ELSEVIER

European Journal of Operational Research 108 (1998) 283–292

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

Improved exploration in Hopfield network state–space through parameter perturbation driven by simulated annealing

G. Papageorgiou [a], A. Likas [b], A. Stafylopatis [a,*]

[a] *Department of Electrical and Computer Engineering, National Technical University of Athens, 9 Iroon Polythecniou Str., 157 73 Zographou, Athens, Greece*

[b] *Department of Computer Science, University of Ioannina, 451 10 Ioannina, Greece*

Received 15 March 1996; revised 1 August 1997

Abstract

An approach is presented for treating discrete optimization problems mapped on the architecture of the Hopfield neural network. The method constitutes a modification to the local minima escape (LME) algorithm which has been recently proposed as a method that uses perturbations in the network's parameter space in order to escape from local minimum states of the Hopfield network. Our approach (LMESA) adopts this perturbation mechanism but, in addition, introduces randomness in the selection of the next local minimum state to be visited in a manner analogous with the case of Simulated Annealing (SA). Experimental results using instances of the Weighted Maximum Independent Set (MIS) problem indicate that the proposed method leads to significant improvement over the conventional LME approach in terms of quality of the obtained solutions, while requirin\cr & g a comparable amount of computational effort. © 1998 Elsevier Science B.V.

Keywords: Discrete optimization; Neural networks; Simulated annealing; Hopfield network; Boltzmann machine

1. Introduction

A large class of problems arising from real world situations can be formulated as optimization problems and thus qualitatively described as a search for the "best" or "optimal" solution among a finite or countably infinite number of alternative solutions. Several interesting combinatorial optimization problems are considered computationally *intractable*, that is, exhaustive search algorithms require at least non-deterministic polynomial time to obtain optimal solutions (class NP) [1,2]. Nevertheless, often, what is truly desired is a very *good* solution, computed in a short time, and not the nominally best. This explains why sub-optimal, polynomial time algorithms have attracted interest. The *Hopfield neural network* model [3,4], the *Simulated Annealing* method [5] and closely related models, such as the *Boltzmann Machine* [6], have proved effective in providing near-optimal solutions to hard optimization problems.

* Corresponding author. Fax: +30-1-772-2109; e-mail: andreas@softlab.ece.ntua.gr.

PII S0377-2217(97)00375-5

The Hopfield network, both discrete and analog, has been widely used for solving combinatorial optimization problems. The objective (or cost) function and the problem constraints are appropriately mapped in an *energy function* and the network is expected to find a configuration which minimizes this energy function. An important property of the Hopfield model is that starting from any initial state, it will always settle to a stable state. However, the Hopfield model gets easily trapped in local minimum states, something that decreases its efficiency, especially in problems of large size. The current work is focused on the *discrete* Hopfield model, where at each step one neuron is randomly selected and examined for possible change of its binary state.

A significant amount of research has been reported on improving the performance of the Hopfield network. The most successful method has been derived from the integration with Simulated Annealing [5]. By introducing a probability for the acceptance of a new state, the network occasionally accepts transitions to states with higher energy and, thus, it can escape from local minima. This combination of the Simulated Annealing algorithm with the discrete Hopfield network is known as the Boltzmann Machine model [6]. Asymptotic convergence to the global minimum state has been proved for Simulated Annealing and, thus, for the Boltzmann Machine. The Boltzmann Machine approach is effective but requires large computational time as the problem size increases. Parallel versions of the model or closely related methods have successfully dealt with this limitation [7–12].

A different consideration is made by Peng et al. [13]. Instead of using randomness in the procedure of accepting a new state, they use a sophisticated method for the generation of a new state. According to this method, the parameters of the Hopfield network, i.e. the connection weights and the thresholds, are perturbated (through noise injection) and produce a new network that, when it relaxes, provides the original network with a new initial state. The original network runs again and reaches a possibly new local minimum state which is accepted if the corresponding energy value is lower than the starting one. Then, the whole procedure is repeated from the beginning. This new method, called the Local Minima Escape (LME) algorithm, provides a mechanism for escaping from high energy local minima but may be trapped in local minimum states with long basin of attraction, that are far from the global minimum state. This is not a serious problem for small problem instances, but it is more clear in the case of large problem instances where there are many regions of the state space that contain such minima and therefore the algorithm may be easily trapped in states with high energy values.

In the approach presented in this paper we adopt the parameter perturbation mechanism introduced by the LME algorithm, but provide an integration of this approach with the Simulated Annealing methodology, which is applied at the new state acceptance part of the search process. The synergy of weight perturbation with the probabilistic acceptance of new local minimum states leads to a more flexible search procedure that is able to explore the state–space more adequately, due to the capability of easily escaping from flat shallow local minima. Experimental results with difficult large instances of the Weighted Maximum Independent Set problem enforce this consideration.

Section 2 provides a brief summary of the Hopfield-type models that are commonly used. The LME algorithm is described in Section 3, while our approach (called LMESA), which is a modified version of the LME algorithm, is presented in Section 4. Experimental results obtained from the application of our method to the Weighted Maximum Independent Set problem are discussed in Section 5. Finally, conclusions and ideas for further research are contained in Section 6.

