Int. J. Appl. Math. and Comp. Sci., 1999, Vol.9, No.4, 921–937

STOCHASTIC NEURAL NETWORKS FOR FEASIBILITY CHECKING

GYÖRGY STRAUSZ*

Complex diagnosis problems, defined by high-level models, often lead to constraint-based discrete optimization tasks. A logical description of large, complex systems usually contains numerous variables. The first test of the logical description is typically to check the feasibility in order to know that there is no contradiction in the model. This can be formulated as an optimization problem and methods of discrete optimization theory can then be used. The purpose of the paper is to show that stochastic neural networks can be applied to this type of tasks and the networks are efficient tools for finding feasible or good-quality configurations. Boltzmann and mean-field neural networks were tested on large-sized complex problems. The paper presents simulation results obtained from a real application task and compares the performance of the neural networks being examined.

Keywords: optimization, neural networks, simulated annealing, mean-field approximation

1. Introduction

Logical analyses of large-sized systems are usually complex tasks. Several application problems can be described by constraints of discrete variables. Application tasks from the field of production line design and control, resource management and control, etc., belong to this class of problems. Feasibility tests of a constraint-based logical description can be solved as an optimization problem, where the optimization criterion is minimization of contradictions, in order to find a feasible configuration if it exists.

Hopfield-type neural networks were shown to be capable (Hopfield and Tank, 1984) of solving discrete optimization tasks. In the case of large-sized application tasks, where modelling the system often leads to NP-complete optimization problems, the deterministic Hopfield network gives poor results. However, stochastic extensions of the network will be proved to be efficient tools for such complex tasks. Simulated annealing can be used to avoid getting stuck in a local minimum. A stochastic extension of the original network is called the Boltzmann or simulated-annealing neural network.

* Department of Measurement and Instrument Engineering, Technical University of Budapest, H–1521 Budapest, Hungary, e-mail: strausz@mmt.bme.hu

The mean-field approximation is a powerful method for finding minimum points of cost or energy functions. The method can be applied to Hopfield-structure simulated annealing networks in order to speed up the slow convergence of the stochastic system. The mean-field neural network is based on a deterministic method that uses some results of spin-glass theory.

Section 2 briefly describes the neural network model that applied to solve the problem. Section 3 introduces the application problem that was used to test network performances. After this it will be shown how the problem can be mapped onto the introduced neural networks. In Section 5 a detailed comparison of the two networks is presented that examines the methods from both theoretical and practical points of view. Section 6 shows the simulation results that were obtained on a test task. Finally, the paper is complemented with conclusions.

2. Extensions of Hopfield-Type Neural Networks

The application task that will be presented in the following section can be considered as a constraint satisfaction problem. In order to take advantage of the features of connectionist machines, the problem is mapped onto a Hopfield-type neural network model (Aarts and Korst, 1989).

The structure of the Hopfield network is a very simple, one-layer recurrent model. Each unit in the layer is binary and fully connected to other units. It was shown that the system is asymptotically stable if the matrix of connection strengths (weights) is symmetric. The dynamic behaviour of the deterministic network can be described by the states of the units:

$$u_i := g(s_i) = g\Big(\sum_j w_{ij}u_j - \Theta_i\Big), \tag{1}$$

where u_i is the value of unit I, w_{ij} is the connection strength between units i and j, and g is the unit-step function:

$$g(u) = \begin{cases} +1 & \text{if} \quad u \geq 0, \\ 0 & \text{if} \quad u < 0. \end{cases} \tag{2}$$

The asynchronous unit updating method is used that corresponds to a local search algorithm for a stable state in the state space. In this case, during the simulation we choose randomly a unit and update its value according to eqn. (1).

In solving optimization problems the advantageous feature of the network can be used, namely we can assign an energy function to the system. The important property of this energy function (a Lyapunov function) is that it always decreases (or it remains constant) as the system evolves according to its dynamic rule. So as it was also shown in (Hopfield, 1982), the network converges to a local minimum of the energy function. The energy function can be written as

$$E = -\sum_i \sum_j w_{ij}u_i u_j. \tag{3}$$

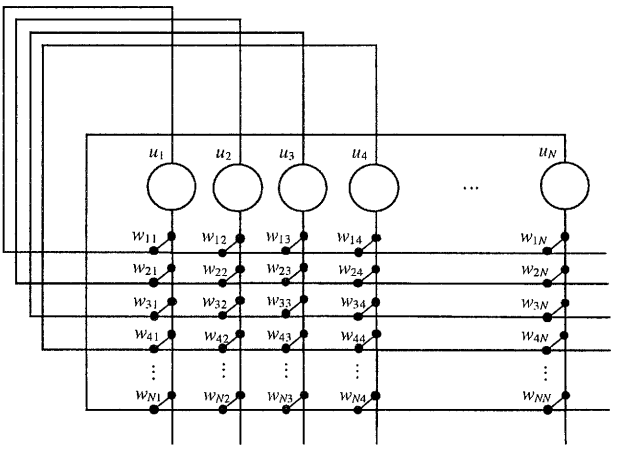

Fig. 1. The structure of the Hopfield-type neural network.

