SOLVING OPTIMIZATION PROBLEMS: MEAN FIELD ANNEALING THEORY AND HOPFIELD NEURAL NETWORK

R. Shankar and R.B. Krishna

IIIrd year, B.Tech (Information Technology) Sri Venkateswara college of Engineering, Sriprumbudur-602105, INDIA. rbkrishna@hotmail.com

1. INTRODUCTION

An optimization problem may have many possible solutions. However, only one of the solutions minimizes the cost, or error, function. This kind of optimization problem is called combinatorial optimization problem. They are often divided into classes according to the time needed to solve them. If an algorithm exists that is able to solve the problem in a time that increases polynomially with the size of the problem, the problem is said to be polynomial (*P*). Non-deterministic polynomial (*NP*) complete problems are a class within the *P* class. Such problems can be tested in polynomial time whether or not a guessed solution of the problem is correct. To solve a *NP* problem in practice, it usually takes time of order exp (N), where N is the size of the problem.

The traveling salesman problem (*TSP*) is a *NP*-complete type of problem in which the shortest path needed for a salesman to visit N cities on a planar surface is to be found out. If you choose a path at random through a given list of cities, you are likely to find that it is not the most efficient in terms of cost for all possible paths, such a computation may take an extremely long time. Given any N cities, there are $N!$ possible tours. If you consider the fact that, for a given tour, it does not matter where you begin, or in which direction you travel, then the total number of independent tours is $N!/2N$. For a small number of cities, you would simply compute the cost of each tour, and choose the one with the minimum. For a tour of more than a few cities, this exhaustive search becomes quite time consuming. For five cities, the number of possible tours is 12. We would easily compute the cost of these twelve tours and select the most efficient one. However if there are ten cities on the tour, the number of different possibilities is 1,81,440 and number of possibilities is of the order of 10^{1217} for 532 cities. Therefore it is of vital importance to develop some heuristics to solve the problem and find near optimal solutions.

2. ARTIFICIAL NEURAL NETWORKS (ANN'S)

ANNs of the type that were described by Hopfield [1] employ one of the simplest possible neural network feedback architectures and are capable of finding good solutions for certain optimization problems. Furthermore, these *ANN's* can also solve certain constraint satisfaction problems. Constraint satisfaction problems can often be modeled as optimization problems that have numerous correct solutions that are of equal value. The neural network model for solving a combinatorial optimization problem was first introduced by Hopfield and Tank in their 1985-paper [2]. They have demonstrated in it that the network had significant emergent computational power: specifically it could solve *NP*-complete, Traveling Salesman type combinatorial optimization problems. Since then neural networks have been shown to provide powerful approaches for a wide variety combinatorial optimization problems. In 1991 Kunz [3] proposed the first Hopfield network model for solving channel assignment problem in the cellular radio network satisfying some frequency constraints. Lee Giles et al. [4] proposed a routing methodology in the optical multistage interconnection networks making use of the Hopfield model network. Nirwan Ansari et al. [5] and Nobuo Funabiki and Seishi Nishikawa [6] have reported methods for optimizing satellite broadcasting schedules also based on the Hopfield neural model.

While solving such problems, An optimization problem is first mapped onto a neural network in such a way that the network configurations correspond to possible solutions to the problem. A function of neural states, called the energy function, is constructed and this energy function is proportional to the cost function of the problem. The dynamics of the network is determined so that the energy functions (therefore the cost function) is minimized as the neuron update.

3. Hopfield Neural Networks

In a neural network we have the neurons that have a certain degree of activation, depending on how and with which intensity they are interconnected. This connection between the neurons is called synopses, and they are not physical, but chemical. The degree of activation of a particular pair of neurons is proportional to the amount of neurotransmitters released by the neuron. The neurotransmitter allows the electric pulse generated by some stimulus to pass from one neuron to others, transmitting the information neuron by neuron until it reaches the brain. The neuron dynamics will be governed by the state of other neurons to which it is connected to by the synapses.