2. The Hopfield and related models

The basic idea in the Hopfield model is to encode the objective function and the problem constraints in terms of an appropriate *energy function* which can be minimized by the network architecture.

The *discrete* Hopfield network performs local search in the discrete space $\{0,1\}^m$. The energy

function that corresponds to a discrete Hopfield neural network with m units, connection weights w_{ij} (with $w_{ii} = 0$) and threshold values θ_i has the form

$$E(\vec{v}) = -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} v_i v_j w_{ij} - \sum_{i=1}^{m} v_i \theta_i, \tag{1}$$

where $\vec{v} = (v_1, \ldots, v_m)$ is the state of the network and $v_i \in \{0, 1\}$. The network operates *sequentially*, that is, at each time instant one unit is selected randomly and the difference in the network's energy, that will result if the selected unit i changes state, is computed. Assuming symmetrical weights ($w_{ij} = w_{ji}$) this energy difference can be written as

$$\delta E_i(\vec{v}) = (2v_i - 1)\left(\sum_{j=1}^{m} w_{ij} v_j + \theta_i\right). \tag{2}$$

If $\delta E_i(\vec{v}) < 0$, then the change is accepted, otherwise it is rejected. For symmetrical weights it is ensured that the network will settle into a state corresponding to a local minimum of the energy function [3], where $\delta E_i(\vec{v}) > 0$ for all $i = 1, \ldots, m$. This final state of the network, however, rarely happens to coincide with or be near the global minimum state, due to the locality of the search. Therefore, research effort is turned to the development of local minima escape techniques.

Simulated Annealing (SA) is a stochastic optimization technique, inspired from condensed matter physics. It uses a stochastic hill-climbing algorithm with the added ability to escape from local minima in the state–space [5], where conventional methods usually get trapped. At each step a new state is considered randomly and the cost difference, that the state transition would cause in the objective function, is computed. Let δC be the difference of the cost of current state and new state, that is $\delta C =$ newcost − oldcost. The probability that a candidate move is accepted is determined by either the *logistic* or the *Metropolis* criterion. In the *logistic* case the change is accepted at temperature T with probability $p_{\text{new}} = 1/(1 + \exp(\delta C)/T)$, while in the *Metropolis* case the change is accepted with the above probability p_{new} only if $\delta C \geqslant 0$, otherwise it is accepted with probability 1 [14]. We shall use the term *trial* to denote the operations of next state generation and the computation of δC and p_{new}. The acceptance of a suggested transition, will be referred to as an *update*. Hence, depending on the acceptance probability, a trial may be eventually followed by an update. The acceptance probability is controlled by a *temperature* parameter, T. At the beginning of the process the value of T is high, thus allowing state transitions that lead to increase of cost. As the algorithm proceeds, T is decreased according to a *cooling schedule*, so that the probability for such transitions finally tends to zero and the algorithm converges to a stable state. SA is commonly described as a sequence of Markov chains, each corresponding to a temperature value. Every computational step of a chain starts only after the previous step has been completed, thus the operation of SA is strictly sequential.

The Boltzmann Machine (BM) is based on an integration of the dynamics of the discrete Hopfield model with the SA methodology [6]. The objective function to minimize is the energy E of the discrete Hopfield network. At each step a new state is considered by randomly selecting a unit i of the discrete Hopfield network and computing the difference $\delta C = \delta E = (2v_i - 1)(\sum_{j=1}^{m} w_{ij} v_j + \theta_i)$. An update takes places according to either the logistic or the Metropolis criterion. Since the BM optimizer constitutes a special case of the SA method, the results concerning asymptotic convergence to the global minimum point under certain assumptions, that have been proved for SA [5], carry over to the BM case too [6]. The same holds for the finite time implementations of the algorithm that attempt to approximate the global minimum. The operation of the BM is strictly sequential and may require large computation time as the size of the problem grows. Moreover, in order for the annealing to be effective, the stationary distribution (or at least a quasi-equilibrium distribution) must be restored at each temperature, thus, sufficient state transitions must take place and consequently a large number of trials are required.

3. The local minima escape method

The LME algorithm was introduced in [13] as an algorithm for improving the exploration

capability of the Hopfield network and avoiding the problem of local minima. As described in Section 2, the Hopfield network operates as a local search algorithm. Given an initial state, it will stabilize at a local minimum state nearby. In order to locate the global minimum of the energy, an initial state must be found that happens to be within the basin of attraction of the global minimum state. In the LME approach, the mechanism for escaping from a local minimum state is based on the search for an initial state that lies out of the basin of attraction of the current local minimum. Such an initial state will finally lead to a new local minimum state which may be of lower energy than the previous one. If this happens, the new state is adopted as current state of the search, otherwise we search for another promising initial state. The search for appropriate initial states is performed through suitable perturbations of the parameters of the network, i.e. weights and thresholds. Therefore the LME algorithm can be considered as a combination of a network disturbing technique and the Hopfield network's local search property.

