ABSTRACT

A New Method to Optimize the Satellite Broadcasting Schedules Using the Mean Field Annealing of A Neural Network

by
Youyi Yu

This thesis reports a new method for optimizing satellite broadcasting schedules based on the Hopfield neural model in combination with the mean field annealing theory. A clamping technique is used with an associative matrix, thus reducing the dimensions of the solution space. A formula for estimating the critical temperature for the mean field annealing procedure is derived, hence enabling the updating of the mean field theory equations to be more economical. Several factors on the numerical implementation of the mean field equations using a straightforward iteration method that may cause divergence are discussed; methods to avoid this kind of divergence are also proposed. Excellent results are consistently found for problems of various sizes.

A NEW METHOD TO OPTIMIZE THE SATELLITE BROADCASTING SCHEDULES USING THE MEAN FIELD ANNEALING OF A NEURAL NETWORK

by
Youyi Yu

A Thesis
Submitted to the Faculty of New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science
Department of Electrical and Computer Engineering
May 1992

APPROVAL PAGE

A New Method to Optimize the Satellite Broadcasting Schedules Using the Mean Field Annealing of A Neural Network

by
Youyi Yu

5/11/92

Dr. Nirwan Ansari, Thesis Adviser
Assistant Professor of the Department of Electrical and Computer Engineering, NJIT

5/11/92

Dr. Edwin Hou, Thesis Co-adviser
Assistant Professor of the Department of Electrical and Computer Engineering, NJIT

5/11/92

Dr. Stanley Reisman, Committee Member
Associate Chairman and Professor of the Department of Electrical and Computer Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Youyi Yu

Degree: Master of Science in Electrical and Computer Engineer.

Date: May, 1992.

Date of Birth:

Place of Birth:

Undergraduate and Graduate Education:

- Master of Science in Electrical Engineering, New Jersey Institute of Technology, Newark, NJ,1992.
- Bachelor of Science in Electrical Engineering, Shanghai Science and Technology University, Shanghai, P. R. China, 1983.

Positions Held:

- Lecturer, Department of Electronics Engineering, The Engineering College of Shanghai University, Shanghai, P. R. China,1986-90.
- Assistant Engineer, Shangahi 640 Institute, Shanghai, P. R. China, 1972-78.

ACKNOWLEDGMENTS

I would like to express my gratitude and thanks to my advisor, Dr. Nirwan Ansari, for his selection of the thesis topic, guidance and encouragement at all stages of the work.

Many thanks are due to my co-advisor, Dr. Edwin Hou; to the distinguished member of the committee, Professor and Associate Chairman of the Dept. of E. C. E., Dr. Stanley Reisman. Their suggestions have considerably improved the quality of the thesis.

Last but not least, I would like to thank my family. Many thanks also go to my mother and my wife for their deep love and encouragement.

TABLE OF CONTENTS

Page

CHAPTER 1 INTRODUCTION 1

CHAPTER 2 MAPPING THE SBS PROBLEM ONTO A NEURAL NETWORK..3

2.1 Problem Statement 3

2.2 Neuron Encoding 5

2.3 Associative Matrix A 6

2.4 Formalization of the Energy Function 7

CHAPTER 3 THE MFT FRAMEWORK FOR THE SBS PROBLEM...12

3.1 Review of MFT 12

3.2 he Mean Field Equations for the SBS Problem 13

CHAPTER 4 ALGORITHM PARAMETERS...............................18

4.1 The Lagrange Parameters — $w_0, w_1, w_2, w_3,$ 18

4.2 The Critical Temperature, T_c..21

4.3 The Annealing Schedule 28

CHAPTER 5 NUMERICAL IMPLEMENTATION AND SOLUTIONS 29

5.1 Numerical Implementation 29

5.2 Convergence..30

5.3 Solutions..34

CHAPTER 6 CONCLUSIONS..39

LIST OF REFERENCES..40

LIST OF FIGURES

Figure **Page**

Figure 2.1 An Orbiting Satellite Communication System..4

Figure 3.1 Depicting the Function of V_{ijk}..17

Figure 4.1 The Dynamics in Obtaining a Solution 22

Figure 4.2 Mapping a Point from y_2 to y_3 23

Figure 4.3 The Solutions at High and Low Temperatures.....................................24

Figure 4.4 The Two Possible Solutions of Equation 4.12..................................... 27

Figure 5.1 Slope of y_2 in the Neighborhood of the Solution is Greater Than 1.............31

Figure 5.2 Slope of y_2 in the Neighborhood of the Solution is Less Than 1.................32

Figure 5.3 The Intersection Point B is Outside the Range [0, 1]..............................33

Figure 5.4 System Energy at different Temperatures.. 37

Chapter 1
Introduction

Optimization of large connectionist problems is a long-standing topic in various disciplines, with many different approaches and applications. The problem discussed here, optimization of the broadcasting time from a set of satellites to a set of ground terminals (the satellite broadcast scheduling (*SBS*) problem), is one of these categories that must be solved for satellite communication systems. In their papers [1],[2], Bourret *et al* solved this problem by using a neural network in which neurons are connected in a three-layer model. To find the optimum, a sequential search is used. The search is controlled by a competitive activation mechanism based on a dynamic prioritization of satellites. The sequential search, which is local in scope, is also very time-consuming. In addition, two additional premises (a set of distinct priorities of satellites and a set of suitable requests which are very difficult to determine for large problems) are also needed. Therefore, alternative efficient optimization methods are explored to solve this problem.

In this thesis, a new method is presented to solve the SBS problem. The work is based on a Hopfield neural network [3] [4], where all neurons are completely connected, in combination with the mean field annealing theory (*MFT*) which was recently found to be an efficient method in solving large connectionist problems [5] [6]. The main advantage of using the MFT method lies in the fact that the search for optima is parallel in the global sense, and hence the execution time is shorter than other stochastic hill-climbing methods [7][8][9]. In contrast to the method mentioned in [1] [2] which requires the two premises mentioned above, they are not required for our method. Using our method, excellent solutions are consistently found for problems of various sizes.

Instead of using a special neuron model (*graded neuron*) [10] to reduce the solution space and to avoid a destructive redundancy, a conventional neuron model clamped by an "associative matrix" is used in this work. This clamping technique is often applied in learning algorithms [11] [5] [12], resulting in a large decrease of the solution space.

Due to the non-linearity of the sigmoid function, a so-called critical temperature T_c exists. Instead of using the "trial and error" approach to determine T_c, a formula for estimating T_c is derived. Experiments show that the estimated values using this formula are within 10% from the experimental (trial & error approach) results.

In this work , a type of divergence caused by the numerical implementation of the mean field equations is analyzed, and some schemes are suggested to avoid this kind of divergence.

This thesis is organized as follows. In Chapter 2, we briefly describe the satellite broadcasting problem, and map it onto a neural network framework. This is followed, in Chapter 3, by a brief review on some recently proposed optimization methods with emphasis on the MFT. A set of mean field equations are also derived in this chapter to solve the SBS problem. In the next chapter, the algorithm of parameters and derivation of the formula for T_c are presented. The numerical calculations and the comparison of the results are discussed in Chapter 5. Finally, conclusions are presented in Chapter 6.

Chapter 2
Mapping the SBS Problem onto a Neural Network

2.1 Problem Statement

The problem discussed here is related to a low level satellite communication system. As shown in Fig. 2.1, this system consists of a set of satellites and a set of ground terminals. Unlike the geostationary communication systems, the satellites here are usually located in a polar orbit with a rather low altitude and they always orbit around the earth. Hence, the ground terminals only need to employ low-power transmitters and portable antennas. The system can provide global communications coverage including the two polar regions which still cannot be achieved by the geostationary systems [13].

Our work is to maximize the broadcasting time for each satellite such that all the following constraints are met:

1. A satellite cannot broadcast to more than one ground terminal at a time ;
2. A ground terminal cannot receive information from more than one satellite at a time;
3. A satellite must broadcast as much as possibly close to its requested time, and the system cannot allocate more time than requested unless the requests time for the rest of the satellites are completely satisfied;
4. A satellite only broadcasts when it is visible from a ground terminal.