Theoretically, it is equivalent to minimizing the energy function or to simulating the processing of the network, but from the point of view of our application using the energy function is more natural, so in what follows we will focus on this approach.

The energy surface of the network is usually a multimodal function containing many local minima (Fig. 2.). The performance of the deterministic network is often not sufficient as it often gets stuck in a local minimum that sometimes belongs to a high-energy state. In order to find a global minimum or at least a 'low-energy state'a local minimum stochastic extension of the network is often used, called the Boltzmann machine (Kanter and Sompolinsky, 1987). The idea is to get rid of spurious local minima by using simulated annealing. The method is analogous to the annealing process used in steel production. Thermal noise is added to the process that forces units to change their values according to a probabilistic rule instead of the deterministic one. Simulations of a high temperature compel the network to behave randomly and a slow cooling process allows the system to settle down in a low energy state. A chosen unit of the Boltzmann network changes its state with the probability

$$P\left(u_i \to 1 - u_i\right) = \frac{1}{1 + \exp(-\Delta E/T)}, \tag{4}$$

where (in the case of binary units)

$$\Delta E = E(1 - u_i) - E(u_i) \tag{5}$$

and T is the 'temperature' parameter that controls the cooling process.

The introduced simulated annealing results in a stochastic search process on the energy surface, instead of the deterministic search carried out by the Hopfield network. Applying the method means that the deterministic neurons of the original network are replaced by stochastic units. The new stochastic network, called the Boltzmann machine, has some new parameters that control the stochastic behaviour

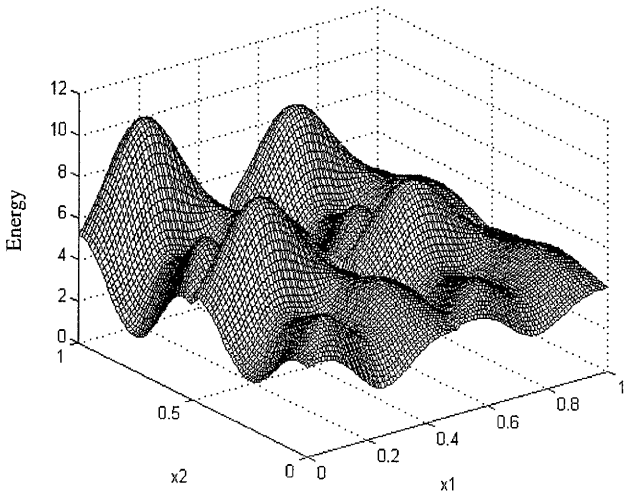

Fig. 2. Multidimensional energy surface.

of the system. Random changes were introduced to the system by the simulated temperature parameter and the control algorithm of the temperature parameter is called the cooling schedule.

The ultimate success of the simulation strongly depends on the parameters of the annealing process. The most important parameters can be divided into three groups:

- starting temperature,
- annealing speed,
- stopping criterion.

The starting temperature should be chosen so that the network behave almost randomly at the beginning. In the case of infinite-dimensional systems, the transition to a random state happens at a given temperature that is called the critical temperature. The parameter that is often referred to as the critical temperature for neural networks too, is determined by the magnitude of the coefficients (weight values) of the simulated system.

The speed of the annealing process should be set as slow as possible. Theoretically, an infinitely slow annealing results in settling the system in the optimal solution. An optimal cooling schedule is also known (Hajek, 1985), but this results in prohibitively long simulations. From the application point of view, the annealing speed is constrained by the time limit imposed by practical considerations. The speed is controlled by two parameters: the number of iterations at a given temperature and the temperature decay rate. There are several cooling functions that were tested to control the cooling process (Hajek, 1988). The most popular method is to use an exponential cooling schedule where the consecutive temperature values are calculated by multiplication by a c coefficient $(c < 1)$ (Ingber, 1995).

The stopping criterion is determined such that it is met when the network no longer changes its state. In asynchronous simulations this parameter is usually set to be $3N \div 10N$ iterations, where N is the number of units.

Simulated annealing usually produces good results, but the process is very time-consuming. As we do not have to know the actual values of the neurons during simulation, it is possible to obtain a faster method: instead of using stochastic variables, we can calculate their average values. This leads to the application of mean-field theory to the Hopfield model (Hertz *et al.*, 1991).

The average value of a unit can be calculated as follows:

$$\langle u_i \rangle \equiv P(u_i = 1)(1) + P(u_i = 0)(0) = g\big(2s_i/T\big) = \frac{1}{1 + \exp(-2s_i/T)}. \tag{6}$$

Applying the mean-field approximation, we estimate the average of functions with random variables by a function of the average of the variables. The input sum of the neurons is substituted with their average values:

$$\begin{aligned} s_i \to \langle s_i \rangle &= \left\langle \sum_j w_{ij} u_j \right\rangle = \sum_j w_{ij} \langle u_j \rangle, \\ \langle u \rangle &= g\big(2\langle s_i \rangle / T\big) = \frac{1}{1 + \exp(-2 \sum_j w_{ij} \langle u_j \rangle)}. \end{aligned} \tag{7}$$

The algorithm uses an approximation where the true fluctuating units are replaced with their average values. So instead of using stochastic binary units, we apply deterministic continuous-valued ones.

