Fall 1995

On the optimization problems in multiaccess communication systems

Gangsheng Wang
New Jersey Institute of Technology

Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

ABSTRACT

ON THE OPTIMIZATION PROBLEMS IN MULTIACCESS COMMUNICATION SYSTEMS

by
Gangsheng Wang

In a communication system, the bandwidth is often a primary resource. In order to support concurrent access by numerous users in a network, this finite and expensive resource must be shared among many independent contending users. Multi-access protocols control this access of the resource among users to achieve its efficient utilization, satisfy connectivity requirements and resolve any conflict among the contending users. Many optimization problems arise in designing a multi-access protocol. Among these, there is a class of optimization problems known as NP-complete, and no polynomial algorithm can possibly solve them. Conventional methods may not be efficient and often produce poor solutions. In this dissertation, we propose a neural network-based algorithm for solving NP-complete problems encountered in multi-access communication systems. Three combinatorial optimization problems have been solved by the proposed algorithms; namely, frame pattern design in integrated TDMA communication networks, optimal broadcast scheduling in multihop packet radio networks, and optimal channel assignment in FDMA mobile communication networks. Numerical studies have shown encouraging results in searching for the global optimal solutions by using this algorithm. The determination of the related parameters regarding convergence and solution quality is investigated in this dissertation. Performance evaluations and comparisons with other algorithms have been performed.

ON THE OPTIMIZATION PROBLEMS IN MULTIACCESS COMMUNICATION SYSTEMS

by
Gangsheng Wang

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

Department of Electrical and Computer Engineering

January 1996

APPROVAL PAGE

ON THE OPTIMIZATION PROBLEMS IN MULTIACCESS COMMUNICATION SYSTEMS

Gangsheng Wang

Nirwan Ansari, Dissertation Advisor — Date
Associate Professor, ECE Department, NJIT

Denis Blackmore, Committee Member — Date
Professor, Mathematics Department, NJIT

Edwin Hou, Committee Member — Date
Assistant Professor, ECE Department, NJIT

Zoran Siveski, Committee Member — Date
Assistant Professor, ECE Department, NJIT

Irving Wang, Committee Member — Date
Member of Technical Staff, AT&T Bell Laboratories

BIOGRAPHICAL SKETCH

Author: Gangsheng Wang

Degree: Doctor of Philosophy

Date: January 1996

Undergraduate and Graduate Education:

- Doctor of Philosophy in Electrical Engineering, New Jersey Institute of Technology, New Jersey, 1996
- Master of Science in Electrical Engineering, the Second Academy, Ministry of Aerospace, Beijing, P. R. China, 1988
- Bachelor of Science in Electrical Engineering, Huazhong University of Science and Technology, Wuhan, P. R. China, 1982

Major : Electrical Engineering

Presentations and Publications :

1. G. Wang and N. Ansari, "Searching for optimal frame patterns in an integrated TDMA communication system using mean field annealing," submitted to *IEEE Transactions on Neural Network.*
2. N. Ansari and G. Wang, "Neural networks for broadcast scheduling in telecommunications," *Fuzzy Logic and Neural Network Handbook* (C. H. Chen ed.), McGraw-Hill, New York, 1995.
3. G. Wang and N. Ansari, "Maximizing data throughput in an integrated TDMA communication system using mean field annealing," IEEE GLOBECOM'94, San Francisco, Dec. 1994.
4. G. Wang and N. Ansari, "A neural network approach to broadcast scheduling in multihop radio networks, " *Proc. of ICNN'94*, pp. 4699-4704, June 1994.
5. N. Ansari, R. Sarasa, and G. Wang, "An efficient annealing algorithm for global optimization in Boltzmann machines," *Journal of Applied Intelligence 3,* pp. 177-192, 1993.

This dissertation is dedicated to my family

ACKNOWLEDGMENT

I would like to express my sincere gratitude to my advisor, Dr. Nirwan Ansari, for his help and supervision during the whole period of the preparation and writing of this dissertation. Without his insight and patient scrutiny of every single detail, this dissertation will never be created.

I would also like to express my thanks to my committee members, Prof. Denis Blackmore, Prof. Edwin Hou, Prof. Zoran Siveski and Dr. Irving Wang for reviewing my dissertation. Their suggestive comments have greatly improved the quality of this dissertation.

I am grateful to my wife for her enduring patience and encouragement during the many long years that went to the dissertation preparation. Thanks are extended to my family for their understanding and support during my difficult times.

I also wish to thank my friends and colleagues for their assistance.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

LIST OF FIGURES

LIST OF FIGURES
(Continued)

CHAPTER 1

INTRODUCTION

With the rapid development of communication techniques, a communication network is expected to serve increasingly many users for data transmissions, information exchanges and communications. In a communication system, the bandwidth is often a primary resource. In order to support the concurrent access by numerous users in a network, this finite and expensive resource must be shared among many independent contending users. A typical example is the time division multiplexing (TDM) system developed in the 1960's, in which a powerful mainframe is accessed by a large number of users. Since each user has relatively small or infrequent demands, a dedicated system will greatly decrease the utilization of the computer resources. The TDM scheme assigns a fixed subset of the time-bandwidth space to each user and the computer system has successfully provided satisfactory services for many simultaneous processing requirements from users.

The communication bandwidth is usually divided into a single or a number of channels. The channel resources can be shared by users in three domains — time, frequency and space. Multiaccess protocols control the access of resources among users to achieve their efficient utilization, satisfy connectivity requirements and resolve any conflict among the contending users. The design of a multiaccess protocol depends on the type of communication network, the characteristics of data traffic in the network and the quality-of-service (QOS) requirement. Multiaccess protocols can be classified into three classes. The first class is *time-division multiple access* (TDMA) in which each user is confined to access the entire channel bandwidth only during the allocated time slots. The second class, known as *frequency-division multiple access* (FDMA), consists of assigning users a fraction of the bandwidth

and limiting their access to the allocated subband. The third class is *code-division multiple access* (CDMA), which realizes orthogonality of signaling waveforms by the use of different signaling codes and allows overlap in transmission in both the frequency and time coordinates. The access mode can be random, fixed or dynamic. Multiaccess protocols also differ by the distributed or centralized nature of the resource allocation algorithms.

Different multiaccess protocols have their own merits and disadvantages. In order to select a proper protocol operating in a network, one must investigate the characteristics of traffic, network topology, network type and operation cost. In a satellite network, for example, a single wideband channel is shared by all earth stations. In light traffic, a random TDMA protocol can provide high throughput and a rather simple way to access the channel. When the network is overloaded, however, the collision probability is higher and the data needed to retransmit will further load the network. In this situation, a fixed TDMA protocol provides better performance.

The earliest multiplexing techniques, such as TDM (time-division multiplexing) and FDM (frequency-division multiplexing), are very successful for stream-type traffic such as voice. However, for data communication, especially for computer communication, the traffic is usually characterized as *bursty*. To serve the bursty-type traffic, Packet communication networks have been developed over the past two decades [2][23][37][38][39]. Packet communication networks can be classified as store-and-forward point-to-point networks, LAN's, packet radio networks, and packet satellite networks [38]. Packet radio (PR) is a technology which applies the packet switching technique to the broadcast radio environment. Since a packet radio network has such advantages as allowing direct communications among mobile users over wide geographical areas, coexistence with different systems in the same frequency

band, protection against multipath effects as well as antijamming protection [23], it has been widely used in computer and mobile communications [2][15][23][26].

1.1 Motivation of the Dissertation

The objective of introducing multiple access protocols in communication networks is to fully utilize the channel resources and minimize the time delay of data. A given type of multiple access protocol usually involves many optimization problems. For example, for an integrated voice/data communication system in which a random TDMA protocol is applied, the different time-slot arrangements for data transmissions within a frame will result in a different data throughput. The optimization problem in this system is to search for the optimal frame patterns (the relative positions of data transmissions in a frame) that can provide the maximum data throughput while keeping satisfactory quality of voice service. We will present a detailed discussion of this problem in the sequel.

In packet communication networks, many complicated optimization problems arise as the demand and sophistication of the area grows. Conventional strategies may no longer meet the new challenges. Especially for constrained optimization problems, conventional methods may get stuck in poor local optima. There is a class of optimization problems known as NP-complete. It has been shown that the computational complexity will exponentially increase with the problem size and no polynomial algorithms can be found for this class of problems. One alternative way to solve the problems is using approximation techniques, i.e., instead of searching the exact global optima with rather complicated computations, the approximation techniques attempt to find solutions that are close to the global optima with much less computational effort. Since neural networks have shown great promise in solving NP-complete problems, such as the traveling salesperson and content-addressable memory problems[19] [20] [31] [32] [33], we propose to apply neural networks to solve

the optimization problems in multiaccess communication systems. In this dissertation, we investigate the feasibility of neural network methods. We propose a *mean field annealing* (MFA) based algorithm in solving difficult optimization problems encountered in multiaccess communication systems. MFA is only a general scheme used in statistical mechanics. Even though some applications in solving optimization problems by MFA have been found in literatures [31]-[35], there are still many open questions regarding convergence, stability and parameter selection. In this dissertation, we use the stability theory to discuss the stability of MFA iterations and the determination of the parameters related to convergence, computational complexity and solution quality. Three optimization problems have been successfully solved by our MFA approach. Numerical results have shown that the proposed algorithm provides much better performance in both computational complexity and solution quality than some existing algorithms. We have proved that two problems encountered in our dissertation are NP-complete. The derived convergence results and criteria are applicable to many other NP-complete problems.

1.2 Arrangement of the Dissertation

The dissertation is organized as follows: After the Introduction, Chapter 2 presents the fundamental theory of NP-completeness. Two known NP-complete problems, namely 3-Satisfiability and CLIQUE, are introduced in order to prove the NP-completeness of the optimization problems encountered in the dissertation. Chapter 3 overviews the annealing process in statistical mechanics and simulated annealing (SA). Afterwards, mean field annealing is investigated. The convergence issue is discussed and the determinations of the related parameters are addressed. In Chapter 4, MFA is applied to searching for the optimal patterns in an integrated TDMA communication system. Chapter 5 presents an optimal broadcast scheduling algorithm based on neural networks in a packet radio communication network.

Chapter 6 proposes an MFA-based optimal channel assignment approach for FDMA mobile communication systems. The results and the achieved performances are demonstrated in each individual chapter. Conclusions are made in the last chapter. In the appendices, we prove that the broadcast scheduling and channel assignment problems are NP-complete.

CHAPTER 2

NP-COMPLETENESS

For many years many researchers have tried to find efficient algorithms for solving various combinatorial optimization problems arising in areas such as telecommunication, computer science and operations research. Among them there is a class of difficult optimization problems called *NP-complete*, for which efficient algorithms rarely exist. NP-complete problems are considered as intractable since the computational complexity increases exponentially with the problems size. In this chapter, we study the complexity class P of polynomial-time solvable decision problems and the class NP of decision problems. The proof procedures for NP-complete problems via polynomial-time "transformation" are addressed. Two NP-complete problems, namely 3-Satisfiability (3SAT) and CLIQUE, are introduced in order to prove the NP-completeness of the optimization problems studied in this dissertation. For detailed NP-complete theory, we refer to [16].

2.1 Polynomial Time and Exponential Time Algorithms

A function $f(n)$ is $O(g(n))$ whenever there exists a positive constant c such that values of $|f(n)| \leq c \cdot |g(n)|$ for all values of $n \geq 0$. A polynomial time algorithm is defined to be the one whose time complexity function is $O(p(n))$ for some polynomial function p, where n is used to denote the input length, i.e., the problem size [16]. A polynomial algorithm is often considered efficient due to the following reasons:

- It usually takes little computing time.
- None of the difficult problems have polynomial algorithms for their solutions.

- If a problem can be solved in polynomial time in one model, it can also be solved in polynomial time in another model.

To illustrate the efficiency, we give the following example: Suppose that there are two algorithms for a solution of a given problem, Algorithm A is of complexity n^2 (polynomial algorithm) and Algorithm B is of 2^n (exponential algorithm), where the complexity functions express execution time in terms of microseconds. If the problem size n is 50, the execution time is 0.0025 second for Algorithm A and 35.7 years for Algorithm B ! This example demonstrates the reason why a polynomial time algorithm is more desirable than its exponential counterpart. A problem is referred to as *intractable* if no polynomial time algorithms can possibly solve it.

2.2 Decision Problems, Encodings and Languages

The theory of NP-completeness restricts attention to decision problems. Such problems have only two possible solutions: "yes" and "no". A decision problem $\prod$ consists simply of a set $D_{\prod}$ of instances and a subset $Y_{\prod} \subseteq D_{\prod}$ of yes-instances, where an instance of a problem is obtained by specifying particular values for all the problem parameters. However, in the real world, many optimization problems are not decision problems. In order to apply the theory of NP-completeness to optimization problems, an optimization problem is usually recast as a decision problem by applying a numerical bound B to the problem, i.e., a minimization problem can be recast as a decision problem by asking whether there exists a structure of the required type having value "no more than" B. In the analogous way, a maximization problem can be recast by finding whether there is a structure of the required type having value "at least" B.

The reason for the restriction to decision problems is that they make it easy to use the machinery of formal-language theory. For any finite set Σ of symbols, a

language L over Σ is defined as any set of strings made up of symbols from Σ. For instance, if $\Sigma = \{0,1\}$, the set of L=$\{1, 11, 101, 111, \cdots\}$ is the language of binary representations of odd numbers. The language of all strings over Σ is denoted by Σ^*. For example, if $\Sigma = \{0,1\}$, then $\Sigma^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, \cdots\}$. ϵ denotes the empty string. An *encoding scheme* is introduced to represent problem instances in a way that computers understand. An encoding of a set S of abstract objects is a mapping e from S to the set of binary strings. For example, the ASCII encoding for alphabet a is $e(a)$=1100001. Encoding schemes establish the correspondence between decision problems and languages. Therefore, a problem $\prod$ and its encoding scheme e partition $\sum^*$ into three classes of strings: those that are not encodings of instances of $\prod$; those that encode instances of $\prod$ for which the answer is "no", and those that encode instances of $\prod$ for which the answer is "yes". The third class of strings is the languages we associate with $\prod$ and e:

$$L[\prod, e] = \{x \subseteq \Sigma^* : \Sigma \text{ is the alphabet used by } e, \text{ and } x \text{ is the encoding under } e \text{ by an instance } I \in Y_\Pi, Y_\Pi \text{ is the set of } \textit{yes} \text{ instances}\}$$

Any decision problem $\prod$ may have many encoding schemes, and different schemes have different *input lengths* which are used to evaluate the time complexity. For example, the binary encoding of decimal 13 is 1101 and its BCD encoding is 0001 0011. The input lengths are 4 and 8, respectively. A function $f : \{0,1\}^* \rightarrow \{0.1\}^*$ is defined *polynomial-time computable* if there exists a polynomial-time algorithm that, given any input $x \in \{0.1\}^*$, produces an output $f(x)$. For some set I of problem instances, we say that two encodings e_1 and e_2 are *polynomially related* if there exist two polynomial-time computable functions f' and f'' such that for any $i \in I$, we have $f'(e_1(i)) = e_2(i)$ and $f''(e_2(i)) = e_1(i)$. If two encodings e_1 and e_2 of a problem are polynomially related, the encoding $e_2(i)$ can be computed from the encoding $e_1(i)$ by a polynomial-time algorithm, and vice versa. Therefore, two encoding schemes for a problem $\prod$ will yield polynomially related input lengths. For a decision problem

$\prod$ on an instance set I, if e_1 and e_2 are polynomially related encodings on I, then $e_1(\prod) \in P$ if and only if $e_2(\prod) \in P$. P denotes the complexity class in which the decision problems can be solved in polynomial time. If the property holds, we do not need to specify any specific encoding schemes any more when we prove the NP-completeness for a given decision problem.

2.3 The Complexity Classes P and NP

"NP" stands for nondeterministic polynomial time. Before we proceed to the definition of "NP class", we need to clarify the following definitions:

Acception: A string $x \in \{0,1\}^*$ is *accepted* by an algorithm A if given input x, the algorithm outputs $A(x) = 1$. The language accepted by the algorithm is the set $L=\{x \in \{0,1\}^* : A(x) = 1\}$.

Rejection: An algorithm A *rejects* a string x if the output $A(x) = 0$.

Decision: A language L is *decided* by an algorithm A if every binary string is either accepted or rejected by the algorithm.

Polynomial-time acceptance: A language L is *accepted in polynomial time* by an algorithm A if for any length-n string $x \in L$, the algorithm accepts x in time $O(n^k)$ for some positive constant k. To accept a language, an algorithm only needs to consider the strings in L.

Polynomial-time Decision: A language L is decided in polynomial time by an algorithm A if, for any length-n string $x \in \{0,1\}^*$, the algorithm decides x in time $O(n^k)$ for some positive constant k. To decide a language, an algorithm must accept or reject every string in $\{0,1\}^*$.

Using the above definitions, we can address the complexity class P:

$P = \{L \subseteq \{0,1\}^* :$ there exists an algorithm A that decides L in polynomial time.$\}$

A *nondeterministic polynomial time algorithm* is one that can *verify* whether a given instance $I \in Y_{\prod}$ in deterministic polynomial time. $Y_{\prod}$ is the set of *yes* instances. Here a verification algorithm is defined as a two-argument algorithm A, one argument is an ordinary input string x and the other is a binary string y called a *certificate*. Such an algorithm A verifies an input string x if there exists a certificate y such that $A(x,y)=1$. The language verified by a verification algorithm A is

$$L = \{x \in \{0,1\}^* : \text{ there exists } y \in \{0,1\}^* \text{ such that } A(x,y) = 1\}$$

Thus, *Class NP* is defined as the class of languages that can be verified by a polynomial-time algorithm. Notice that polynomial time verifiability does not imply polynomial time solvability. In short, *Class P* consists of problems that can be solved in polynomial time, whereas the class NP consists of problems for which a solution can be verified rather than solved in polynomial time. Obviously, if $L \in P$, then $L \in NP$, e.g., $P \subseteq NP$ since, if a language L is decided by an algorithm A, it automatically verifies the language. Intuitively we know that to solve a problem is more difficult than to verify (check) a presented solution. This analogy extends to the class P and NP, and thus, NP includes languages that in P.

2.4 Transformability and NP-Completeness

As shown above, we have $P \subseteq NP$. Thus, every decision problem solvable by a polynomial time deterministic algorithm is also solvable by a polynomial time nondeterministic algorithm. It is believed that $P \neq NP$. All problems in P can be solved by polynomial time algorithms, whereas all problems in $NP - P$ are intractable. The theory of NP-completeness for any NP-complete problem Π focuses on proving results "if $P \neq NP$, then $\prod \in NP - P$".

2.4.1 Transformability

A problem $\prod$ can be transformed to another problem $\prod'$ if any instance $x \in \Pi$ can be mapped into an instance $x' \in \prod'$. Thus, if a problem $\prod$ can be transformed to another problem $\prod'$, then $\prod$ should be no harder to solve than $\prod'$. This transformation is denoted by $\Pi \propto \Pi'$.

A language L_1 is *polynomial-time transformable* to language L_2, denoted by $L_1 \propto_p L_2$, if there exists a polynomial-time computable function $f : \{0,1\}^* \rightarrow \{0,1\}^*$ such that for all $x \in \{0,1\}^*$, $x \in L_1$ if and only if $f(x) \in L_2$.

The function f is called a *transformation function.* It provides a polynomial-time mapping so that if $x \in L_1$, then $f(x) \in L_2$. Moreover, if $x \notin L_1$, then $f(x) \notin L_2$. Thus, the transformation function maps any instance x represented by L_1 to an instance $f(x)$ of the problem represented by L_2. Answering whether $f(x) \in L_2$ directly provides the answer to whether $x \in L_1$.

Lemma 2.1 *[16] If* $L_1, L_2 \subseteq \{0,1\}^*$ *and* $L_1 \propto_p L_2$, *then* $L_2 \in P$ *implies* $L_1 \in P$ *(equivalently* $L_1 \notin P$ *implies* $L_2 \notin P$*)*

2.4.2 NP-Completeness

A language $L \in \{0,1\}^*$ is defined to be *NP-complete* if $L \in NP$ and $L' \propto_p L$ $\forall L' \in NP$. It is equivalent to say that, a decision problem $\prod$ is NP-complete if $\prod \in NP$ and for all other decision problems $\prod' \in NP$ and $\prod' \propto_p \prod$. Therefore, NP-complete problems are the hardest problems in NP. If any one NP-complete problem can be solved in polynomial time, then every problem in NP has a polynomial-time solution. If any problem in NP is intractable, then so are all NP-complete problems. If $L' \propto_p L$ for every $L' \in NP$, but not necessarily with $L \in NP$, then L is called *NP-hard.*

Lemma 2.2 *[16] If $L_1, L_2 \in NP$, L_1 is NP-complete, and $L_1 \propto_p L_2$, then L_2 is NP-complete.*

This lemma provides us an insight for proving new problems NP-complete. Once we have at least one known NP-complete problem available, to prove that $\prod$ is NP-complete, we merely show that

1. $\prod \in NP$, and
2. Some known NP-complete problem $\prod'$ transforms to $\prod$.

2.5 NP-completeness Proofs

For a given problem $\prod \in NP$, we can prove it NP-complete if we can show that some already known NP-complete problem $\prod'$ can be transformed to $\prod$. Therefore Lemma 2.2 provides us with an approach to prove that problem $\prod$ represented by a language L is NP-complete:

1. Prove $L \in NP$
2. Select a known NP-complete language L'
3. Find a transformation function f which satisfies $x \in L'$ if and only if $f(x) \in L$ for all $x \in \{0,1\}^*$
4. Prove that the function f runs in polynomial time.

2.6 3-Satisfiability Problem (3SAT)

The satisfiability problem (SAT) is a decision problem which asks whether there is a satisfying truth assignment for a given collection C of clauses. Let $U = \{u_1, u_2, \cdots, u_k\}$ be a set of *boolean* variables. Variable u or its negation $\overline{u}$ is defined

as a *literal* over U. A *clause* over U is a set of literals over U such as $\{u_1, \overline{u}_3, u_4\}$. A *truth assignment* for U is a boolean function $\theta: U \rightarrow \{0,1\}$. If $\theta(u) = 1$, we say that u is "true" under θ; if $\theta(u) = 0$ we say that u is "false". The literal u is true under θ if and only if the variable u is true under θ; the literal $\overline{u}$ is true if and only if the variable u is false. A clause over U is *satisfied* by a truth assignment if and only if at least one of its members is true under that assignment. For example, the clause C=$\{u_1, \overline{u}_3, u_4\}$ is satisfied by θ except $\theta(u_1) = 0$, $\theta(u_3) = 1$ and $\theta(u_4) = 0$.

A collection C of clauses over U is satisfiable if and only if there exists some truth assignment for U that simultaneously satisfies all the clauses in C. Such a truth assignment is called a *satisfying truth assignment* for C. Equivalently if we express a *boolean formula* Φ as an AND of clauses, each of which is the OR of one or more literals, then a truth assignment is satisfiable if it results in $\Phi = 1$. For example, $U = \{u_1, u_2\}$, $C = \{\{u_1, \overline{u}_2\}, \{\overline{u}_1, u_2\}\}$, a satisfying truth assignment is given by $\theta(u_1) = 1, \theta(u_2) = 1$ since

$$\Phi = (u_1 \vee \overline{u}_2) \wedge (\overline{u}_1 \vee u_2) = (1 \vee \overline{1}) \wedge (\overline{1} \vee 1) = (1 \vee 0) \wedge (0 \vee 1) = 1$$

where $\wedge$ and $\vee$ are logic AND and OR operations. Therefore, for a given instance consisting of a set U of boolean variables and a collection C of clauses over U, the satisfiability asks whether there is a satisfying truth assignment for an arbitrary boolean formula.

The 3SAT problem is just a restricted version of SAT in which all instances have exactly three literals per clause. The 3SAT problem can be described as

INSTANCE: Collection $C = \{C_1, C_2, ..., C_m\}$ of clauses on a finite set U of variables such that $|C_i| = 3$ for $1 \leq i \leq m$.

QUESTION: Is there a truth assignment for U that satisfies all the clauses in C?

The naive algorithm to determine whether an arbitrary boolean formula is satisfiable does not run in polynomial time. There are 2^n possible assignment S in

a formula Φ with n variables. As the following theorem shows, the 3SAT problem is NP-complete, so a polynomial-time algorithm is unlikely to exist.

Theorem 2.1 *[16] 3SAT is NP-complete*

The proof can be found in [16]. Its simple structure makes it useful to prove other NP-completeness results. Problem CLIQUE is to be proved NP-complete based on the known 3SAT problem in the next section.

2.7 The CLIQUE Problem

In an undirected graph $G = (V, E)$, a *clique* is a subset $V' \subseteq V$ of vertices, each pair of which is connected by an edge in E. Therefore, a clique is a complete subgraph of G. The size of a clique is the number of vertices it contains. The clique problem is an optimization problem of finding a clique of maximum size in a graph. Recasting this optimization as a decision problem, we can describe the CLIQUE problem as

INSTANCE: A graph $G = (V, E)$ and a positive integer $J \leq |V|$

QUESTION: Does G contain a clique of size J or more such that a subset $V' \in V$, $|V'| \geq J$ and every two vertices in V' are jointed by an edge in E?

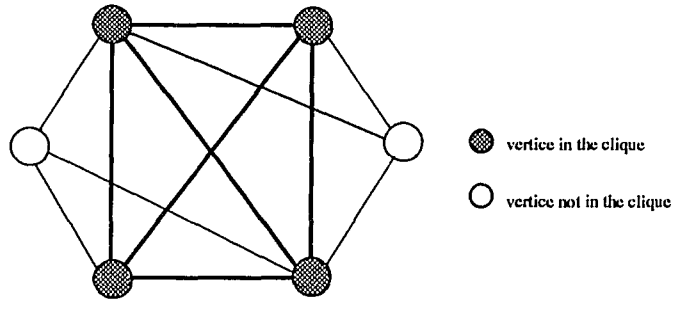


Figure 2.1 An example of a clique

Fig. 2.1 is an example of a clique in which the maximum clique has size 4.

Theorem 2.2 *The CLIQUE problem is NP-complete.*

Proof:

1. Show that CLIQUE $\in$ NP

   For a given graph $G = (V, E)$, we use the set $V' \subseteq V$ of vertices in the clique as a certificate for G. Verifying whether V' is a clique can proceed by checking whether, for every pair $u, v \in V'$, the edge $(u, v) \in E$. The computation is limited up to the following number of checking operations

   $$(n-1) + (n-2) + ... + 2 + 1 = n(n-1)/2.$$

   The computional complexity of checking whether the set $V' \subseteq V$ is a clique is $O(n^2)$, thus the verification can be accomplished in polynomial time, therefore CLIQUE $\in$NP.

2. Transformation from 3SAT to CLIQUE

   Here, we show that the known NP-complete problem 3SAT can be transformed to CLIQUE. i.e., 3SAT $\propto$CLIQUE, where $a \propto b$ stands for the transformation from a to b.