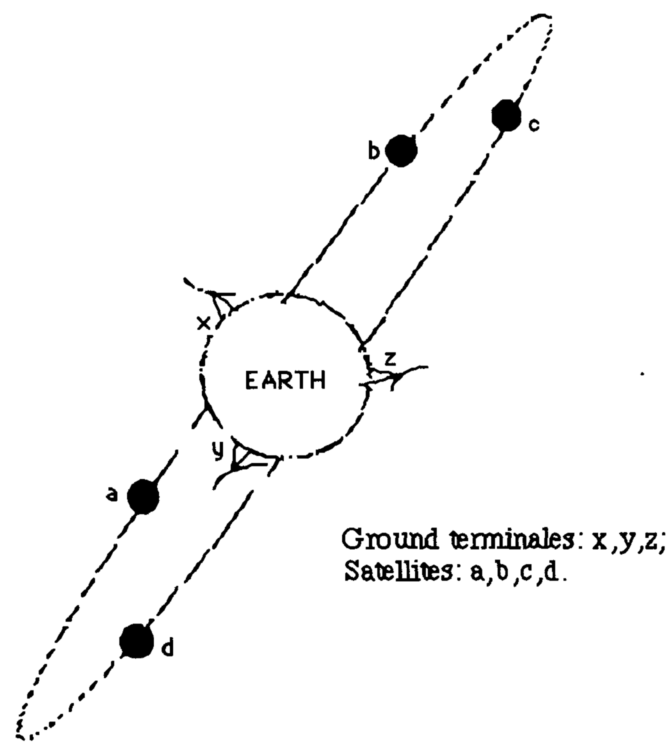

Fig. 2.1. Orbiting Satellite Communication System.

To solve this problem, we adopt the following notation.

S is the set of satellites consisting of N_S elements (satellites);

S $= \{\ a,\ b,\ c,\ d,\ \ldots\ \} = \{\ 1,\ 2,\ 3,\ \ldots,\ i,\ \ \ldots,\ N_S\}$

Here, *a, b, c, d, ...* denote the different satellites each of which can be indexed by an integer number, i, ranging from 1 to N_S.

A is the set of ground terminals consisting of N_A elements (terminals);

A $= \{\ z,\ y\ ,x\ ,w,\ \ldots\ \} = \{\ 1,\ 2,\ 3,\ \ldots,\ j,\ \ldots,\ N_A\}$

Here, *z, y ,x ,w, ...* are different ground terminals; each of which can also be indexed by an integer number, j, ranging from 1 to N_A;

T is the set of time slots consisting of N_T elements (time slots); each of which can be indexed by an integer number, k, ranging from 1 to N_T;

R is a vector denoting the set of requested number of time slots given by the problem. It consists of N_s elements (time slots).

R = $[r_1, r_2, r_3, \ldots \; r_{N_s}]^T$

Here, $r_1, r_2, r_3, \ldots \;, r_{N_s}$, are the requested time slots for satellite $1, 2, 3, \ldots, N_s$, respectively.

U is a vector denoting the set of maximum number of time slots for each satellite allocated by the system. It consists of N_s elements.

U = $[u_1, u_2, u_3, \ldots, u_{Ns}]^T$

Here $u_1, u_2, u_3, \ldots \; u_{Ns}$ are the number of time slots allocated for satellites $1, 2, 3, \ldots, N_S$, respectively.

Our goal is to find the optimal schedule satisfying the following two criteria simultaneously:

(a) The schedule must be legal, that is, all constraints are fulfilled;

(b) The distance between the vectors, **U** and **R**, must be minimized.

2.2 Neuron Encoding

In this thesis, we denote a neuron by S_{ijk}. Each neuron is turned "on" or "off " depending on Whether or not satellite i is assigned to transmit to terminal j during time slot k. Thus, S_{ijk} is mathematically defined by

$$S_{ijk} = \begin{cases} 1 & \text{if satellite } i \text{ is assigned to terminal } j \text{ during time slot } k\text{;} \\ 0 & \text{otherwise.} \end{cases} \tag{2.1}$$

2.3 Associative Matrix A

From the above definition of neurons, it is clear that some neurons are always fixed to zero because of Constraint 4 mentioned in Section 2.1. This is due to the fact that no ground terminal is visible to the satellite even when all ground terminals are idle. Usually, the number of neurons which are nulled owing to Constraint 4 is large. We should reflect this constraint into the neural network by clamping those neurons which do not meet Constraint 4 to zero throughout the optimization. To do so, we define an associative matrix A with $N_S \times N_A$ rows and N_T columns.

$$A = \begin{bmatrix} A_1 \\ \cdot \\ \cdot \\ \cdot \\ A_2 \\ \cdot \\ \cdot \\ \cdot \\ A_{N_S} \end{bmatrix}$$

$$= \begin{bmatrix} a_{111} & a_{112} & \cdots & \cdots & a_{11N_T} \\ \cdot & \cdot & \cdots & \cdots & \cdot \\ a_{1N_A1} & a_{1N_A2} & \cdots & \cdots & a_{1N_AN_T} \\ \cdot & \cdot & \cdots & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdots & \cdot \\ a_{N_SN_A1} & \cdot & \cdot\cdot & \cdot\cdot & a_{N_SN_AN_T} \end{bmatrix} \tag{2.2}$$

where A_i is the sub-matrix associated with Constraint 4 imposed on satellite i;

a_{ijk} takes zero or one according to:

$$a_{ijk} = \begin{cases} 1 & \text{if satellite } i \text{ is visible to terminal } j \text{ at time slot } k \ ; \\ 0 & \text{otherwise.} \end{cases} \tag{2.3}$$

From the definition of $\boldsymbol{A}$ and the problem constraints, two important relations are observed:

(a) The maximum number of requested time slots $r_{max(i)}$ for satellite i must be less than or equal to the number of nonzero columns of the sub-matrix $\boldsymbol{A}_i$;

$$r_{max(i)} \leq \sum_{i=1}^{N_T} (\text{"non-zero columns in } \boldsymbol{A}_i\text{"}) \tag{2.4}$$

(b) The usable time slots u_i must be less than or equal to $r_{max(i)}$.

$$u_i = \sum_{j}^{A} \sum_{k}^{T} S_{ijk} < r_{max(i)} \tag{2.5}$$

The above relations are useful since they can be used to check the illegality of a solution.

2.4 Formalization of the Energy Function

In optimization problems, one needs to formalize a particular objective function which is to be optimized. Our problem is a constrained optimization problem. In order to map a constrained optimization problem onto a Hopfield neural network [3][4], we have to embed the constraints onto one function known as the energy function which consists of two terms : the cost term and the constraint term. The cost term is the optimization cost (objective) function that is independent of the constraint term. This constraint term is the penalty imposed on for violating the constraints.

$$E = w_c \times \text{" cost "} + w_p \times \text{" penalty "} \tag{2.6}$$

where w_c and w_p are the Lagrange parameters [Lue84].

These two terms must counteract each other. In our case, the cost term is negative and the constraint term is positive. The optimization is then achieved by minimizing the energy function. Here the cost term or the energy due to the cost, E_0, is defined by

$$E_0 = -\frac{1}{2} \sum_{i}^{S} \sum_{j}^{A} \sum_{k}^{T} S_{ijk} \cdot S_{ijk} \tag{2.7}$$

which reflects the idea of maximizing the total broadcasting time. The negative sign implies that minimization is to be applied.

The following penalty terms will be defined according to the four constraints, :

(1) A satellite cannot broadcast to more than one ground terminal at a time.

The statement implies that all of the following equations must be satisfied simultaneously because they represent all possible violations.

$$\sum_{i}^{S} \sum_{k}^{T} \sum_{j}^{A} \sum_{j_1 \neq j}^{A} S_{ijk} \cdot S_{ij_1k} = 0 \tag{2.8}$$

$$\sum_{i}^{S} \sum_{k}^{T} \sum_{j}^{A} \sum_{j_1 \neq j}^{A} \sum_{j_2 \neq j_1 \neq j}^{A} S_{ijk} \cdot S_{ij_1k} \cdot S_{ij_2k} = 0 \tag{2.9}$$

.
.
.
.
.
.

$$\sum_{i}^{S}\sum_{k}^{T}\sum_{j}^{A}\sum_{j_1\neq j}^{A}\sum_{j_2\neq j_1\neq j}^{A}\cdots\sum_{j_m\neq\ldots\neq j_2\neq j_1\neq j}^{A} S_{ijk}\cdot S_{ij_1k}\cdot S_{ij_2k}\cdot\cdots\cdot S_{ij_mk} = 0 \qquad (2.10)$$

Obviously, when the total number of ground terminals increases, the number of equations required to impose this constraint increases. Fortunately, however, it can be shown that if the first equation is satisfied , the remaining equations are also satisfied simultaneously.

Lemma 1: If Equation (2.8) is satisfied, Constraint (1) is met.

Proof : Consider Equation (2.9)

$$\begin{aligned}&\sum_{i}^{S}\sum_{k}^{T}\sum_{j}^{A}\sum_{j_1\neq j}^{A}\sum_{j_2\neq j_1\neq j}^{A} S_{ijk}\cdot S_{ij_1k}\cdot S_{ij_2k}\\ &= \sum_{i}^{S}\sum_{k}^{T}\sum_{j_2\neq j_1\neq j}^{A} S_{ij_2k}\sum_{j}^{A}\sum_{j_1\neq j}^{A} S_{ijk}\cdot S_{ij_1k}\end{aligned} \qquad (2.11)$$

By definition, each neuron takes on either 1 or 0. Thus, if Equation (2.8) is true, then every term in Equation (2.8) must be zero.

$$S_{ijk}\cdot S_{ij_1k} = 0 \qquad \forall \qquad i, k, j_1 \neq j \qquad (2.12)$$

Substituting Equation (2.12) into Equation (2.9), we obtain the following:

$$\sum_{i}^{S}\sum_{k}^{T}\sum_{j}^{A}\sum_{j_1\neq j}^{A}\sum_{j_2\neq j_1\neq j}^{A} S_{ijk}\cdot S_{ij_1k}\cdot S_{ij_2k} = 0$$

By deduction, if Equation (2.8) is true, the remaining equations required to impose Constraint (1) are all satisfied simultaneously. Thus, only Equation (2.8) is needed to impose Constraint (1). Q.E.D.