The mean-field approximation is proved to be precise for large-sized, homogeneous systems. The method was successfully used to model the behaviour of magnetic materials. For smaller systems the precision of the approximation is not known.

3. An Assignment Problem for Radio Link Frequencies

Several logical test problems can be described as optimization tasks. If the values of the system variables are to be selected from a finite domain and the logical structure of the problem is defined by constraints, the task becomes equivalent to a graph colouring problem. In the graph coloring problems we should assign colours (values) to the nodes (variables) of a graph such that the colouring satisfies the constraints assigned to the edges of the graph. A typical graph colouring problem is the radio link frequency assignment problem (RLFAP). The problem was introduced by a European research project (Tiourine *et al.*, 1995). Large-sized, real data bases are publicly available for the problem to test the developed methods.

As a matter of fact, there are several kinds of RLFAP according to the kinds of constraints that have to be taken into account and to the optimization criteria (if any).

The basic problem is only a constraint satisfaction problem. Let L be a set of radio links $(l_1, l_2, \ldots, l_M)$ and F_i be a set of frequencies $(f_i^1, \ldots, f_i^P)$ which can be assigned to l_i given a kind of devices which provide the radio link l_i.

The frequencies assigned to the links must meet mainly two kinds of constraints:

(i) Equality constraints:

$$|l_i - l_j| = d$$

This kind of constraints has to be met for two links providing a two-way radio contact.

(ii) Binary inequalities constraints:

$$|l_i - l_j| > d_{ij}$$

The constraints of this kind come from the computation of the electromagnetic wave propagation according to the landscape of the area where the related links have to be established.

The RLFAP can be described in a very general way as follows: Find a solution meeting all the constraints and minimizing the number of different frequency values used, and if there is no feasible solution, minimize a criterion based on the violation of the constraints.

The problem is considered to be difficult, and also good as a test problem for validating our method, because it is a large-sized NP complete one. It can be shown that the application problem is identical with the graph colouring problem that is a well-known and popular problem to test stochastic methods for finding approximate solutions of NP complete tasks.

There are numerous test problems and in all problems frequencies should be assigned to several hundreds of links (200–900). The frequency value for each link should be chosen from a given set of possible values. The number of different frequency values is 23 or 24, and the number of constraints that should be met is over a thousand in each test data base.

4. Mapping the Problem to Neural Networks

The RLFAP constraint satisfaction problem is formulated here as an optimization problem, where the primary aim of the optimization is to meet all the constraints. Let us consider first the basic problem where the optimization criterion is to minimize the number of violated constraints. In this case, we can use the number of violated constraints as the cost function of the problem. The solution can be characterized by a finite set of discrete variables (radio links) $L = \{l_1, \ldots, l_M\}$ and each of them should take a value from a finite domain (frequencies) $F = \{f_1, \ldots, f_P\}$.

A general approach to map a discrete optimization problem onto a neural network can be summarized by the following steps (Aarts and Korst, 1989):

- Formulate the optimization problem as a 0–1 programming problem by introducing new binary variables.

- Define a neural network such that the state of each neuron determines the value of the introduced binary variables.
- Define the connections and the corresponding weights of the neural network such that the energy function of the neural network is feasible and order-preserving with the cost function of the original optimization problem. Feasibility and order-preserving features apply here to the coherent corresponding states of the original optimization problem and the neural network.

The RLFAP is mapped onto a network such that a neuron being 'on' corresponds to a certain decision. Each unit indicates whether a variable of the problem takes a given value. A neuron indicates if a frequency value f_k is assigned to a given link l_i. (Two indices will be used for the neurons to indicate that each neuron belongs to a given link-frequency pair.) Thus we have

$$u_{ik} = \begin{cases} 1 & \text{if} \quad l_i = f_k, \\ 0 & \text{if} \quad l_i \neq f_k. \end{cases} \tag{8}$$

Constraints are embedded in the connection weights such that the global minimum of the network's energy function is characterized by a state where no constraint is violated. In order to restrict the space of allowed states, the so-called graded units (Peterson and Soderberg, 1989) were used that force exactly one unit to be active in each group. It automatically guarantees that each variable takes exactly one value in the solution.