   Given a 3SAT instance with boolean variable set $u = (u_1, u_2, u_3, \cdots, u_k)$ and clauses $C = \{C_1, C_2, \cdots, C_m\}$ ($|C_i| = 3\ \forall i$), a boolean formula is formed by $\Phi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$, each clause C_i has exactly three distinct literals l_1^i, l_2^i and l_3^i. We can construct a graph $G = (V, E)$ as follows. For each clause C_i=$(l_1^i \vee l_2^i \vee l_3^i)$ in C, we place a triple of vertex v_1^i, v_2^i and v_3^i in V. An edge between two vertices v_x^i and v_y^j is added ($1 \leq x, y \leq 3$; $i, j \leq m$) if $i \neq j$ and $l_x^i \neq \overline{l_y^j}$. To illustrate the graph construction, Fig. 2.2 shows a graph G derived from the 3SAT boolean formula $\Phi = C_1 \wedge C_2 \wedge C_3$, where $C_1 = (u_1 \vee \overline{u}_2 \vee u_3), C_2 = (u_1 \vee \overline{u}_2 \vee \overline{u}_3)$ and $C_3 = (u_1 \vee u_2 \vee u_3)$. The maximum size of a clique is 3.

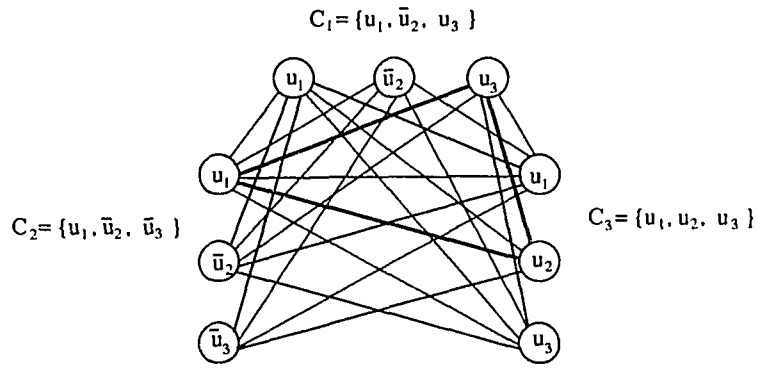


Figure 2.2 The derived graph from Φ. The clique with size 3 is $l' = \{1, 2, 3\}$

We show that, if a graph G is constructed in this way, a boolean formula Φ is satisfiable if and only if G has a clique of size m.

Assume that the boolean formula Φ is satisfied by a truth assignment. Then, each clause C_i should have at least one literal $l_x^i (x = 1, 2, 3, i = 1, 2, \cdots, m)$ that is assigned 1, and each such literal corresponds to a vertices v_x^i. A set of V' of m vertices can be formed if one "true" literal from each clause is picked up. For any two vertices $v_x^i, v_y^j \in V'$, where $i \neq j$, both corresponding literals l_x^i, l_y^j are mapped to 1 by the given satisfying assignment, and thus the literals cannot be complementary. Thus, by the construction of G, the edge $(v_x^i, v_y^j) \in E$. Therefore, if the formula Φ has a satisfying assignment, then the constructed graph G has a clique V' of size m. On the other hand, suppose that G has a clique V' of size m. Since there are no edges in the same triple (clause) and V' contains exactly one vertex per clause, there is no edge connecting two vertices v_x^i, v_y^j based on the rule for a graph if $l_x^i = \overline{l_y^j}$. Thus, a literal and its complement will not be assigned 1 simultaneously. Therefore, each clause C_i is satisfied, which results in a satisfied Φ. Thus, if the constructed graph G has a clique of size m , then the boolean formula Φ is satisfied.

In the example of Fig. 2.2, a satisfying assignment of Φ is $u_1 = 1, u_2 = 1, u_3 = 1$. A corresponding clique of size $k = 3$ consists of the vertices $\{u_1, u_2, u_3\}$. Based on the above observation, we can see that the 3SAT problem can be transformed to the CLIQUE problem, i.e., 3SAT $\propto$CLIQUE.

3. Polynomial Transformation

The transformation from 3SAT to CLIQUE can be completed in polynomial time. For a given formula $\Phi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$ and C_i=(l_1^i, l_2^i, l_3^i) $\forall i$, each clause has three literals, therefore, each clause forms triple vertices. The number of vertices in the triple graph equals to $3m$. The maximum number of added edges is $3|V| = 9m(m-1)$. Therefore the complexity transformation from 3SAT to CLIQUE is $O(m^2)$, a polynomial time transformation.

Since we have shown CLIQUE $\in NP$ and the transformation 3SAT$\propto$ CLIQUE is of polynomial time. We have proved that CLIQUE is NP-complete.

CHAPTER 3

MEAN FIELD ANNEALING THEORY

3.1 Combinatorial Optimization Problems

In a combinatorial optimization problem, cost is defined as a function of discrete variables representing configurations. A combinatorial optimization problem is described by $\Pi = (f, \mathcal{S})$, where $\mathcal{S} = \{s\}$ is a finite set of configurations and f is the cost function, $s \in \mathbf{Z}^n$ and $f : s \rightarrow \mathbf{R}^+$. The objective is to find an optimal configuration s_{opt} which provides the minimum cost, i.e.,

$$f_{opt} = f(s_{opt}) = \min_{s \in \mathcal{S}} f(s). \tag{3.1}$$

3.2 Statistical Mechanics

Since there exists a significant analogy between statistical mechanics and the procedure of solving complicated combinatorial optimization problems, we first review the annealing process in statistical mechanics.

Statistical mechanics is concerned with the properties of a large number of particles in samples of liquid or solid matter. Since the number of particles is quite large per cubic centimeter, only the behavior of the system in thermal equilibrium at a given temperature is observable. Different position placements of particles in a liquid or solid matter will yield different energies. At each temperature all particles randomly move around until thermal equilibrium is reached. If a state is defined by the set of particle positions, then, at thermal equilibrium, the probability of the system being in state i is represented by the *Gibbs distribution* [1][43]:

$$\pi_i = Pr\{s = i\} = \frac{exp\left(-\frac{E(i)}{k_b T}\right)}{\mathcal{Z}}, \tag{3.2}$$

where $\mathcal{Z} = \sum_{i \in S} exp\left(-\frac{E(i)}{k_b T}\right)$ is called the *partition function*, k_b is the *Boltzmann constant*, T is the temperature and $E(i)$ is the energy of state i, $\mathcal{S}$ is the state space, and $k_b, T, E(i) \in \mathbf{R}^+$. It is easy to find that [1],

$$\lim_{T \to \infty} \pi_i = \lim_{T \to \infty} \frac{exp\left(-\frac{E(i)}{k_b T}\right)}{\sum_{j \in \mathcal{S}} exp\left(-\frac{E(j)}{k_b T}\right)} = \frac{1}{|\mathcal{S}|}, \tag{3.3}$$

implying that, at a very high temperature, all states are equally probable.

On the other hand, we have

$$\begin{aligned}
\lim_{T \to 0} \pi_i &= \lim_{T \to 0} \frac{exp\left(-\frac{E(i)-E_{min}}{k_b T}\right)}{\sum_{j \in \mathcal{S}} exp\left(-\frac{E(j)-E_{min}}{k_b T}\right)} \\
&= \lim_{T \to 0} \frac{exp\left(-\frac{E(i)-E_{min}}{k_b T}\right)}{\sum_{j \in \mathcal{S}_{min}} exp\left(-\frac{E(j)-E_{min}}{k_b T}\right) + \sum_{j \notin \mathcal{S}_{min}} exp\left(-\frac{E(j)-E_{min}}{k_b T}\right)} \\
&= \lim_{T \to 0} \frac{exp\left(-\frac{E(i)-E_{min}}{k_b T}\right)}{\sum_{j \in \mathcal{S}_{min}} exp\left(-\frac{E(j)-E_{min}}{k_b T}\right)} \\
&= \begin{cases} \frac{1}{|\mathcal{S}_{min}|} & \text{if } i \in \mathcal{S}_{min} \\ 0 & \text{otherwise,} \end{cases}
\end{aligned} \tag{3.4}$$

where $\mathcal{S}_{min} = \{i : E(i) = E_{min}\}$ and $E_{min} = \min_{j \in \mathcal{S}} E(j)$.

From this equation, we can see that, as the temperature approaches zero, the system will concentrate on the states with the minimum energy, i.e., the states with the minimum energy are more probable.

In statistical mechanics, the crystalline lattice structure in a solid usually yields lower energy. A physical process called *annealing* is often performed in order to form a crystal. In the annealing process, a solid in a *heat bath* is heated up by increasing the temperature of the heat bath until it is melted into liquid, then the temperature is slowly lowered. At each temperature, all particles randomly arranges themselves until thermal equilibrium is reached. If the cooling is slow enough to allow the solid to reach thermal equilibrium at each temperature, the low energy crystalline solid would be formed when the system is frozen ($T \to 0$). If the annealing is too fast, the solid may become glass-like with non-crystalline structure or consist of defective crystals with meta-stable amorphous structures.

The *entropy* at the equilibrium is defined [1]:

$$H(T) = -\sum_{i \in S} \pi_i \cdot \ln \pi_i, \tag{3.5}$$

From Eqs. (3.3) and (3.4), we obtain:

$$\lim_{T \to \infty} H(T) = -\sum_{i \in \mathcal{S}} \frac{1}{|\mathcal{S}|} \ln \frac{1}{|\mathcal{S}|} = \ln |\mathcal{S}|, \tag{3.6}$$

$$\lim_{T \to 0} H(T) = -\sum_{i \in \mathcal{S}_{min}} \frac{1}{|\mathcal{S}_{min}|} \ln \frac{1}{|\mathcal{S}_{min}|} = \ln |\mathcal{S}_{min}|, \tag{3.7}$$

and also,

$$\begin{aligned} \frac{\partial H(T)}{\partial T} &= -\sum_{i \in \mathcal{S}} [\ln \pi_i + 1] \cdot \frac{\partial \pi_i}{\partial T} \\ &= -\sum_{i \in \mathcal{S}} \left[-\frac{E(i)}{k_b T} - \ln Z + 1 \right] \cdot \frac{\partial \pi_i}{\partial T}, \end{aligned} \tag{3.8}$$

If we define the average energy

$$\overline{E}_T = \sum_{i \in \mathcal{S}} \pi_i \cdot E(i). \tag{3.9}$$

and the variance

$$\sigma_T^2 = \sum_{i \in S} \left(E(i) - \overline{E}_T \right)^2 = \overline{E^2}_T - \overline{E}_T^2. \tag{3.10}$$

then, we have

$$\frac{\partial \pi_i}{\partial T} = \frac{\pi_i}{k_b T^2} \left[E(i) - \overline{E}_T \right]. \tag{3.11}$$

$$\begin{aligned} \frac{\partial H(T)}{\partial T} &= \sum_{i \in S} \left[\frac{E(i)}{k_B T} + \ln Z - 1 \right] \cdot \frac{\pi_i}{k_b T^2} \left[E(i) - \overline{E}_T \right] \\ &= \frac{1}{k_b^2 T^3} \sum_{i \in S} E(i) \pi_i [E(i) - \overline{E}_T] + \frac{(\ln Z - 1)}{k_b T^2} \sum_{i \in S} \pi_i \left[E(i) - \overline{E}_T \right] \\ &= \frac{1}{k_b^2 T^3} \left[\sum_{i \in S} E^2(i) \cdot \pi_i - \overline{E}_T \sum_{i \in \mathcal{S}} E(i) \cdot \pi_i \right] = \frac{\sigma_T^2}{k_b^2 T^3}. \end{aligned} \tag{3.12}$$

Since $T > 0$ and $\sigma_T^2 \geq 0$, we have $\frac{\partial H(T)}{\partial T} \geq 0$. Combining the Eqs. (3.3) and (3.4), we can see that the entropy is monotonically decreased as the temperature is lowered (See Fig. 3.1). In statistical mechanics, entropy is used to measure the order

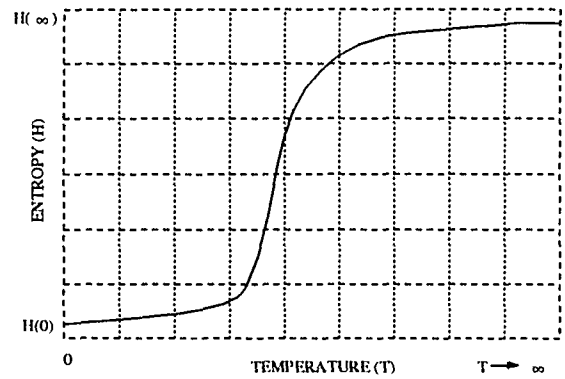


Figure 3.1 The entropy $H(T)$ changes with the temperature T

of a physical system: the larger the entropy, the more chaotic the system. In the annealing process, the entropy monotonically decreases, so it will lead the system to the ordered (crystalline) if the temperature is lowered slowly enough so that the system is allowed to relax to the equilibrium at each temperature.

3.3 Simulated Annealing

Based on the annealing process in statistical mechanics, Kirkpatrick et al. [25] proposed an algorithm, namely *simulated annealing* (SA) for solving complicated combinatorial optimization problems. In the SA algorithm, a simulation of the annealing process is performed. The cost function and configuration in optimization correspond to the energy function and state of statistical physics, respectively. The temperature is introduced as a control parameter.

Suppose that a cost function $f : s \rightarrow \mathbf{R}^+$, $s \in \mathcal{S}$, to be minimized is defined on some finite set $\mathcal{S}$. For each configuration $s \in \mathcal{S}$, there is a neighboring set $\mathcal{N}(s) \subseteq \mathcal{S}$, which is generated by a small perturbation of s.

In SA, given the current state $s(k)$, a neighboring state $s'(k)$ is randomly selected from $\mathcal{N}(s)$, where k is the k-th trial. The transition probability from state

$s(k)$ to $s'(k)$ is given by the *Metropolis criterion* [29]:

$$P[s(k), s'(k)] = Pr\{s(k) \rightarrow s'(k)\} = exp\left[-\frac{[f(s'(k)) - f(s(k))]^+}{T}\right], \tag{3.13}$$

where

$$[x]^+ = \max\{0, x\}. \tag{3.14}$$

Eq. (3.13) can be written in another form:

$$Pr\{s(k+1) = s'(k)\} = \begin{cases} 1 & \text{if } f(s'(k)) < f(s(k)) \\ exp\left(-\frac{f(s'(k))-f(s(k))}{T}\right) & \text{otherwise.} \end{cases} \tag{3.15}$$

From Eq. (3.15), it can be seen that the Metropolis criterion while performing the local search for the minimum cost at a fixed temperature T allows occasional transition from a lower cost configuration to a higher cost configuration with certain probability, thus preventing the system from getting stuck in local minima. The random process $\chi = (s(k) : k \geq 0)$ produced in SA can be characterized by a discrete time homogeneous Markov chain [1]. The one-step transition matrix is

$$\begin{aligned} P(x,y) &= Pr[s(k+1) = y | s(k) = x] \\ &= \begin{cases} 0 & \text{if } y \notin \mathcal{N}(x) \text{ and } y \neq x, \\ G(x,y)\min\left\{1, exp(-\frac{[f(y)-f(x)]}{T})\right\} & \text{if } y \in \mathcal{N}(x) \text{ and } y \neq x, \\ 1 - \sum_{x' \neq x} G(x,x')\min\left\{1, exp(-\frac{[f(x')-f(x)]}{T})\right\} & \text{if } y = x. \end{cases} \end{aligned}$$

where $G(x,y)$ is the probability of generating configuration y from x.

If the generation probability of any configuration x is uniformly distributed in its neighboring configuration set $\mathcal{N}(x)$ and the configuration transition is based on Eq. (3.16), the corresponding Markov chains are irreducible, aperiodic, and recurrent [1]. Under these conditions, the stationary equilibrium distribution π_i for configuration i exists after infinite number of transitions,

$$\begin{aligned} \pi_i(T) &= \lim_{k\to\infty} Pr\{s(k) = i | T\} \\ &= \lim_{k\to\infty} Pr\{s(k) = i | s(0) = s_0, T\} \\ &= \frac{exp(-\frac{f(i)}{T})}{\sum_{j\in\mathcal{S}} exp\left(-\frac{f(j)}{T}\right)}. \end{aligned} \tag{3.16}$$

From Eq. (3.4), we know that

$$\pi_i^* = \lim_{T \to 0} \pi_i(T) = \begin{cases} \frac{1}{|S_{min}|} & \text{if } i \in S_{min}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.17}$$

Therefore,

$$\lim_{T \to 0} \left[\lim_{k \to \infty} P(s(k) \in S_{min}) \right] = \lim_{T \to 0} \sum_{i \in S} \pi_i(T) = \sum_{i \in S_{min}} \pi_i^* = 1. \tag{3.18}$$

Eq. (3.18) states that the SA algorithm asymptotically converges to the configurations with the minimum cost, i.e., if the temperature is slowly lowered and at each temperature the system performs a sufficient number of transitions, the configurations (solutions) with the global minimum cost will be found with probability one.

3.4 Mean Field Annealing Theory

Even though SA is proved to be able to reach the global optima asymptotically, it is often time-consuming to reach thermal equilibrium at each temperature. Finite numbers of transitions at each temperature cannot guarantee convergence to the global optima. In statistical physics, *mean field* approximation is often used. Mean field annealing (MFA) uses a set of deterministic equations to replace the stochastic process in SA. It uses saddle point approximation in the calculation of the stationary probability distribution at equilibrium, and reaches equilibrium at each temperature much faster than SA. Even though this approximation method may not be guaranteed to converge to global minima, it does provide a good approximation in finding near-optimal solutions with much less computing effort.

As shown in the previous section, the stationary probability distribution at equilibrium for configuration $\mathbf{s}'$ is given by

$$\pi_{\mathbf{s}'}(T) = \frac{exp\left(-\frac{f(\mathbf{s}')}{T}\right)}{\mathcal{Z}},$$

$$\mathcal{Z} = \sum_{\mathbf{s}} exp\left(-\frac{f(\mathbf{s})}{T}\right).$$

where $\mathbf{s}, \mathbf{s}' \in \mathbf{Z}^n$ are configurations and $\mathbf{Z}$ is the integer set. For a large optimization problem, exact calculation of the partition function $\mathcal{Z}$ is prohibitive. The saddle point approximation [35] is used. Note that the Dirac delta function, $\delta(\cdot)$, can be expressed as:

$$\delta(x) = \frac{1}{2\pi i}\int_{\mathbf{I}} e^{xy} dy \tag{3.19}$$

where the integral is taken along the imaginary axis. Hence,

$$\begin{aligned}
\mathcal{Z} &= \sum_{\mathbf{s}} exp\left(-\frac{f(\mathbf{s})}{T}\right) \\
&= C\sum_{\mathbf{s}} \int_{\mathbf{R}} d\mathbf{v} \int_{\mathbf{I}} e^{-\frac{f(\mathbf{v})}{T}} \cdot e^{\mathbf{u}(\mathbf{s}-\mathbf{v})} d\mathbf{u} \\
&= C\int_{\mathbf{R}} d\mathbf{v} \int_{\mathbf{I}} e^{-\frac{f(\mathbf{v})}{T}-\mathbf{u}\mathbf{v}} \cdot \sum_{\mathbf{s}} e^{\mathbf{u}\cdot\mathbf{s}} d\mathbf{u} \\
&= C\int_{\mathbf{R}} d\mathbf{v} \int_{\mathbf{I}} e^{-\frac{f(\mathbf{v})}{T}-\mathbf{u}\mathbf{v}+\ln\sum_{\mathbf{s}} e^{\mathbf{u}\cdot\mathbf{s}}} d\mathbf{u} \\
&= C\int_{\mathbf{R}} d\mathbf{v} \int_{\mathbf{I}} e^{-f_e(\mathbf{u},\mathbf{v})} d\mathbf{u},
\end{aligned} \tag{3.20}$$

where

$$f_e(\mathbf{u},\mathbf{v}) = \frac{f(\mathbf{v})}{T} + \mathbf{u}\mathbf{v} - \ln\sum_{\mathbf{s}} e^{\mathbf{u}\cdot\mathbf{s}}, \tag{3.21}$$

C is a complex constant, and f_e is called the *effective energy* in statistical mechanics. At saddle points,

$$\frac{\partial f_e}{\partial \mathbf{u}} = \mathbf{v} - \frac{\sum_{\mathbf{s}} \mathbf{s}\cdot e^{\mathbf{u}\cdot\mathbf{s}}}{\sum_{\mathbf{s}} e^{\mathbf{u}\cdot\mathbf{s}}} = 0,$$

and

$$\frac{\partial f_e}{\partial \mathbf{v}} = \frac{1}{T}\frac{\partial f(\mathbf{v})}{\partial \mathbf{v}} + \mathbf{u} = 0. \tag{3.22}$$

Therefore

$$\begin{aligned}
\mathbf{v} &= \bar{\mathbf{s}}_T = \frac{\sum_{\mathbf{s}} \mathbf{s}\cdot e^{\mathbf{u}\cdot\mathbf{s}}}{\sum_{\mathbf{s}} e^{\mathbf{u}\cdot\mathbf{s}}}, \\
\mathbf{u} &= -\frac{1}{T}\frac{\partial f(\mathbf{v})}{\partial \mathbf{v}},
\end{aligned} \tag{3.23}$$

where $\bar{\mathbf{s}}_T$ is the thermal average of $\mathbf{s}$ at temperature T.

In statistical physics, $\mathbf{h} = -\frac{\partial f(\mathbf{v})}{\partial \mathbf{v}}$ is called the *mean field*. If a configuration $\mathbf{s} = [s_1, s_2, \cdots, s_n]^T$ is represented by a sequence of binary values, i.e., $\mathbf{s} \in \{0,1\}^n$, then we have $\mathbf{v} = [v_1, v_2, \cdots, v_n]^T$ and

$$v_i = \frac{\sum_{s_i=0}^{1} s_i \cdot e^{u_i s_i}}{\sum_{s_i=0}^{1} e^{u_i s_i}} = \frac{e^{u_i}}{1+e^{u_i}} = \frac{1}{2}\left[1 + \tanh\left(\frac{u_i}{2}\right)\right], \tag{3.24}$$

where $\mathbf{u} = [u_1, u_2, \cdots, u_n]^T$ and $u_i = -\frac{1}{T} \cdot \frac{\partial f(\mathbf{v})}{\partial v_i}$.

For the binary system, we have the following MFA equations:

$$v_i = \frac{1}{2}\left[1 + \tanh\left(\frac{h_i}{2T}\right)\right], \tag{3.25}$$

$$h_i = -\frac{\partial f(\mathbf{v})}{\partial v_i}. \tag{3.26}$$

In 1982, Hopfield [20] defined the following energy function of the Hopfield net for optimization,

$$f_h(\mathbf{s}) = -\frac{1}{2}\sum_i \sum_j T_{ij} s_i s_j - \sum_i s_i I_i, \tag{3.27}$$

where $s_i \in \{0,1\}$. In the Hopfield model, the system is represented by a network composed of n neurons. Each neuron i can be represented by an operational amplifier, s_i is the output of neuron i, and T_{ij}, which is symmetric ($T_{ij} = T_{ji}$ and $T_{ii} = 0$), represents the synaptic connection between neuron i and j. I_i is the input current to amplifier i. The stable states of the network correspond to the 2^n corners of the hypercube $\{0,1\}^n$, the local minima of the energy function defined in Eq. (3.27). For the MFA approximation, if the energy is formulated as in Eq. (3.27), the mean field h_i and the thermal average v_i become

$$h_i = -\frac{\partial f_h(\mathbf{v})}{\partial v_i} = \sum_j T_{ij} v_j + I_i, \tag{3.28}$$

$$v_i = \bar{s}_i = \frac{1}{2}\left[1 + \tanh\left(\frac{h_i}{2T}\right)\right]. \tag{3.29}$$

In MFA, the iterative procedure to reach thermal equilibrium at each temperature is called *relaxation*, in which the mean field is updated by

$$h_i(t+\Delta t) = h_i(t) + \Delta t \left[-\frac{\partial f_h(\mathbf{v})}{\partial v_i} - h_i(t) \right],$$

Taking the limit, we have

$$\frac{dh_i}{dt} = \lim_{\Delta t \to 0} \frac{h_i(t+\Delta t) - h_i(t)}{\Delta t}, \tag{3.30}$$

or

$$\frac{dh_i}{dt} = -\frac{\partial f_h(\mathbf{v})}{\partial v_i} - h_i(t) = \sum_j T_{ij} v_j + I_i - h_i. \tag{3.31}$$

The MFA relaxation operation at each temperature should lead the system to stable equilibrium. The stability and convergence of MFA will be analyzed in the next section. The MFA procedure can be summarized in the flow chart shown in Fig. 3.2.

3.5 Convergence of MFA

Before we prove the convergence of MFA, we need to review stability theory [30]. Consider a differential equation:

$$\frac{dx}{dt} = f(x). \tag{3.32}$$

where $x \in \mathbf{R}^n, f : x \to \mathbf{R}^{+n}$

Definition 3.5.1 *(Equilibrium Point): x^* is called an equilibrium point of Eq. (3.32) if $f(x^*) = 0$. That is, at the equilibrium point, the system will no longer change with time.*

Since such an equilibrium point may be stable or unstable, we need to define the stability.