Hence, the penalty term for this constraint is:

$$E_1 = \sum_{i}^{S} \sum_{k}^{T} \sum_{j}^{A} \sum_{j_1 \neq j}^{A} S_{ijk} \cdot S_{ij_1k} \tag{2.13}$$

2. A ground terminal cannot receive information from more than one satellite at a time.

   Constraint 2 is a dual to Constraint 1. This can be seen by simply replacing the *z,y,x,w,* with *a, b, c, d* in Equations (2.8) through (2.10) respectively.Thus, the penalty term for this constraint is similarly defined by:

$$E_2 = \sum_{j}^{A} \sum_{k}^{T} \sum_{i}^{S} \sum_{i_1 \neq i}^{S} S_{ijk} \cdot S_{i_1jk} \tag{2.14}$$

3. A satellite must broadcast as much as possibly close to its requested time slots, and the system cannot allocate more time than requested unless the requests for the rest of the satellites are completely satisfied;

   The first part of the statement implies that the distance between **U** and **R** should be minimized. Thus, the penalty term, E_3, corresponding to this statement is:

$$\begin{aligned} E_3 &= \sum_{i}^{S} \Big(\sum_{j}^{A} \sum_{k}^{T} S_{ijk} - r_i \Big)^2 \\ &= \sum_{i}^{S} (u_i - r_i)^2 \end{aligned} \tag{2.15}$$

   where

$$u_i = \sum_{j}^{A} \sum_{k}^{T} S_{ijk} \tag{2.16}$$

   The second part of the statement implies that $u_i - r_i \begin{cases} \geq 0 \\ \leq 0 \end{cases} \forall i$. Note that this has been incorporated in the cost term E_0.

4. A satellite broadcasts only when it is visible from a ground terminal.

This constraint is imposed by the clamping technique which will be discussed in Chapter 3. That is, neurons are forced to 0 (complied to Constraint 4) by using the associative matrix.

The total energy function for the SBS problem defined in the Hopfield framework becomes:

$$E = w_0 \cdot E_0 + w_1 \cdot E_1 + w_2 \cdot E_2 + w_3 \cdot E_3 \tag{2.17}$$

where w_0, w_1, w_2, w_3 are the Lagrange parameters used to weight the significance of E_0, E_1, E_2 and E_3, respectively.

Chapter 3
The MFT Framework for the SBS Problem

3.1 Review of MFT

In the previous chapter, the SBS problem has been mapped onto the neural network framework, and the energy function has been formalized. The remaining task is to employ a robust method to minimize the energy function. Our problem is a large scale combinatorial optimization problem in which the energy function to be optimized is a function of discrete variables. A search for the optimal configuration is computationally expensive, if not impossible.

Conventional methods such as gradient-based methods which are local in scope are not applicable. Recently, several robust methods such as simulated annealing [7] [15] and genetic algorithm [16] have been proposed to solve large scale problems. However, not every problem can be mapped onto the framework suitable for these methods. MFT[5] [6] [17] [18] is derived from the Stochasticly Simulated Annealing (*SSA*) by incorporating the SSA mechanism with the Hopfield Energy function. It has been shown to be robust in solving large scale problems, and more efficient than SSA.

The main difference between SSA and conventional methods is that SSA searches for the global minimum by using the gradient descent method in a stochastic manner. It allows, under certain conditions, the search to climb uphill, thus providing the SSA a mechanism to escape from local minima.

In SSA, there are two conceptual operations involved: a thermostatic operation which schedules the decrease of the temperature (an algorithm parameter), and a random relaxation process which search for the equilibrium solution at each temperature.

In MFT the two operations are still needed. The thermostatic operation is the same as in SSA. However, the relaxation process in searching for the equilibrium solution has been replaced by searching for the average (mean) value of the solutions. Equilibrium can be reached faster by using the mean [18], and thus the MFT speed up by several tens to hundreds times over the SSA.

The remaining question is whether the two solutions obtained from the two respective relaxations are approximately equal to each other. It has been proved by Peterson [5] [6] that for large size problems which are really what we are interested in and also, by experiments, even for small size problems, the answer is true.

3.2 Mean Field Equations for the SBS problem

In this section, we first briefly review the general mean field equations. The notations are adopted from [5] [6] [10] in which the detailed derivation can be found. The relaxation in both SSA and MFT are made according to the Boltzmann distribution [7]

$$\boldsymbol{P}\ (\mathbf{S}') = e^{-E(S')/T} / Z \tag{3.1}$$

where $\mathbf{S}'$ is any one of the possible configurations specified by the corresponding neuron set;

E($\mathbf{S}'$) is the energy of the corresponding configuration;

T is the parameter called temperature;

Z is the partition function given by

$$Z = \sum_{S'} e^{-E(S')/T} \tag{3.2}$$

and the summation covers all possible neuron configurations.

In the mean field theory, instead of concerning the neuron variables directly, we shall investigate their means (average) by defining:

$$
\begin{aligned}
V_i &= \langle S_i \rangle \\
&= 1 \cdot P_T(S_i{=}1) + 0 \cdot P_T(S_i{=}0) \\
&= P_T(S{=}1)
\end{aligned}
\tag{3.3}
$$

and $$\mathbf{V}' = \langle \mathbf{S}' \rangle \tag{3.4}$$

where S_i is a neuron;

V_i is the mean of neuron S_i;

$P_T(S_i{=}1)$ and $P_T(S_i{=}0)$ are the probabilities for $S_i{=}1$ or $S_i{=}0$, respectively;

$\mathbf{S}'$ is any one of possible configurations;

$\mathbf{V}'$ is the mean configuration corresponding to $\mathbf{S}'$.

Thus, in the mean field, Equation (3.1) becomes

$$\boldsymbol{P}\,(\mathbf{V}') = e^{-E(\mathbf{V}')/T} \,/\, Z \ , \tag{3.5}$$

and the discrete sum in Equation (3.2) can be replaced by multiple nested integrals over the continuous variables V_i and U_i [5] [6]:

$$Z = c \prod_{i=1}^{N} \int_{-\infty}^{\infty} dV_i' \int_{-j\infty}^{j\infty} dU_i' \quad e^{-F(\mathbf{V}',\mathbf{U}',T)} \tag{3.6}$$

where c is a complex constant,

$$\mathbf{U}' = \{U_1, U_2, U_3, \dots U_N\}; \tag{3.7}$$

F is called the effective energy given by

$$F(\mathbf{V}',\mathbf{U}',T) = E(\mathbf{V}')/T + \sum_{i=1}^{N} U_i\ V_i\ - log\ (\sum_{s_i = -1}^{+1} e^{S_i U_i}) \tag{3.8}$$

As has been indicated by Peterson [5] [6], by using a saddle point expansion of F, one could see that the partition function Z is actually dominated by the saddle point, i.e.,

$$Z \simeq C\ \ e^{-F(V_0', U_0', T)} \tag{3.9}$$

where C is a constant, and (V_0, U_0) is the saddle point of Equation (3.7).

Thus, the statistical mechanism of the MFT governed by Equation (3.5) are likewise determined by the mechanism of the saddle points. The saddle point can be obtained as follows:

$$\frac{\partial(F)}{\partial(U_i)} = 0 \tag{3.10}$$

$$\frac{\partial(F)}{\partial(V_i)} = 0 \tag{3.11}$$

Substituting Equation (3.8) into Equation (3.10) ,

$$\frac{\partial(F)}{\partial(U_i)} = V_i - (\sum_{S=\{0,1\}} S\ e^{S\ u_i} \Big/ \sum_s e^{S\ u_i}\ ) = 0$$

we obtain

$$V_i = (\sum_S S\ e^{S\ U_i}) \Big/ \sum_s e^{S\ u_i} \tag{3.12}$$

where s = {0,1}.

Substituting Equation (3.8) into Equation (3.11),

$$\frac{\partial(F)}{\partial(V_i)} = \frac{\partial(E)}{\partial(V_i)}\frac{1}{T} + U_i = 0,$$

we have

$$U_i = -\frac{\partial(E)}{\partial(V_i)}\frac{1}{T} \tag{3.13}$$

Equation (3.11) and Equation (3.12) are known as the general MFT equations.