Among the many neural network models developed to simulate some features of the human brain like recognition and learning, Hopfield proposed a neuron dynamics, which minimizes the following energy function.

$$E=-1/2*\Sigma T_{ij}V_iV_j,$$

where V_i is the state of the i-th neuron of the network and T_{ij} is the synaptic strength between neurons i and j. The dynamics, which minimizes this energy function, is

$$V_i(t+1)=sgn(\textstyle\sum_j T_{ij}V_j).$$

One first simplification of the behaviour of real neurons is to set the state of the neuron as a Boolean variable, that means to indicate whether it is activated or not (V_i =-1,1). One particular feature of Hopfield-like neural networks is that they can modify their behavior in response to their environment (feedback mechanism) which means, given a set of inputs to the neurons, they self-adjust to produce consistent responses. In order to simulate the features of the human brain, one has to implement a learning stage. One of the most used training algorithms is the Hebb rule, which is an unsupervised learning where the synaptic strength between two neurons is increased when both the source and destination neuron are activated. We can write this as $T_{ij}=\sum_\mu{}^N\xi^\mu{}_i\xi^\mu{}_j$, where ξ^μ is one of the N patterns stored in the brain. We represent one pattern by a set of N neurons in a particular configuration.

This learning stage is appropriate for pattern recognition by a neural network, which means searching for a local minimum in the energy surface, which is different from the problem of optimization, where the search is for global minimum.

As this network device works as a multiple-feedback circuit, which is not governed by any variable parameter, like the artificial temperature in Simulated Annealing (*SA*) or mutations in Genetic Algorithm (*GA*), it is claimed as "silicon implementable," that is, it is possible to develop microchips that execute this function.

Hopfield neural networks are said to be recurrent because of the dynamical feedback mechanism of changing the input as a function of the output of the network. For a stable network, after successive iterations the changes on the output will become smaller and smaller until it reaches an equilibrium state where all the outputs will become constant in the successive iterations. A network can be proven to be stable if the synaptic matrix is symmetric with zeros on its main diagonal, that means, $T_{ij}=T_{ji}$, and $T_{ii}=0$.

4. Traveling Salesman Problem

The realization of the traveling salesman problem in Hopfield neural networks consists on attributing to the distance between two cities A and B a symmetric synaptic strength d_{ab} and so on to all pairs of cities on a unit square (we normalize the maximum distance, and then the solutions given can be compared with any others with no need for rescaling). The problem is mapped onto a network by representing each city by a row of N elements, where N is the number of cities. Then, the state of a neuron, V_{ij}, means the ith city visited in the j-th order. If it is set to one, the city is really visited in that order, if it is set to zero, then this assertion is false. Fig. 1 is an example where the city B is visited first,

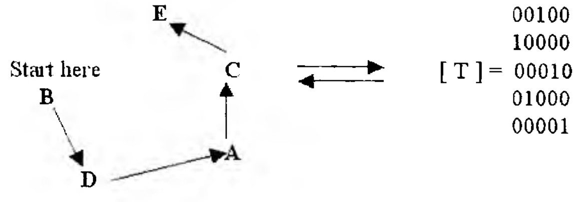

Figure 1. Matrix representation of the TSP problem

followed by *D, A, C and E*. The resulting perimeter, or energy, is $d_{B,D}+d_{D,A}+d_{A,C}+d_{C,E}$. It can be verified that this $N{\times}N$ matrix is sparse (more exactly, it has only N elements).

The way of defining the energy function is not unique and requires specification of the problem and what is to be minimized. In the Traveling Salesman Problem, it is required that each row and each column have only one non-zero element. This means that a city is to be visited only once and only one city per time is visited. To map this constrained optimization problem onto a Hopfield neural network, we have to embed the constraints onto one function known as the energy function which consists of two terms: the cost term and constraint term. The cost term is the optimization cost (objective) function that is independent of the constraint term and the constraint term is the penalty imposed on for violating the constraints. These two terms must counteract each other. The optimization is then achieved by minimizing the energy function [2].

Even though Hopfield and Tank reported success for a 30-city problem, the convergence of the network to an acceptable tour is difficult to reach and that this possibility decreases rapidly with the number of cities. This is because many terms of the energy function compete with each other to be minimized. Further more, the (strict descent) dynamics of the Hopfield network result in convergence to the first local minima encountered. There is no guarantee that the network will converge to the global minimum rather than to a local minimum and even whether the minimum found will respect all the constraints (providing an acceptable tour) or not.