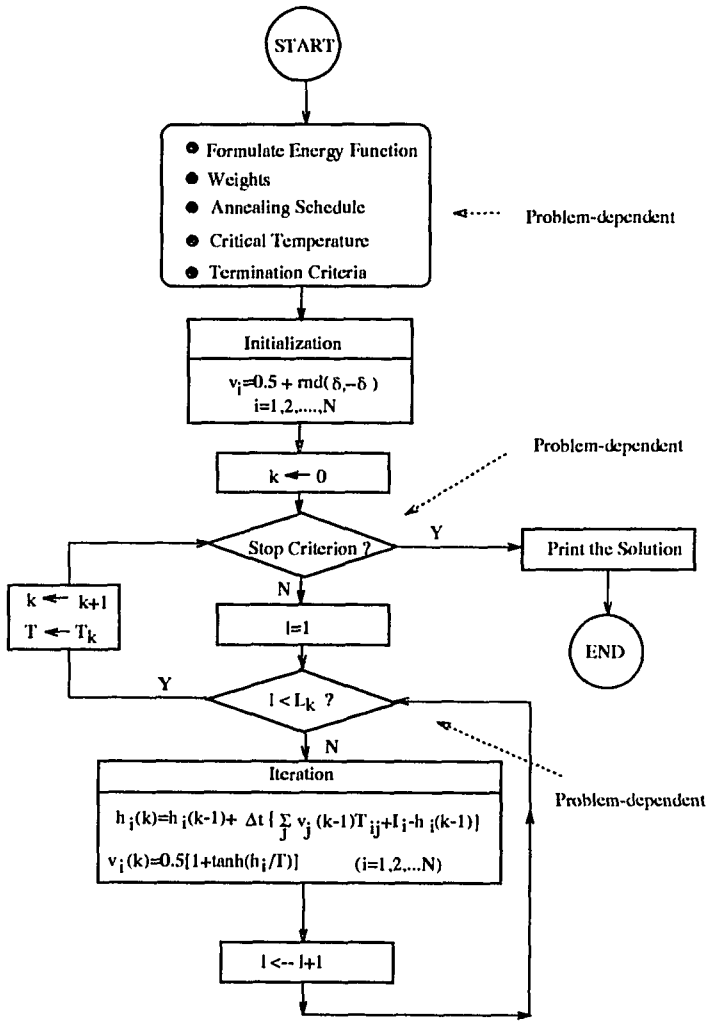


Figure 3.2 The MFA iteration procedure

Definition 3.5.2 *(Stability): Let $x^*(t)$ be a solution $\dot{x} = f(x)$. An equilibrium point x^* is stable if, for every neighborhood set $\mathcal{N}$ of x^*, there is a neighborhood $\mathcal{N}_1$ of x^* such that every solution $x(t)$ with initial point $x(t_0)$ in $\mathcal{N}_1$ is defined in both $\mathcal{N}_1$ and $\mathcal{N}$ for all $t > t_0$. That is, for any $x_0 = x(t_0)$ and any given value $\epsilon > 0$, there exists an arbitrarily small value $\delta > 0$ so that if $||x(t_0) - x^*|| < \delta, then ||x(t) - x^*|| < \epsilon$, where $||x||$ is the Euclidean norm, i.e.,*

$$||x_i - x_j|| = \left[\sum_k (x_{ik} - x_{jk})^2\right]^{\frac{1}{2}}. \tag{3.33}$$

Definition 3.5.3 *(Asymptotically Stability): If x^* is stable and $\lim_{t\to\infty} x(t) = x^*$, then x^* is asymptotically stable.*

Theorem 3.1 *(Liapunov's Stability Theorem)[30]: Let x^* be an equilibrium point for Eq. (3.32). Let $E : \mathcal{N} \to \mathbf{R}$ be a continuous and differentiable function defined on a neighborhood $\mathcal{N}$ of x^* such that if*

(a) $E(x^*) = E_{min}$ *and* $E(x) > E(x^*)$ *if* $x \neq x^*$,

(b) $\dot{E} \leq 0 \quad \forall x \in \mathcal{N}$, *then x^* is stable. Furthermore, if also*

(c) $\dot{E} < 0 \ \forall x \in \mathcal{N}$,

then x^* is *asymptotically stable.* where

$$\dot{E}(x) = \frac{dE}{dt} = \sum_j \frac{\partial E}{\partial x_j} \cdot \frac{dx_j}{dt} = \nabla E^T \cdot \dot{x}, \tag{3.34}$$

$$\nabla E(x) = \left[\frac{\partial E}{\partial x_1}, \frac{\partial E}{\partial x_2}, \cdots, \frac{\partial E}{\partial x_n}\right]^T. \tag{3.35}$$

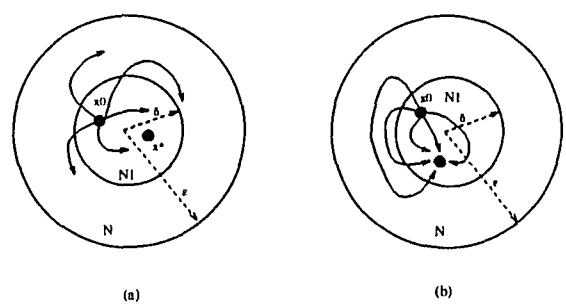


Figure 3.3 The illustration of stability. (a) stability. (b) asymptotically stable

A function $E(x)$ satisfying (a) and (b) is called a Liapunov function for x^*. If (c) also holds, we call $E(x)$ a strict Liapunov function.

For MFA, if we construct

$$E(\mathbf{v}) = f_h + \sum_{i=1}^{n} \int_0^{v_i} h_i(y)dy, \tag{3.36}$$

then we have

$$\frac{\partial E(\mathbf{v})}{\partial v_i} = \frac{\partial f_h(\mathbf{v})}{\partial v_i} + h_i = -\sum_j T_{ij} v_j - I_i + h_i, \tag{3.37}$$

or

$$\nabla E(\mathbf{v}) = -\mathbf{T}\mathbf{v} - \mathbf{I} + \mathbf{h}, \tag{3.38}$$

Where $\mathbf{T} = \{T_{ij} : \forall i, j\}$ and $\mathbf{v}, \mathbf{I}, \mathbf{h} \in \mathbf{R}^n$

From Eq. (3.31), we can rewrite

$$\dot{\mathbf{h}} = \mathbf{T} \cdot \mathbf{v} + \mathbf{I} - \mathbf{h}. \tag{3.39}$$

If we construct

$$\Psi(\mathbf{v}) = E(\mathbf{v}) - E(\mathbf{v}^*), \tag{3.40}$$

and assume that $\mathbf{v}^*$ is an equilibrium point and a local minimum. Then we have:

(a) $\Psi(\mathbf{v}^*) = E(\mathbf{v}^*) - E(\mathbf{v}^*) = 0$, and

$\Psi(\mathbf{v}) = E(\mathbf{v}) - E(\mathbf{v}^*) > 0, \qquad \forall \mathbf{v} \in \mathcal{N}(\mathbf{v}^*)$

(b) From Eq.(3.25), we have

$$v_i = \frac{1}{2}\left[1 + \tanh\left(\frac{h_i}{2T}\right)\right],$$

Therefore

$$\frac{\partial v_i}{\partial h_i} = \left(\frac{1}{1 + e^{-h_i/T}}\right)' = \frac{e^{h_i/T}}{T(1 + e^{-h_i/T})^2} > 0.$$

$$\begin{aligned}
\dot{\Psi}(\mathbf{v}) &= \sum_i \frac{\partial \Psi}{\partial v_i} \cdot \frac{dv_i}{dt} \\
&= \sum_i \frac{\partial \Psi}{\partial v_i} \cdot \frac{\partial v_i}{\partial h_i} \frac{dv_i}{dt} \\
&= \sum_i \left(\frac{\partial f_h}{\partial v_i} + h_i\right) \cdot \frac{\partial v_i}{\partial h_i} \left(-\frac{\partial f_h}{\partial v_i} - h_i\right)
\end{aligned}$$

$$
\begin{aligned}
&= -\sum_i \frac{\partial v_i}{\partial h_i}\left(\frac{\partial f_h}{\partial v_i} + h_i\right)^2 \\
&= -\frac{e^{h_i/T}}{T(1+e^{-h_i/T})^2}\left(\frac{\partial f_h}{\partial v_i} + h_i\right)^2 \\
&= -\nabla E^T \cdot \mathbf{W} \cdot \nabla E < 0. \qquad (3.41)
\end{aligned}
$$

where $\mathbf{W} = diag\left(\frac{\partial v_1}{\partial h_1}, \frac{\partial v_2}{\partial h_2}, \cdots, \frac{\partial v_n}{\partial h_n}\right)$. It is shown that if an equilibrium point $\mathbf{v}^*$ is a local minimum, it will be asymptotically stable. $\Psi(\mathbf{v})$, therefore, $E(\mathbf{v})$, is a strict Liaponov function. i.e., at each temperature, the evolution of Eq. (3.31) will lead the system to converging to a local minimum.

3.6 Further Discussion of MFA

As shown in Eq. (3.27), for a binary neural network $\mathbf{s} = \{(s_i) : s_i \in \{0,1\}\forall i\}$, the energy function is described by

$$f_h(\mathbf{s}) = -\frac{1}{2}\sum_i\sum_j T_{ij}s_is_j - \sum_i s_iI_i.$$

By taking the average of the energy, we have

$$\overline{f_h(\mathbf{s})} = -\frac{1}{2}\sum_i\sum_j T_{ij}\overline{s_is_j} - \sum_i \overline{s_i}I_i.$$

In order to find the mean field based on Eq. (3.26), we have to calculate $\overline{s_is_j}$. Since there exist interactions between neurons, any neurons i and j need not be independent. Therefore the complexity to calculate their expected values will be $O(2^N)$ (taking into account all configurations), where N is the number of neurons in a neural network. If we ignore the interactions, the average energy can be approximated by:

$$\overline{f_h(\mathbf{s})} \approx -\frac{1}{2}\sum_i\sum_j T_{ij}\overline{s_i}\cdot\overline{s_j} - \sum_i \overline{s_i}I_i. \qquad (3.42)$$

Then, the mean field can be obtained by

$$h_i = -\frac{\partial\overline{f_h(\mathbf{s})}}{\partial\overline{s_i}} = \sum_j T_{ij}\overline{s_j} + I_i \qquad (3.43)$$

$$\overline{s_i} = \sum_{s_i=0}^{1} s_i Pr\{s_i\} = \sum_{s_i} s_i \frac{\exp(h_i \cdot s_i/T)}{1+\exp(h_i \cdot s_i/T)} = \frac{1}{2}\left[1+\tanh\left(\frac{h_i}{2T}\right)\right]. \qquad (3.44)$$

Eqs. (3.43) and (3.44) are exactly the same as the MFA equations (3.25) and (3.26). From the approximation in Eq. (3.42), it can be seen that the complexity of computing the mean field h_i is reduced from $O(2^N)$ to $O(N)$ (see Eq. (3.43)). This approximation is good, especially for a large system (optimization problem). In SA, random perturbations based on the *Metropolis* criterion move the system towards its thermal equilibrium at each temperature, i.e., $\overline{f_h} \rightarrow h_{eq}(T)$. Instead of the stochastic process in SA, MFA adopts a set of deterministic equations as shown in Eqs. (3.43), (3.44) by considering the thermal averages of neurons $\bar{s}$ at equilibrium. Even though this approximation may no longer guarantee the system convergence to the global minima, experimental results [8], [9], [31]-[33] have shown that this approximation is effective in finding near-optimum solutions whereas the computational complexity is greatly reduced as compared to SA.

3.6.1 The Related Parameters in MFA

Before performing the MFA iterations, several parameters need to be determined in order to obtain good solutions:

a. Weights: A combinatorial optimization problem can be described by an energy function in the form of

$$E = w_1 \cdot \text{“}cost\text{”} + w_2 \cdot \text{“}constraints\text{”}$$

Assuming $w_1, w_2 \in \mathbf{R}^+$, we consider the following extreme situations:

1. $w_1 \rightarrow +\infty$, the first term (cost) will dominate and the MFA iterations will lead the system to invalid solutions.
2. $w_1 = 0$, the MFA iterations may stop at a valid solution, however, the solution could be very far from the global mimimum.

Therefore, the proper determination of the relationship between weights will greatly affect the solution quality.

b. **Critical Temperature:** Critical temperature is defined as one at which on the average each neuron begins to move predominantly towards 1 or 0. Obviously, at high temperature, the iterations based on Eqs. (3.43), (3.44) are trivial and all averages of neurons are disorderly distributed around 0.5. Therefore, starting iterations at too high a temperature simply introduces computational cost without any progress towards a solution. On the other hand, if the initial temperature is too low, the evolution of neuron averages is just like the *quenching* process in statistical mechanics and will result in a poor solution. Thus, the determination of a critical temperature at which the iterations start becomes crucial to the efficiency of the MFA approach.

c. **Final temperature or stopping criterion:** The system must decide when to stop the iterations, which is equivalent to deciding the final temperature at which the iterations terminate.

d. **Annealing Schedule:** Annealing schedule is the way the temperature is lowered. Fast decrement of temperature might lead the system to poor solutions, whereas slow decrement will bring about computational inefficiency.

The determination of these parameters is correlated with each other and the choice of parameters is still an open question. We will investigate these issues in the following individual sections.

CHAPTER 4

SEARCHING FOR THE OPTIMAL FRAME PATTERNS IN AN INTEGRATED TDMA COMMUNICATION NETWORK

The integration of data and voice in an integrated services data network (ISDN) has received extensive attention in recent years in order to efficiently share the system resources such as transmission, switching and control facilities. Many research works have been directed to the time-division multiple access (TDMA) strategy [3][11][14][18][24][42]. In an integrated TDMA system, transmission channels are shared by the circuit-switched type of traffic such as voice and the packet-switched data. To avoid conflicts in accessing channels, contending users must be assigned to use the channels at different times. The time axis is divided into frames and each frame consists of a certain number of fixed-length time slots. A certain portion of the time slots in a frame are assigned to voice transmission and the remaining portion is reserved for data. Many studies in the literature model the voice traffic as a lossy system and data as a queuing system. Therefore voice traffic will be blocked with no transmission if it cannot find an available time slot at the instant of its arrival. For data traffic, on the other hand, arrivals can be buffered and transmitted whenever a time slot is available. Hence the objective of the system design is to minimize the blocking probability of voice traffic as well as the time delay of data traffic. Two approaches, namely fixed-length boundary (FB) and movable boundary (MB) are generally used. In the FB scheme, a TDMA frame is partitioned into two regions, each consisting of a certain number of time slots, one for voice traffic and the other for data traffic. The idle time slots assigned to voice cannot be used to transmit data. Obviously, this scheme does not fully utilize system facilities. In contrast, the

MB scheme utilizes any residual voice time slots to transmit data. As a result the queuing delay is expected to decrease. Fig. 4.1 shows the two schemes.

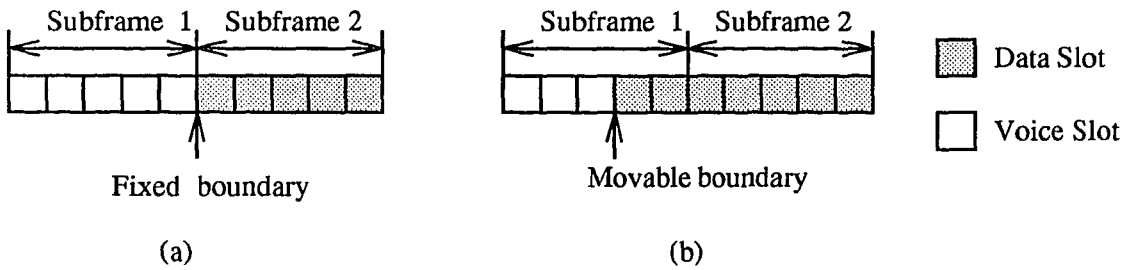


Figure 4.1: Integrated voice and data TDMA frame format: (a) FB scheme. (b) MB scheme.

4.1 The Multiaccess Protocol

In this chapter, we are concerned with the single-hop *multiaccess/broadcast* packet transmission network. In this type of network, a single wideband channel is shared by all users. A slotted ALOHA random access protocol is employed for transmission, i.e., whenever a user is ready to transmit a packet, it simply goes at will. The only constraint is that users must start to transmit at the beginning of each time slot. When two or more users attempt to transmit at the same time slot, collision occurs and all of the collided packets are required to be retransmitted. Although the maximum achievable channel utilization is low, the slotted ALOHA scheme is superior to fixed assignment schemes when there is a large population of bursty users [37]. The ALOHA multiaccess protocol provides a simple control scheme for the channel access. Typical examples of this type of network are the ALOHA network [2], SATNET [22], and ETHERNET [28]. Fig. 4.2 shows a packet satellite network in which a slotted ALOHA protocol can be used. A single satellite in geosynchronous orbit is used to provide connectivity among a number of earth stations. A global-coverage antenna is used on the satellite, allowing each station to communicate with

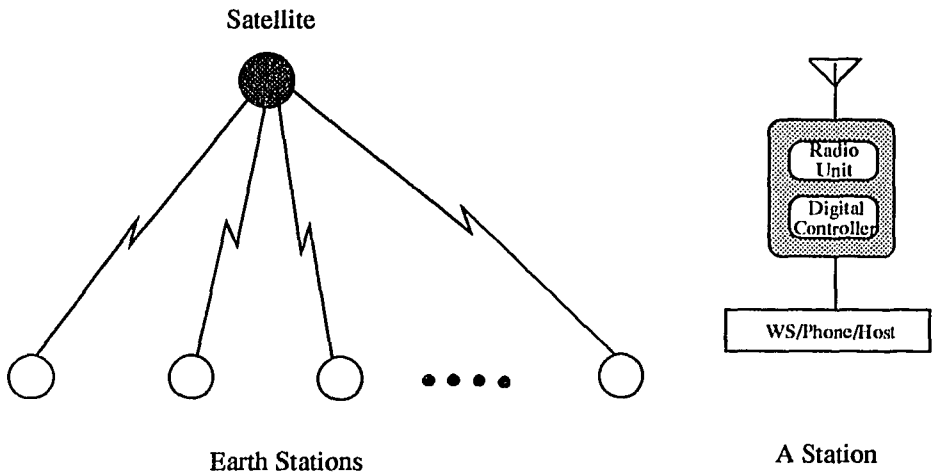


Figure 4.2 A packet satellite network

any others in time. A single transponder is used on the satellite to repeat received uplink packets on the downlink. All earth stations share a single wideband channel in the uplink transmissions, which is from an earth station to the satellite.

4.2 Frame Pattern and Data Throughput

For the ALOHA-type networks, we assume that each user has two types of traffic - data and voice. Since voice traffic is very sensitive to time delay, once a user with voice traffic wins access to the channel at a certain time slot, the same time slot in the successive frames will be reserved for this user until the voice communication is completed. If a voice user finds no available time slots, its call will be rejected. A time frame consists of N time slots, some for voice and others for data. The MB scheme is adopted in our approach. Therefore, data traffic can be transmitted in both its nominal slots and the silent voice slots at a time. For each data slot, all data-type users access the channel based on contention. The collided data packets must be retransmitted after a random number of time slots. Instead of the distinct boundary as shown in Fig. 4.1 (b), the positions of data slots are arranged in different

places within a frame. The frame format is shown in Fig. 4.3. Each frame consists of N time slots, including N_d slots for data transmission. We denote:

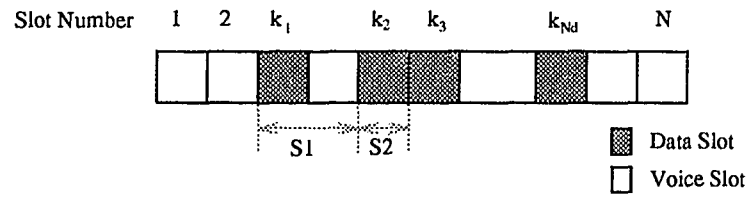


Figure 4.3 A frame format.

N_d : The number of time slots that can be used for data packets at a given frame, which includes the nominal data slots and silent voice slots. The silent voice slots are the ones which are unused at certain time.

k_i: The slot number of the ith available data slot, where $1 \leq k_i \leq N$, $1 \leq i \leq N_d$.

s_i: The interdistance between the ith data slot and its first successor $(i+1)$th data slot, where

$$s_i = \begin{cases} k_{i+1} - k_i, & \text{if } i = 1, 2, \cdots, N_d - 1, \\ k_1 + N - k_{N_d} & \text{if } i = N_d. \end{cases}$$

$\mathbf{s} = (s_1, s_2, \cdots, s_{N_d})$ is called a *frame pattern*, which represents the relative positions of data slots. Obviously,

$$s_1 + s_2 + \cdots + s_{N_d} = N, \tag{4.1}$$

and

$$s_i \geq 1 \qquad \forall i. \tag{4.2}$$

Before we get an insight into the frame pattern, we make the following assumptions:

1. The holding time of the voice call is much longer than the frame time so that the queuing behavior of data for a given frame pattern can reach steady state.

2. The slotted ALOHA random access protocol is used for data transmission. The total data traffic, new and retransmitted, constitutes a *Poisson* process with a mean arrival rate of G packets/slot.

3. Packets have a fixed length, and the length of a time slot equals the time required to transmit a packet.

Based on the above assumptions, the probability that there are l data packets in the interval $[k, k + s_i]$ is given by

$$P(l) = \frac{(G \cdot s_i)^l \cdot e^{-G \cdot s_i}}{l!} \tag{4.3}$$

and the probability of no collision in the interval s_i is the probability that no Poisson data traffic is generated during the time interval s_i (see Fig. 4.4), i.e.,

$$\Pr(\text{no collision in the time interval } s_i) = e^{-G \cdot s_i}$$

and

$$\Pr(\text{packets successfully transmitted in the time interval } s_i) = G \cdot s_i \cdot e^{-G \cdot s_i}.$$

Therefore the average data throughput is given by

$$\gamma_s = \frac{1}{N_d} \sum_{i=1}^{N_d} G \cdot s_i e^{-G \cdot s_i}. \tag{4.4}$$

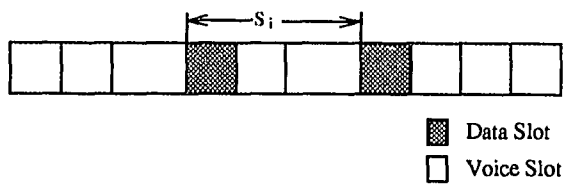


Figure 4.4 Illustration of no-collision interval.

From Eq. (4.4), it can be seen that the relative positions of voice and data will generate different data throughput. Fig. 4.5 is an example of the throughput

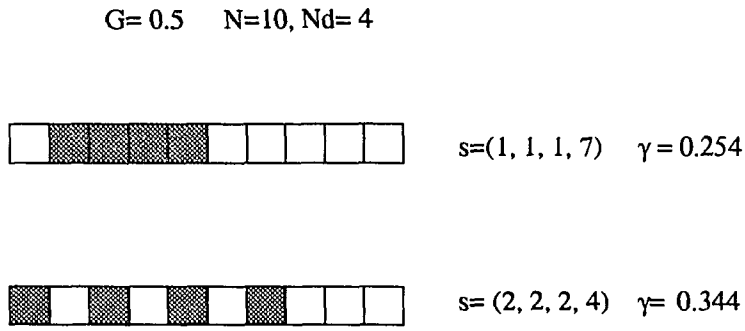


Figure 4.5 The throughput generated by the two frame patterns.

generated by two frame patterns. For a given N and N_d, there are $C_{N_d}^{N}$ frame patterns. For example, assuming $N = 40$ and $N_d = 10$, the total number of feasible frame patterns is $C_{10}^{40} = 8.4 \times 10^8$. An optimal frame pattern is the one which provides the maximum data throughput. The computational complexity using exhaustive search for finding the optimal frame pattern among the set of all frame patterns becomes intractable as the problem size increases.

4.3 Random Search for the Optimal Frame Patterns

The general way to obtain a good frame pattern is to pick up a bunch of feasible frame patterns that satisfy the constraint Eq. (4.1), then calculate the throughput for every single frame pattern. The one having the highest throughput among all the selected frame patterns will be used as the optimal frame pattern. Obviously, the random search is a local search scheme. Its performance strongly depends on the initial selection of a frame pattern, the way used to pick up another frame pattern from the current one and the number of the selected frame patterns. In evaluating the performance of the random search, we take the average throughput of the randomly selected frame patterns.

4.4 Searching for the Optimal Frame Patterns by Using Mean Field Annealing

As shown in Section 4.2, maximizing data throughput is equivalent to finding a specific frame pattern $\mathbf{s}^{opt} = \{s_1^{opt}, s_2^{opt}, \cdots, s_{N_d}^{opt}\}$ such that

$$\gamma_{\mathbf{s}^{opt}} = \max_{\mathbf{s} \in \mathrm{S}} \gamma_{\mathbf{s}} = \max_{\mathbf{s} \in \mathrm{S}} \frac{1}{N_d} \sum_{i=1}^{N_d} G \cdot s_i \cdot e^{-G \cdot s_i}, \tag{4.5}$$

subject to

$$\sum_{i=1}^{N_d} s_i = N, \text{ where } 1 \leq s_i \leq N - N_d + 1. \tag{4.6}$$

and

$$s_i \geq 1 \qquad \forall i \tag{4.7}$$

The MFA procedure for solving this constrained optimization problem is summarized as follows:

1. Form an energy function which reflects both data throughput to be maximized and the constraints.

2. Select the weights that keep the balance between maximization of data throughput and satisfaction of the constraints.

3. Determine the annealing schedule.

4. Determine the critical temperature T_c to achieve fast convergence.

5. Define the termination criterion.

6. Perform the iterative procedure to find optimal solutions:

   (a) Initialize the average of neuron ij with $v_{ij} = 0.5 + 0.001 \times rand[-1, 1] \ \forall i, j$, and start annealing from the critical temperature. Neuron ij is defined in the sequel.

(b) At each temperature T, update v_{ij} $\forall i,j$ according to Eq. (3.25) and Eq. (3.26) through the synchronous or asynchronous method until a certain criterion is satisfied. One complete updating of all neurons at a fixed temperature is referred to as a *sweep*.

(c) Decrease the temperature T according to the annealing schedule, repeat Step 6(b) until the convergence criterion is satisfied. Each temperature corresponds to an iteration which consists of a bunch of sweeps.

To map this optimization problem onto the MFA framework, we need to determine the parameters defined in the Step 1-Step 5 above.

4.4.1 Energy Function

$$\begin{aligned} E(\mathbf{s}) &= -\frac{w_1}{N_d}\sum_{i=1}^{N_d} G\cdot s_i\cdot e^{-G\cdot s_i} + \frac{w_2}{2}\left(\sum_{i=1}^{N_d} s_i - N\right)^2 + w_3\sum_{i=1}^{N_d}\sum_{j=1}^{m} s_{ij}(1-s_{ij}) \\ &= -w_1\cdot E_1 + w_2\cdot E_2 + w_3\cdot E_3 \end{aligned} \tag{4.8}$$

and

$$s_i = \sum_{j=1}^{m} s_{ij}\cdot 2^j \tag{4.9}$$

where $m = \lceil \log_2(N - N_d + 1)\rceil + 1$ and $\lceil x\rceil$ is the ceiling of x. For example, for $N = 40$, if $N_d = 9$, then $m = 5$, and if $N_d = 5$, then $m = 6$. $s_{ij} \in \{0,1\}\forall i,j$ are binary neurons. $w_l > 0$, $l = 1,2,3$, are called the *weights*.

The interdistance between the ith data slot and its first successive data slot is denoted by s_i. Since s_i is an integer and $1 \le s_i \le N - N_d + 1$, s_i can be expressed in terms of m binary neurons defined by Eq. (4.9). The first term in Eq. (4.8) is the negative weighed data throughput, and therefore maximizing data throughput is equivalent to minimizing the negative throughput. The second term introduces a penalty for constraint violation. If the constraint in Eq. (4.6) is satisfied, the

energy introduced by the second term is zero. The third term equals zero only if all neurons converge to either 0 or 1. If the weights (w_1,w_2,w_3) are properly chosen, the annealing procedure will lead the system to a configuration with the minimum energy corresponding to the optimal frame pattern.

4.4.2 Determination of the Weights

Taking the derivative of E_1 and setting it equal to yields

$$\frac{\partial E_1}{\partial s_i} = (1 - G \cdot s_i) \exp(-G \cdot s_i) = 0.$$

Therefore, the frame pattern $\mathbf{s} = \{s_i = \frac{1}{G} \ \forall i\}$ provides the maximum throughput $\gamma_{max} = e^{-1}$. However, this frame pattern may not satisfy the constraint of Eqs. (4.6) and (4.7). On the other hand, the minimum value of E_2 is zero when $\sum_{i=1}^{N_d} s_i = N$, but any combinations of s_i would not achieve the maximum data throughput. Therefore, there exists a profound relationship between w_1 and w_2.