For our SBS problem, replacing S_{ijk} defined in Equation (2.1) by V_{ijk} and substituting it into Equation (3.12) and then into Equation (3.13), we now get

$$V_{ijk} = \frac{1 \cdot e^{1 \cdot U_i} + 0 \cdot e^{0 \cdot U_i}}{e^{1 \cdot U_i} + e^{0 \cdot U_i}}$$

$$= \frac{e^{U_i}}{1+e^{U_i}}$$

$$= \frac{1}{2} + \frac{e^{U_i}}{1+e^{U_i}} - \frac{1}{2}$$

$$= \frac{1}{2} + \frac{2e^{U_i} - 1 - e^{U_i}}{2(1+e^{U_i})}$$

$$= \frac{1}{2} + \frac{1}{2}\frac{e^{U_i} - 1}{1+e^{U_i}}$$

$$= \frac{1}{2} + \frac{1}{2} \mathit{tanh}\,(U_i)$$

$$= 0.5 + 0.5\,\mathit{tanh}\left(-\frac{\partial(E)}{\partial(V_{ijk})}\frac{1}{2T}\right) \tag{3.14}$$

By incorporating the clamping technique discussed in Section 2.3, we obtain the MFT equations for the SBS problem as follows:

$$V_{ijk} = a_{ijk} \left(0.5 + 0.5 \tanh \left(- \frac{\partial(E)}{\partial(V_{ijk})} \frac{1}{2T} a_{ijk} \right) \right) \quad (3.15)$$

Equation (3.15) is depicted below in Fig. 3.1.

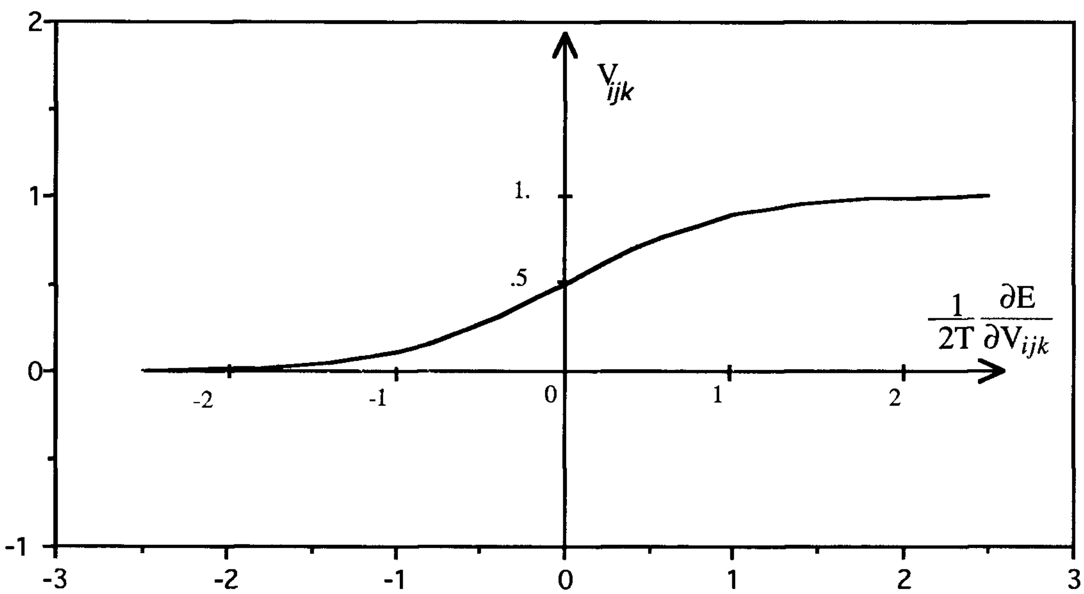

Fig. 3.1. Depicting the Function of V_{ijk}

Chapter 4
Algorithm Parameters

Before solving the MFA equations, several parameters must be specified. They are the Lagrange parameters,w_0, w_1, w_2, w_3, the critical temperature T_c, the saturation temperature and the annealing schedule. These are discussed below.

4.1 The Lagrange Parameters — w_0, w_1, w_2, w_3

In general, good solutions can be obtained for a reasonably wide domain in the space of w_0, w_1, w_2, w_3. However, some guidelines are suggested here in order to assure that our choices of the parameters lie within this domain.

In the mean field domain, all the energy functions E_0, E_1, E_2, E_3 become the functions of mean field variables as follows:

$$E_0 = -\frac{1}{2} \sum_i^S \sum_j^A \sum_k^T V_{ijk} \cdot V_{ijk} \tag{4.1}$$

$$E_1 = \sum_i^S \sum_k^T \sum_j^A \sum_{j_1 \neq j}^A V_{ijk} \cdot V_{ij_1k} \tag{4.2}$$

$$E_2 = \sum_j^A \sum_k^T \sum_i^S \sum_{i_1 \neq i}^S V_{ijk} \cdot V_{i_1jk} \tag{4.3}$$

$$E_3 = \sum_i^S \left(\sum_j^A \sum_k^T V_{ijk} - r_i \right)^2 \tag{4.4}$$

Consider the derivative of the total energy function in the mean field domain:

$$\frac{\partial(E)}{\partial(V_{ijk})} = w_0 \frac{\partial(E_0)}{\partial(V_{ijk})} + w_1 \frac{\partial(E_1)}{\partial(V_{ijk})} + w_2 \frac{\partial(E_2)}{\partial(V_{ijk})} + w_3 \frac{\partial(E_3)}{\partial(V_{ijk})} \quad . \tag{4.5}$$

The parameter w_0 governs the relative balance between the "cost" and" "constraint" terms. w_1, w_2, w_3 reflect the relative importance among Constraints 1 through 3. Since $\frac{\partial(E_1)}{\partial(V_{ijk})}$ and $\frac{\partial(E_2)}{\partial(V_{ijk})}$ are similar in nature and much more important than the others, they are thus weighted equally, and are weighted heavier than the others. For example, we may choose:

$w_0 = 0.4;$

$w_1 = 2.0;$

$w_2 = 2.0.$

Consider the effect of each individual parameter on any neuron. Note that, from Equations (4.2) and (4.3), $\frac{\partial(E_1)}{\partial(V_{ijk})}$ and $\frac{\partial(E_2)}{\partial(V_{ijk})}$ are always positive, and thus by Equation (3.14), the value of neuron V_{ijk} due to E_1 and E_2 approaches "0".From Equation (4.1) $\frac{\partial(E_0)}{\partial(V_{ijk})}$ is always negative, making neuron V_{ijk} approach "1". $\frac{\partial(E_3)}{\partial(V_{ijk})}$ may be positive or negative depending on whether the requested time slots have been satisfied, thus making the neuron approach "0" or "1," respectively.

Since we have already determined w_1 and w_2, we are left to determine the relationship between w_0 and w_3. In other words, we may now assume Constraints 1 and 2 are already satisfied, i.e., E_1=E_2=0 $\Rightarrow \frac{\partial(E_1)}{\partial(V_{ijk})} = \frac{\partial(E_2)}{\partial(V_{ijk})} = 0$. Consider the extreme case in which V_{ijk} takes on either 0 or 1. In this case, for each fixed i, if the number of neurons having values "1" are more than the requested time slots (see Equation (4.4)), this implies that the system tries to allocate more time slots than requested. Thus, we should try to force the system to turn "off" a neuron. Note that the neuron V_{ijk} that is "on" has $\frac{\partial(E_0)}{\partial(V_{ijk})} = -1$ (see Equation (4.1)), and $\frac{\partial(E_3)}{\partial(V_{ijk})} = 2$ (see Equation (4.4)). To turn off this neuron (see Equation (4.5)),

$$\frac{\partial(E)}{\partial(V_{ijk})} > 0$$

$$\Rightarrow \qquad w_0 \frac{\partial(E_0)}{\partial(V_{ijk})} + w_3 \frac{\partial(E_3)}{\partial(V_{ijk})} > 0$$

$$\Rightarrow \qquad w_0(-1) + 2w_3 > 0$$

$$\Rightarrow \qquad w_3 > 0.5w_0 \tag{4.6}$$

Now consider the other extreme case in which the number of time slots allocated by the network is less than the requested time slots. In this case, the network should try to turn on a neuron. Note that the neuron V_{ijk} that is "off" has $\frac{\partial(E_0)}{\partial(V_{ijk})} = 0$; and $\frac{\partial(E_3)}{\partial(V_{ijk})} < 0$. Thus, $\frac{\partial(E)}{\partial(V_{ijk})} < 0$; and the neuron is turned "on" as long as $w_3 > 0$.