5. Satellite Broadcasting Scheduling Problem

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. Here, a method that is based on a Hopfield network in combination with the mean field annealing theory (*MFT*) is presented. *MFT* was recently found to be an efficient method in solving large connectionist problems. The main advantage of using the *MFT* method lies in the fact that the search for optima is parallel in the global sense.

Most commercial satellite systems are launched onto the geostationary orbit. As the demand for various telecommunications applications becomes increasingly sophisticated, there are situations where orbits other than geostationary orbit become desirable. In this case, it is necessary to schedule the "hand-over" from one satellite to another- this is the problem addressed in this paper. It emerges as an important problem since potential advantages of a low altitude system such as reduced satellite power requirement and antennas, smaller propagation delay and high resolution images have prompted the industry to build such a system. For example, ORBCOMM proposes to operate a system of about 20 low altitude satellites, and GLOBALSTAR is considering a larger system of about 40 satellites.

Communication links between satellites and ground terminals are provided in a repetition of time slots. A time slot has a unit time to broadcast information from a satellite to a ground terminal when they are visible to each other. The goal of our algorithm is to find the broadcasting schedule of satellites with maximum number of broadcasting time slots under the following four constraints.

1. A satellite cannot broadcast information to more than one ground terminal at the same time slot.
2. A ground terminal cannot receive information from more than one satellite at the same time slot.
3. A satellite cannot be assigned to more time slots than requested for it, unless all the requests are allocated for all the satellites. The satellite must broadcast as much as possibly close to its requested time.
4. A satellite can broadcast only when it is visible from a ground terminal.

To solve this problem the following notation has been adapted.

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

$$S=\{1,2,3,\ldots,i,\ldots,N_S\}.$$

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

$$A=\{1,2,3,\ldots,j,\ldots,N_A\}.$$

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

A set of requests for broadcasting time of satellites is given by a request vector R. The i-th element r_i

represents the number of broadcasting time slots requested for satellite i, given by the problem. It consists of N_S elements (time slots)

$$R=\{r_1, r_2, r_3,\ldots,r_{N_S}\}^t.$$

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

$$U=\{u_1, u_2, u_3,\ldots,u_{N_S}\}^t.$$

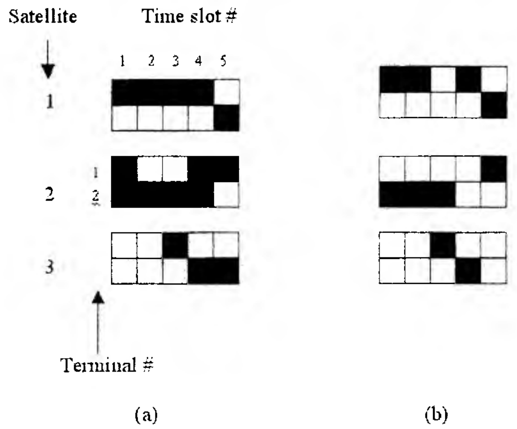

Figure 2. A satellite broadcast scheduling problem.

(a) An associative matrix.

(b) The optimum solution for the request vector (2,3,2).

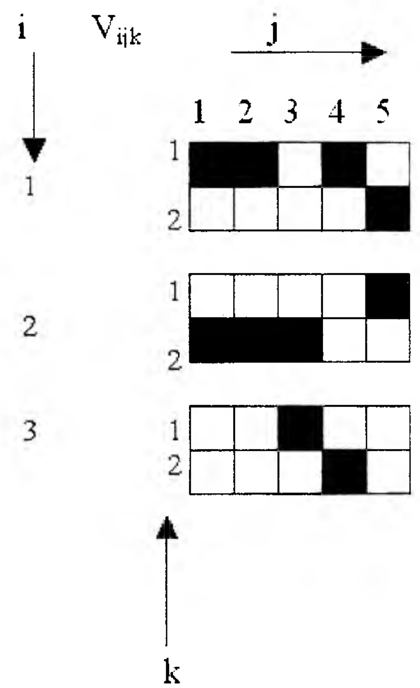

Figure. 3. The neural network for the problem in fig. 2.