The annealing procedure attempts to relax the system into a state with the minimum energy, and simultaneously with all constraints satisfied. A state (frame pattern) $\mathbf{s}'$ that violates the constraint should yield higher energy than a state $\mathbf{s}$ which satisfies the constraints. Consider the situation that a state $\mathbf{s}$ satisfies the constraints Eqs. (4.6) and (4.7), and each neuron has converged to either 0 or 1. If a neighboring state $\mathbf{s}'$ of $\mathbf{s}$ differing from $\mathbf{s}$ only in one element violates the constraint such that

$$s_i' = \begin{cases} s_k - 1 & \text{if } i = k, \text{ for a certain } k, \\ s_i & \forall i \quad \text{except } i = k. \end{cases} \tag{4.10}$$

then

$$\sum_{i=1}^{N_d} s_i' = N - 1 \neq N,$$

and

$$E(\mathbf{s}') = -\frac{w_1}{N_d} \sum_{i=1}^{N_d} G \cdot s_i' \cdot e^{-G \cdot s_i'} + \frac{w_2}{2}, \tag{4.11}$$

$$E(\mathbf{s}) = -\frac{w_1}{N_d}\sum_{i=1}^{N_d} G \cdot s_i \cdot e^{-G \cdot s_i}. \tag{4.12}$$

According to the above statement, $E(\mathbf{s}') > E(\mathbf{s})$. Therefore from Eq. (4.11) and (4.12), we have

$$\begin{aligned}
E(\mathbf{s}') - E(\mathbf{s}) &= -\frac{w_1}{N_d}\sum_{i=1}^{N_d} G \cdot s_i' \cdot e^{-G \cdot s_i'} + \frac{w_2}{2} + \frac{w_1}{N_d}\sum_{i=1}^{N_d} G \cdot s_i \cdot e^{-G \cdot s_i} \\
&= \frac{w_2}{2} - \frac{w_1 \cdot G}{N_d}\left[s_k' \cdot e^{-G \cdot s_k'} - s_k \cdot e^{-G \cdot s_k}\right] \\
&= \frac{w_2}{2} - \frac{w_1 \cdot G}{N_d}\left[(s_k - 1) \cdot e^{-G \cdot (s_k - 1)} - s_k \cdot e^{-G \cdot s_k}\right] \\
&= \frac{w_2}{2} - \frac{w_1 \cdot G}{N_d} \cdot e^{-G \cdot s_k}\left[(s_k - 1) \cdot e^{G} - s_k\right] \\
&> \frac{w_2}{2} - \frac{w_1 \cdot G}{N_d} \cdot e^{-G \cdot s_k}\left[(s_k - 1) \cdot e^{G} - (s_k - 1)\right] \\
&= \frac{w_2}{2} - \frac{w_1 \cdot G}{N_d} \cdot e^{-G \cdot (s_k - 1)} \cdot (s_k - 1) \cdot [1 - e^{-G}]
\end{aligned}$$

Using the fact that

$$x \cdot e^{-x} \leq e^{-1},$$

$$\begin{aligned}
E(\mathbf{s}') - E(\mathbf{s}) &> \frac{w_2}{2} - \frac{w_1}{N_d} \cdot \left(1 - e^{-G}\right) \cdot e^{-1} \geq 0. \\
\Rightarrow \qquad w_2 &\geq \frac{2w_1}{N_d} \cdot \left(1 - e^{-G}\right) \cdot e^{-1}.
\end{aligned} \tag{4.13}$$

Here only one specific case is considered, indicating that the selection of w_1 and w_2 is related to G and N_d. The adjustment of the weights according to G and N_d is required to obtain better solutions. The term, w_3, is a weak constraint, and $w_3 = 1$ in this paper.

4.4.3 Critical Temperature

According to Eqs. (3.43) and (3.44), each neuron is updated as follows

$$v_{ij}(t+\delta t) = \frac{1}{2} + \frac{1}{2}\tanh\left[-\frac{1}{T}\frac{\partial E(\mathbf{v},t)}{\partial v_{ij}}\right] \qquad \forall i,j. \tag{4.14}$$

where

$$v_{ij} = \overline{s_{ij}}. \qquad 1 < i < N_d, 1 \le j \le m,$$

$$\mathbf{v} = \{v_1, v_2, \cdots, v_{N_d}\}, \qquad v_i = \{v_{i1}, v_{i2}, \cdots, v_{im}\}.$$

The critical temperature is defined as the temperature at which the sharp state transition starts. In Eq. (4.14), a very large value of T leads each v_{ij} to fluctuate around $\frac{1}{2}$, and the state transition is very slow. Therefore there must exist a critical temperature at which quick state transitions are expected to start. In this paper, the critical temperature is obtained by *trial-and-error*. That is, temperature is slowly decreased from a very high value. At each temperature, only one sweep is taken. At the end of each sweep, compute the average absolute error

$$\epsilon = \frac{1}{N_d \cdot m}\sum_{i=1}^{N_d}\sum_{j=1}^{m}\left|v_{ij}(t+\delta t) - v_{ij}(t)\right|, \tag{4.15}$$

where t stands for the time a sweep starts and $t+\delta t$ for the time a sweep ends. When $\epsilon \ge 0.1$, the above procedure stops, and the corresponding temperature is the critical one.

4.4.4 Annealing Schedule

The following annealing schedule is employed.

$$T_{n+1} = \frac{T_n}{1+\alpha \cdot n}. \tag{4.16}$$

where α is a small positive value, and n is the iteration index.

4.4.5 Termination of Sweeps

At each temperature, each neuron is updated according to Eq. (4.14). The sweep is terminated when $\epsilon \leq \delta_1$, where δ_1 is a small positive value. On the other hand, at some temperature, the condition may not be satisfied after a large number of sweeps. To avoid infinite sweeps, the sweep procedure is forced to end after a fixed number of sweeps n_{sweep}. Then the temperature is further decreased, and a new iteration begins.

4.4.6 Convergence Criterion

All v_{ij} should converge to either 0 or 1 after the last iteration. Therefore, we define the convergence criterion as

$$\frac{1}{m \cdot N_d} \sum_{i=1}^{N_d} \sum_{j=1}^{m} v_{ij}(1 - v_{ij}) < \delta_2, \tag{4.17}$$

where δ_2 is a small positive value. When the criterion is satisfied, all neurons are clamped, and the interdistances for the optimal frame pattern are found to be

$$s_i = \sum_{j=1}^{m} 2^j \cdot U\left(v_{ij} - \frac{1}{2}\right) \qquad \forall i. \tag{4.18}$$

where $U(\cdot)$ is a step function.

4.5 Numerical Results

Four instances with $N_d = 5, 8, 10, 15$, and $N = 40$ are tested by using the MFA algorithm. To demonstrate the advantage of the MFA scheme, a comparison with the random search and the SA approach [10] is made in terms of computational complexity and throughput optimality.

1. Random Search (RS): In random search, a frame pattern is randomly selected from the frame pattern space, and the one that yields the largest throughput is kept until termination. There is no fixed rule for terminating the procedure.

Usually, the procedure terminates after a certain number of iterations. Here an *iteration* consists of a frame pattern generation and a throughput comparison.

2. Simulated Annealing: To make the comparison fair, an *iteration* in [10] consists of a pattern generation and transition test based on the Metropolis criterion.

3. Mean Field Annealing: In MFA, to exploit the parallelism of neural networks, synchronous updating is adopted, i.e., the current value for each neuron is updated by using the previous neurons' values. Therefore, neurons in the neural network operate in parallel, and an *iteration* implies that the whole network is updated once, i.e., all neurons are updated once.

The three algorithms are implemented and compared. Each algorithm is executed 1000 times, and the throughput is averaged over the 1000 runs. Each *run* is defined as an execution of an algorithm. Fig. 4.6 shows the first 100 iterations of a run at the arrival rate G= 0.5. It can be seen that the MFA approach reaches its steady throughput within the first 100 iterations. Fig. 4.7 is the results after 500 iterations. Fig. 4.8 shows that, after 1000 iterations, all of the algorithms reach the steady throughput. Fig. 4.9 shows the average data throughput over an ensemble of 1000 runs when the arrival rate G is 0.5. Fig. 4.10 shows the average throughput achieved by the three algorithms over an ensemble of 1000 runs at different arrival rates. It can be seen that the average throughput achieved by both MFA and SA are very close. At some arrival rates, MFA can achieve even better average throughput than its counterpart because finite-time implementation (1000 iterations) of SA does not guarantee convergence to global optima. The RS algorithm is not effective in searching for the optimal pattern. It can be seen from these results that MFA achieves faster convergence than SA while achieving sub-optimal performance.

4.6 Summary

Searching for the optimal patterns in an integrated TDMA communication system is a combinatorial optimization problem. As the problem size gets large, the computational complexity becomes intractable. SA is a good algorithm in finding global optimal solutions, but it is usually time-consuming. MFA, which uses saddle point approximation, is proposed to solve for the optimal patterns. It is computationally efficient, and is able to acquire (sub) optimal solutions comparable to those obtained by SA. The determination of related parameters are addressed, and comparisons with the RS and SA approaches are presented. Numerical results have shown that MFA is 5-6 times faster than SA and at the same time achieves comparable solutions.

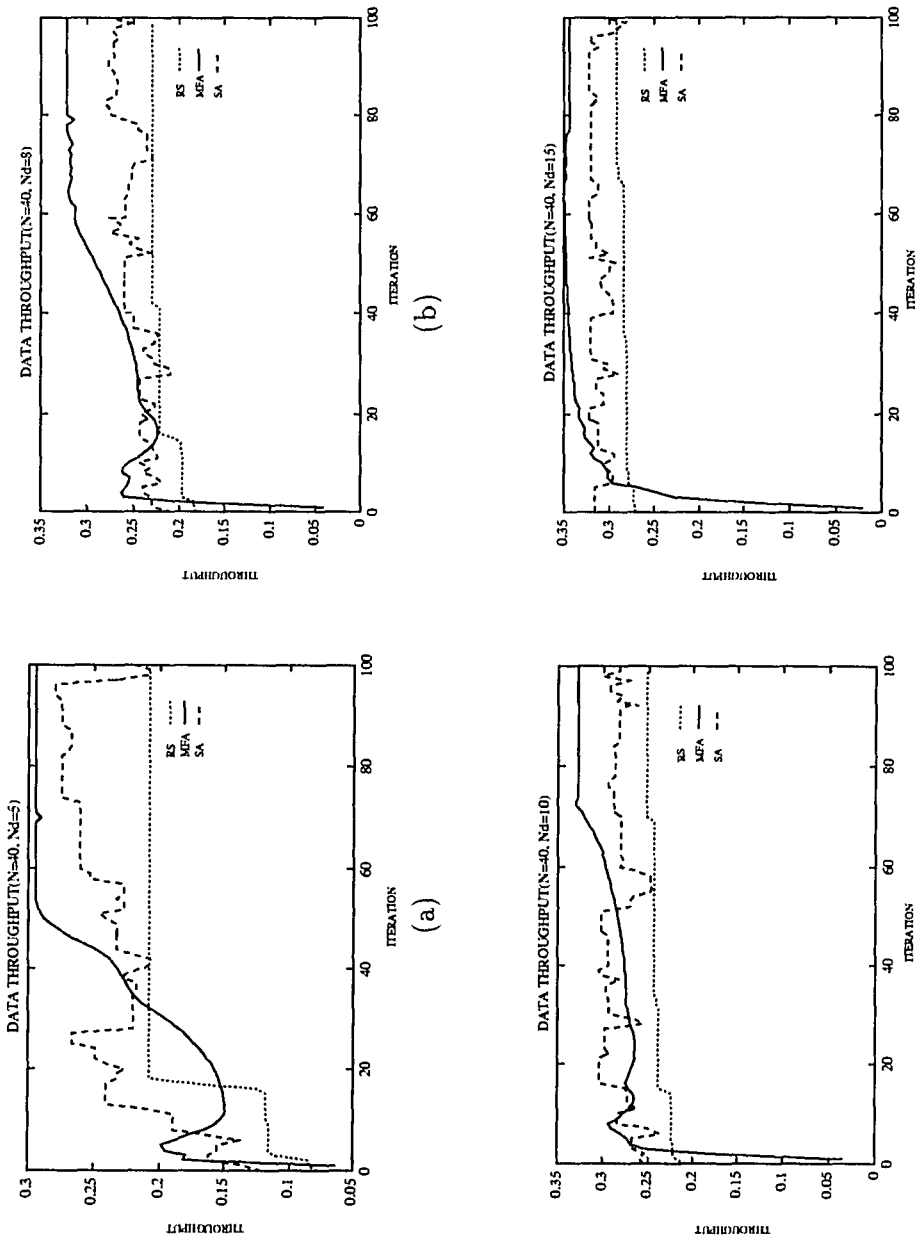


Figure 4.6 1 run and 100 iterations (G=0.5)

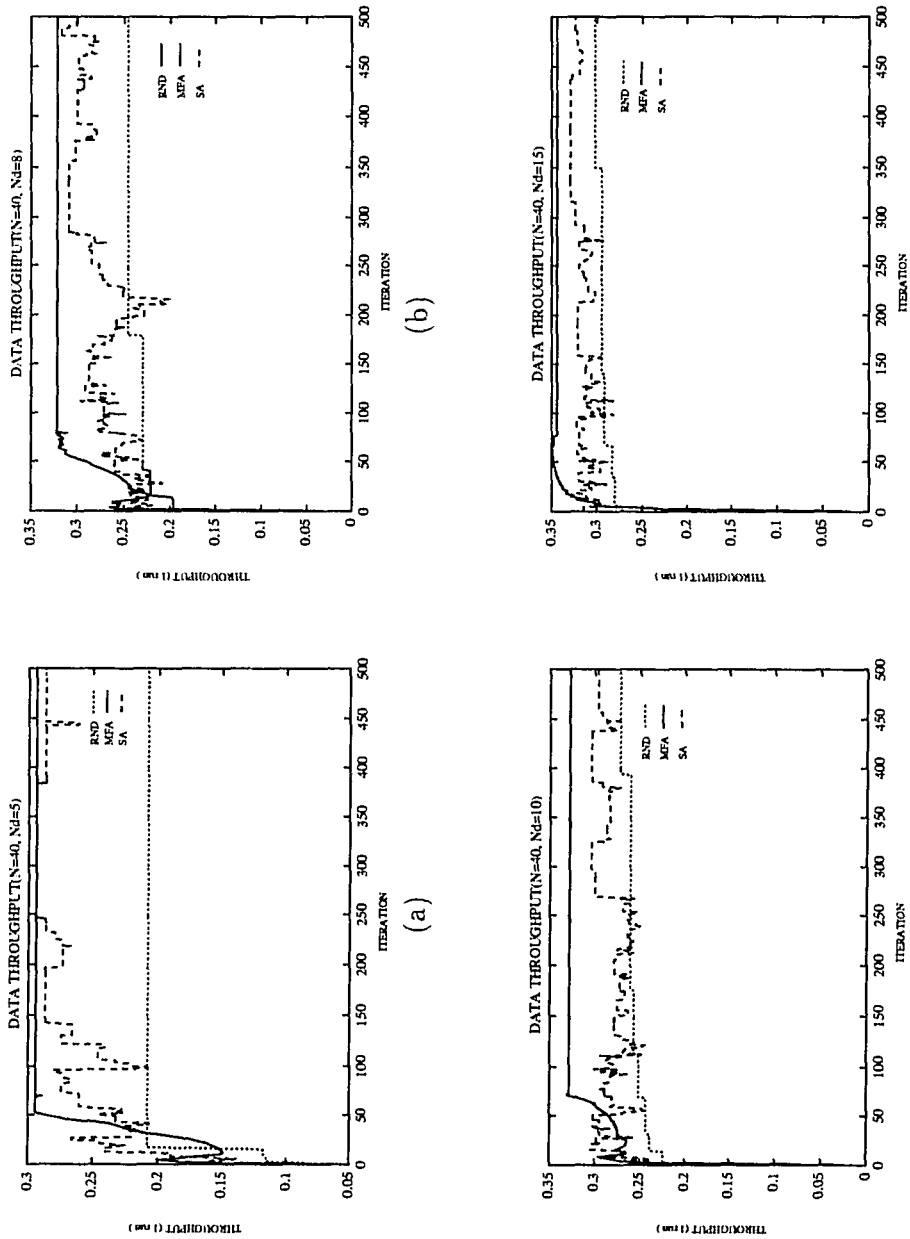


Figure 4.7 1 run and 500 iterations (G=0.5)

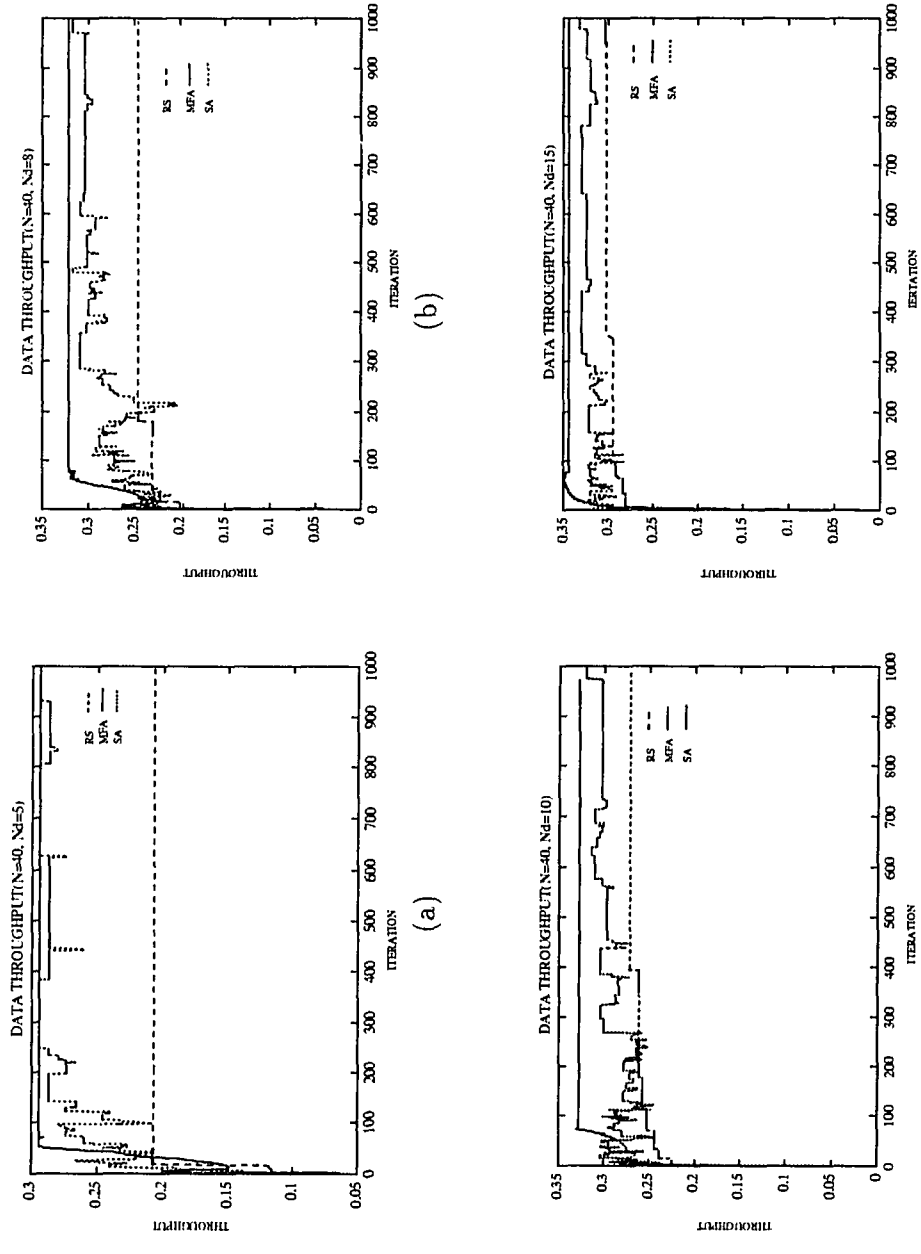


Figure 4.8 1 run and 1000 iterations (G=0.5)

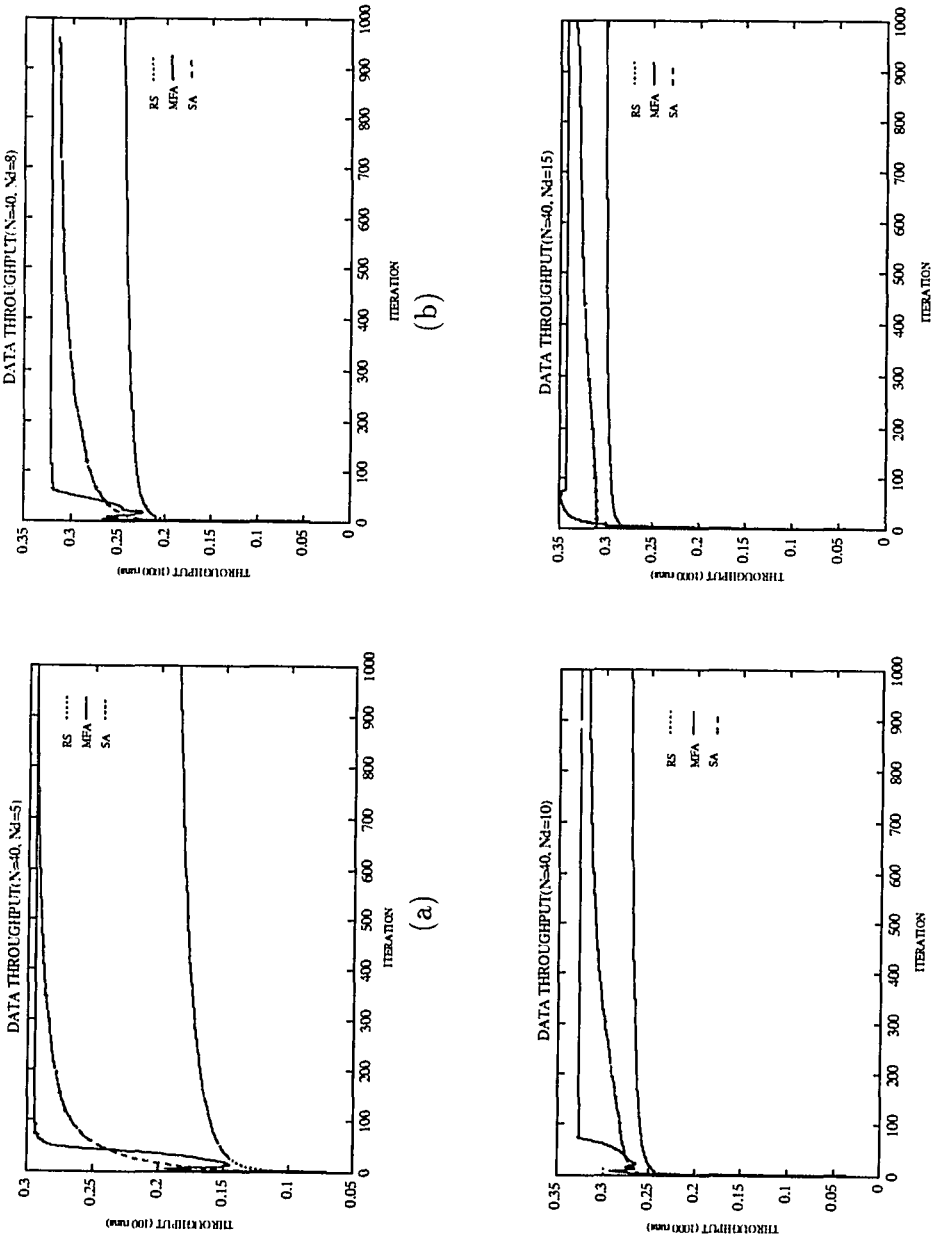


Figure 4.9 Ensemble average throughput for 1000 runs and 1000 iterations (G=0.5)

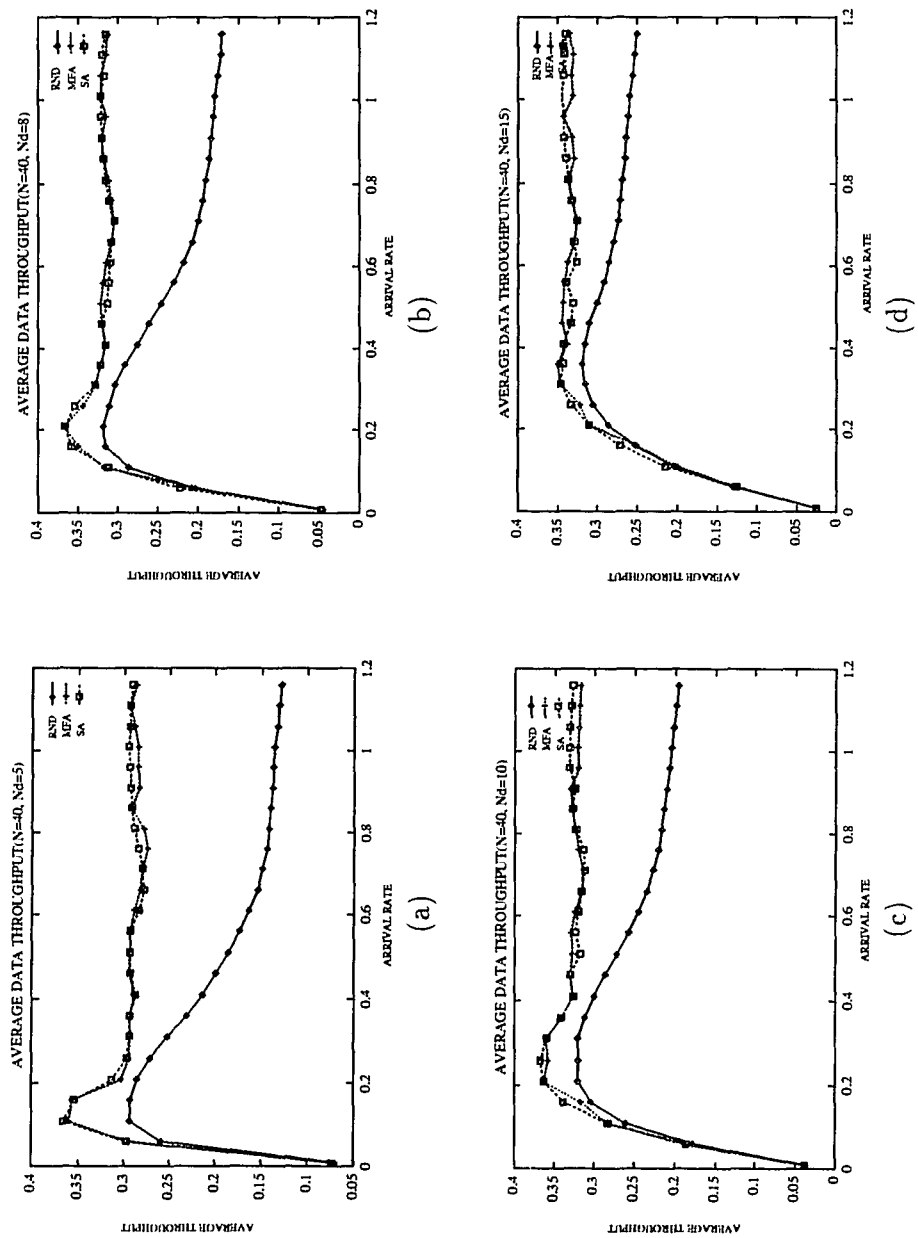


Figure 4.10 Average throughput for different N_d (1000 runs and 1000 iterations for each run)

CHAPTER 5

OPTIMAL BROADCAST SCHEDULING IN MULTIHOP PACKET RADIO NETWORKS

Another type of packet network is the *multihop store-and-forward multiaccess/broadcast* network. In this type of network, each station is equipped with a transmitter/receiver and a control unit. Stations communicate with each other via broadcast radio. The control unit performs the packet switching functions. Connectivity between neighboring stations is established, which means that the neighboring stations can directly receive transmissions from each other. All stations employ omni-directional antennas and share a high speed radio channel. When a station intends to transmit, it broadcasts through its antenna. Each neighboring station receives the transmission. If the transmission is destined for it, the neighboring station will absorb the packets. Otherwise, it will store the packets in its buffer and send them out later. Therefore, for any two distant stations where direct connectivity does not exist, the intermediate stations act as repeaters and perform store-and-forward functions. The typical example of the multihop store-and-forward packet radio is the packet radio network (PRNET) sponsored by the Advanced Research Projects Agency (ARPA) [23]. It permits mobile communication over a wide geographic area, provides efficient multiaccess for bursty-type traffic, and allows coexistence with different systems in the same frequency band and antijam protection. Fig. 5.1 shows a packet radio network in the internet. The communication between geographically separated hosts are established through the PRNET. Fig. 5.2 illustrates the broadcast transmission of data packets.