In conclusion, we may use the following rule of thumb:

$$w_0 = 0.4 \tag{4.7}$$

$$w_1 = w_2 = 2 \tag{4.8}$$

$$w_1 > w_3 > 0.5w_0 \tag{4.9}$$

4.2 The Critical Temperature, Tc

In MFT, our task is to solve for the neuron value V_{ijk} at different temperatures through a set of nonlinear equations Equation (3.14). For convenience, this equation is rewritten as follows:

$$V_{ijk} - 0.5\, a_{ijk} = 0.5\, tanh\left(-\frac{\partial(E)}{\partial(V_{ijk})}\frac{1}{2T}\, a_{ijk}\right) \tag{4.10}$$

In order to gain insight on the dynamics in obtaining a solution, consider Fig.4.1. In this figure, while the abscissa represents the neuron V_{ijk}, the ordinate represents various functions of V_{ijk}. On the other hand, while the ordinate represents an arbitrary variable X, the abscissa represents 0.5 *tanh(-X/2T).* The straight line labeled by y1 represents the left-hand side of Equation (4.10). Here we only need to consider the case when $a_{ijk}=1$, otherwise, $V_{ijk} = 0$. The straight line labeled by y2 represents the function $\frac{\partial(E)}{\partial(V_{ijk})}$.

Note that in our SBS problem, from Equations (4.1) through (4.5) , we obtain

$$\frac{\partial(E)}{\partial(V_{ijk})} = m\,(V_{ijk} - B\,),$$

where m and B are constants, which is a straight line. However, the discussion here is also applicable to the case when $\frac{\partial(E)}{\partial(V_{ijk})}$ is no longer a straight line. As mentioned above, the dashed curve represents the function (0.5 *tanh(-X/2T)*) in which the ordinate is X, and the abscissa is 0.5*tanh(X/2T)*. It is readily seen from Equation (4.10) that we can map y2 through the hyperbolic tangent function [i,e., 0.5 *tanh* (-y2/2*T*)] to obtain the curve labeled by y3. y3 corresponds the right-side of Equation (4.10). Fig. 4.2 shows an example of mapping a point on y2 to the corresponding point on y3 through the mapping, 0.5 *tanh(-X/2T)* .

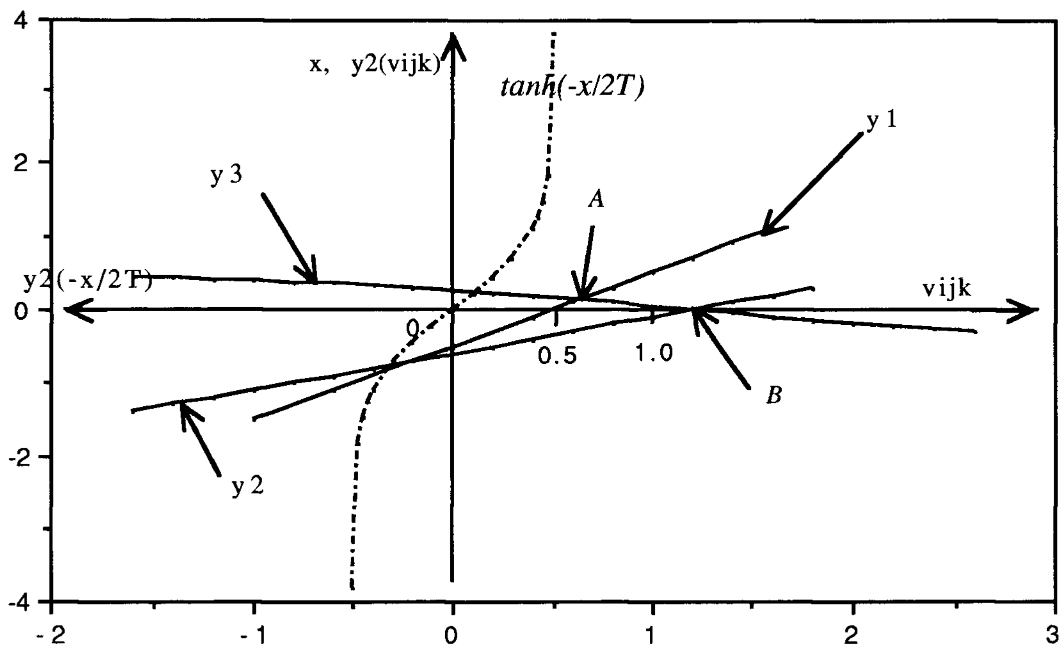

Fig. 4.1. The Dynamics in Obtaining a Solution

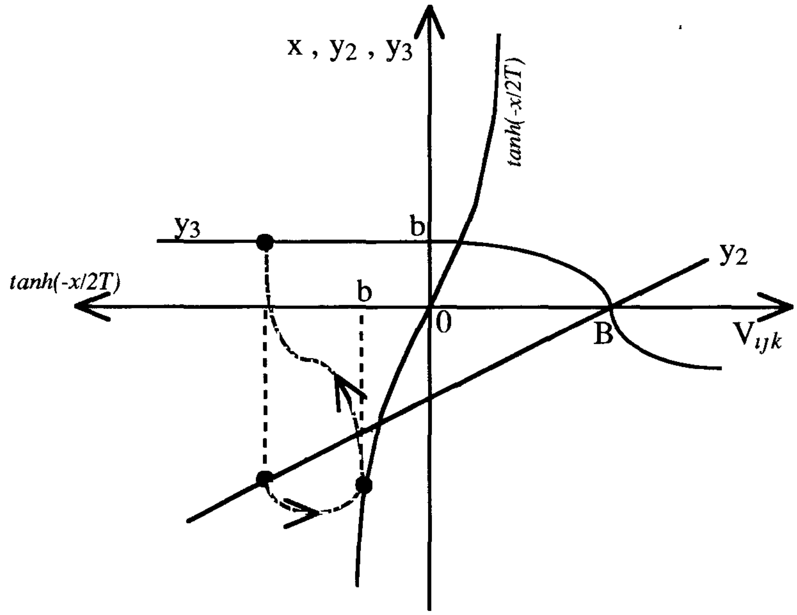

Fig. 4.2. Mapping a Point From y_2 to y_3

Note that Curve y3 and line y2 intersect on the abscissa axis at a point labeled B. The value of B depends on the state of the network. Curve y3 and Line y1 intersect at A which is the solution for Equation (4.10). The abscissa value of A is the neuron value of V_{ijk} at temperature T.

Fig. 4.3 shows the behavior of Curve y3 at different temperatures. Note that at high temperature, Curve y3 becomes a straight line with slope equal to approximately zero. Thus, the solution at high temperature is V_{ijk} =0.5, i.e., the intersection point between y1 and y3 is (0.5, 0). We thus have the following Lemma.

Lemma 2 All neurons except those clamped by the associative matrix, have values of 0.5 at high temperature.

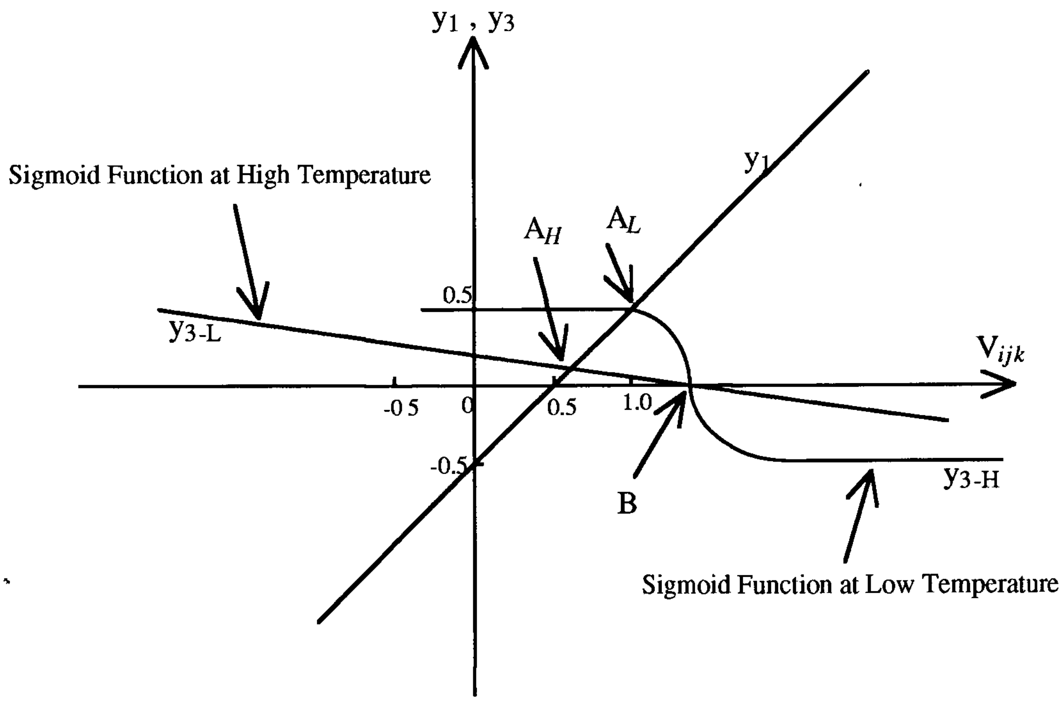

Fig. 4.3. Solutions at High and Low Temperatures

However, as the temperature is decreasing, the dash curve and therefore y3 is becoming a signum function as shown in Fig. 4.3. If B is greater than 1, it can be seen that in this case, the intersection which is the solution is (1, 0.5), i.e., $V_{ijk}=1$. Similarly, it can be shown that if B is less than 0, the solution is $V_{ijk}=0$.