The visibility between satellites and terminals is described by an associative matrix. The ijk-th element is "1" when satellite i is visible to terminal j at time slot k, and the element is "0" when it is not there. Fig. 2(a) shows an associative matrix for the satellite broadcast-scheduling problem with three satellites, two terminals, and five time slots. The black square indicates "1" and the white one does "0". For example satellite 1 is visible to terminal 1 at 4 time slots 1-4. When a request vector (2,3,2) is given to this matrix, the optimum solution is composed of ten broadcasting time slots as shown in 2(b). Neural network for *SBS* problem basically follows the approach by Hopfield et al. [2]. The output V_{ijk} of neuron represents whether satellite i should be assigned to terminal j at time slot k or not. The "one output ($V_{ijk}=1$)" represents the assignment, and the "zero output ($V_{ijk}=0$)" does no assignment. Thus, a total of $S\times A\times T$ neurons is required for the problem with S satellites, A terminals, and T time slots. Fig. 3 shows the neural network composed of $30(=3\times2\times5)$ neurons, and the convergence state to the solution corresponding to fig. 2(b). The black square indicates the one output and the white one does the zero output.

6. Formulation of the Energy Function

This problem is also a constrained optimization problem. Here the cost term or the energy due to the cost, E_o, is defined by

$$E_o=-\tfrac{1}{2}\sum_i \sum_j \sum_k S_{ijk}\cdot S_{ijk}$$

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

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

1. To satisfy the first statement the penalty term for the first constrain is

$$E_1=\sum_i \sum_k \sum_j \sum_{j1\neq j}(S_{ijk}\cdot S_{ij1k}).$$

2. The penalty term for the second constraint can be defined by

$$E_2=\sum_j \sum_k \sum_i \sum_{i_1\neq i}(S_{ijk}\, S_{i1jk}).$$

3. The penalty term to satisfy the third constraint can be given by

$$E=\sum_i (\sum_j \sum_k S_{ijk}-r_i)^2=\sum_i (u_i-r_i)^2$$

Where $u_i=\sum_j \sum_k S_{ijk}$.

4. The fourth constrain is imposed using clamping technique in the next section. The total energy function for the SBS problem defined in the Hopfield framework becomes

$$E=w_0E_0+w_1E_1+w_2E_2+w_3E_3$$

Where w_0, w_1, w_2, w_3 are the Larange parameters used to weigh the significance of E_0, E_1, E_2, E_3, respectively.

7. The MFT Framework for the SBS Problems

Conventional methods are not efficient in solving large size optimization problems. A new method for optimizing *SBS* based on the Hopfield neural network in combination with mean field annealing theory (*MFT*) has been reported in this paper. 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 *MFT* equations to be more economical. *MFT* is derived from the Stochasticly Simulated Annealing (*SSA*) by incorporating the *SSA* mechanism with the Hopfield function. *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* mechanism to escape from local minima. In *SSA*, there are two conceptual operations involved: a thermostatic operation that schedules the decrease of the temperature (an algorithm parameter), and a random relaxation process which searches for the equilibrium solution at each temperature. In *MFT* the two equations are still needed. The thermostatic operation is 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.

Usually, the number of neurons that are nulled owing to constraint (4) is large. This constraint is reflected into the neural network by clamping those neurons that do not meet constraint (4) to zero throughout the optimization. By incorporating the clamping technique the final *MFT* equations for the *SBS* problem is given below.

$$V_{ijk}=a_{ijk}\,(0.5+0.5tanh(-\{\delta E/\delta V_{ijk}\}\{1/2T\}a_{ijk})$$

In the mean field domain, S_{ijk} is replaced by V_{ijk} in all the energy functions E_0, E_1, E_2, E_3. Considering the importance among the constraints the weights for w_0, w_1, and w_2, are used as 0.4, 2, 2, respectively and $w_1<w_3>0.5w_0$.

The Critical Temperature, T_c

In *MFT*, our task is to solve for the neuron value V_{ijk} at different temperatures through a set of nonlinear equations for V_{ijk}. All neurons except those clamped by the associative matrix, have values of 0.5 at high temperature. Now the critical temperature at which at least one neuron V_{ijk} reaches 1 or 0 from its original trival state, i.e., 0.5. The critical temperature is obtained by solving for the parameter T in the equation for V_{ijk}.