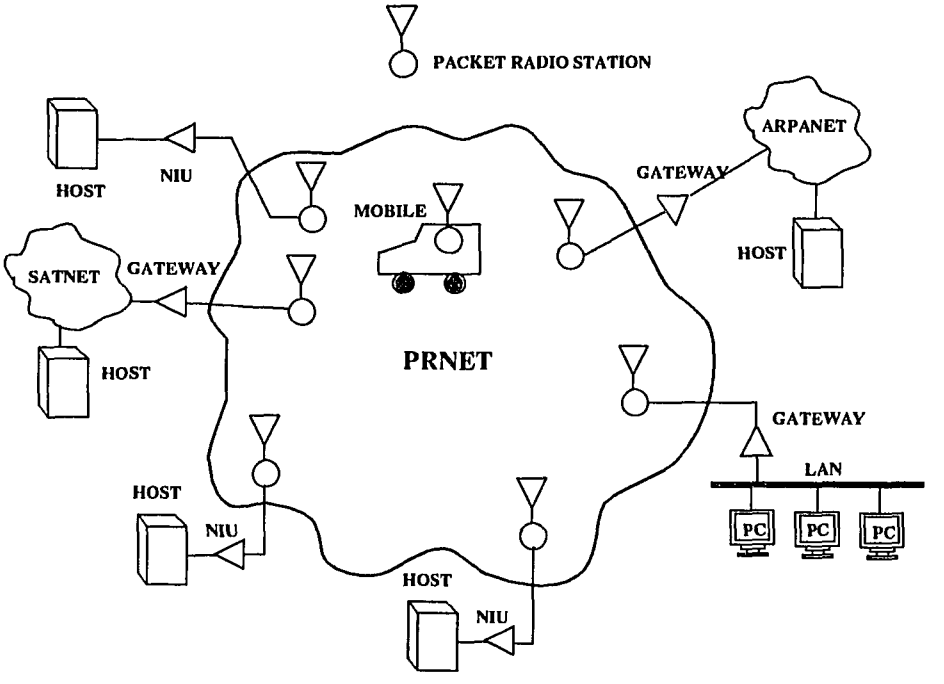


Figure 5.1 A PRNET in the internet.

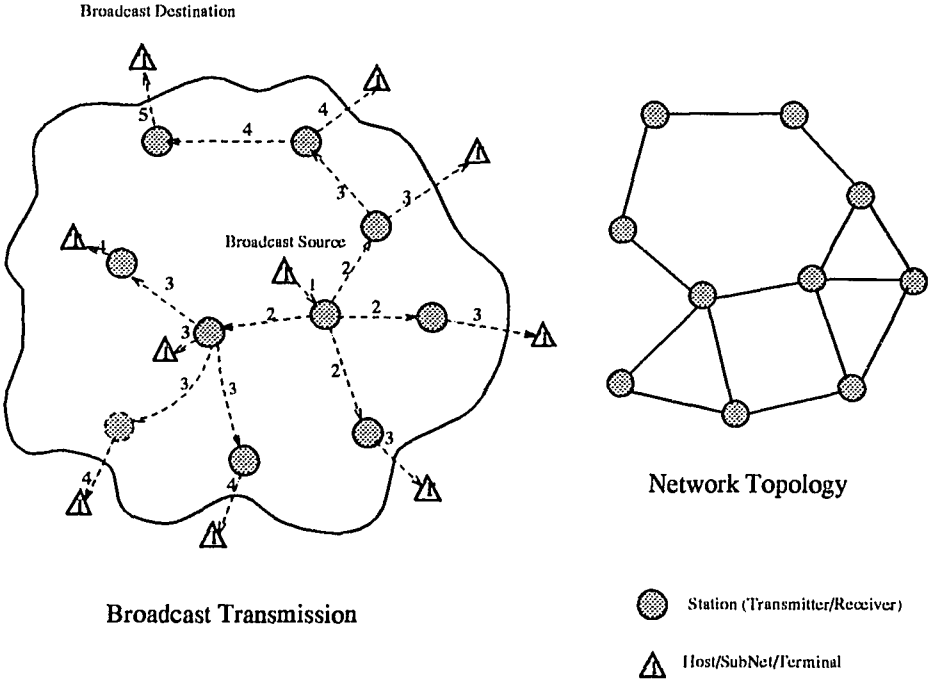


Figure 5.2 The broadcast transmission in a PRNET.

5.1 The Spatial TDMA Protocol and the Scheduling Problem

In a multihop PR network, since a single channel (usually broadband) is shared by all users, the transmission for each station must be scheduled to avoid any collision or interference. Based on the characteristics of a multihop network, the single radio channel can be shared by all stations in both time and space domains. A multiaccess protocol, namely spatial TDMA, can be used to schedule conflict-free transmission [26]. In the spatial TDMA network, time is divided into frames which consist of fixed-length time slots. When certain stations transmit simultaneously, collision or interference will occur. Therefore, any two stations that may result in collision or interference must be scheduled to transmit at different time slots, while the stations some distance away may be arranged to transmit at the same time slots without causing interference. Since the primary objective of the PR network is to provide high throughput with low delay, a scheduling scheme must provide a schedule which can achieve maximum channel utilization as well as lower delay. For a fixed-topology PR network in which locations of stations are fixed, the problem is to schedule a frame in which each station transmits at least once. Additional transmissions can be added into the frame if the addition does not cause any collision. There are two approaches for the scheduling. The first approach is to schedule the transmissions within a frame under the condition that the number of time slots in a frame (frame length) equals N, the number of stations in a PR network. The optimal schedule is the one which can provide the maximum number of collision-free transmissions in a frame. The second approach is to find an optimal frame schedule which has the minimum frame length while keeping a maximum number of transmissions. Such a scheduling problem is NP-complete (Appendix A), which implies a good algorithm rarely exists. In this chapter, an approximation algorithm based on mean field annealing (MFA) is presented to solve the scheduling problem.

5.2 Problem Formulation

A PR network can be represented by a graph $G = (V, E)$ where the vertices in V are network stations, and E is a set of edges. We assume that the network has a fixed topology. The total traffic passing through station i consists of packets received from other stations which will be routed through station i and the packets from the terminals attached to it. The spatial TDMA protocol is adopted in which a single wideband channel is shared by all stations of the network. Time is divided into unit-length slots. Each frame consists of a fixed number of time slots. Data can be transmitted in successive frames. The transmission time of stations in a frame is scheduled to avoid any collision. We are concerned with the fixed assignment of transmission for stations in a frame. Thus, once the optimal transmission patterns (the arrangement of transmissions) are determined, the frame is repeated in the time axis. Without loss of generality, we assume that a time slot is equal in length to the amount of time for a station to transmit one packet over the channel. We also assume that all stations have the same transmission range R and they are synchronized. Zero-capture is assumed, i.e., when some stations receive two or more overlapping packets, regardless of the difference of received signal power between the stations, collision occurs and all of the packets are destroyed. For any two stations i, $i' \in V$, if the distance between them is less than R, they can receive the packets transmitted from each other. Therefore, there exists an undirected edge $e = (i, i') \in E$ incident to station i and i', and the two stations are one-hop apart. If $(i, i') \notin E$ and there is such an intermediate station j as $(i, j) \in E$ and $(i', j) \in E$, then station i and i' are two-hop apart. The topology of a PR network can be described by an $(N \times N)$ symmetric binary matrix C, where $N = |V|$ is the number of stations in the network. The matrix, $C = (c_{ij})$ $(i, j = 1, \cdots N)$, also known as the connectivity matrix, is defined by

$$c_{ij} = \begin{cases} 1 & \text{if } (i, j) \in E \text{ and } i \neq j, \\ 0 & \text{otherwise.} \end{cases} \tag{5.1}$$

To ensure that a packet is correctly received in a station, the following constraints must be satisfied:

1. A station cannot have transmission and reception status simultaneously, i.e., if $(i, i') \in E$, station i and i' must be scheduled to transmit in different time slots.

2. A station is not allowed to receive two or more transmissions simultaneously, i.e., if $(i, j) \in E$ and $(j, k) \in E$, but $(i, k) \notin E$, station i and k must transmit in different time slots in order to avoid collision in station j.

If the first constraint is violated, *Primary interference* (collision-type) occurs. *Secondary interference* happens if the second constraint is violated. Fig. 5.3 illustrates the two types of interference. The optimal schedule must guarantee interference-free transmissions. In short, a station and its one-hop or two-hop neighboring stations must be scheduled to transmit in different time slots.

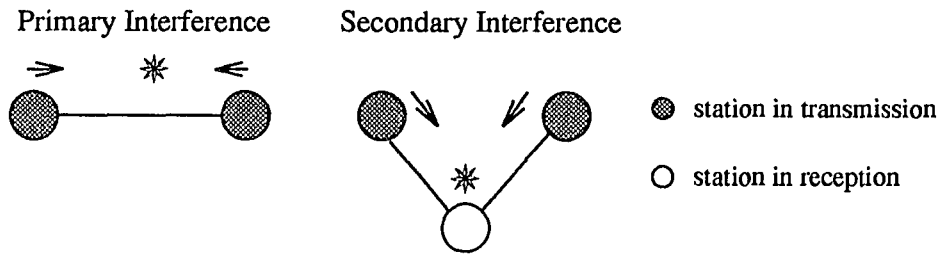


Figure 5.3 Illustration of interferences.

We can form a new $(M \times N)$ matrix called the compatibility matrix $F = (f_{ij})$ from matrix C, where

$$f_{ij} = \begin{cases} 1 & \text{if stations } i \text{ and } j \text{ are one-hop or two-hop apart,} \\ 0 & \text{otherwise.} \end{cases} \tag{5.2}$$

Note that $f_{ii} = 0\ \forall i$ and F is symmetric, i.e., $f_{ij} = f_{ji}$.

Therefore, for any two stations i and j, if $f_{ij} = 0$, both stations can transmit in the same slot with no collision. We assume that each frame consists of M time slots. In a frame, each station must be scheduled to transmit at least once (one time slot). Additional transmissions can be arranged provided that the addition does not cause collision. We use an $(M \times N)$ binary matrix $S = (s_{ij})$ to express a transmission schedule, where

$$s_{ij} = \begin{cases} 1 & \text{if station } j \text{ transmits at the } i\text{-th slot in a frame,} \\ 0 & \text{otherwise.} \end{cases} \tag{5.3}$$

Let ρ_k be the channel utilization for station k, then

$$\rho_k = \frac{\text{number of transmission slots assigned to station } k}{\text{frame length}} = \frac{\sum_{i=1}^{M} s_{ik}}{M}. \tag{5.4}$$

The channel utilization for the whole network, ρ, is given by

$$\rho = \frac{1}{N}\sum_{j=1}^{N} \rho_k = \frac{1}{NM}\sum_{i=1}^{M}\sum_{j=1}^{N} s_{ij}. \tag{5.5}$$

Denote S' as a set of collision-free schedules where $S' = \{S^1, \cdots, S^N\}$, and each feasible schedule S^i is an $M \times N$ binary matrix defined by Eq. (5.3). Define ρ_{S^i} as the channel utilization achieved by schedule S^i. Therefore, the optimal scheduling problem is described as follows.

Find the optimal schedule $S^{opt} \in S'$ so that

1. It has the frame length $M = N$ (Approach 1), or

   It has the minimum frame length M (Approach 2),

2. It satisfies the constraints

$$\sum_{i=1}^{M} s_{ij}^{opt} \geq 1 \quad (j = 1, 2, \cdots, N), \tag{5.6}$$

and

$$\sum_{k=1}^{M}\sum_{i=1}^{N}\sum_{j=1}^{N} f_{ij} s_{ki}^{opt} s_{kj}^{opt} = 0, \quad \text{and} \tag{5.7}$$

3. it yields the maximum channel utilization, i.e.,

$$\rho_{S^{opt}} = \max_{S^i \in \mathcal{S}'} \rho_{S^i}. \tag{5.8}$$

For a given M and N, there are 2^{MN} schedule configurations. An exhaustive search for the optimal schedules is prohibitive when M and N are large.

For the second approach, the minimum frame length depends on the topology of the network and is generally unknown for a PR network. However, a tight lower bound for a frame length can be found analytically, thus allowing one to estimate the minimum required frame length. By defining the degree of a vertex i as the number of edges incident to it and denoting the degree as $\deg(i)$, we have the following lemma:

Lemma 5.1 *The frame length M satisfies*

$$M \geq X(G) + 1, \tag{5.9}$$

where

$$X(G) = \max_{\forall i \in V} \deg(i). \tag{5.10}$$

Proof: It is obvious that $\deg(i)$ equals the number of one-hop neighbors of station i. Denote $B(i)$ as the set of one-hop neighbors of station i. For any two stations $j, j' \in B(i)$, since $c_{ij} = 1$ and $c_{ij'} = 1$, station j and j' are one-hop neighbors if $c_{jj'} = 1$ and two-hop neighbors if $c_{jj'} = 0$. According to the constraints mentioned above, station i and all its $\deg(i)$ one-hop neighbors must be arranged to transmit in different distinct time slots in order to obtain collision-free transmissions. Any two stations j, $j' \in B(i)$ cannot transmit in the same slot. Therefore, the required number of time slots for transmission for station i and its one-hop neighbors is $\deg(i)+1$, and the least required number of time slots of a frame for the network, $\Delta(G)$, is given by

$$\Delta(G) = X(G) + 1 = \max_{\forall i \in V} \deg(i) + 1. \tag{5.11}$$

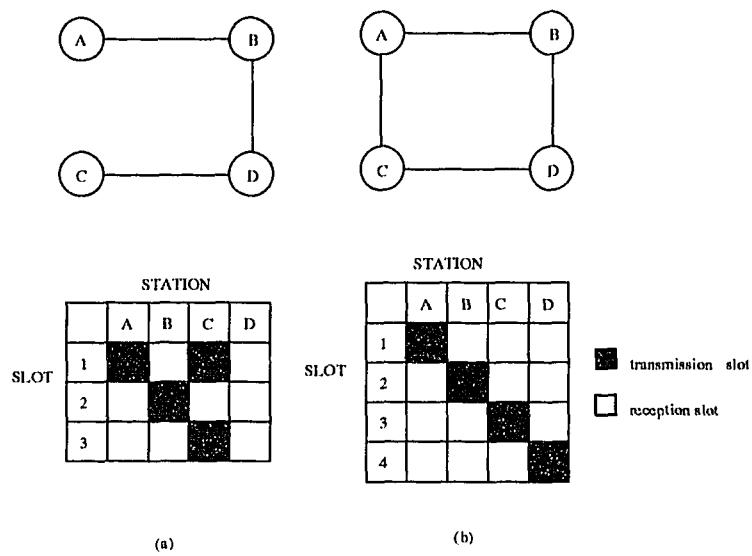


Figure 5.4 Two PR networks and their transmission schedules.

5.3 The Optimal Scheduling Algorithm Based on MFA

Eq. (5.11) only provides a lower bound for the frame length. For a given network, the frame length for any of the collision-free schedules is always greater than or equal to $\Delta(G)$, i.e., the inequality $M \geq \Delta(G)$ holds. The real frame length for an optimal schedule depends on the topology of a network. For certain networks, a feasible schedule with exact frame length $\Delta(G)$ may not exist. Therefore, a longer frame length is required. The example shown in Fig. 5.4 is used to illustrate this point. In this figure, two networks and the corresponding optimal schedules are given. In Fig. 5.4 (a), $X(G) = 2, \Delta(G) = 3, M = X(G)+1 = 3$, and the equality $M = \Delta(G)$ holds. In Fig. 5.4 (b), however, $X(G) = 2, \Delta(G) = 3, M = 4$, and the inequality $M > \Delta(G)$ holds. Thus, a feasible schedule with the frame length $\Delta(G)$ does not exist. From this example, we can see that the minimum required frame length is dependent on the connectivity of a network, and cannot in gereral, be predetermined. The lower bound for the frame length in Eq. (5.9) provides useful information when collision-free transmissions are scheduled. We can start to search for the optimal schedules

with a frame length equal to the lower bound $\Delta(G)$. If no feasible schedules with this length can be found, we will increase the frame length, and then search further for feasible solutions. In this way, the scheduled frame length would be minimized. Once the frame length is determined, the optimal scheduling procedure will continue until a collision-free schedule with maximum channel utilization, defined in Eq. (5.5), is found. This schedule provides the optimal solution for the scheduling problem. In the sequel, we will discuss how to use neural networks to solve such an optimal scheduling problem.

5.3.1 Energy Function

To solve the optimal scheduling problem by using MFA, we first need to map the channel utilization to be maximized and the constraints into an energy function. We assume that the frame length is M and there are N stations in a PR network. $M \times N$ neurons are required to represent a schedule. Each neuron s_{ij} ($i = 1, \cdots, M$; $j = 1, \cdots, N$) is defined in Eq. (5.3). The following energy function is derived

$$E = -\frac{w_1}{2}\left(\frac{1}{NM}\sum_{i=1}^{M}\sum_{j=1}^{N} s_{ij}^2\right) + \frac{w_2}{2}\sum_{k=1}^{M}\sum_{i=1}^{N}\sum_{j=1}^{N} f_{ij}s_{ki}s_{kj} + w_3\sum_{i=1}^{M}\sum_{j=1}^{N} s_{ij}(1 - s_{ij}). \quad (5.12)$$

where weights $w_i > 0, i = 1, 2, 3$. The first term in Eq. (5.12) is negatively weighted channel utilization. The second term is a penalty function for constraint violations. When the constraint is satisfied, it becomes zero. The third term is used to force neurons to converge to either 0 or 1 (if all $s_{ij} = 0$ or 1, the third term equals zero). The mean field of neuron ij is

$$h_{ij}^{MFT} = -\frac{\partial E}{\partial v_{ij}} = \frac{w_1}{NM} \cdot v_{ij} - w_2\sum_{k=1}^{N} f_{jk} \cdot v_{ik} - w_3(1 - 2v_{ij}), \quad (5.13)$$

where

$$v_{ij} = \overline{s_{ij}}.$$

5.3.2 The MFA Scheduling Algorithm

The updating of the neuron average is given by

$$v_{ij}(t+\Delta t) = \frac{1}{2}\left\{1+\tanh\left[\frac{1}{2T}\left(\frac{w_1}{NM}\cdot v_{ij} - w_2\sum_{k=1}^{N} f_{jk}\cdot v_{ik} - w_3(1-2v_{ij})\right)\right]\right\}. \tag{5.14}$$

Then the MFA iteration proceeds until freezing occurs. For the first approach, the frame length is equal to the number of stations in a PR network. For the second approach, since the exact frame length is unknown, we can start to schedule the frame with length $\Delta(G)$, the lower bound of the frame length shown in Eq. (5.11). The proposed scheduling algorithm includes three steps:

Step 1: Presetting neurons

Approach 1: The frame length equals the number of stations in a PR network. Set $v_{ii} = 1 \ \ \forall i \in V$

Approach 2: Find the station p which has the maximum degree $X(G)$, then set the initial frame length $M = \Delta(G)$ as defined in Eq. (5.11), and assign station p and its one-hop neighboring stations $j \in B(p) = \{k : c_{pk} = 1\}$ to the different distinct time slots. For example, set $v_{1p} = 1$ and $v_{ij} = 1$ $\forall j \in B(p)$ $(i = 2, \cdots, \Delta(G))$. For the i-th slot, since $v_{ij} = 1$, The k-th neuron with $f_{jk} = 1$ must be set to $v_{ik} = 0$ to resolve collisions (see Eq. (5.7)). The preassigned neurons no longer need to be updated and their values will be used to update the other neurons.

Step 2: Performing the MFA iterations based on Eq. (5.14)

The iteration continues until freezing occurs and the freezing state should provide the maximum channel utilization within the frame length M.

Step 3: Applying the heuristic algorithm for unassigned stations (only for Approach 2)

After completing the above two steps, some stations might remain unassigned for transmission due to the collision-free constraint. The number of unassigned stations depends on the topology of the network. Usually, after the first two steps, only a few stations are unassigned. Extra time slots are needed to arrange the remaining transmissions. We use the following heuristic algorithm to schedule the transmissions of the unassigned stations. Denote the unassigned stations as $U = \{U_1, \cdots, U_q\}$

1. Sort the stations in U in a descending order of station degree such that $\deg(U_i) \geq \deg(U_{i+1})$.
2. Add a time slot for the frame, and assign the stations in U to transmit in the slot. The priority of assigning a station's transmission is based on the order of U; i.e., the priority of U_i is greater than that of U_{i+1}. The stations arranged in the slot must be conflict-free. Repeat the above procedure until U is empty.
3. The actual frame length M equals $\Delta(G)$ + the number of added time slots.
4. Check the stations which have been assigned to transmit in the first $\Delta(G)$ time slots. If any of the stations can transmit in the added time slots without conflict, assign the transmissions of the stations in the corresponding time slots.

After the three steps are completed, the optimal schedule represented by v_{ij} $\forall i, j$ is translated into the actual transmission assignment (i.e., $v_{ij} = 1$ means that station j can transmit in slot i).

To illustrate the scheduling algorithm, we use the network shown in Fig. 5.2(a) as an example. Fig. 5.2(b) is the result from the first step. The value in the i-th row and j-th column is v_{ij}, the value of neuron ij. $v_{ij} = 1$ means that station j can

transmit in slot i, whereas $v_{ij} = 0$ implies that it is prohibited from transmitting. The values for the empty neurons are decided in the successive steps. Fig. 5.5(c) is the result after the second step is performed. For this example, after the second step is completed, all stations have been arranged for transmission. Therefore, the third step is not executed.

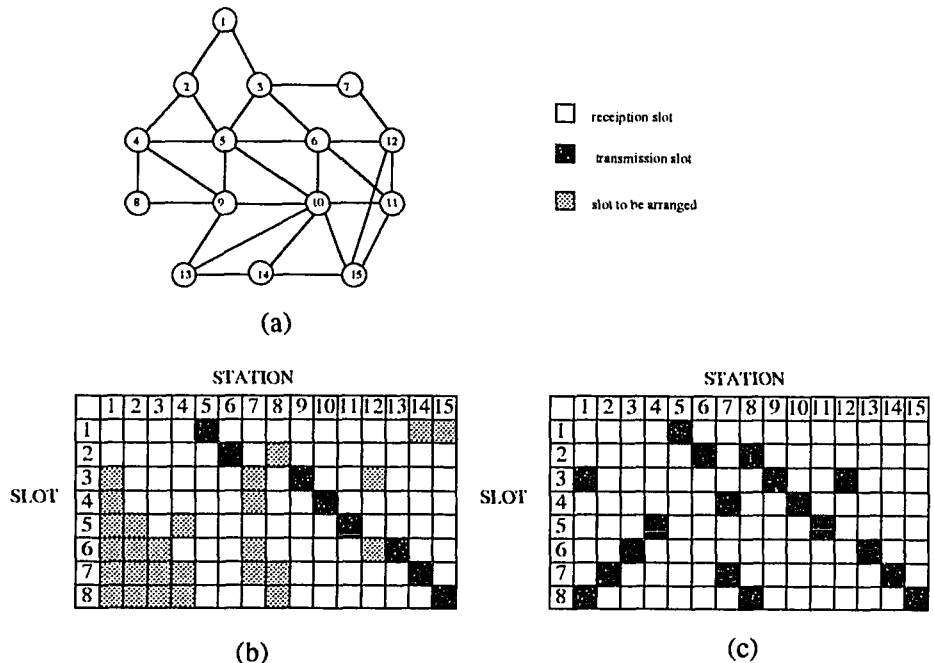


Figure 5.5: Illustration of the procedure involved in the MFA algorithm. (a) the 15-station network, (b) the result from the first step. (c) the result after the second step is executed.

5.3.3 Critical Temperature

Each neuron is updated according to:

$$v_{ij}^{(n)} = \frac{1}{2} + \frac{1}{2}\tanh\left(-\frac{1}{2T}\frac{\partial E(v^{(n-1)})}{\partial v_{ij}^{(n-1)}}\right) \qquad \forall i,j, \tag{5.15}$$

where $v_{ij} =< s_{ij} >_T$, and $< \cdot >$ is the averaging operator. $v = (v_{ij})$ is the neuron matrix of dimension $M \times N$, and n stands for the n-th iteration.

From Eq. (5.15), it is seen that iterations starting at too high a temperature result in trivial solutions. Starting at too low a temperature, on the other hand, might force the system into a poor or invalid solution. Critical temperature is defined as the temperature at which fast state transitions begin.