Our goal is to determine the temperature parameter known as the critical temperature at which a remarkable state transition takes place resulting in a deep drop of system energy. From **Lemma 2** all neurons except those which are clamped have the same initial value of 0.5, and thus the remarkable state transition likely occurs when neurons start acquiring a value of 1 or 0, at which case the neurons start competing for 1 or 0. We thus propose the following definition for the critical temperature.

Definition: The critical temperature is the highest temperature at which at least one neuron V_{ijk} reaches 1 or 0 from its original trivial state,i.e., 0.5.

Lemma 3 The critical temperature, T_c, for our SBS problem is approximately equal to :

$$T_c = m(2b - y - 1)/4y \qquad (4.11)$$

where m , B and y are derived from the following system of equations:

$$\begin{cases} 2V_{ijk} - 1 = y \\ -(1/2T)\, m(V_{ijk} - B) = y \end{cases} \qquad (4.12)$$

Proof: Expanding *tanh* (-x/2T) by a Taylor series at x=0, we have

$$tanh(-x/(2T)) = -\frac{x}{2T} - \frac{1}{24\,T^3} x^3 + \ \dots \qquad (4.13)$$

If we only use the first order term to approximate *tanh* (- x/ (2T)), the solution point A is obtained by solving the following system of equations:

$$\begin{cases} 2V_{ijk} - 1 = y \\ -(1/2T)\, m(V_{ijk} - B) = y \end{cases} \qquad (4.14)$$

where we let $x = \dfrac{\partial(E)}{\partial(V_{ijk})} = m(V_{ijk} - B)$ (4.15) and $$m = \frac{\partial\left(\frac{\partial(E)}{\partial(V_{ijk})}\right)}{\partial(V_{ijk})} = 2w_3 - w_0 \tag{4.16}$$

which is a constant .

Solving Equation (4.14), the critical temperature

$$T_c = m(2B - y - 1)/4y \tag{4.17}$$

where y is +1 or -1 because the remarkable transition occurs when the neuron reaches 1 or 0 from its original trivial value; i.e., $V_{ijk} = 1$ or $0 \Rightarrow y = \pm 1$. Furthermore, as shown in Fig. 4.4, if $B > 1$ and $m > 0$, the solution $V_{ijk} = 1$ ($y=1$). Likewise, if $B < 0$ and $m > 0$, the solution $V_{ijk} = 0$ ($y=-1$). Similarly, other conditions can be summarized below:

	$B > 0.5$	$B < 0.5$
$m > 0$	$+1$	-1
$m < 0$	-1	$+1$

(4.18)

Strictly speaking, when $0<B<1$, V_{ijk} does not take on 0 or 1. However, within a few iterations the particular V_{ijk} will converge to 1 or 0.

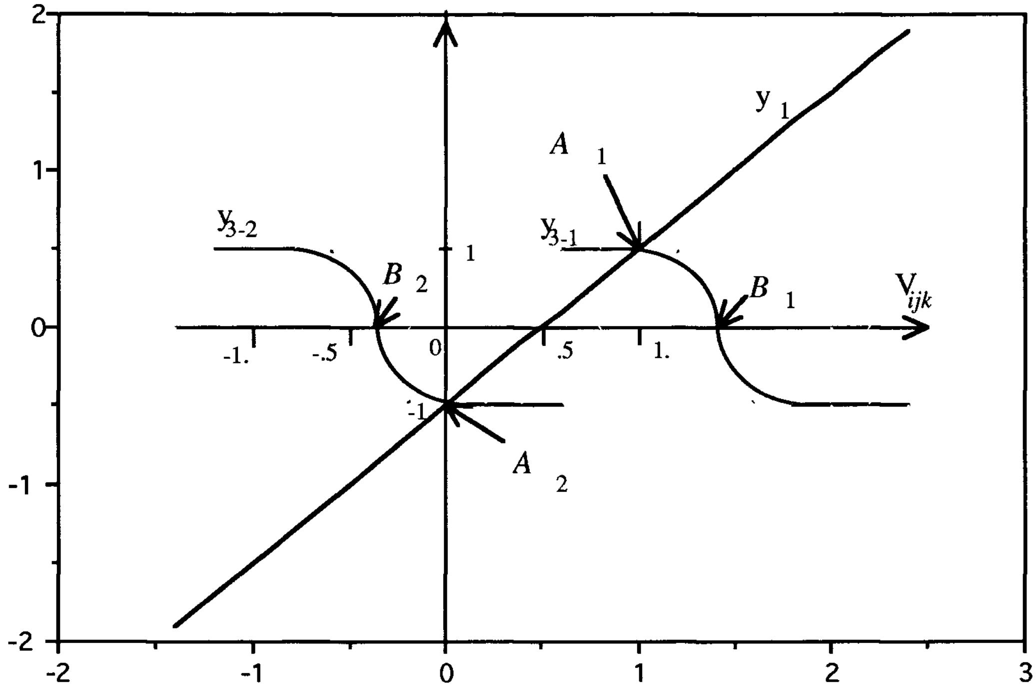

Fig. 4.4. The two possible solutions of Equation 4.12.

[Example 4.1] Given the following conditions for which the solution will be discussed in Example 5.3

$$w_0 = 0.3, \quad w_1 = 2.00, \quad w_2 = 2.00, \quad w_3 = 0.2$$

Here, $m = 2 w_3 - w_0$

$= 0.4 - .03$

$= 0.1 > 0$

$B = (0+0 +2 \times 0.2 \times (0.5 \times 10 - 8.0) / (-0.1)$

$= 12 > 0.5$

Since $m > 0$ and $B > 0.5$, then $y = 1$.

$T_c = 0.5 \times m \times (B - 1)$

$= 0.55.$

The experimental result which will be discussed in Example 5.3 show that the critical temperature is 0.51.

4.3 The Annealing Schedule

We adopt the following linear annealing schedule starting from the critical temperature:

$$T(n+1) = 0.9\, T(n),$$

where $$T(0) = T_c . \tag{4.19}$$

The stopping criterion for the annealing procedure is defined by the temperature at which the network is saturated. The network is saturated if the following conditions are met.

(1) All neuron values are within the range [0.0,0.2] or within the range of [0.8, 1.0] without any exception;

(2) $$\sum_{i}^{S}\sum_{j}^{A}\sum_{k}^{T} (S_{ijk})^2 \,/\, N > 0.95 \tag{4.20}$$

where N is the number of neurons that have values within the range between 0.8 and 1.0.

Chapter 5
Numerical Implementation and Solutions

5.1 Numerical Implementation

When implementing the MFA algorithm numerically, the straightforward iteration method below is used at each temperature to obtain the steady state neuron values.

$$(V_{ijk})^{(n+1)} = a_{ijk} \left(0.5 + 0.5 \, tanh \left(- \frac{\partial(E)}{\partial(V^{(n)}_{ijk})} \frac{1}{2T} \; a_{ijk} \right) \right) \tag{5.1}$$

The superscript n indicates the iteration index. For each iteration, there are many neurons to be updated. We can either update all neurons synchronously or one after another asynchronously. In practice, it is found that asynchronous updating has a better performance. The procedure to schedule the satellite broadcasting times using MFT is summarized below:

(1) For a given SBS problem, establish the associative matrix *A* described in Section 2.3.

(2) Set the coefficients w_0, w_1, w_2, w_3 as discussed in Section 4.1;

(3) Determine the critical temperature T_c according to Equations (4.17) through (4.18);

(4) Initialize neurons with random numbers as follows:

$$V_{ijk} = \{ 0.5 + 0.001 rand \, [\, 1, -1 \,] \} \, a_{ijk} ; \tag{5.2}$$

(5) Anneal the network until the network is saturated according to the saturation criterion defined by Equation (4.20)

(6) At each temperature, iterate the MFT equations until the following convergence

criterion is met .

$$\sum_{ijk}^{SAT} \left| S_{ijk}(n) - S_{ijk}(n) \right| < 10^{-3} N_n \tag{5.3}$$

where N_n is the number of non-zero neuron elements. That is, we require that the averaged difference of a neuron between two iterations to be within 10^{-3}.

