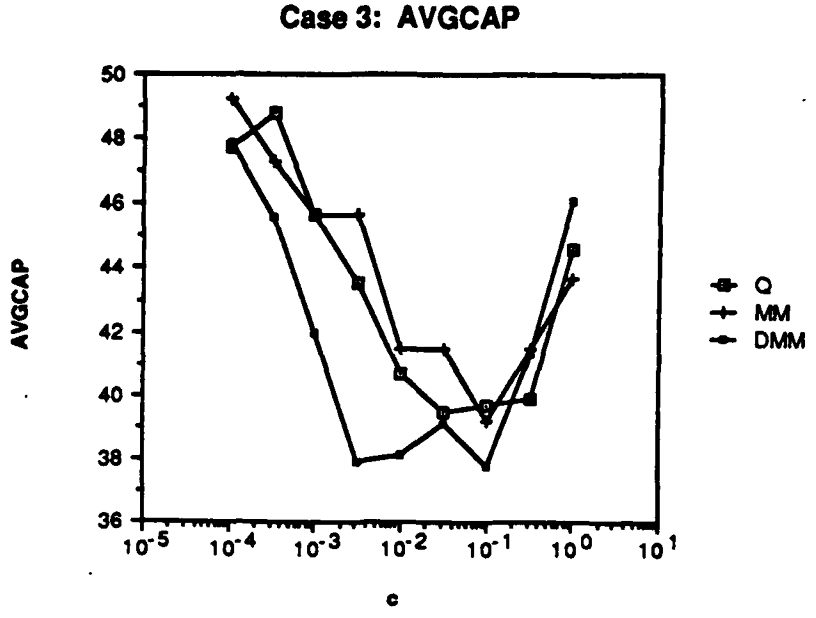


Figure 5.2.7. Graphs of results from computer experiments for Case 2 ($n = 240$, $m = 300$, $K = 2$, $\sigma_1 = 160$, $\sigma_2 = 80$). AVGCAP = the average capacity of the final partition and AVG#TDEC = the average number of times T was decremented. Q = quadratic penalty method; MM = method of multipliers; DMM = dynamic method of multipliers; and HL = hardlimiter penalty method.

		AVGCAP			AVG#TDEC		
R	$\log_{10}c$	QP	ALM	DALM	QP	ALM	DALM
75	-4.0	47.7	49.2	47.9	74.1	72.4	48.8
75	-3.5	48.8	47.3	45.6	71.5	63.1	42.4
75	-3.0	45.7	45.7	42.0	68.9	66.5	60.1
75	-2.5	43.5	45.7	37.9	53.5	46.8	53.6
75	-2.0	40.7	41.5	38.1	52.2	57.9	53.6
75	-1.5	39.5	41.5	39.1	50.3	53.0	44.7
75	-1.0	39.7	39.2	37.8	42.5	41.1	32.5
75	-0.5	39.9	41.5	41.3	25.9	34.1	26.7
75	0.0	44.6	43.7	46.1	28.5	28.7	25.1
75	0.5	94.4	86.0	91.0	21.5	27.3	22.0
75	1.0	132.3	129.9	132.4	19.6	25.9	20.3

HL		
R	AVGCAP	AVG#TDEC
75	39.3	75

Figure 5.2.8. Results from computer experiments for Case 3 ($n = 160$, $m = 200$, $K = 4$, $\sigma_1 = \sigma_2 = \sigma_3 = \sigma_4 = 40$). AVGCAP = the average capacity of the final partition and AVG#TDEC = the average number of times T was decremented. Q = quadratic penalty method; MM = method of multipliers; DMM = dynamic method of multipliers; and HL = hardlimiter penalty method.