For the scheduling problem,

$$\begin{aligned} v_{ij}^{(n)} &= \frac{1}{2} + \frac{1}{2}\tanh\left(-\frac{1}{2T}\frac{\partial E(v^{(n-1)})}{\partial v_{ij}^{(n-1)}}\right) \\ &= \frac{1}{2} + \frac{1}{2}\tanh\left\{-\frac{1}{2T}\left[-\frac{w_1}{MN}v_{ij}^{(n-1)} + w_2\sum_{k=1}^{N} f_{jk}\cdot v_{ik}^{(n-1)}) + w_3(1-2v_{ij}^{(n-1)})\right]\right\}. \end{aligned} \tag{5.16}$$

It is seen from Eq. (5.16) that the state of each neuron remains relatively unchanged at high temperatures. The iteration procedure in Eq. (5.16) should start at a temperature (the critical temperature) at which fast transition begins as illustrated in Fig. 5.6 until steady state is reached. Thus,

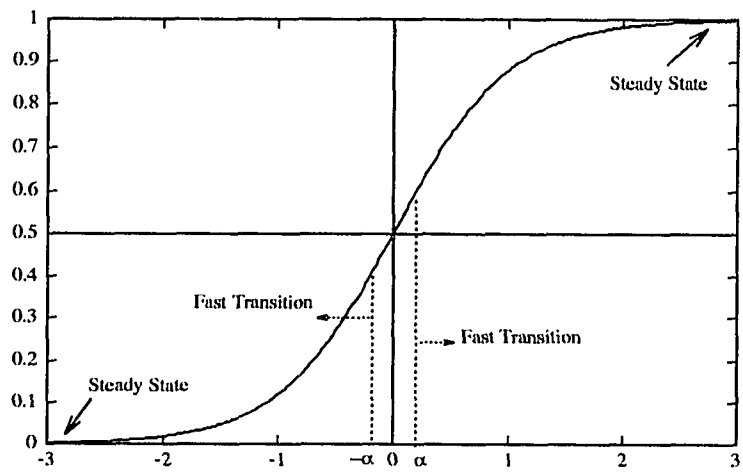


Figure 5.6 The illustration of state transition.

$$\left|\frac{1}{2T_c}\frac{\partial E}{\partial v_{ij}^{(0)}}\right| = \alpha,$$

where $0.1 \leq \alpha \leq 0.2$ is chosen, at which state transition becomes rapid. Since all neurons v_{ij} are initialized to $0.5 \pm \delta 1$ ($\delta 1$ is a small random number), we choose

$$\begin{aligned} T_c &= \frac{1}{2\alpha}\left|\frac{\partial E}{\partial v_{ij}^{(0)}}\right| \\ &= \frac{1}{2\alpha}\left|-\frac{0.5W_1}{MN} + W_2\sum_{k=1}^{N} 0.5 f_{jk} + W_3(1-2v_{ij}^{(0)})\right| \\ &\approx \frac{1}{2\alpha}\left|-\frac{0.5W_1}{MN} + W_2\sum_{k=1}^{N} 0.5 f_{jk}\right|. \end{aligned} \tag{5.17}$$

Thus,

$$T_c \geq \frac{1}{4\alpha}\left(-\frac{W_1}{MN} + W_2 \cdot \min_{\mathrm{j}} \sum_{k=1}^{N} f_{jk}\right),$$

or

$$T_c \leq \frac{1}{4\alpha}\left(-\frac{-W_1}{MN} + W_2 \cdot \max_{\mathrm{j}} \sum_{k=1}^{N} f_{jk}\right).$$

The lower bound for the temperature is taken as

$$T_c = \frac{1}{4\alpha}\left(-\frac{W_1}{MN} + W_2 \cdot \min_{\mathrm{j}} \sum_{k} f_{jk}\right). \tag{5.18}$$

The derived critical temperature is suitable only for the synchronous mode, in which all $v_{ij}^{(n)}$ are updated simultaneously using the previous $v^{(n-1)}$. For the asynchronous mode, each neuron is sequentially updated. The critical temperature for asynchronous iteration can be estimated by a *trial-and-error* method, i.e., the iteration starts at a very high temperature and the temperature is gradually lowered. At each temperature, each neuron is sequentially updated once. At the end of each iteration, the absolute average value

$$\epsilon 1 = \frac{1}{MN^2}\sum_{i}\sum_{j}\left|v_{ij}^{(n)} - v_{ij}^{(n-1)}\right| \tag{5.19}$$

is checked. At high temperatures, $\epsilon 1 \ll \alpha$ for some constant α. When $\epsilon 1 \geq \alpha$, significant state transitions begin. Therefore, when $\epsilon 1 \geq \alpha$, the trial process ends, at which point the corresponding temperature is critical.

5.3.4 Annealing Schedule

The annealing schedule reflects the way the temperature is reduced, and the following empirical annealing schedule

$$T_{n+1} = 0.9 \cdot T_n \tag{5.20}$$

is adopted in our simulations.

5.3.5 Stopping Criterion

At a very low temperature, all neurons converge to either 0 or 1. Let

$$\epsilon 2 = \frac{1}{MN} \sum_i \sum_j v_{ij}^{(n)} \left(1 - v_{ij}^{(n)}\right). \tag{5.21}$$

Thus, the iterative process may stop either when the error is approaching 0 (in our case, $\epsilon 2 \leq 0.01$) or when the temperature reaches 0. The final values of neurons represent the schedule.

5.4 Numerical Examples and Performance Analysis

5.4.1 Channel Utilization and Average Time Delay

The performance of the resulting schedules can be evaluated by two criteria: channel utilization and average time delay. The channel utilization is defined in Eq. (5.5). Before we derive the average time delay, the following assumptions are made:

1. Packets have a fixed length, and the length of a time slot equals the time required to transmit a packet.

2. The interarrival time for each station i is statistically independent from other stations, and packets arrive according to a *Poisson* process with a rate of λ_i (packets/slot). The total traffic in stations i consists of the traffic incoming from other stations and the data from terminals attached to it. Packets are stored in buffers in each station and the buffer size is infinite.

3. The probability distribution of the service time of station i is deterministic and statistically independent from other stations. The average service rate is μ_i (packets/slot).

4. Packets can be transmitted only at the beginning of each time slot.

Under the above assumptions, a network can be modeled as N M/D/1 queues, where N is the number of stations. According to the *Pollaczek-Khinchin formula* [7], the average delay for each queue i is given by

$$D_i = \overline{X}_i + \frac{\lambda_i \overline{X_i^2}}{2(1-\rho_i)}, \tag{5.22}$$

where $\overline{X_i} = \frac{1}{\mu_i}$ is the average service time for station i ,

$\rho_i = \frac{\lambda_i}{\mu_i}$ is the utilization factor for station i, and

$\overline{X_i^2}$ is the second moment of service time for station i.

Since the service time is deterministic, the variance equals zero, and thus

$$\overline{X_i^2} = \overline{X_i}^2 = \frac{1}{\mu_i^2}, \tag{5.23}$$

and

$$\mu_i = \frac{\sum_{j=1}^{m} v_{ji}}{M} \text{ (packets/slot)}. \tag{5.24}$$

The total time delay is given by

$$D = \frac{\sum_{i=1}^{N} \lambda_i D_i}{\sum_{i=1}^{N} \lambda_i}. \tag{5.25}$$

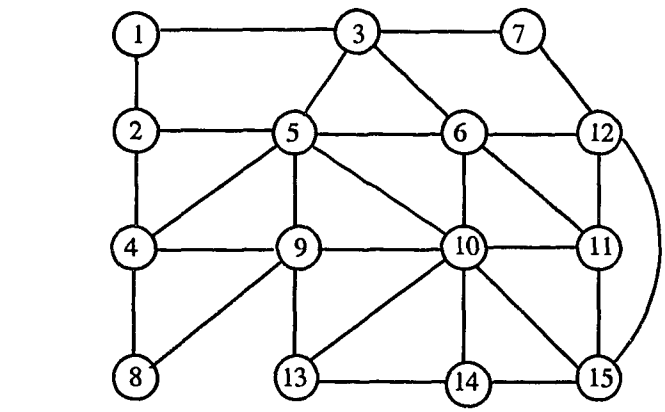
5.4.2 Numerical Results

Three PR networks with 15, 30, and 40 stations are scheduled in our numerical examples. The networks are shown in Fig. 5.7. The scheduling results for the two networks are shown in Fig. 5.8 for Approach 1 and in Fig. 5.9 for Approach 2. For the three networks in Approach 2, $\Delta(G) = 9, 10, 9$ and the scheduled frame length is 9,11,10 respectively, which are close to the lower bound shown in Eq. (5.9).

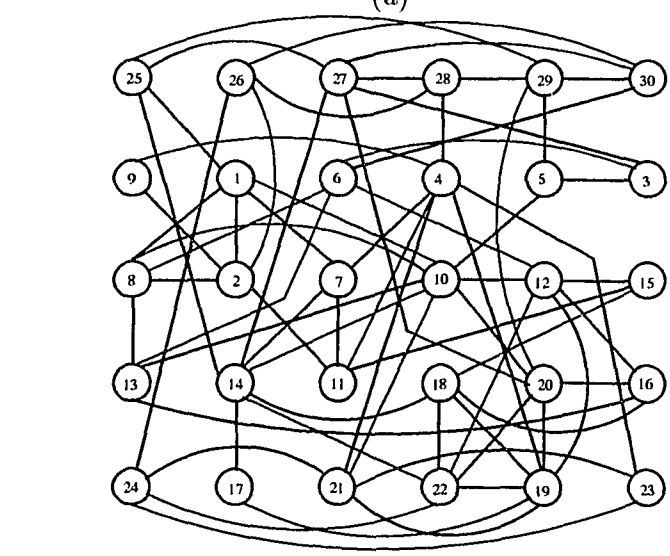
We compare the performance achieved by the MFA scheduling algorithm with the other two scheduling algorithms [13], [41] in which the objective of scheduling is to achieve the maximum channel utilization. The time delay and channel utilization are plotted in Fig. 5.10. From this figure, it is seen that the time delay experienced by the MFA schedule is much less than that of the other two scheduling algorithm s which have the same time delay, and the channel utilization achieved by MFA is a little bit less than the other two.

5.5 Summary

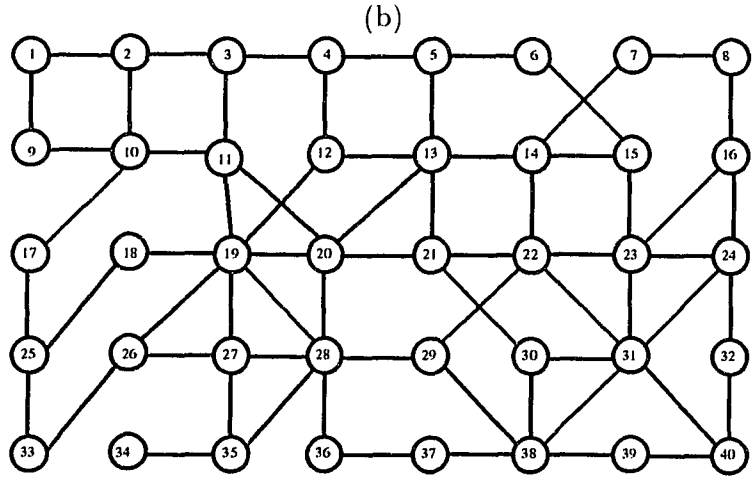
In this chapter, an MFA-based algorithm is proposed to solve the broadcast scheduling problem in PR networks. Two approaches are used in the scheduling. Approach 1 provides the highest channel utilization among all of the approaches, but the time delay is much larger than Approach 2. Approach 2 has the minimum time delay at the expense of lower channel utilization. As compared with the E&T algorithm [13], the proposed MFA approaches provide superior performance in both channel utilization and average time delay.



(a)

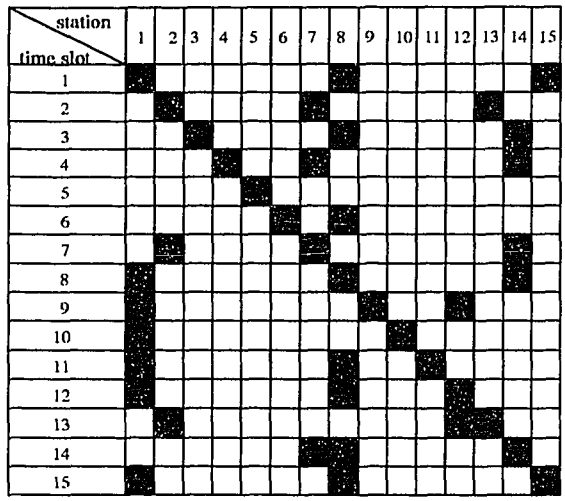


(b)

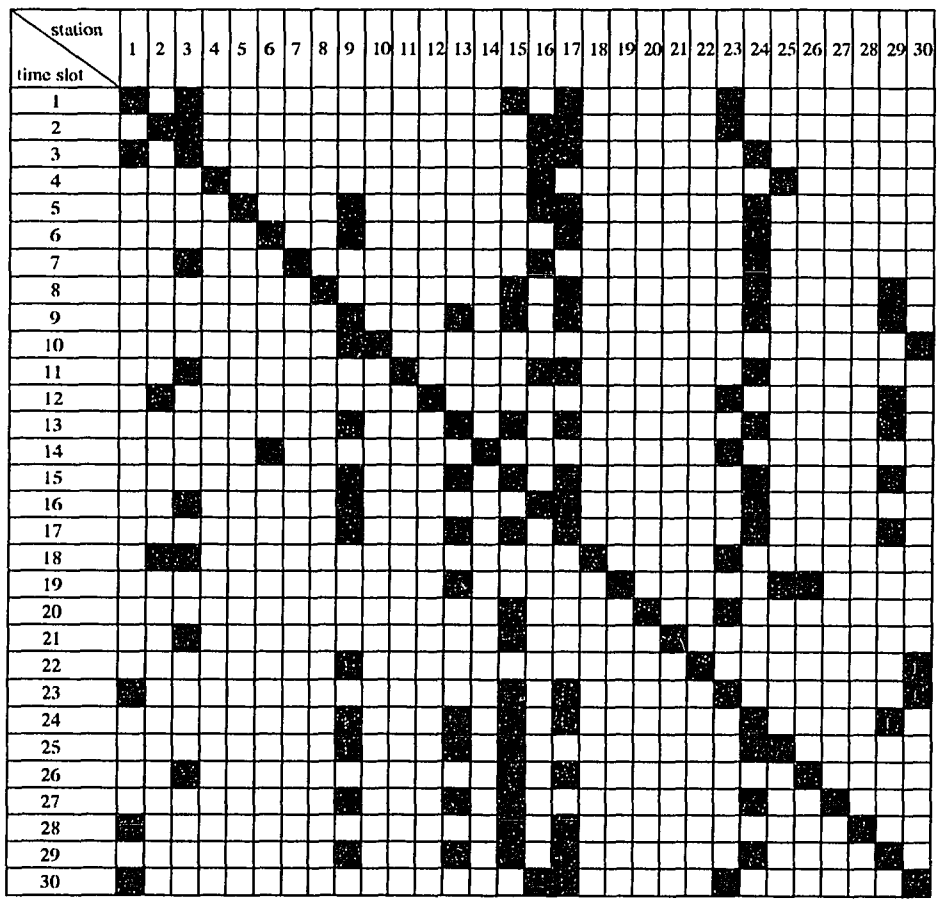


(c)

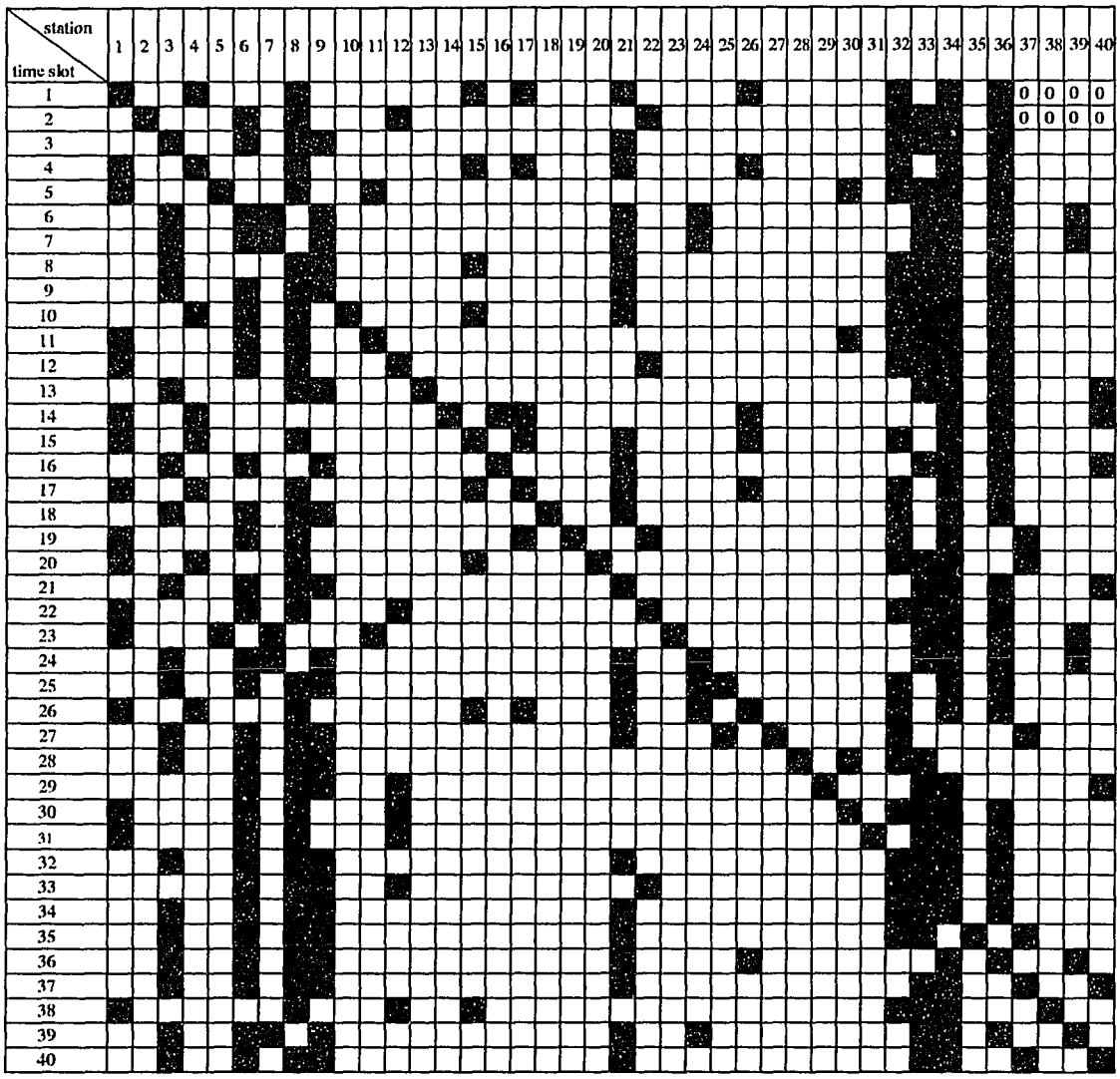
Figure 5.7: The radio networks used in the simulation. (a) the 15-station network. (b) the 30-station network. (c) the 40-station network.


station
time slot
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

(a)

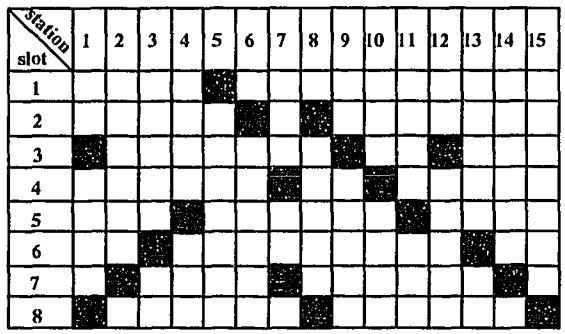

station
time slot
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30

(b)

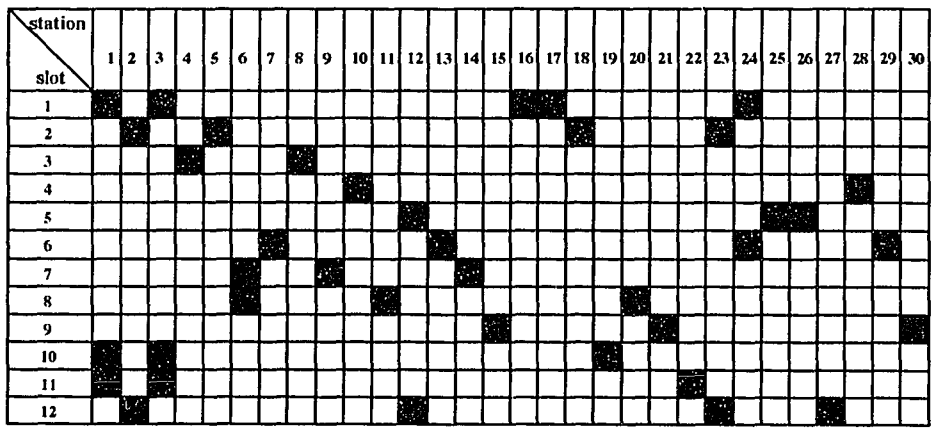


(c)

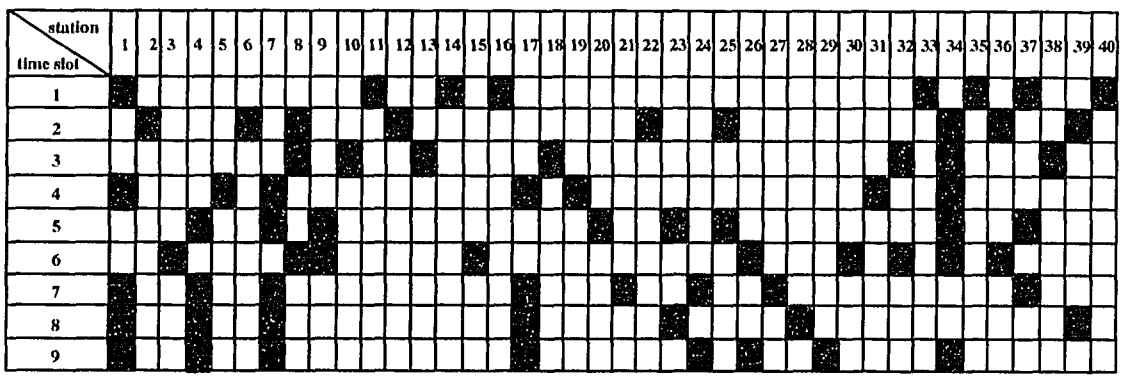
Figure 5.8: The MFA broadcast schedules for Approach 1: (a) the 15-station network. (b) the 30-station network. (c) the 40-station network.



(a)



(b)



(c)

Figure 5.9: The MFA schedules (Approach 2): (a) the 15-station network. (b) the 30-station network. (c) the 40-station network.

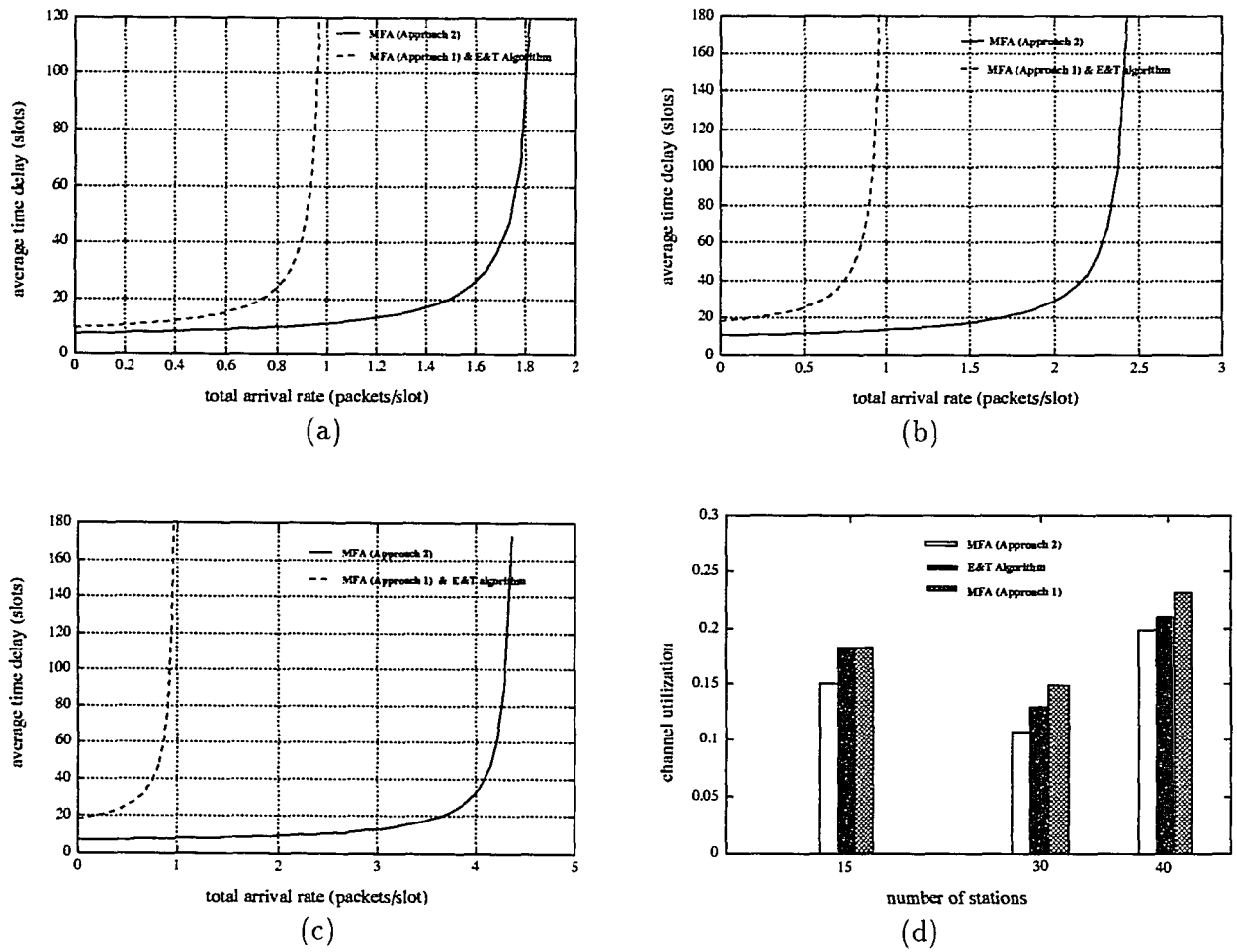


Figure 5.10 Time delay for different schemes: (a) 15-station. (b) 30-station (c) 40-station. (d) Channel utilization

CHAPTER 6

OPTIMAL CHANNEL ASSIGNMENTS IN FDMA MOBILE COMMUNICATION NETWORKS

In the FDMA protocol, the whole available bandwidth is subdivided into a number of subbands called *channels*. Each user in the network is confined to access an allocated channel only. Although FDMA takes a fraction of the bandwidth to achieve adequate frequency separation, it is relatively easy to implement and requires no real time coordination. One of the FDMA examples is cellular communication systems, in which the frequency band is allocated by the Federal Communications Commission (FCC) to be on 824-849 MHz for uplink transmissions (from a mobile to a base station) and on 869-894 MHz for downlink transmissions (from a base station to a mobile). The frequency band is subdivided into a certain number of narrowband channels, each capable of supporting one phone circuit that can be accessed by any user. The channels are indexed by a sequence of numbers $\{1, 2, \cdots, N\}$. Channel i and Channel $i+1$ are called adjacent. The channel spacing is 30 kHz. Therefore this frequency band can accommodate 832 duplex channels [12][17][27][36]. Furthermore, the 832 duplex channels are equally divided into Bands A and B. Voice and control channels are assigned at each base station from the allocated spectrum, either A or B band. Therefore, there are 416 channels for each band, including 21 control channels for call setup. Thus, each band has 395 available traffic channels for voice transmissions. Fig. 6.1 shows a cellular communication system in which a geographical area is divided into hexagonal *cells*. The number of cells K, which cannot use the same channel, is called the frequency reuse factor. This prevents adjacent cells from interfering with one another. A frequency pattern, or cluster, is determined by the equation $K = i^2 + ij + j^2$, where $i, j \in \mathbf{Z}^+$. The cluster shown in Fig. 6.1 corresponds

$i = 1$ and $j = 2$, or $K = 7$. K is related to the distance, D, between two adjacent cells which use the same frequency, and the radius of a cell r, by

$$K = \frac{1}{3}\left(\frac{D}{r}\right)^2.$$

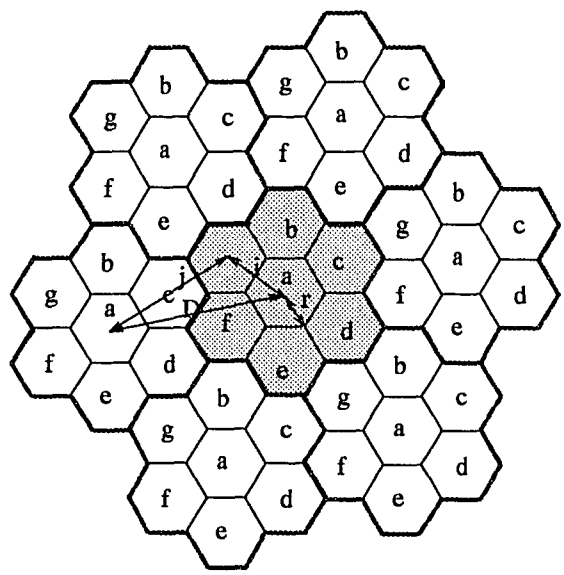


Figure 6.1 A cellular network with $K = 7$.