5.2 Convergence

Divergence is one of the most difficult problems encountered in numerical calculations. There are several types of factors which lead the calculation to divergence. Next, we shall discuss the factors that will cause divergence in the straightforward iteration method used to solve the SBS problem. Rewriting the straightforward iteration described in Equation (5.1), we obtain

$$V_{ijk} - 0.5\, a_{ijk} = 0.5 \tanh\left(- \frac{\partial(E)}{\partial(V_{ijk})} \frac{1}{2T} a_{ijk} \right) \tag{5.4}$$

Let the left -hand side and the right-hand side of Equation (5.4) be y_1 and y_2, respectively.

$$y_1 = V_{ijk} - 0.5\, a_{ijk} \tag{5.5}$$

$$y_2 = 0.5 \tanh\left(- \frac{\partial(E)}{\partial(V_{ijk})} \frac{1}{2T} a_{ijk} \right) \tag{5.6}$$

The solution of Equation (5.4) is the intersecting point between Line y_1 and Curve y_2 as shown in Fig. 5.1. Consider the two following cases in which the slope of y_2 in the neighborhood of the solution are less than and greater than 1.

Case 1. $\quad |\text{ the slope of } y_2 \,| > 1$;

Case 2. $\quad |\text{ the slope of } y_2 \,| < 1$.

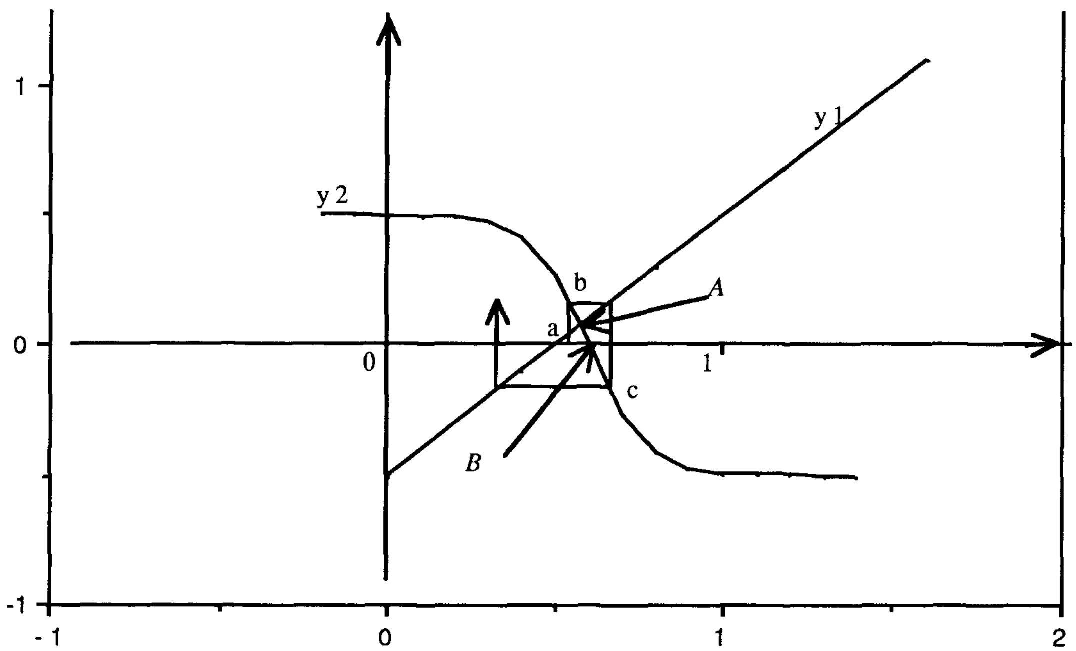

Fig. 5.1. Slope of y_2 in the neighborhood of the solution is greater than 1

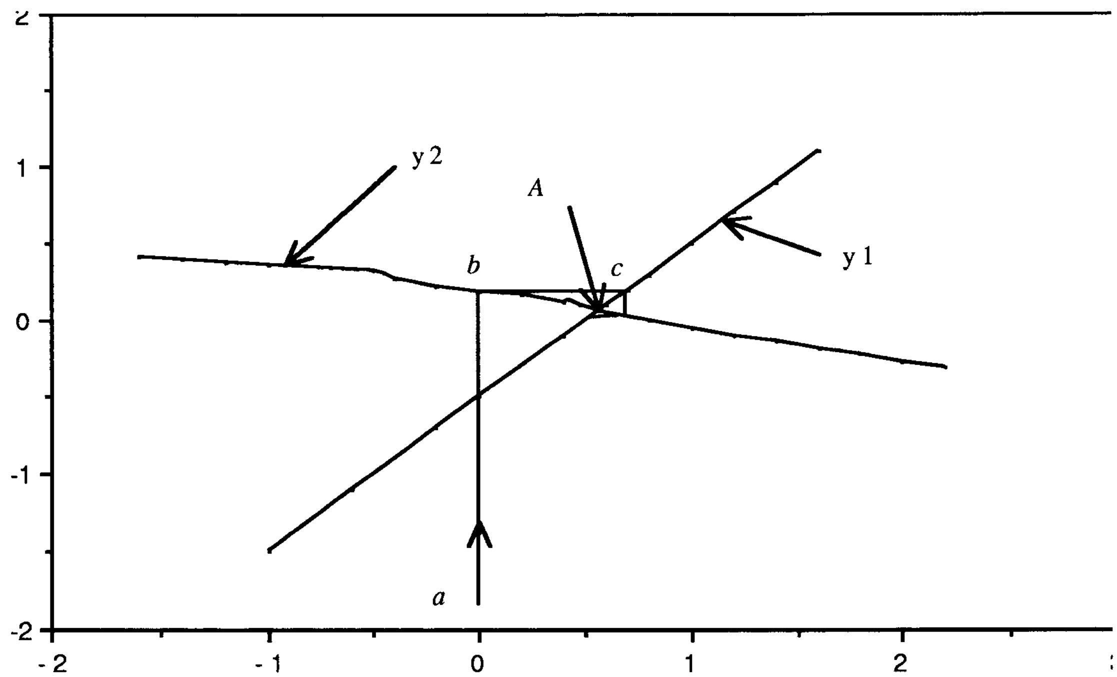

Fig. 5.2. Slope of y_2 in the neighborhood of the solution is less than 1

Fig. 5.1 depicts the condition corresponding to Case 1 and likewise, Fig. 5.2 to Case 2. These figures show how the solution evolves through the iteration procedure (Equation (5.4)), indicated by the arrows. Here A is the solution. The sequence of arrow a-b-c represents one iteration. As shown in Fig. 5.1, the iteration procedure diverges from solution A, while the procedure converges to solution A in Fig. 5.2.

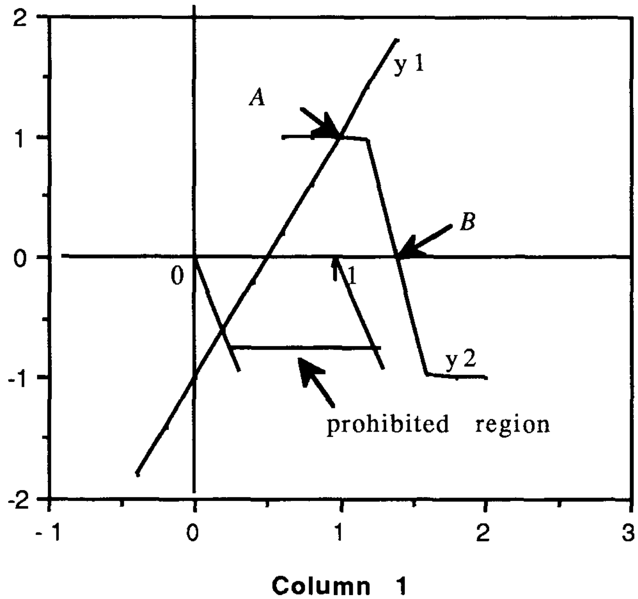

Fig.5.3. The intersection point B is outside the range [0,1]

If y2 is moved such that the intersecting point B between y2 and the abscissa is outside the range [0,1], it can be shown (see Fig. 5.3) that the iterating procedure will converge to the solution. Unfortunately, the exact location of the intersection point B is unknown because it will move dynamically during the iterating process. In Case 2, the divergence caused by one iteration is called local divergence. Local divergence may not be a fatal divergence because the intersection point B may move out of the range [0,1] after a sweep of iterations (all neurons are updated once). If the intersection point B always lies within the range [0,1], then the local divergence becomes global. We shall avoid global divergence.

In the previous section, we point out that asynchronous iteration is better than synchronous iteration. One reason is that the asychronous method has more chances for the intersection point B to jump out of the [0,1] range.

To avoid global divergence, one may adjust some parameters such that the intersection point B is out of the range [0,1]. The following are a few suggestions:

(a) Adjust w_0, w_1, w_2, w_3;

(b) Simply increase r_i;

(c) Use other iteration methods.

5.3 Solutions

We have implemented the proposed method to solve the SBS problem of various sizes. We consider cases when the requested broadcasting time is less than the maximum capacity the network can allocate, as well as cases when the requested broadcasting time exceeds the maximum capacity of the network. Cases of first type are known as "small request" cases, and cases of second type are known as "large request" cases.