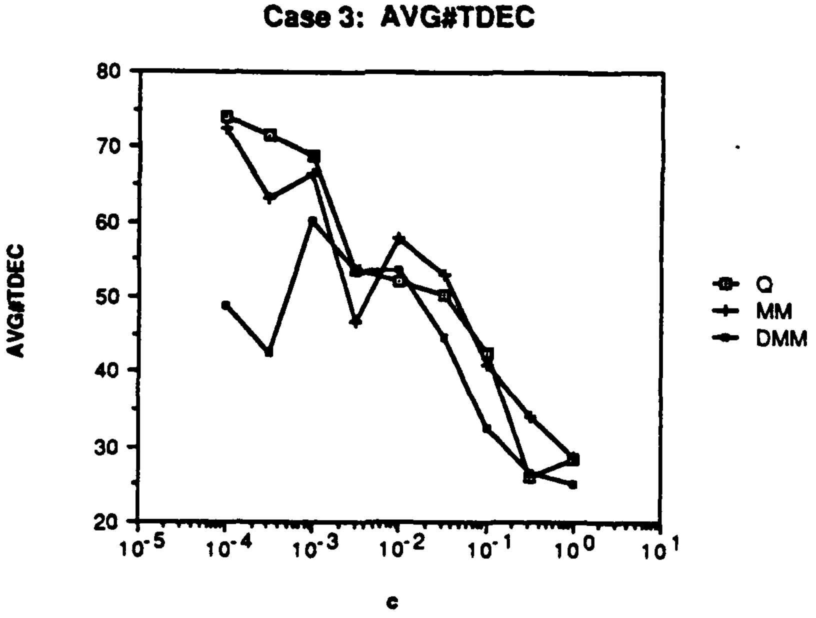


Figure 5.2.9. Results from computer experiments for Case 3 (n = 160, m = 200, K = 4, $\sigma_1 = \sigma_2 = \sigma_3 = \sigma_4 = 40$). AVGCAP = the average capacity of the final partition and AVG#TDEC = the average number of times T was decremented. Q = quadratic penalty method; MM = method of multipliers; DMM = dynamic method of multipliers; and HL = hardlimiter penalty method.

from satisfying the equality constraints and the greedy fix-up algorithm did not produce near optimal solutions. Very large values of c lead to poor solutions, because the large quadratic term in the augmented Lagrangian function restricted the movement of the annealing process.

For all three cases, the dynamic method of multipliers produced low-valued AVGCAP values over the widest range of values of c than the quadratic penalty method or the method of multipliers. In Figure 5.2.5 note that when c equaled 0.001 the dynamic method of multipliers produced a smaller value of AVGCAP and a larger value of AVG#TDEC than the quadratic penalty method or the method of multipliers. The implication is that there may be a quality of solution versus running time tradeoff in Case 1. However, there does not seem to be such a tradeoff in the other two cases.

Note that the dynamic method of multipliers yields better solutions than the quadratic method for small values of c, while the reverse is true for large values of c. This can partly be explained by the fact that in the dynamic method of multipliers the linear term of the augmented Lagrangian function behaves roughly like an extra quadratic term by the way it is updated. Hence, the dynamic method of multipliers is like a quadratic penalty method but with a larger penalty parameter.

A final observation is that the hardlimiter penalty method performed well when compared to the other penalty methods. For Case 1, the AVGCAP value of the hardlimiter method, when R equals 250, is smaller than the other penalty methods, and the AVGCAP value of the hardlimiter method, when R equals 50, is within 4% of the smallest AVGCAP value of the other methods. For Case 2, the AVGCAP value of the hardlimiter method, when R equals 250, is within 8% the smallest AVGCAP value, and the AVGCAP value of the hardlimiter method, when R equals 40, is within 11% of the best AVGCAP value. For Case 3, the AVGCAP value of the hardlimiter penalty method is within 4% of the smallest AVGCAP value of the other penalty methods.

Although the hardlimiter penalty method performs well over all cases, for certain values of c the dynamic method of multipliers produces better solutions for Cases 2 and 3.

We will digress to make a comment on the sensitivity of the quadratic penalty method, the method of multipliers, and the dynamic method of multipliers to the value of c in our experiments. The reason why c should not be too small can be explained if we assume that the capacity of partition $(V_1, \ldots, V_K)$ is well approximated by $(1-\sum_{i=1}^{K}\left[\frac{|V_i|}{n}\right]^2)m$, which is the average capacity of $(V_1, \ldots, V_K)$ over all graphs in G(n,m). Then the augmented Lagrangian function is well approximated by $\Gamma(g_1, g_2, \ldots, g_K)$

$$= (1 - \sum_{i=1}^{K} g_i^2)m + \sum_{i=1}^{K} \lambda_i(\frac{\sigma_i}{n} - g_i)n + \sum_{i=1}^{K}\frac{c}{2}\left[\frac{\sigma_i}{n} - g_i\right]^2 n^2,$$

where $g_i = |V_i|/n$. The Hessian matrix of Γ is $(-2m+cn^2)I$, where I is the identity matrix. Therefore, Γ is convex if $c \geq \frac{2m}{n^2}$ and concave if $c \leq \frac{2m}{n^2}$. In Figures 5.2.5, 5.2.7, and 5.2.9, note that as parameter c decreases, the quality of solutions found by the quadratic penalty method begins to degrade after c crosses the value of $\frac{2m}{n^2}$ ($\frac{2m}{n^2}$ is equal to 0.01 for Cases 1 and 2 and is equal to 0.016 for Case 3).