There are currently 395 traffic channels available for a cell at each band. To account for interference, a cell can only use $395/K$ channels. Each cell is further divided into 3 sectors for better reception. Thus, a cell sector can support $395/3K$ traffic channels. For example, for $K = 7$ as shown in Fig. 6.1, each cell can support $395/7 = 56$ traffic channels and each sector in a cell can only support $56/3 = 18$ traffic channels (calls).

As the demand for communications increases, hundreds of channels may be required to serve thousands of concurrent users. To meet this requirement, channels must be reused in a certain way. In a cellular mobile-telephone system, the whole geographical communication area is divided into *cells*. Each cell covers a sub-area.

The same channel used in one cell might be reused in another cell provided that the two cells are separated by a certain distance in space. Interference may occur when the same cell or different cells use certain pairs of channels. In order to avoid any interference, three types of interference constraints, namely *co-channel, adjacent-channel* and *co-site*, must be satisfied. We will address the constraints in the next section. The channel assignment task, given a group of available channels, is to find an assignment that satisfies the users' communication demands and various constraints.

6.1 The Optimal Channel Assignment Problem

For an n-cell inhomogeneous cellular radio network, the cell system is expressed by $X = \{x_1, x_2, \cdots, x_n\}$, where $x_i \, \forall i$ is called a cell. The requirement on X is an n-vector $R = (r_i)$ where r_i is the number of required channels by cell x_i. The interference constraints can be described by an $n \times n$ non-negative matrix $C = (c_{ij})$. C is called a *compatibility matrix* on X and it is defined as follows.

1. Co-channel constraint $c_{ij} = 1$ $(i \neq j)$: if f, f' are the channels assigned to cell x_i and x_j respectively, then they must satisfy $|f - f'| \geq 1$. That is, the pair of cells x_i and x_j cannot use the same channel.

2. Adjacent-channel constraint $c_{ij} = 2$ $(i \neq j)$: in this case, it requires $|f - f'| \geq 2$. The use of adjacent channels (*i.e.*, $|f - f'| = 1$) in x_i and x_j is prohibited.

3. Co-site constraint $c_{ii} = l$: here l is a positive integer. If the channels f and f' are allocated to cell x_i, $|f - f'| \geq l$ must hold, i.e., the channels used in the same cell must be separated by a certain distance in the frequency domain.

We define an n-vector $F = (F_i)$, where F_i is a bundle of channels assigned to cell x_i and $|F_i|$ is the number of channels assigned to cell x_i. A triple $\Pi_C = (X, R, C)$ is

called a channel assignment problem. F will be called a *feasible channel assignment* for Π_C if the following conditions are satisfied.

1. $|F_i| = r_i, \forall i$;

2. $|f - f'| \geq c_{ij}$, for all $x_i, x_j \in X, f \in F_i, f' \in F_j$.

If m successive channels are assigned to cells in an assignment, then m is called the span of the assignment. For a given channel assignment problem P, the objective of the assignment is to find a feasible channel assignment F that has the minimum span while satisfying all interference constraints. Like the broadcast scheduling problem, the minimum span cannot be decided in a straightforward way, hence some lower bound is expected to assist the searching for the optimal assignment.

In the simplest form of the channel assignment problem where only a co-channel constraint is considered, it is shown to be equivalent to a graph coloring problem, and therefore an NP-complete optimization problem. The extension of the constraints will show that the channel assignment is NP-complete. We will prove this NP-completeness in Appendix B.

6.2 Neural Network Solutions to the Channel Assignment Problem

Since the channel assignment problem is NP-complete, we use the effective MFA approach once again. To map the problem onto a neural network, we define that

$$s_{ij} = \begin{cases} 1 & \text{if channel } j \text{ is assigned to cell } x_i, \\ 0 & \text{otherwise.} \end{cases} \tag{6.1}$$

For a channel assignment problem $\Pi_C = (X, R, C)$, assume that there are n cells and m channels (to be decided); the energy function can be formulated as follows.

$$E(S) = E_1 + E_2 = \frac{w_1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{p=1}^{m} \sum_{q=g(p-c_{ij}+1)}^{g(p+c_{ij}-1)} c_{ij} s_{ip} s_{jq} + \frac{w_2}{2} \sum_{i=1}^{n} \left(\sum_{j=1}^{m} s_{ij} - r_i \right)^2, \tag{6.2}$$

where

$$g(x) = \begin{cases} 1 & \text{if } x < 1, \\ x & \text{if } 1 \le x \le m, \\ m & \text{if } x > m. \end{cases} \tag{6.3}$$

$S = (s_{ij})$ is an $n \times m$ matrix representing a neural network, and s_{ij} expresses the status of neuron ij. The steady state of the neural network S corresponds to an assignment. The first term E_1 in Eq. (6.2) is the penalty function for constraint violations. It consists of

$$E_1 = \frac{w_1}{2} \sum_{i=1}^{n} \sum_{p=1}^{m} \sum_{q=g(p-c_{ii}+1)}^{g(p+c_{ii}-1)} c_{ii} s_{ip} s_{iq} + \frac{w_1}{2} \sum_{i=1}^{n} \sum_{j=1 j \neq i}^{n} \sum_{p=1}^{m} \sum_{q=g(p-c_{ij}+1)}^{g(p+c_{ij}-1)} c_{ij} s_{ip} s_{jq}. \tag{6.4}$$

The first term in Eq. (6.4) reflects the co-site constraint. The second term represents the co-channel and adjacent-channel constraints. For an interference-free assignment, E_1 is equal to zero. The E_2 term in Eq. (6.2) reflects the requirement constraint. When an assignment meets the requirement R, E_2 is zero. Therefore, the minimum energy of the neural network is zero. The optimal assignment will yield the minimum energy.

6.3 The Determination of the Frequency Span

For a given compatibility matrix C and the channel requirement vector R, we first need to determine the minimum frequency span (the number of required successive channels), which is denoted by m. However, as it is shown in Appendix B, the determination of the minimum span is polynomially-related to the graph coloring problem which is an NP-complete problem. Therefore, the problem itself is NP-complete. In order to apply the MFA scheme to solve the channel problem, the following two methods are used to approximately determine the minimum span. After this, the MFA algorithm is used to search for the feasible channel assignment. If no satisfiable assignments can be found, the span m is incremented by one or more, depending on how far the assignments are from the channel requirement R. The two methods are described as follows.

6.3.1 Method 1 of Determining the Frequency Span m

The output of Method 1 is denoted O_1. Usually, c_{ii}, the minimum frequency separations for any channels assigned to Cell i $(\forall i)$, are larger than c_{ij}, the frequency separations for the channels assigned to any two different cells i, j. If $c_{ii} \gg c_{ij}$, we can simply decide m by the formula

$$\begin{aligned} O_1 &= C_{ii} \cdot (r_i - 1) + 1 \\ i &= \{k : \ r_k = \max_{\forall j \neq k} r_j\} \end{aligned} \tag{6.5}$$

6.3.2 Method 2 of Determining the Frequency Span m

The output of Method 2 is denoted O_2. If c_{ii} is close to c_{ij} for all i, j, Method 1 may not work properly. The following heuristic algorithm is used to determine the minimum span m. It can provide a tighter value close to the minimum required span only when $c_{ii} = 2$ and $c_{ij} = 0$ or 1. A denotes the set of cells which have been counted in the calculation of O_2, and U denotes the set of the remaining uncounted cells.

1. $O_2 = c_{pp}(r_p - 1) + 1$; $A \leftarrow x_p$, $U \leftarrow X - x_p$, where p satisfies the inequality $c_{pp}(r_p - 1) \geq c_{ii}(r_i - 1)\ \forall x_i$; $y \leftarrow 0$ (if $y = 0$, O_2 remains unchanged in the next step; if $y = 1$, O_2 will be incremented).

2. Select $q \in U$ such that $c_{qq}(r_q - 1) \geq c_{ii}(r_i - 1)\ \forall\ x_i \in U$

   If $c_{qi} = 1\ \forall\ x_i \in A$ and $y = 1$ then $O_2 \leftarrow O_2 + c_{qq}(r_q - 1)$, $y \leftarrow 0$.

   If $c_{qi} = 1\ \forall\ x_i \in A$ and $y = 0$ then $y \leftarrow 1$.

3. $A \leftarrow A + q$, $U \leftarrow U - q$, repeat Step 2 until U is empty.

In our numerical examples, both methods are applied to computing m and they provide satisfactory results. The determination of the exact required frequency span is still an open question.

6.4 Convergence of the MFA Channel Assignment Algorithm

If we define

$$Z_m(a,b) = \left\{ l : l, a, b \in \mathbf{Z}^+, a \le l \le b \right\} \tag{6.6}$$

and

$$\overline{Z}_m(a,b) = \{ l : l \in Z_m(1,m) \text{ and } l \notin Z_m(a,b) \} \tag{6.7}$$

Then

$$\begin{aligned}
E(S) &= \frac{w_1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{p=1}^{m} \sum_{q=g(p-c_{ij}+1)}^{g(p+c_{ij}-1)} c_{ij} s_{ip} s_{jq} + \frac{w_2}{2} \sum_{i=1}^{n} (\sum_{p=1}^{m} s_{ip} - r_i)^2 \\
&= \frac{w_1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{p=1}^{m} \sum_{q=g(p-c_{ij}+1)}^{g(p+c_{ij}-1)} c_{ij} s_{ip} s_{jq} + \frac{w_2}{2} \sum_{i=1}^{n} \left[\left(\sum_{p=1}^{m} s_{ip} \right)^2 - 2 \sum_{p=1}^{m} s_{ip} \cdot r_i + r_i^2 \right] \\
&= \frac{w_1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{p=1}^{m} \sum_{q=1}^{m} c_{ij} s_{ip} s_{jq} \left(\prod_{l=\overline{Z}_m(g(p+cij-1),g(p+cij-1))} (1 - \delta_{q_l}) \right) + \\
&\quad + \frac{w_2}{2} \sum_{i=1}^{n} \left[\sum_{p=1}^{m} s_{ip}^2 + 2 \sum_{p=1}^{m-1} \sum_{r=p+1}^{m} s_{ip} s_{ir} - 2 \sum_{p=1}^{m} s_{ip} \cdot r_i + r_i^2 \right] \\
&= \frac{w_1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{p=1}^{m} \sum_{q=1}^{m} c_{ij} s_{ip} s_{jq} \left(\prod_{l=\overline{Z}_m(g(p+cij-1),g(p+cij-1))} (1 - \delta_{q_l}) \right) + \\
&\quad + \frac{w_2}{2} \sum_{i=1}^{n} \left[\sum_{j=1}^{n} \sum_{p=1}^{m} \sum_{q=1}^{m} s_{ip} s_{jq} \delta_{ji} \delta_{qp} - 2 \sum_{p=1}^{m} s_{ip} \cdot r_i + r_i^2 \right] \\
&\quad + w_2 \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{p=1}^{m} \sum_{q=1}^{m} s_{ip} s_{jq} \delta_{ji} (1 - \delta_{pm}) \left(\prod_{l \in Z_m(1,p)} (1 - \delta_{q_l}) \right) \\
&= \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{p=1}^{m} \sum_{q=1}^{m} \frac{w_1}{2} c_{ij} s_{ip} s_{jq} \left(\prod_{l=\overline{Z}_m(g(p+c_{ij}-1),g(p+c_{ij}-1))} (1 - \delta_{q_l}) \right) + \\
&\quad \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{p=1}^{m} \sum_{q=1}^{m} \left\{ \frac{w_2}{2} s_{ip} s_{jq} \delta_{ji} \delta_{qp} + w_2 s_{ip} s_{jq} \delta_{ji} (1 - \delta_{pm}) \left(\prod_{l \in Z_m(1,p)} (1 - \delta_{q_l}) \right) \right\} \\
&\quad - w_2 \sum_{i=1}^{n} \sum_{p=1}^{m} s_{ip} \cdot r_i + \frac{w_2}{2} \sum_{i=1}^{n} r_i^2 \\
&= - \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{p=1}^{m} \sum_{q=1}^{m} T_{ip,jq} \; s_{ip} s_{jq} - \sum_{i=1}^{n} \sum_{p=1}^{m} I_{ip} s_{ip} + \frac{w_2}{2} ||R||^2.
\end{aligned}$$

where

$$T_{ip,jq} = -\frac{w_1}{2}c_{ij}\left(\prod_{l=\overline{Z}_m(g(p+cij-1),g(p+cij-1))}(1-\delta_{ql})\right) - \\ -\frac{w_2}{2}\delta_{ji}\delta_{qp} - w_2\delta_{ji}(1-\delta_{pm})\left(\prod_{l\in Z_m(1,p)}(1-\delta_{ql})\right)$$

$$I_{ip} = w_2 r_i$$

$$||R||^2 = \sum_{i=1}^{n} r_i^2$$

Since $\frac{w_2}{2}||R||^2$ is a constant for a given w_2 and does not affect the iteration procedure, this term can be ignored, and the energy function can be written in the form of

$$E(S) = -\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{p=1}^{m}\sum_{q=1}^{m} T_{ip,jq}\ s_{ip}s_{jq} - \sum_{i=1}^{n}\sum_{p=1}^{m} I_{ip}s_{ip} \tag{6.8}$$

Note that

$$T_{ip,jq} = T_{jq,ip} \tag{6.9}$$

In Chapter 3, we have proved that, if the energy function can be written in a form as Eq. (3.27), the MFA iterations based on Eq. (3.29) will guarantee the convergence at each temperature. Since Eq. (6.8) is written in the form of the Hopfield energy function as defined in Eq. (3.27) and the matrix $(T_{ip,jq})$ is symmetric, the MFA approach to solve the channel assignment problem will lead the iterations to converge to the local minima at each temperature.

By taking the derivative of the energy, we have

$$\frac{\partial E(s)}{\partial s_{ip}} = w_1\sum_{k=1}^{n}\sum_{l=g(p-c_{ik}+1)}^{g(p+c_{ik}-1)} c_{ik}s_{kl} + w_2\left(\sum_{l=1}^{m} s_{il} - r_i\right). \tag{6.10}$$

Therefore, the MFA iterations are

$$v_{ij}^{(n)} = \frac{1}{2} + \frac{1}{2}\tanh\left\{-\frac{1}{2T}\left[w_1\sum_{k=1}^{n}\sum_{l=g(p-c_{ik}+1)}^{g(p+c_{ik}-1)} c_{ik}v_{kl}^{(n-1)} + w_2\sum_{l=1}^{m}\left(v_{il}^{(n-1)} - r_i\right)\right]\right\}. \tag{6.11}$$

6.5 The Determinations of MFA Parameters

We use approaches similar to those presented in the previous chapters to determine such parameters as critical temperature, annealing schedule and iteration termination criteria. Only minor changes are made in the determinations and the details are skipped in this chapter.

6.6 Numerical Examples and Results

Three instances with 5, 10 and 25 cells have been tested by the proposed MFA channel assignment algorithm. Fig. 6.2 list the compatibility matrix and the requirement vectors for all of the instances. Assume that the available channels are numbered from 1 to m. The assignments are shown in Fig. 6.3. In Fig. 6.3, Cell i represents x_i. In Fig. 6.3 (a), (b), the frequency spans m are estimated by Method 1 described in Section 6.3.1 and they are 67 and 204, respectively, which are equal to the spans of the actual assignments. In Fig. 6.3 (c), the estimation of m by Method 2 is 69 and the actual required span of the assignment is 73, so the estimate is a good one. Although the determination of m is NP-complete, the methods we proposed provide satisfactory results in our simulations. The running time for all of the three examples is only a few seconds on a Sun Sparc 20.

6.7 Summary

The channel assignment problem arises when the scarce and expensive frequency resource must be fully utilized. It is proved that the assignment problem is NP-complete. In this chapter, an MFA-based algorithm is proposed to solve the difficult optimization problem. Three interferences constraints, namely co-cell, adjacent, and co-site, are considered. The energy function can be written in the form of Hopfield net, which has been proved to guarantee the MFA iteration convergence. Two

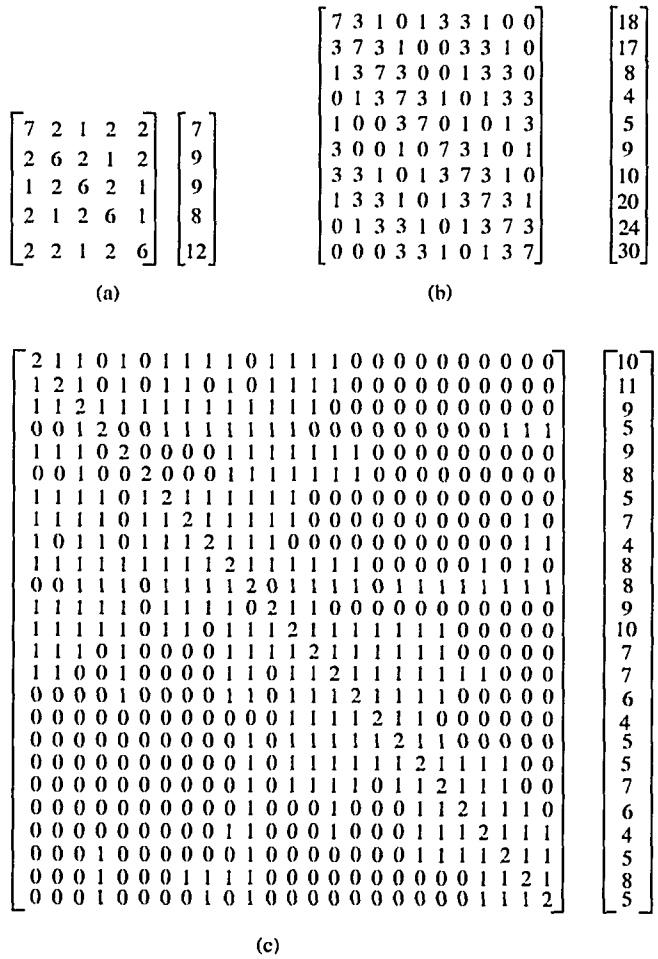


Figure 6.2: Compatibility matrix and requirement matrix. (a) C1 and R1, (b) C2 and R2, (c) C3 and R3

methods are presented to estimate the minimum span and they provide satisfactory results. The related parameters encountered in the MFA iterations are determined in the similar ways similar to those in the previous chapters. Three instances are tested by the algorithm and it provides feasible solutions for all of instances.

cell	channel number
1	3 10 17 27 34 41 51
2	5 15 21 29 39 45 53 59 65
3	2 8 18 24 32 42 48 56 62
4	12 22 30 36 44 54 60 66
5	1 7 13 19 25 31 37 43 49 55 61 67

(a)

cell	channel number
1	66 73 80 87 94 101 108 115 122 129 138 147 154 161 168 175 183 190 198
2	76 83 90 97 104 111 118 125 144 151 158 165 172 180 187 194 201
3	141 148 155 162 169 176 197 204
4	179 186 193 200
5	4 11 18 25 32
6	3 10 17 24 31 38 45 52 59
7	6 13 20 27 34 41 48 55 62 69
8	2 9 16 23 30 37 44 51 58 65 72 79 86 93 100 107 114 121 128 135
9	5 12 19 26 33 40 47 54 61 75 82 89 96 103 110 117 124 131 138 145 152 159 166 173
10	1 8 15 22 29 36 43 50 57 64 71 78 85 92 99 106 113 120 127 134 141 148 155 162 169 176 183 190 197 204

(b)

cell	channel number	cell	channel number
1	1 9 17 25 33 41 49 57 64 70	14	8 16 24 32 40 48 56
2	2 10 18 26 34 42 50 58 65 71 73	15	3 11 19 27 35 43 51
3	3 11 19 27 35 43 51 59 66	16	26 34 42 50 58 66
4	1 9 17 25 33	17	2 10 18 41
5	4 12 20 28 36 44 52 60 67	18	20 25 28 33 36
6	4 12 20 28 41 49 57 64	19	1 4 6 9 12
7	36 44 52 60 67	20	5 13 17 21 29 37 45
8	8 16 24 32 40 48 56	21	53 55 58 60 62 65
9	26 34 42 50	22	30 68 71 73
10	5 13 21 29 37 45 53 61	23	3 11 19 27 35
11	2 10 18 41 49 57 64 70	24	38 43 51 59 65 67 69 72
12	6 14 22 30 38 46 54 62 68	25	4 6 12 14 22
13	7 15 23 31 39 47 55 63 69 72		

(c)

Figure 6.3 Channel assignments (a) $F1$, (b) $F2$, (c) $F3$.

CHAPTER 7

CONCLUSIONS

In this dissertation, mean field annealing theory is applied to solve difficult combinatorial optimization problems encountered in multiaccess communication systems. Three problems, namely optimal frame pattern search, broadcast scheduling and optimal channel assignment, are thoroughly investigated. The main conclusions are as follows:

1. In order to meet the increasing demand for telecommunication services, the limited frequency spectrum resources must be efficiently utilized. Multiaccess protocols have been proposed in order to reuse the scarce and expensive channel resource in frequency, time and space domains. It has been shown that most optimization problems encountered in the multiaccess systems are NP-complete, and so polynomial algorithms for finding solutions are unlikely to exist.

2. In solving the combinatorial optimization problems, heuristic algorithms usually provide fast search procedures for finding solutions. However, as the problem size increases, there will be many local extrema (maxima or minima), and heuristic algorithms may get stuck in local optima which may be far away from the global optima. Also, heuristic algorithms are usually problem-dependant, that is, one efficient heuristic algorithm that efficiently solves optimization problem may not be applicable or effective for another problem. In contrast, the simulated annealing approach usually yields good approximate global optima for a wide range of problems. The annealing process allows the system search to occasionally jump out from local minima while keeping the local search behavior at each temperature. As the

temperature decreases, the probability for the system to stay in the global optima approaches one. However, the iterations in reaching global optima in SA are very time-consuming. Usually, the efficiency of an algorithm is evaluated by the computational complexity and the solutions quality, where solutions quality measures how close the solutions found are to the global ones. Mean field annealing, which uses the approximation technique, is a good trade-off between the computational complexity and the solutions quality. Instead of the stochastic searches in SA, MFA finds the thermal averages at each temperature by a sequence of deterministic equations, leading to fast convergence to the thermal equilibrium at each temperature. On the other hand, the analogous annealing process is applied to MFA to avoid its getting stuck in local optima.

3. The convergence and stability of the MFA approach are discussed in this dissertation. It has been shown that MFA uses saddle point approximation to calculate the partition function, which results in the MFA iteration equations updating the thermal averages of neurons. The MFA iteration process is proved to be able to quickly reach thermal equilibriums at each temperature and finally to reach steady states near optima.

4. In this disertation, MFA is applied to the optimal frame pattern design in an integrated TDMA communication system in order to provide the maximum data throughput. It is shown that different frame patterns may yield different data throughputs. The number of frame patterns dramatically increase as the problem size increases, and the complexity becomes intractable in searching for the optimal frame patterns which provide maximum data throughputs in the frame pattern space. Numerical examples have shown that the proposed MFA-based searching algorithm is efficient in both solutions quality and computational complexity. As compared with the random search algorithm, it provides

much better solutions with comparable complexity. In comparison with the SA approach, its computational complexity is one order of magnitude less and the solutions are quite close to the SA's. This implies that the MFA approach can reach near-optimal solutions with acceptable complexity.

5. An MFA-based algorithm is proposed to find the broadcast schedules in a packet radio network. It has been proved that the problem of searching for the optimal interference-free transmission schedules is NP-complete, so efficient polynomial algorithms rarely exists. Two scheduling approaches are investigated. The channel utilization and time delay performance are evaluated and compared with one existing heuristic algorithm. The MFA provides better performance than the existing algorithm.

6. Cellular networks offer challenging and attractive research areas in a variety of aspects. Channel assignment in the frequency domain is one of the practical optimization problems. It has been proved that the channel assignment problem is NP-complete when three types of interferences, namely co-channel, adjacent and co-site, are considered. It is also shown that it is NP-complete even for the simple case in which only a co-channel constraint is considered. The feasible assignments provide efficient use of the spectrum resource and meet the increasing service requirement for communications. In this dissertation, the proposed MFA algorithm succesfully solved the assignment problem.

7. The convergence speed and the solutions quality strongly depend on the selection of parameters such as weights, annealing schedule, critical temperature, and criteria of iteration termination. The determination of the related parameters is discussed for the different problems. The selections derived from the determination criteria provide satisfactory results in all of the numerical and simulation results presented in this dissertation.

Our major contributions of this dissertation are pointed out in (3)-(7) above. Our primitive targets are directing to solve the difficult optimization problems in multiaccess communication systems. Our motivation to use MFA approaches comes from the its applications [31]-[35]. However, after we recast our optimization problems into the structures required by MFA, we have found that there are many variables that may affect the MFA applicability and efficiency. The determinations of the parameters are still open questions. Therefore, we have worked on the open questions and attempted to find a general way to determine the related parameters. Our derivations and numerical tests have shown satisfactory results in searching for the optimal solutions and reducing computational complexity. Hopefully, more applications by our approach can be found in the near future.

APPENDIX A

NP-COMPLEXNESS OF THE BROADCAST SCHEDULING PROBLEMS

We denote the two approaches of broadcast scheduling problems described in Chapter 4 as Π_{B1} and Π_{B2}. Π_{B1} is the scheduling problem that, given a packet radio network, finds an interference-free frame transmission schedule with the maximum throughput under a fixed frame length N, where N is the number of stations in the packet radio network. Π_{B2} is defined as finding an interference-free transmission schedule with the minimum frame length. The NP-completeness of the two problems are proved individually in the following sections.

A.1 NP-completeness of the Broadcast Scheduling Problem Π_{B1}

INSTANCE: The scheduling problem $\Pi_{B1} = [G, K]$, where $G = (V, E)$ represents a packet radio network and K is a positive integer ($K \leq |V|$).

QUESTION: Is there an interference-free schedule which provides the average number of transmissions $\gamma_{av} \geq K$?

where

$$\gamma_{av} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} s_{ij},$$

and

$$s_{ij} = \begin{cases} 1, & \text{if station } j \text{ transmits at time slot } i, \\ 0 & \text{otherwise.} \end{cases} \tag{A.1}$$

Using graph theory to assist the proof of NP-completeness of the scheduling problem, we form an augmented graph $G_a = (V_a, E_a)$ in such a way that

$$V_a = V,$$

$$E_a = E \bigcup_{\substack{(i,j) \in E \\ (j,k) \in E \\ (i,k) \notin E}} (i,k).$$

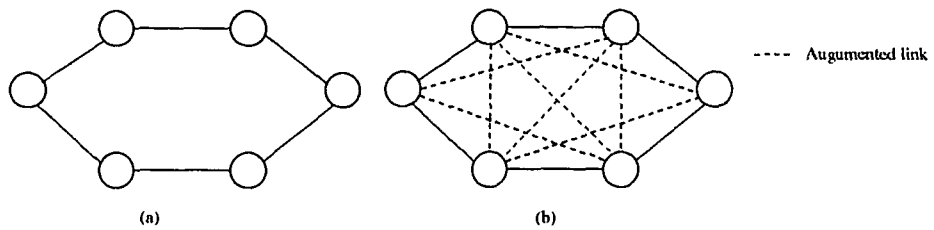


Figure A.1: The derivation of an augmented graph from a graph G. (a) Graph G, (b) Graph G_a

Fig. A.1 illustrates the augmented graph G_a derived from a graph G. For problem Π_{B1}, finding an interference-free transmission schedule in any time slot within a frame is equivalent to finding a set of vertices $V' \subseteq V$ in G_a such that, if any $i, j \in V'$, then $(i,j) \notin E_a$. All stations (vertices) in the set V' can transmit simultaneously with no interference.