[Example 5.1] Consider the SBS problem with four satellites, three ground terminals and nine time slots. Constraint 4 is defined by the Associate Matrix $\boldsymbol{A}$ and the requested broadcasting time for each satellite is defined by **R** as follows:

$$A = \begin{matrix} 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{matrix}$$

$$\mathbf{R} = [\ 2 \quad 2 \quad 2 \quad 2\ ]^T$$

SOLUTION 1

$w_0 = 0.5$, $w_1 = 2.00$, $w_2 = 2.00$, $w_3 = 0.2$

SATELLITE 1

0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000
0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000

SATELLITE 2

0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000

SATELLITE 3

0.000	0.000	0.000	0.000	0.000	1.000	1.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000

SATELLITE 4

0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000
1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

$$\mathbf{U} = [\ 3 \quad 3 \quad 3 \quad 3\ ]^T$$

Here **U** is the time slots allocated by the network.

SOLUTION 2

$w_0 = 0.3$, $w_1 = 2.00$, $w_2 = 2.00$, $w_3 = 0.2$

SATELLITE 1

0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.999	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.999	0.000	0.000

SATELLITE 2

0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.999	0.000	0.000
0.000	0.000	0.000	0.002	0.000	0.999	0.000	0.000	0.000

SATELLITE 3

0.004	0.000	0.000	0.000	0.000	0.999	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.999
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

SATELLITE 4

0.000	0.000	0.999	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.004	0.000
0.999	0.000	0.000	0.003	0.000	0.000	0.000	0.000	0.000

$$\mathbf{U} = [\ 2\ \ 2\ \ 2\ \ 2\ ]^T$$

Both solutions are legal, and sufficient to meet the requested time slots, i.e.,

$$\mathbf{U} \geq \mathbf{R}$$

Solution 1 allocates more time slots than requested because w_0 is larger, i.e., more emphasis is placed to maximize the capacity.

[Example 5.2] Consider the SBS problem with four satellites, three ground terminals and nine time slots. Constraint 4 is defined by the Associate Matrix *A* and the requested broadcasting time for each satellite is defined by **R** as follows:

$$A = \begin{matrix} 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{matrix}$$

$$\mathbf{R} = [\ 9 \quad 8 \quad 7 \quad 6\ ]^T$$

$w_0 = 0.3$, $w_1 = 2.00$, $w_2 = 2.00$, $w_3 = 0.2$

SATELLITE 1

0.000	1.000	0.000	1.000	0.000	1.000	1.000	0.000	1.000
0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000

SATELLITE 2

1.000	0.002	0.002	0.001	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	1.000	1.000	0.000
0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	1.000

SATELLITE 3

0.000	0.000	0.003	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000
0.000	0.000	0.000	0.000	1.000	0.000	1.000	1.000	0.000

SATELLITE 4

0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000

$$\mathbf{U} = [\ 6 \quad 5 \quad 4 \quad 3\ ]^T$$

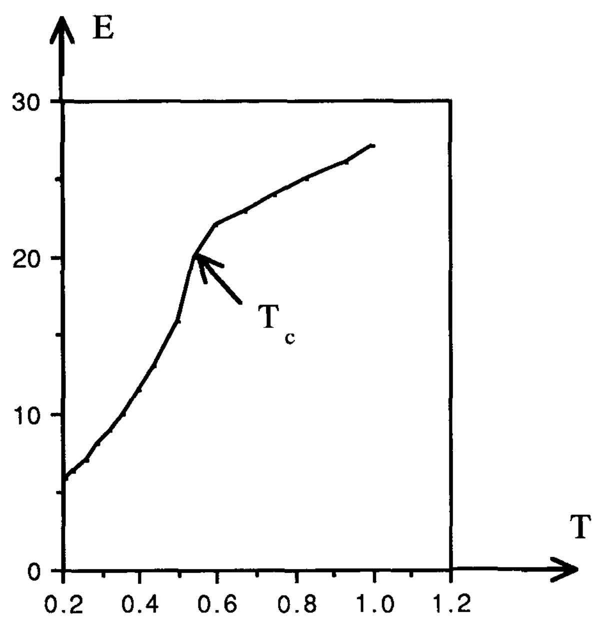

Fig. 5.4 System Energy at Different Temperatures

In this example, the requested time slots are more than the system can allocate. Though the system cannot meet the request, it provides the maximum under the given constraints.Fig. 5.4 shows the system energy at different temperatures. The experiment shows that the critical temperature is 0.51 at which a remarkable transition of system (a deep drop of the system energy) takes place.

Chapter 6
Conclusions

In this thesis, we have presented a new method to solve the satellite broadcast scheduling problem. The problem was first mapped onto a neural network from which an energy function is derived. Optimization is achieved by minimizing the energy by Mean Field Annealing. Our key contributions include :

(1) Formalize an appropriate energy function.

(2) Introduce the clamping technique, and thus reduce the computation.

(3) Derive the estimated critical temperature of the algorithm.

(4) Discuss and suggest alternatives to avoid the divergence of the numerical implementation of the proposed method.

(5) Demonstrate the robustness of our method by having achieved good solutions for problems of various sizes.

As compared to the previous method [1] [2], our method excels in the following ways:

(1) Our method using MFT is parallel and global in scope, thus achieving good performance and computational efficiency.

(2) Our method does not need to specify a set of distinct priorities for the satellites to broadcast, and no assumption is made on requiring a set of suitable requests.

In conclusion, MFT has been demonstrated to be an effective and robust optimization technique in solving the SBS problem.

LIST OF REFERENCES

[1] Bourret, P., Goodall, S. and Samuelides, M., "Optimal Scheduling Competitive Activation: Application to the Satellite Antennas Scheduling Problem," *Proc. IJCNN'89*, pp. I565-572, 1989.

[2] Bourret, P., Rem, and F., Goodall, S., "A Special Purpose Neural Network for Scheduling Satellite Broadcasting Times,"*Proc. IJCNN'90*, pp. II535-538, 1990.

[3] Hopfield, J. J.,"Neurons with Graded Response have Collective Computational Properties Like Those of Two-state Neurons," *Proceedings of the National Academy of Sciences*, Vol. 81, pp. 3088-3092, 1984.

[4] Hopfield, J. J. and Tank, D. W., "Neural Computation of Decisions in Optimization Problems," *Biol. Cybernetics* , Vol. 52, pp. 141-152, 1985.

[5] Peterson, C., "A Mean Field Theory Learning Algorithm for Neural Networks,"*Complex System,* Vol. 1, pp. 995-1019, 1987.

[6] Peterson, C., "Neural Networks and NP-Complete Optimization Problem; a Performances Study on the Graph Bisection Problem,"*Complex System,* Vol. 2, pp. 59-89, 1988.

[7] Kirpatrick, S., Gelatt, C. D. and Vecchi, M. P., "Optimization by Simulated Annealing," *Science,* Vol. 220, pp. 671-680, 1983.

[8] Van, P.J., Laarhoven, M. and Arts, E. H. L., *Simulated Annealing: Theory and Applications*. Dordrecht, Holland: Reidel Publishing Conpany, 1987.

[9] Hinton, G. E. and Sejnowski, T. J., "Learning and Relearning in Boltzmann Machines" in *Parallel Distributed Processing , vol.1: Foundations,* ed. D. E. Rumelhart, et al., pp. 282-317, Cambridge, MA: The MIT Press, 1986.

[10] Peterson, C., "A New Method for Mapping Optimization Problems onto Neural Networks," *International Journal of Neural Systems,* Vol. 1, No. 1, pp. 3-22, 1989.

[11] Anderson, J. A. and Rosenfeld, E., Eds., *Neurocomputing: Foundations of Research.* Cambidge, MA: M I T Press, 1988.

[12] Nilsson, N., *Learning Machines.* New York: McGraw-Hill,1965.

[13] Staniforth, P. R., "Store-and-forward Satellite Communications System on UOSAT-2,"*The Journal of the Institution of Electronic and Radio Engineers,*" Vol. 57, p. 43, Jan. 1987.

[14] Luenberger, D. G., *Linear and Nonlinear Programmimg Reading*. MA: Addison Wesley,1984.

[15] Geman, S. & Geman, D., "Stochastic Relaxation, Gibbs Distribution,and the Bayessian Restoration of Images," *IEEE tran. on Pattern Analysis and Machine Intelligence,* vol. PAM1-6, pp. 721-741, 1984

[16] Goldberg, D. E., *Genetic Algorithm in Speech, Optimization & Machine Learning, Reading,* MA: Addison Wesley, 1989

[17] Peterson, C. and Hartman, E., "Explorations of the Mean Field Theory Learning Algorithm" *Neural Network,* Vol.2, pp. 475-494, 1989.

[18] Bilbro, G., et al, "Optimization by Mean Field Annealing," in *Advances in Neural Information Processing System* 1, ed. D. S . Touvetzky, pp. 91-98, 1989.