5.3. Conclusions

In this chapter, we considered solving the equality constrained problem by simulated annealing and penalty methods. In particular, we focused on the quadratic penalty method and adaptive variations of it. Our concern with the quadratic penalty method was that resulting solutions were sensitive to its penalty parameter values. Since simulated annealing is a very time consuming algorithm, finding a good penalty parameter value experimentally could be impractical.

This leads us to investigate adaptive penalty methods that would perhaps be less sensitive to parameter values. The two adaptive penalty methods we considered were based on the method of multipliers, and required the minimization of an augmented Lagrangian function. However, the multiplier vector of the function is dynamically and periodically adjusted. Of these two adaptive penalty methods our experiments showed that the dynamic method of multipliers worked best and produced low-valued solutions over a wider range of penalty parameter values than the quadratic penalty method. We think further investigation of adaptive penalty methods should be done.

CHAPTER 6

CONCLUSIONS

6.1. Summary of Thesis

This thesis consists of a collection of results, most of which concern the finite-time behavior of simulated annealing.

In Chapters 2 and 3, we analyzed simulated annealing when it is applied to the matching problem. In Chapter 2, we showed that in the worst case simulated annealing solves the problem in average time that is at least exponential in the number of nodes of the graph if (a) the simulated annealing algorithm is the basic simulated annealing algorithm in Subsection 2.2.1 or (b) if we restrict our attention to the constant-temperature schedules. An upper bound on the average time it takes the basic simulated annealing algorithm of Subsection 2.2.1 to find a near maximum matching is also given. If we only require the algorithm to find a matching of size at least a fixed fraction of the size of the maximum matching, this upper bound is polynomial in the number of nodes of the graph. An estimate on the average time the basic simulated annealing of Subsection 2.2.1 will solve the matching problem for a "typical" graph is given in Chapter 3. We also presented computer simulation data that demonstrated that this estimate was reasonable. If we restrict our attention to graphs that have at least as many edges as there are nodes, then the estimate is a polynomial function of the number of nodes of the graph, which contrasts the results in Chapter 2 and which is encouraging to proponents of simulated annealing.

Since there are efficient algorithms available to solve the matching problem [9], it is doubtful that simulated annealing will be used on that problem in practice. However, our analysis seem to be the first thorough theoretical analysis of the average time complexity of simulated annealing applied to a nontrivial combinatorial optimization problem. Also, we can at least test our intuition and experience against these results. For example, since simulated annealing is a simple heuristic, we would not expect it to outperform more sophisticated methods in the worst case, and our exponential average-time lower bounds compared with the $O(\sqrt{|V|}|E|)$ time algorithm of [9] certainly supports this expectation.

In Chapter 4, we presented a collection of results. The template method and some examples of its use were given in Section 4.2, and the threshold random search algorithm was given in Section 4.3. In Section 4.4, we presented conditions which imply that no monotone decreasing temperature schedule is optimal.

The use of adaptive penalty methods to solve equality constrained problems by simulated annealing was investigated in Chapter 5. One of these methods (the dynamic method of multipliers) was shown, through experiments, to provide low-valued solutions over a wider range of parameter values than the static penalty method (quadratic penalty method) we considered. We believe further study of adaptive penalty methods should be done.

6.2. Directions for Future Research on Simulated Annealing

One direction for future research is to extend the results of Section 2.4. Rather than restricting ourselves to constant temperature schedules we may consider monotone decreasing temperature schedules. Since we are relaxing the constraints for the temperature schedules, we will have to consider a smaller set of transition probability matrices than the set R(G) of Section 2.4 in order to maintain the same lower bound for Ω.

A second direction for future research is to use a "physicist's" approach to analyze the behavior of simulated annealing for some NP-Complete problem. What we mean by a physicist's approach is to make reasonable assumptions to model the annealing process by a process that can be analyzed. Then experiments should be used to check the accuracy of the model. An example of this type of analysis, but on a known polynomial-time problem, was done in Chapter 3.