In the case of a Boltzmann machine, forming groups of neurons can solve the problem of introducing graded units. For mean-field networks this leads to the Potts glass model (Peterson and Soderberg, 1989), where the nonlinear transfer function is modified by a vector generalization of the sigmoid function:

$$g(u_{ij}^l) = \frac{\exp(u_{ij}^l)}{\sum_k \exp(u_{ij}^k)}, \tag{9}$$

where l and k are the neuron indices in a group and

$$u_{ij} = -\frac{\delta E}{\delta u_{ij}} \frac{1}{T}. \tag{10}$$

Consequently, $u_{ij}^k \in \{0, 1\}$ and $\sum_k u_{ij}^k = 1$.

To find solutions that meet all the constraints of the RLFAP, we should choose the appropriate weights of the networks. An appropriate selection of the weights (for both Boltzmann and mean-field neural networks) can be described in the following way:

$$w_{ik,jl} = \begin{cases} 1 & \text{if } |f_k - f_l| < d_{ij} \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad w_{ik,ik} = 0.$$

It can then be easily seen that the energy function

$$E = \sum_{i,k} \sum_{j,l} w_{ik,jl} u_{ik} u_{jl} \tag{11}$$

is appropriate to solve the original problem, as it exactly gives the number of violated constaints.

So far it has been shown how to map the basic RLFAP to a neural network. In our case we had to complete another task as well. Among the possible solutions an optimal one should be found, where the optimization criterion is to minimize the number of used different values (frequencies). The difficulty of this optimization problem comes from the fact that there is a large number of different configurations (solutions) that belongs to the same frequency usage distribution. This results in large (horizontal) plateaux of the energy surface, which makes the search ineffective.

To accomplish this task, the model has been extended with new neuron-like elements and the energy function has also been completed with a second term. For each frequency value in the domain there is an integer element (t_v) in the energy function that outputs the number of variables currently taking the value. An efficient energy function term has been found for this problem as

$$E_2 = -c \sum_{v}^{D} \sum_{z}^{D} t_v t_z, \tag{12}$$

where D is the number of values in the domain and c is a sufficiently small factor to keep the magnitude of this term less than unity. (The reason to use c is that the primary criterion is still not to violate the constraints.)

Extending the energy function with this new term changes the energy function such that the gradient of the function is non-zero at each point. This makes even the stochastic search much more effective. It can be proved that the optimum of the energy function is not changed by the new term, but its real strength is that it gives a gradient that is proportional to the distance between the current and optimal states.

This term is feasible and, although not order preserving in a general case (if we consider the cost function to be the number of used values), it keeps the local order preserving feature for all possible consecutive states, which is sufficient when we use asynchronous updating.

5. Comparison of the Boltzmann and Mean-Field Approaches

Boltzmann and mean-field neural networks were introduced as possible tools for discrete optimization. Both the methods are extensions of the Hopfield neural networks that apply a stochastic search in order to avoid getting stuck in local minima. The tested methods are alternatives to each other, so it is an important question to answer which approach is more efficient. It is a hard task to give a clear answer to this question, even in the case of the example problem under consideration. There are

only a few papers which deal with this problem, and some of them arrive at controversial results (Elmohamed *et al.*, 1997). Therefore it is useful to present a detailed comparison that evaluates both the approaches using many points of view.

In the following, we shall examine and compare the theoretical background, precision, quality of the results, speed, convergence time and applicability of the neural networks in question. The statements are partly based on well-known, analytically proved features of the two neural networks and partly on the results of experiments that were carried out on the introduced large-sized, difficult problem instances.

Theoretical background of the methods. Although both the approaches contain heuristic elements, it is important to summarize what features of the algorithms are analytically proved. The Boltzmann machine that uses simulated annealing converges asymptotically to an optimal solution and even an optimal cooling schedule is known (Hajek, 1988). An algorithm is available that is able to give a limit of the precision for the results as a function of the executed iteration. Although the complexity of such algorithms is so high (they are usually exponential-time methods) that the implementation is impractical for real problems, these theoretical results provide guidelines for determining the values of the heuristic parameters of the methods.

In the case of the mean-field approach such analytical results are not available. The reason for this is that the precision of the mean-field estimation for finite-sized systems is not known. The applicability of the mean-field approximation depends not only on the size of the problem, but also on its homogeneity. The applicability of the approach is deduced from the results of the successive models built for large-sized, homogeneous systems in statistical physics.

The quality of the results. Boltzmann machines are capable of finding low-cost solutions, by a temperature controlled stochastic search on the cost function. Experiments show that it is very often sufficient to choose a cooling schedule that is faster in a magnitude than a theoretically necessary cooling schedule. In the case of the mean-field method we lose information with averaging and inaccurate determination of the average values. Therefore the search process is less sensitive to the details of the search process, which results in a decrease in the probability of finding low-cost solutions. It is especially true when the energy or cost surface contains many deep valleys of small diameters.

The speed of the search process. It is useful to check the speed of the methods from two points of view. First, it is reasonable to compare how long simulations are necessary at a given temperature to estimate the equilibrium state, and then, of course, it is important to see how long the whole search process is.

The mean-field network converges to a thermal equilibrium at a given temperature much faster, because this method does not follow each state change of the stochastic system. The mean-field network is still significantly faster if we take into account the fact that each iteration needs more calculations (in the case of software simulations) for this network.