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

$$(V_{ijk})^{(n+1)}=a_{ijk}(0.5+0.5tanh(-\{\delta E/\delta V_{ijk}^{(n)}\}\{1/2T\}a_{ijk})$$

The superscript n indicates the iteration index. For each iteration, there are many neurons to be updated.

The updating can be done synchronously or asynchronously. The procedure to schedule the satellite broadcasting times using *MFT* is summarized below:

1. For the given SBS problem input the associative matrix and request vector.
2. Establish the weights.
3. Initialize the temperature to critical temperature determined.
4. Initialize the neurons with decimal numbers randomly distributed between 0.4 to 0.6.
5. Iteration process is set in a loop.
6. Update all the unclamped neurons using the derived equations.
7. Decrease the temperature value using the equation based on annealing schedule.
8. Calculate the total energy after the updating.
9. Repeat the process till there is no significant change in the Neuron State.

Simulations were carried out in Java programming language for a few examples of the *SBS* problems. We were able to obtain solutions for the optimization problems to which solutions could not be obtained using only Hopfield model.

8. Conclusion

A discussion has been carried out on the capabilities of neural networks in solving optimization problems. Examples of optimization problems like Traveling Salesman Problem and Satellite Broadcasting Scheduling problem have been explained. The neural network in combination with simulated annealing mechanism could solve large sized optimization problems. We have implemented the proposed method using Java for the Satellite Broadcasting Scheduling problem. Here we have used Hopfield net in combination with mean field annealing. This technique allows fast convergence and escape from local minima in search for global optima.

REFERENCES

[1] J.J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, in Proc. *Nat. Acad. Sci.*, 1982, vol. 79, pp. 2554-2558.

[2] J.J. Hopfield and D.W. Tank, Neural computation of decisions in optimization problems, *Biological Cybernetics*, vol. 52, pp. 141-152, 1982.

[3] D. Kunz, Channel assignment for cellular radio using neural networks, *IEEE Trans. Veh. Tech.*, vol. 40, pp. 188-193, Feb 1991.

[4] C. Lee Giles, Routing in optical multistage interconnection networks: A neural network solution, *IEEE Journal Lightwave Tech.* Vol. 13, pp. 1111-1115, June 1995.

[5] N. Ansari, A new method to optimize the satellite broadcasting schedules, *IEEE Trans. Neural Networks*, vol. 6, pp. 470- 483, Mar. 1995.

[6] Nobuo Funabiki and Seishi Nishikawa, A binary Hopfield neural network approach for satellite broadcast scheduling problems, *IEEE Trans. Neural Networks,* vol. 8, pp. 441-444, mar. 1997.

FORMING OF THE INVESTMENT PORTFOLIO USING SELF ORGANIZING NEURAL NETWORKS

G. Stankevicius

Dept. Process Control, Kaunas University of Technology, Studentu 48-327, Kaunas, LT-3031, Lithuania, giedrius@earthling.net

ABSTRACT

The problem of comparison of different companies is facing, when looking for possible candidates for the investment portfolio. Screening of the companies, using “well-known” trading strategy parameters, is one of the ways to solve this problem. Actually, much more companies appear on the list, than the trader is willing to buy. To define the best companies or group of the best companies self-organizing (Kohonen's) neural network could be used. Using fundamental financial parameters as inputs, the output of neural network forms the different groups of companies located into a number of disjoint clusters.

Then, by the special averaging technique, the 3D map of quality of investment could be formed. Investing portfolios could be formed by simple technical analysis approach.

Non-linear ranging technique was applied as an alternative to self-organizing neural network procedure. The certain meanings of weights were given to the factors, which characterize the companies. Then, by estimation of all weights, companies were assigned to their place in the general listing.

Four different portfolios were formed as a result of these researches. The performance of these portfolios showed which of the researched techniques gave better result. The real data from USA stock markets was used for the realization of the whole idea.

1. INTRODUCTION

Due to the globalization of financial markets, expansion of the electronic trade and the growth of information about the market, the specialists of investment funds more frequently try to use artificial intelligence methods for the market