In graph theory, a subset $V' \subseteq V$ is called an *independent set* in a graph $G = (V, E)$ if, for all $i, j \subseteq V'$, the edge $(i,j) \notin E$. Vertices are said to be *independent* from each other in the subset V'. Therefore, the problem Π_{B1} is identical to finding a maximum independent set $V' \subseteq V$ in graph G_a. We denote an independent set problem as Π_{IND} and use $a \Leftrightarrow b$ to represent equivalence between a and b, then we have $\Pi_{B1} \Leftrightarrow \Pi_{IND}$

Theorem A.1 Π_{B1} *is NP-complete.*

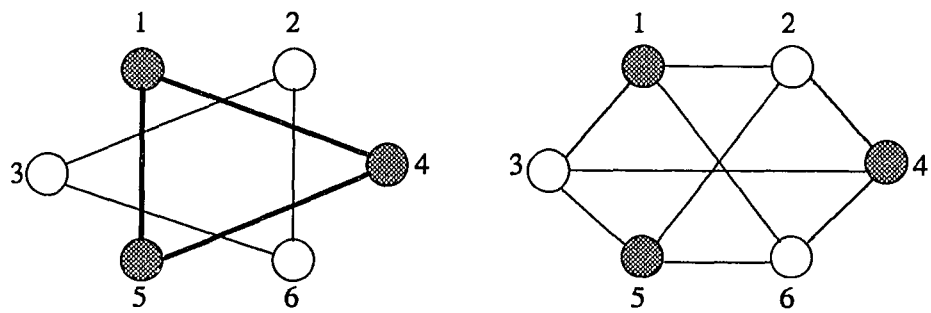
Proof: To prove Π_{B1} is NP-complete, we first need to show $\Pi_{B1} \in NP$, and then give a polynomial transformation from the known NP-complete CLIQUE problem.

1. $\Pi_{B1} \in NP$.

   For a given graph $G = (V, E)$, we can arbitrarily *guess* a schedule and use it as a certificate of G. Equivalently a subset $V' \subseteq V$ of vertices is used as a certificate of G. Here a vertex $j \in V'$ if $s_{ij} = 1$ for the i-th slot in the schedule. An augmented graph G_a is derived from G, which can be formed in polynomial time $O(N^2)$, where $N = |V|$. Checking whether V' is an interference-free set of vertices can be proceed by checking whether, for every $i, j \in V'$, the edge $(i, j) \in E_a$. This check performs up to $N(N-1)$ operations, which has polynomial time complexity. Therefore, $\Pi_{B1} \in NP$.

2. Polynomial Transformation CLIQUE $\propto_p \Pi_{IND}$

   We prove that Π_{B1} is NP-hard by showing that CLIQUE $\propto_p \Pi_{IND}$. This transformation is based on the notion of the *complement* of a graph. For an undirected graph $G = (V, E)$, the complement of G is defined as $G^c = (V, E^c)$, where $E^c = \{(i, j) : (i, j) \notin E\}$. Fig. A.2 shows a graph and its complement and the transformation from CLIQUE to Π_{IND}.



(a) Clique {1,4,5} (b) Independent set {1,4,5}

Figure A.2 Illustration of the complement of a graph (a) Graph G, (b) Graph G^c

The transformation takes an instance of the clique problem $[G, K]$ as *input*. It computes the complement G^c, which is easily derived in polynomial

time $O(N^2)$. The *output* of the transformation algorithm is the instance $[G^c, K]$ of the problem Π_{IND}. To show this transformation is valid, we need to prove that the graph G has a clique of size K if and only if the graph G^c has an independent set of size K.

Suppose that G has a clique $V' \subseteq V$ with $|V'| = K$, we claim that V' is an independent set in G^c. Based on the formation of G^c, if any $i, j \in V'$ and $(i, j) \in E$, we have $(i, j) \notin E^c$, which implies that vertices i, j are independent in G^c. Since (i, j) is chosen arbitrarily from E and every edge $(i, j) \in E$ is not in E^c, therefore all vertices $K = |V'|$ in a clique V' are independent in E^c. Hence, the set V' of size K forms an independent set in G^c. Conversely, suppose that G^c has an independent set $V' \subseteq V$ of size $K = |V'|$, then, for all $i, j \in V'$, we have $(i, j) \notin E^c$, therefore, $(i, j) \in E$ in G, which forms a complete subgraph of $V' \in V$ in E. In other words, V' is a clique with size K in G.

From above statement, we conclude that a graph G has a clique V' if and only if the graph G^c has an independent set V'. The derivation of G^c from G can be completed in polynomial time, therefore CLIQUE $\propto_p \Pi_{IND}$. Equivalently, CLIQUE $\propto_p \Pi_{B1}$ and Π_{B1} is NP-hard.

Since we have shown that $\Pi_{B1} \in NP$ and Π_{B1} is NP-hard, therefore we have proved that Π_{B1} is NP-complete.

A.2 NP-completeness of the Broadcast Scheduling Problem Π_{B2}

The broadcast scheduling problem Π_{B2} can be described as a decision problem:

INSTANCE: The scheduling problem $\Pi_{B2} = (G, K)$, where $G = (V, E)$ represents a packet radio network. K is a positive integer ($K \leq |V|$).

QUESTION: Is there a schedule which has the frame length K or less and each station transmits at least once in such a frame ?

Theorem A.2 Π_{B2} *is NP-complete.*

Proof: In the same way as for Π_{B1}, we first form an augmented graph $G_a = (V, E_a)$ from graph G described in the preceding section. Thus, scheduling an interference-free transmission in any time slot within a frame is equivalent to finding such a set $V' \subseteq V$ of vertices such that, if any $i, j \in V'$, then $(i, j) \notin E_a$. To prove Π_{B2} is NP-complete, we first need to show $\Pi_{B2} \in$ NP, and then give a polynomial transformation from a known NP-complete problem.

1. $\Pi_{B2} \in$ NP

   To prove $\Pi_{B2} \in$ NP, we first need to *guess* a frame schedule with an arbitrary frame length $M \leq N$ (N is the number of stations in the packet radio network), then form the augmented graph $G_a = (V, E_a)$ from graph G at each time slot t. Denote $V_t' = \{i : i \in V, s_{ti} = 1\}$, where s_{ti} is defined as Eq. (A.1). We need to check whether $(i, j) \notin E$ for any $i, j \in V_t'$. At the same time, we need to check if the guessed schedule length $M \leq K$. It is easy to verify that the checking process can be completed in polynomial time $O(N^2)$. Therefore, $\Pi_{B2} \in$ NP.

2. $\Pi_{B2} \propto_p$ CLIQUE

   Finding an interference-free schedule with the minimum frame length is equivalent to coloring vertices in G_a so that, for any pair of vertices $i, j \in V$, if $(i, j) \in E_a$, then i and j must be assigned to different colors. The coloring problem is to find the minimum number of colors that can cover all vertices in a graph. Here the color corresponds to a time slot and the frame length to the total number of colors n_c. In order to find the minimum n_c, we should assign a color to as many vertices as possible provided the coloring does not cause conflict. Therefore, solving the problem Π_{B2} can be performed as follows:

   a) Form an augmented graph G_a based on G, $t \leftarrow 1$.

b) Find the maximum independent set V_t' for $G_a = (V, E_a)$ and assign a color c_t to the set V_t'.

c) $V \Leftarrow V - V_t'$, $E_a \Leftarrow E_a - \{(i,j) : i \in V', (i,j) \in E \ \ \forall j\}$ and ,

if G_a is not empty, then $t \leftarrow t + 1$ and repeat Step b), else the coloring procedure ends, the results are encoded into a frame schedule by

$$s_{tj} = \begin{cases} 1, & \text{if } j \in V_t', \\ 0 & \text{otherwise.} \end{cases} \tag{A.2}$$

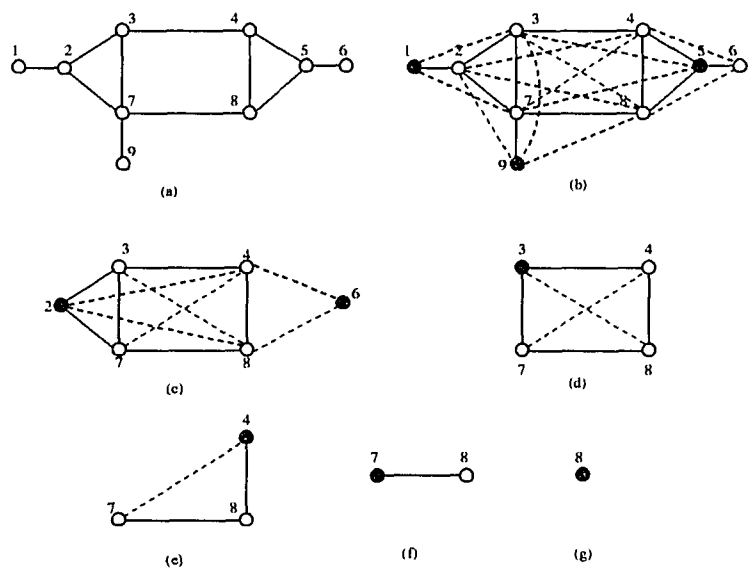


Figure A.3 The decomposition of the scheduling problem

Fig. A.3 illustrates the formulation of the graph G_a and the coloring process. We have $V_1' = \{1,5,9\}$, $V_2' = \{2,6\}$, $V_3' = \{3\}$,$V_4' = \{4\}$, $V_4' = \{7\}$, $V_6' = \{8\}$. Fig. A.4 is the decoding of the graph coloring result.

We notice that the coloring problem is equivalent to finding the maximum independent set in G_a. Therefore, the broadcast scheduling problem Π_{B2} can be transformed into the independent set problem Π_{IND} by finding an augmented graph G_a from G, which is easily generated in polynomial time. Thus, we have $\Pi_{IND} \propto_p$

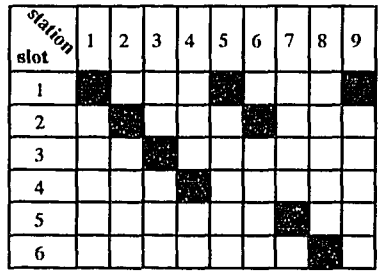


Figure A.4 The decoding of the coloring graph.

Π_{B2}. We have proved that $CLIQUE \propto_p \Pi_{IND}$, therefore, CLIQUE $\propto_p \Pi_{B2}$. Since the CLIQUE problem is NP-complete, we can conclude that Π_{B2} is NP-hard.

As we proved $\Pi_{B2} \in NP$ and Π_{B2} is NP-hard, we can conclude that Π_{B2} is NP-complete.

APPENDIX B

NP-COMPLETENESS OF THE CHANNEL ASSIGNMENT PROBLEM Π_C

As addressed in Chapter 6, the channel assignment problem, Π_C, is to assign channels to cell sites while satisfying channel constraints described by the compatibility matrix $C = (c_{ij})$. In the FDMA cellular communication systems, the frequency band is subdivided into a certain number of narrowband channels, each capable of supporting one phone circuit that can be accessed by any user. The channels are labeled as a sequence of numbers $\{1, 2, \cdots, N\}$. Channel i and Channel $i+1$ are called *adjacent*. The *frequency distance* between channel i and j is $|i-j|$. c_{ij} represents the channel constraint that the frequency distance $\triangle_{ij}$ between the channels assigned to cell i and cell j must be greater than or equal to c_{ij}. Each diagonal element c_{ii} in C represents the minimum separation distance between any two channels assigned to cell i. We use the vector $X = (x_1, x_2, \cdots, x_n)$ to represent the n cell sites for a given cellular communication system. $R = (r_1, r_2, \cdots, r_n)$ is the requirement vector, where r_i represents that cell x_i is requesting for r_i channels. $F = (F_1, F_2, \cdots, F_n)$ is a feasible channel assignment, where $F_i = (f_{i1}, f_{i2}, \cdots, f_{i,r_i})$, and f_{ik} is the channel number assigned to cell x_i. For example, if three channels numbered 2, 5, 7 are assigned to cell x_i and $r_i = 3$, then $F_i = (2, 5, 7)$. Thus, the problem Π_C can be defined as a decision problem:

INSTANCE: $\Pi_C = [X, R, C, K]$, where K is a positive integer.

QUESTION: Is there a feasible channel assignment vector F such that the frequency span is k or less ? Here the *frequency span* is defined as the frequency distance between the largest channel number and the smallest channel number assigned to cells.

We group Π_C into two sub-categories: *cochannel* and *cosite*. Those where only cochannel constraints occur are called cochannel cases. Those where all channel constraints - cochannel, adjacent channel and cosite, occur, are called cosite cases. The channel assignment problems for both cochannel and cosite cases are denoted as Π_{C1} and Π_{C2}.

Theorem B.1 Π_{C1} *is NP-complete.*

Proof: To prove Π_{C1} is NP-complete, we first need to show $\Pi_{C1} \in NP$, then show that Π_{C1} is NP-hard by finding a polynomial transformation from a known NP-complete problem.

1. $\Pi_{C1} \in NP$

   For a given instance $\Pi_{C1}= [X, R, C]$, we can check whether a given assignment is feasible in polynomial time. The checking procedure is as follows:

   For the assignment F, check whether the frequency distance $|f_{ik} - f_{jl}| \geq c_{ij}$ for $i, j = 1, 2, \cdots, n$, $k = 1, 2, \cdots, r_i$ and $l = 1, 2, \cdots, r_j$. Then we check if the frequency span is K or less . Obviously, the checking operations can be completed in polynomial time. Therefore, $\Pi_{C1} \in NP$.

   Let Π_{CLIQUE} denote the CLIQUE problem. We prove Π_{C1} is NP-hard by showing that $\Pi_{C1} \Leftrightarrow \Pi_{IND}$ and $\Pi_{CLIQUE} \propto_p \Pi_{IND}$, where $a \Leftrightarrow b$ means that solving problem a is equivalent to solving problem b.

2. $\Pi_{CLIQUE} \propto_p \Pi_{C1}$

   In the problem Π_{C1}, only cochannel constraints are considered. Therefore, $c_{ii} = 1, c_{ij} \in \{0, 1\}$ for all i, j $(i \neq j)$. We can use a graph $G = (V, E)$ to represent Π_{C1}. Vertex $v \in V$ represents a *requirement* and a link (u,v) connects pairs of requirements u, v that cannot be assigned the same channel;

i.e., $(u,v) \in E$ if $c_{uv} = 1$ and $(u,v) \notin E$ if $c_{uv} = 0$. For example, a given instance of Π_{C1} is

$$X = \{x_1, x_2, x_3, x_4, x_5\}$$

$$c = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \qquad R = \begin{bmatrix} 2 \\ 1 \\ 1 \\ 2 \\ 1 \end{bmatrix}$$

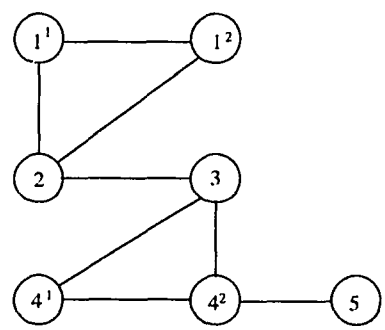


Figure B.1 The construction of the graph

where i^j stands for the j-th requirement by cell x_i. The derived graph is shown in Fig. B.1. In the derived graph G, every connected pair of vertices cannnot be assigned the same channel. The channel assignment problem Π_{C1} is to find the minimum number of channels which satisfy the cochannel constraints and the cells' requirements. By observation, we find that Π_{C1} is equivalent to the graph coloring problem. In a graph coloring problem, inter-connected vertices may not have the same color. The objective is to find the minimum number of colors required to color all vertices. In Π_{C1}, the vertices represent requirements and the colors represent channels. Obviously, if we use one color to cover as many vertices as possible, the number of required colors will be minimum. Since the unconnected vertices can use the same color, the graph coloring problem is transformed into the independent set problem Π_{IND}; i.e., finding the minimum

number of colors is equivalent to the problem Π_{IND} by simply finding the maximum independent set for the graph. Then the independent set V' and the links attatched to the vertices in V' are removed. For the reduced graph, repeat the search procedure for a new maximum indepent set until the graph is reduced to *empty*. We have shown that $\Pi_{CLIQUE} \propto_p \Pi_{IND}$ and $\Pi_{C1} \Leftrightarrow \Pi_{IND}$, therefore we prove that $\Pi_{CLIQUE} \propto_p \Pi_{C1}$.

Since $\Pi_{C1} \in NP$ and $\Pi_{CLIQUE} \propto_p \Pi_{C1}$, we can conclude that Π_{C1} is NP-complete.

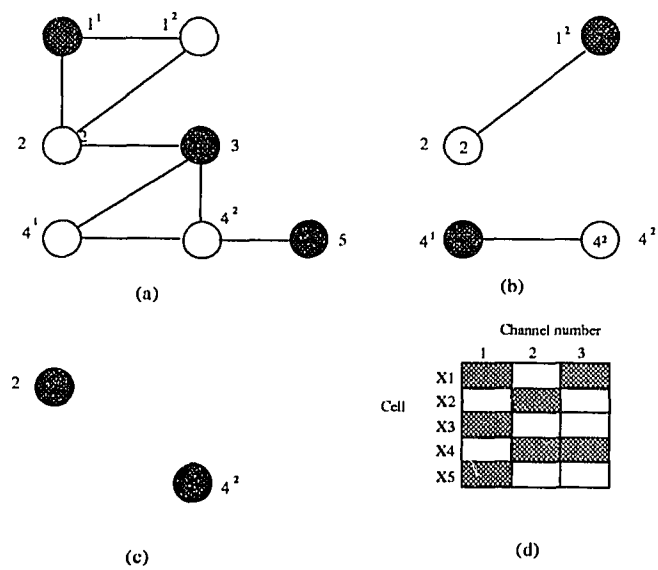


Figure B.2 The decompostions of graph (a)-(c). (d) the the channel assignment.

In the example shown in Fig. B.1 , the graph can be decomposed into a bunch of subgraphs. In each subgraph, the maximum independent set is found. The decomposition and coloring procedure are illustrated in Fig. B.2 (a)-(c). The shaded circles (cells) will be assigned to use the same channel. The feasible channel assignment is shown is Fig. B.2 (d).

Theorem B.2 Π_{C2} *is NP-complete.*

Proof:

1. $\Pi_{C2} \in NP$.

   We can show that $\Pi_{C2} \in NP$ by performing the same checking procedure as in the proof of $\Pi_{C1} \in NP$. Therefore, $\Pi_{C2} \in NP$.

2. $\Pi_{C1} \propto_p \Pi_{C2}$

   Once we have proved that Π_{C1} is NP-complete, the NP-complete proof of Π_{C2} is relatively easy. Since we have shown $\Pi_{C2} \in NP$, what we still need to prove is that there is a transformation $\Pi_{C1} \propto_p \Pi_{C2}$.

   All of the constraints in Π_{C2} can be described by the compatability matrix $C = (c_{ij})$ with $c_{ij} \in \mathbf{Z}^+$. Π_{C1} is just the subset of Π_{C2} with $c_{ij} \in \{0, 1\}$. Therefore, every instance of Π_{C1} can be directly mapped to an instance of Π_{C2}. i.e., every instance $I \in \Pi_{C1}$ if and only if $f(I) \in \Pi_{C2}$. Where $f(I) = I$. Therefore, we have $\Pi_{C1} \propto_p \Pi_{C2}$.

Since Π_{C1} is NP-complete and $\Pi_{C2} \in NP$ and $\Pi_{C1} \propto_p \Pi_{C2}$, we can conclude that Π_{C2} is NP-complete.

REFERENCES

1. E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machine — A Stochastic Approach to Combinatorial Optimization and Neural Computing,* New York: John Wiley & Sons, 1989.

2. N. Abramson, "The ALOHA system — another alternative for computer communication," in *AFIPS Conf. Proc.*, vol. 37, FJCC, pp. 281-285, 1970.

3. D. Akerberg, "Properties of a TDMA pico cellular office communication system," in *Proc. IEEE GLOBECOM,* Hollywood, FL, 1988, pp. 1343-1349.

4. N. Ansari, R. Sarasa, and G. Wang, "An efficient annealing algorithm for global optimization in Boltzmann machines," *Journal of Applied Intelligence 3,* pp. 177-192, 1993.

5. N. Ansari and G. Wang, "Neural networks for broadcast scheduling in telecommunications," *Fuzzy Logic and Neural Network Handbook* (C. H. Chen ed.), McGraw-Hill, New York, 1995.

6. E. Arikan, "Some complexity results about packet radio networks," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 681-685, July 1984.

7. D. Bertsekas and R. Gallager, *Data Networks,* New Jersey: Prentice-Hall, Inc., 1987.

8. G. Bilbro et al, "Optimization by Mean Field Annealing," Optimization by Mean Field Annealing," in *Advances in Neural Information Processing Systems*, pp. 91-98, Morgan-Korgan-Kaufmann, 1989.

9. D. E. V. D. Bout and T. K. Miller, "Graph Partitioning Using Annealed Neural Networks," *IEEE Trans. on Neural Networks,* vol. 1, no. 2, pp. 192-203, June 1990.

10. C. Chang and C. Wu, "Optimal frame pattern design for a TDMA mobile communication system using a simulated annealing algorithm," *IEEE Trans. on Vehicular Technology,* vol. 42, pp. 205-211, May 1993.

11. D. C. Cox, "Universal digital portable radio communications," *Proc. IEEE,* vol. 75, pp. 436-477, Apr. 1987.

12. V. H. M. Donald, "The cellular concept," *Bell Syst. Tech. J.*, vol. 58, no. 1, Jan. 1979.

13. A. Ephremides and T. V. Truong, "Scheduling broadcasts in a multihop radio network," *IEEE Trans. on Communications*, COM-38, pp. 456-460, April 1990.

14. M. J. Fischer and T. C. Harris, "A model for evaluating the performance of an integrated circuit- and packet-switched multiplex structure," *IEEE Trans. Communications*, vol. COM-24, pp.195-202, Feb. 1976.

15. H. Frank, I. Gitman, and R. VanSlyke, "Packet radio system—network considerations," AFPS Conf. Proc., pp. 217-231, May 1975.

16. M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, New York: W. H. Freeman, 1979.

17. W. K. Hale, "Frequency assignment: Theory and applications, " *Proc. IEEE*, vol 68, pp.1497-1514, Dec. 1980.

18. E. A. Harrington, "Voice/data integration using circuit switched networks," *IEEE Trans. Communications*, vol. COM-28, pp. 781-793, June 1980.

19. J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," *Biological Cybernetics*, vol. 52, no. 4, pp. 141-156, 1985.

20. —, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Science*, vol. 79, pp. 2541-2554, 1982.

21. —, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proceedings of the National Academy of Science*, vol. 81, pp. 3088-3092, May 1984.

22. I. M. Jacobs, R. Binder, and E. V. Hoversten, "General purpose packet satellite networks," *Proc. IEEE*, vol. 66, Nov. 1978.

23. R. Kahn, S. Gronemeyer, J. Burchfiel, and R. Kunzeman, "Advances in packet radio technology," *Proc. IEEE*, vol. 66, no. 11, pp. 1468-1496, Nov. 1978.

24. K. Kinoshita, M. Hata, and K. Hirade, "Digital mobile telephone system using TD/FDMA scheme," *IEEE Trans. Vehicul. Tech.*, vol. VT-31, pp. 153-157.

25. S. Kirkpatrick, "Optimization by Simulated Annealing," pp. 671-680, *Science*, 220, 1983.

26. L. Kleinrock and F. A. Tobagi, "Packet switching in radio channels: part I—carrier sense multiple access modes and their throughput delay characteristics," *IEEE Trans. Communications,* COM-23, pp. 1400-1416, Dec. 1975.

27. W. C. Y. Lee, *Mobile Cellular Telecommunications Systems,* New York: McGraw-Hill, 1989.

28. R. M. Metcalfe and D. R. Boggs, "ETHERNET: Distributed packet switching for local computer networks," *Communi Ass. Comput. Mach.,* vol. 19, pp. 395-403, 1976.

29. N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller and E. Teller, "Equation of State Calculations by Fast Computing Machines," pp. 1087-1092, *J. of Chem. Physics,* 21, 1953.

30. W. H. Morris and S. Smale, *Differential Equations, Dynamical Systems, and Linear Algebra,* Academic Press, 1974.

31. C. Peterson, "A mean field theory learning algorithm for neural networks," *Complex System,* vol. 1, pp. 995-1019, 1987.

32. —, "Applications of mean field theory neural networks," *Dept. of Theoretical Physics, Technical Report CS-1153,* Univ. of Lund, pp. 1-27, August 1989.

33. C. Peterson and J. Anderson, "Neural networks and NP-complete optimization problems: a performance study on the graph bisection problem," *Complex Systems,* vol. 2, pp. 59-71, 1988.

34. C. Peterson and E. Hartman, "Explorations of the mean field theory learning algorithm," *Neural Networks,* vol. 2, pp. 475-494, 1989.

35. C. Peterson and B. Soderberg, "A new method for mapping optimization problems onto neural network," *International Journal of Neural Systems,* vol. 1, no. 1, pp. 3-22, 1989.

36. A. C. Stocker, "Small-cell mobile phone system," *IEEE Trans. Veh. Technol.,* vol. VT-33, no. 4, Nov. 1984.

37. F. A. Tobagi, "Multiaccess protocols in packet communication systems," *IEEE Trans. Communications,* vol. COM-28, pp. 468-488, April 1980.

38. F. A. Tobagi, R. Binder and B. Leiner, "Packet radio and satellite networks," *IEEE Communcations Magazine,* vol. 22, pp. 24-40, Nov. 1984.

39. F. A. Tobagi and L. Kleinrock, "Packet switching in radio channels: part II—the hidden terminal problem in carrier sense multiple access and the busy tone solution," *IEEE Trans. Communications,* COM-23, pp. 1417-1433, Dec. 1975.

40. G. Wang and N. Ansari, "Maximizing data throughput in an integrated TDMA communication system using mean field annealing," *IEEE GLOBECOM'94*, San Francisco, Dec. 1994.

41. G. Wang and N. Ansari, "A neural network approach to broadcast scheduling in multihop radio networks, " *Proc. of ICNN'94*, pp. 4699-4704, June 1994.

42. G. Wu and J. W. Mark, "Capacity allocation for integrated voice/data transmission at a packet switched TDM," *IEEE Trans. Communications*, vol. COM-40, pp. 1059-1069, June 1992.

43. G. Ho. Wannier, *Statistical Physis,* Ch. 4, New York: John Wiley & Sons, 1966.