Consider the system (S, c, R), where S is a set of states, c is a cost function on S, and R is a probability transition matrix over S. Let (X_k: k ≥ 0) be an annealing on the system (S, c, R) with temperature schedule T. Most theoretical research is concemed with determining a good temperature schedule. However, it may be that the perfomance of simulated annealing is more dependent on the choice of R. A third direction for future research is to study how the choice of R affects the performance of simulated annealing. A related direction for future research is to come up with ways to systematically modify R that may improve the performance of the simulated annealing algorithm. For example, using R^2 rather than R may be preferable in certain cases. The template method may be used to illustrate these cases.

REFERENCES

[1] S. Kirkpatrick, C.D. Gelett and M.P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, 1983, pp. 621-630.

[2] V. Černy, "A thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm," *J. Opt. Theory and Appl.*, vol. 45, 1985, pp. 41-51.

[3] A. A. El Gamal, L. A. Hemachandra, I. Shperling, and V. K. Wei, "Using simulated annealing to design good codes," *IEEE Trans. Inform. Theory*, vol. IT-33, 1987, pp. 116-123.

[4] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, 1984, pp. 721-741.

[5] D. Mitra, F. Romeo, and A. Sangiovanni-Vincentelli, "Convergence and finite-time behavior of simulated annealing," *Adv. Appl. Prob.*, vol. 18, 1986, pp. 747-771.

[6] S. B. Gelfand and S. K. Mitter, "Analysis of simulated annealing for optimization," *Proc. 24th Conf. on Decision and Control*, IEEE Press, 1985, pp. 779-786.

[7] M. Lundy and A. Mees, "Convergence on an annealing algorithm," *Math. Prog.*, vol. 34, 1986, pp. 111-124.

[8] S. M. Ross, *Stochastic Processes*. New York: Wiley, 1983.

[9] S. Micali and V. V. Vazirani, "An $O(\sqrt{|V|}\cdot|E|)$ Algorithm for Finding Maximum Matching in General Graphs," *Proc. 21st Symp. Foundations of Computer Science*, IEEE Press, 1980, pp. 17-27.

[10] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Englewood Cliffs: Prentice Hall, 1982.

[11] B. Hajek, "Cooling schedules for optimal annealing," *Math. Oper. Res.*, 1987, to appear.

[12] S. Anily and A. Federgruen, "Simulated annealing methods with general acceptance probabilities," *J. Appl. Probl.* to appear.

[13] B. Hajek, "Hitting time and occupation time bounds implied by drift analysis with applications," *Adv. Appl. Prob.*, vol. 14, 1982, pp. 502-525.

[14] J. W. Greene and K. J. Supowit, "Simulated annealing without rejected moves," *IEEE Trans. Comput.-Aided Design*, vol. 5, 1986, pp. 221-228.

[15] J. E. Hopcraft, and R. M. Karp, "An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs," *SIAM J. Comput.*, vol. 2, 1973, pp. 225-231.

[16] Y. S. Chow and H. Teicher, *Probability Theory: Independence, Interchangeability, Martingales*. New York: Springer-Verlag, 1978.

[17] P. Erdos and J. H. Spencer, "On the existence of a factor of degree one of a connected random graph," *Act Math. Acad. Sci. Hungar.*, vol. 17, 1966, pp. 359-368.

[18] E. M. Reingold, J. Nievergelt, and N Deo, *Combinatorial Algorithms: Theory and Practice*. Englewood Cliffs: Prentice-Hall, 1977.

[19] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco: W. H. Freeman and Co., 1979.

20] C. R. Aragon, D. S. Johnson, L. A. McGeoch and C. Shevon, "Optimization by simulated annealing: an experimental evaluation," *List of Abstracts, Workshop on Statistical Physics in Engineering and Biology*. Yorktown Heights, April 1984.

[21] D. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, New York: Academic Press, 1982.

[22] E. H. L. Aarts and P. J. M. van Laarhoven, "Statistical cooling: a general approach to combinatorial optimization problems," *Philips J. Res.*, vol. 40, 1985, pp. 193-226.

VITA

Galen Hajime Sasaki was bom on September 4, 1959. He attended the University of Hawaii at Manoa from 1977 to 1981, where he received the Bachelor of Science degree in Electrical Engineering. He continued his education at the University of Illinois at Urbana-Champaign, where he received both the Master of Science degree, in 1984, and the Doctor of Philosophy degree, in 1987, in Electrical Engineering.