Experiments show that the mean-field method is less sensitive to the choice of the parameters of the cooling schedule and the effectiveness of the mean-field network decreases less with the speed up of the cooling. This difference between the performances of the networks can be explained by examining the shapes of the energy surfaces of the two neural systems. The mean-field method leads to a continuos-valued Hopfield neural network and this type of network has a smoother energy surface than the discrete network containing the same weights. This results in faster convergence to minima of the energy surface.

It is worth noticing that there is a possibility to skip the slow annealing process if we apply the mean-field method. An alternative is to calculate the values of the mean-field variables for a well-chosen temperature and then to use a heuristic search method to find a low-cost solution close to the state that was provided by the neural network. This approach could be a very fast method if we were able to find an efficient heuristic algorithm for the given problem. Unfortunately, there is no general guideline as for how this algorithm should be constructed. The theoretical background for this approach is also unclear, i.e. it is not known what temperature to choose for the simulation and how to interpret the values of the neurons after the simulation.

Applicability of the methods. The Boltzmann neural network can be applied to both smaller and larger problems, and also to problems that can be considered as simpler or harder. Of course, the characteristics of the problem influence the simulation time and the ultimate success of the process. On the other hand, the mean-field method is based on the assumption that we can estimate the average values of the parameters of a large-sized, homogeneous system. Therefore, in the case of small-sized problems or if the foregoing assumptions do not hold, the mean-field approximation can result in solutions that are far from the optimal ones.

The invoked arguments should be considered differently according to a given application or circumstances. It is hard to give a general summary, because there are several contradictory arguments. For example, if we consider only the speed and precision as criteria, we can mention that the exactness of the more precise Boltzmann machine can be much worse if we should speed up the cooling. On the other hand, the speed of the faster mean-field method could be slowed down if we should apply a slower cooling schedule in order to find acceptable quality solutions. The question which method is faster if the criterion is a given precision or the question which method gives solutions of better quality in a given time interval usually can be answered only on the basis of simulations.

Among the above-mentioned problems regarding the mean-field neural network, those are of little importance which regard the applicability of the method for small-sized problems, because this kind of problems can be solved by several algorithms using a deterministic, exhaustive search. However, those disadvantages of the mean-field approach that pertain to the homogeneity of the problem have a significant effect on applications.

The reasoning has not led so far to a clear conclusion, but it can give arguments to consider selected methods or they can help us to design tests and experiments to

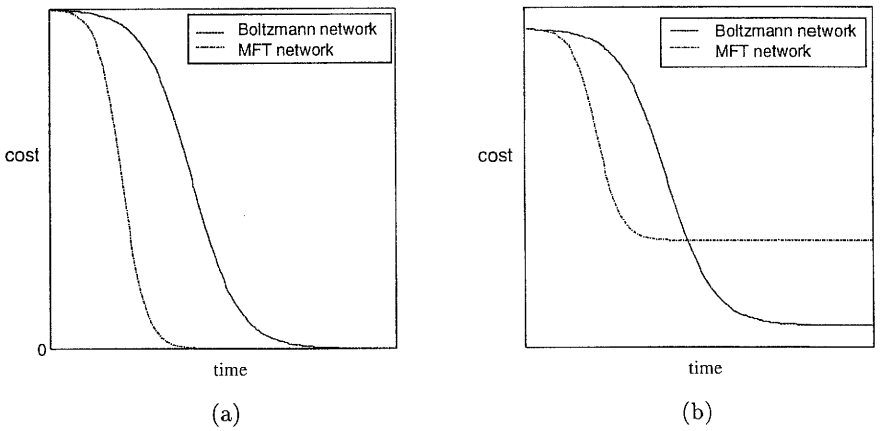

Fig. 3. Typical cooling curves for Boltzmann and mean-field networks: for simple (a) and difficult problems (b).

make the selection between the two networks easier in the case of a given application task.

As a summary of the above reasoning that contains several heuristic arguments, two pairs of curves can be seen in Fig. 3 that demonstrate a typical behaviour of the networks on problems with different difficulties. It can be seen that in the case of a large-sized but relatively simpler problems a higher speed of the mean-field networks can result in a better performance. Meanwhile, for more difficult problems the Boltzmann machine can give better-quality results, although there also exists a time limit till the mean-field approach provides better solutions.

If we have the possibility of choosing between the introduced two neural networks, then both of the following two approaches can be used. Tests can be started with the mean-field network and if acceptable quality results are not obtained, then it is reasonable to apply Boltzmann neural networks. The other possibility is to check first the results of the Boltzmann network and pass to the mean-field approach if we cannot get satisfying solutions in an acceptable time. Anyway, if we apply methods in which the theoretical background and the evaluation of the results contain several heuristic elements, then simulations are necessary for building heuristic rules for setting the parameters of the methods and for evaluating the performance of the systems. The following part of the paper presents a summary of the simulations that were carried out in order to support the statements of this part of the paper.

6. Simulation Results

Simulations were carried out on instances of the radio link frequency assignment problem. These problems are appropriate tests of the demonstrated methods, because these are large-sized, difficult constraint-based optimization problems. These problems are publicly available, so researchers working in different fields can compare

the efficiency of their methods. Unfortunately, there is no generally defined criterion that would help the evaluation of the results. Therefore we shall present several characteristics of the simulations and show the best currently known results.

First, we shall present optimization results of the examined problem instances. After a summary of the results obtained, the effects of the settings of the cooling parameters that will show an efficient strategy for the selection of these parameters are discussed.

In the EUCLID CALMA project eleven problem instances of the radio link frequency assignment problem were introduced. The main characteristics of the problem instances are summarized in Table 1. The size of the problems and the number of constraints defined for the problem are important parameters to describe the difficulties encountered, but it is not possible to define which problem should be considered harder or simpler based only on these parameters. Therefore, the selection of hard problems has been carried out by using simulation experiments. The best results presented in the table have been obtained with a tailored constraint-programming method by long simulations (Tiourine *et al.*, 1995). The presented results based on neural networks have been calculated within several minutes.

There are different optimization criteria for the different instances, but 10 of them are equivalent to the graph colouring problem, so they represent an important class of discrete optimization problems. For all the problems, the primary criterion was to satisfy the constraints of the problem or, if no feasible solution exists, to minimize the number of violated constraints. There were problem instances where the weights were defined for each constraint and the minimization criterion was a weighted sum of the violated constraints. In the case of feasible instances, the minimization of the number of different used frequencies is the optimization criterion.

Several simulations were carried out on each instance of the problem. The performance of the Boltzmann neural network and the mean-field neural network were tested. A summary of the results is presented in Table 2. The result data of the table were obtained via 1–2 minute long simulations on a Sun SparcStation 20 workstation. It can be seen from the table that the simulations with the applied relatively fast cooling schedules resulted in optimal or low-cost, good-quality sub-optimal solutions. Besides the best results, the average cost values of the solution after 10 experiments are also shown.

Both the Boltmann and mean-field neural networks were able to produce good-quality results, but the results obtained from the Boltzmann network were in most cases somewhat closer to the optimum. This difference was more significant in the case of more complex problem instances. As the simulations were not able to find optimal solutions for each problem instance, longer experiments were also carried out. The quality of the results was improved, but significantly better results were obtained only by significantly longer simulations (many hours). The experience fits to the known characteristics of this approach. Stochastic neural networks are efficient methods to find low-cost solutions fast, but they are not the best choice if we should find an optimal solution.

The success of the optimization depends on the appropriate settings of the parameters of the cooling schedule. A typical cooling process starts from a high temperature, where the network behaves stochastically, and with decreasing the temperature parameter the system converges to a deterministic network. Figure 4 shows typical characteristics of the simulated annealing. The acceptation rate shows the probability that the network behaves in an iteration in the same manner as a deterministic system would do. Thus the 0.5 acceptation rate belongs to a random state and the 1 value shows a deterministic behaviour. It can be seen that above the so-called critical temperature the system behaves randomly (the average cost of the visited solutions is constant). Close to the zero temperature the system behaves like a deterministic one and converges to a stable state.

Table 1. Characteristics of the CALMA data-bases.[1]

Number of the problem instance	Number of variables (radio link)	Number of values (radio frequencies)	Number of constrains	Number of neurons in the network	Best known solution	Feasible solution exists?
1	916	24	5548	21984	8	Yes
2	200	23	1235	4600	7	Yes
3	400	23	2760	9200	7	Yes
4	680	24	3967	16320	23	Yes
6	200	23	1322	4600	3437	No
7	400	23	2865	9200	343594	No
8	916	24	5744	21984	262	No
9	680	24	4103	16320	15571	No
10	680	24	4103	16320	31516	No
11	680	24	4103	16320	11	Yes

As was discussed in the first part of the paper, simulated annealing is used to avoid the poor behaviour of the deterministic Hopfield network. The main problem with the original deterministic system is that it realizes a local search on the energy surface that converges to a closest local minimum. Table 3 compares the behaviour of the deterministic and stochastic networks. It can be seen that even with the applied fast cooling schedule the performances of the stochastic systems are significantly better.

The best choice for the starting temperature is the critical temperature where the system ignores the gradient information during the search and behaves perfectly randomly. Figure 5 shows the effects of different choices of the starting temperature. The figure contains three curves, one belonging to an annealing started from the

[1] In the column of the best known solutions, the number of used frequencies is shown in the case of feasible instances and the weighted sum of the violated constrains is shown for the other instances.

critical temperature, the second representing a simulation started from a significantly higher temperature and the third started from a significantly lower temperature than the critical one. It can be seen that there is no reason to initialize the temperature parameter higher than the critical value, because the performance of the network does not become better, but the simulation time is increased. The reason behind this is that the search process does not have any correlation with the structure of the problem over the critical temperature, therefore the iterations executed in this temperature region do not increase the probability of finding optimal solutions. The figure also demonstrates that if one starts the simulation from a low temperature, the final result will be poorer, because the system will not be able to search the whole energy surface, but just its small part.

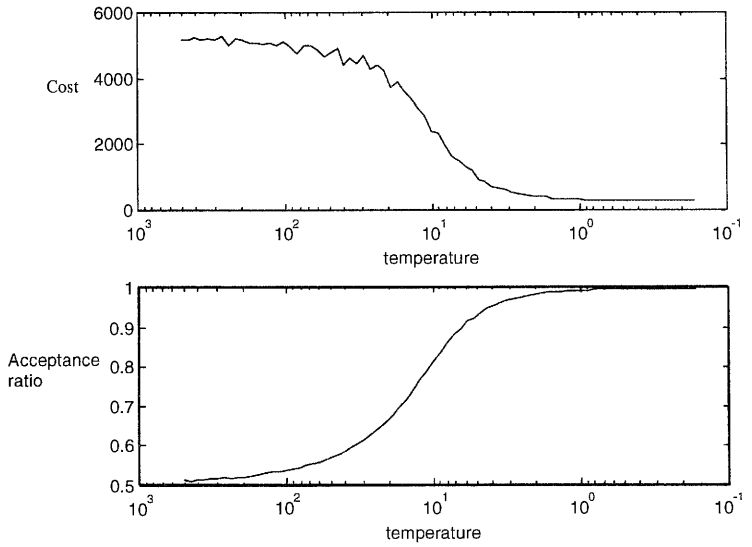

Fig. 4. Characteristics of the simulated annealing.

No algorithm for constructing an optimal finite-time cooling schedule is known. There are several heuristically built cooling schedules to determine the best speed of the cooling, but generally no proof is known to determine which one is the best. Experience shows that the shape of the cooling curve has a small effect on the performance of the search, therefore the simplest method (that was introduced in the first part of the paper), the exponential (also called geometrical) cooling is usually used in applications. Figure 6 shows the effect of the cooling speed. After the experiments that are represented by the curves of the figure, it can be concluded that a very fast cooling schedule deteriorates the performance of the search process. The expected cost of the solution increases with the slow-down of the cooling process, but this improvement is not significant over a given speed. Unfortunately, there is no general method to determine what cooling parameters refer to a 'very fast' cooling, so only previous experiences and experiments made it possible to find good settings.

Table 2. Simulation results on the CALMA data-base.[2]

Number of the problem instance	Best solution	Difference from the best known solution	Average cost solution	Best solution	Difference from the best known solution	Average cost solution
	Boltzmann network			MFT network		
1	9	1	10.5	11	3	11.5
2	7	0	7.1	7	0	7.3
3	8	1	8.7	9	2	9.5
4	23	0	23	23	0	23
6	4278	24%	5120	5961	73%	6315
7	487547	41%	553468	573234	66%	591344
8	282	7%	328	363	38%	393
9	15951	2%	16121	17342	11%	16789
10	31516	0	31588	34234	8%	33124
11	14	3	16	16	5	17

Table 3. Comparison of the deterministic and stochastic networks.

Number of the problem instance	Best/average solution with the Hopfield network	Best/average solution with the Boltzmann network	Best/average solution with the mean-field network
1	15/17	9/10.5	11/11.5
2	9/10	7/7.1	7/7.3
8	555/839	282/328	363/393

7. Conclusions

Some effors towards testing the efficiency of stochastic neural networks for large-sized discrete-type optimization problems have been presented. The Boltzmann and mean-field networks have been used to solve large-sized, difficult problems. For all the tested problems the quality of the solution is good.

The results achieved are similar for both the types of networks, but the Boltzmann machines give more precise solutions. Boltzmann machines are also proved to provide

[2] In the column of the best known solutions, the number of frequencies used is shown in the case of feasible instances and the weighted sum of the violated constrains is given for the other instances.

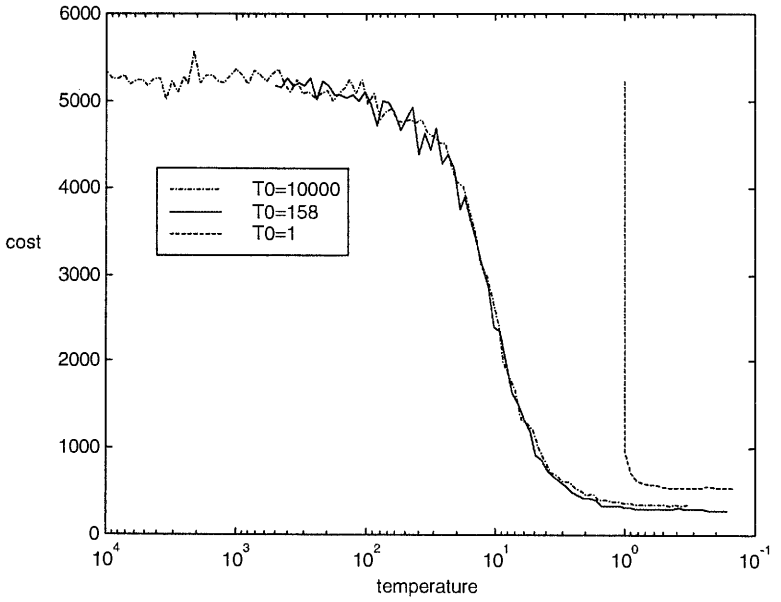

Fig. 5. Simulations started from a different initial temperature (Problem 8).

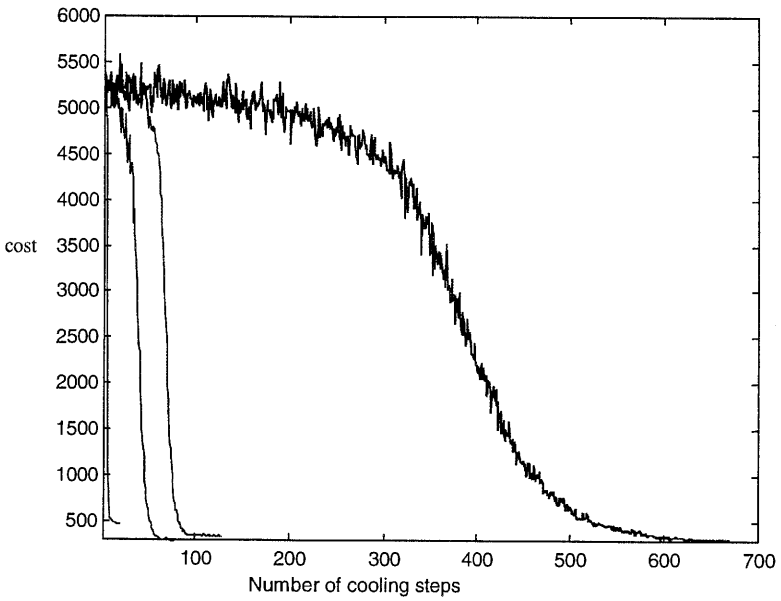

Fig. 6. The annealing processes with different cooling speeds (Problem 8).

more robust methods that can be applied to a wide variety of problems. On the other hand, the mean-field approach offers the possibility of speeding up the simulations with little loss of the attainable quality.

One of the most important advantages of the methods is the possibility of a parallel implementation. Simulations could be executed significantly faster if the networks were implemented on a dedicated hardware or on a general, massive parallel structure.

The tested methods contain several heuristic elements. An important point is the selection of cooling parameters. It is a theoretically difficult task to define an optimal, finite-time cooling schedule for a given problem. It is planned to continue conducting research to build cooling schedules such that they take the complexity of a given problem into account. Such an algorithm could help us to avoid too long simulations for simpler problems and to avoid poor results obtained from too short simulations of hard tasks.

References

Aarts E.H.L. and Korst J. (1989): *Simulated Annealing and Boltzmann Machines.* — New York: Wiley.

Cosnard M. and Goles E. (1997): *Discrete state neural networks and energies.* — Neural Networks, Vol.10., No.2, pp.327–334.

Elmohamed S., Coddington P. and Fox G. (1997): *A comparison of annealing techniques for academic course scheduling.* — NPAC Technical Report, SCCS–777, Syracuse University, USA.

Hajek B. (1985): *A tutorial survey of theory and application of simulated annealing.* — Proc. 24th Conf. *Decision and Control*, Ft. Lauderdale, pp.755–760.

Hajek B. (1988): *Cooling schedules for optimal annealing.* — Math. Ops. Res., Vol.13, No.2, pp.311–329.

Hertz J., Krogh A. and Palmer R.G. (1991): *Introduction to the Theory of Neural Computation.* — New York: Addison-Wesley.

Hopfield J.J. (1982): *Neural networks and physical systems with emergent collective computational abilities.* — Proc. Nat. Acad. Sci. USA, Vol.81, pp.3088–3092.

Hopfield J.J. and Tank D.W. (1984): *Neural computation of decisions in optimization problems.* — Biol. Cybern., Vol.52, pp.141–152.

Ingber L. (1995): *Simulated annealing: practice versus theory.* — J. Math. Comp. Modell., Vol.18, No.11, pp.29–57.

Kanter I. and Sompolinsky H. (1987): *Graph optimization problems and the Potts glass.* — J. Physics, Vol.A20, pp.L673–L679.

Peterson K. and Soderberg, B. (1989): *A new method for mapping optimization problems onto neural networks.* — Int. J. Neur. Syst., Vol.1, No.1, pp.3–22.

Tiourine S., Hurkens C. and Lenstra J.K. (1995): *An overview of algorithmic aproaches to frequency assignment problems.* — Techn. Rep., Eindhoven University of Technology.

Received: 6 January 1999
Revised: 20 April 1999