Let us consider a Hopfield network H and assume it has relaxed at some local minimum state. Through random perturbation of the connection weights and thresholds [13], a new Hopfield network H' can be obtained:

$$w'_{ij} = (1 + \alpha^w n^w_{ij}) w_{ij} + \beta^w n^w_{ij}, \quad 1 \leqslant i,\ j \leqslant m,\ i \neq j, \tag{3}$$

$$\theta'_i = (1 + \alpha^\theta n^\theta_i)\theta_i + \beta^\theta n^\theta_i, \quad 1 \leqslant i \leqslant m, \tag{4}$$

where α^w, β^w, α^θ and β^θ are positive constants which control the strength of disturbance, while n^w_{ij} and n^θ_i are standard Gaussian noises. To ensure the symmetry of connection weights after disturbance, we consider $n^w_{ij} = n^w_{ji}$. Since the networks H' and H have the same architecture, the states of the two networks can be easily mapped to each other, unit by unit.

At each iteration of the LME algorithm a new disturbed network H' is randomly specified using the above equations and the current local minimum state of H is set as the initial state of H'. The network H' converges to a stable state. This stable state of H' is then used as the new initial state of H, which operates and converges to a possibly new local minimum state. If the new local minimum of H is of lower energy than the previous one, it is considered as the new current local minimum state. Otherwise the former local minimum state of H is kept. Iterations are repeated until a prespecified maximum number of iterations are exceeded or if no state better than the current one is discovered for a certain number of consecutive iterations. Moreover, in order for the method to be effective, an appropriate choice of the parameters α^w, β^w, α^θ and β^θ controlling the strength of the perturbation must be made (experimentally).

4. LME search augmented by simulated annealing

Both the LME method and the BM approach suggest ways of escaping from local minimum states of the Hopfield network. They differ in two basic aspects. The first is in new state generation from the current state. In the basic BM formulation the new state is generated by randomly selecting one unit and changing its state. In the LME case there is sophistication in new state generation resulting from the perturbation in the network's parameter space (weights and thresholds). This leads to new states that may be far from the current one, thus allowing the network to escape from local minima.

The second difference lies in the way of performing the transition from the current state to the new generated state. In the BM case a transition is accepted according to the SA methodology, i.e. transitions are allowed that may lead to energy increase. On the contrary, the acceptance mechanism for the LME method is deterministic, i.e. a new state is accepted only if the corresponding energy is lower than the energy of the current state. This may lead to local minimum states that are locally optimal in the sense that, although the energy of these states may be far from the global energy minimum, the perturbation mechanism of the LME method fails to locate a new state of lower cost.

The approach proposed here is an extension to the LME method based on an acceptance mechanism that incorporates randomness and provides greater flexibility in the exploration of

the problem state–space, by allowing transitions that lead to energy increase. This is obtained by augmenting the LME method through the incorporation of an SA scheme operating at a higher level.

As already described in Section 3, in each iteration of the LME method, relaxation takes place twice: first for the perturbated network and then for the original network, which is initialized at the stable state of the former. Let ΔE denote the energy difference between the stable state attained by the original network on the last relaxation and the state of the original network at the end of the previous iteration (from which started the relaxation of the last perturbated network). If $\Delta E < 0$, i.e. the network has settled to a better local minimum, then this state is accepted and becomes the current state of the network. If $\Delta E \geq 0$, i.e. a state of lower quality has been obtained, the new state is *not* rejected, as is the case in the original LME method, but is accepted with probability $p = 1/1 + e^{\Delta E/T}$. This probability is controlled by the *temperature* parameter T. As T is initially high, the probability p may take high values, so many transitions that lead to energy increase are accepted. This leads to a more effective examination of the state–space, since it is possible to follow many alternative paths in the solution

- *Map problem on Hopfield network H.*
- *Initialize temperature T.*
- *Initialize output vector $\vec{v}(0)$ to random binary values.*
- *Let network H relax and evaluate the stable state $\vec{v}_{stable}(0)$.*
- $t = 1$.
- *Repeat steps 1–7 until terminating criterion is satisfied:*
  1. *Perturbate network H and produce H'.*
  2. *Set the current stable state of H as initial state of H' (i.e. $\vec{v'}(t) = \vec{v}_{stable}(t-1)$) and let network H' relax.*
  3. *Set the stable state of H' as initial state of H (i.e. $\vec{v}(t) = \vec{v'}_{stable}(t)$), let network H relax and evaluate the new stable state $\vec{v}_{temp}(t)$.*
  4. *Set $\Delta E = Energy(\vec{v}_{temp}(t)) - Energy(\vec{v}_{stable}(t-1))$.*
     - *If $\Delta E < 0$ then set $\vec{v}_{stable}(t) = \vec{v}_{temp}(t)$.*
     - *If $\Delta E \geq 0$ then with probability $p = \frac{1}{1+e^{\Delta E/T}}$ set $\vec{v}_{stable}(t) = \vec{v}_{temp}(t)$ or with probability $1-p$ set $\vec{v}_{stable}(t) = \vec{v}_{stable}(t-1)$.*
  5. *Update the best found configuration $\vec{v}_{best}$.*
  6. *Periodically update control parameter T, according to cooling schedule.*
  7. $t = t + 1$
- *Output best configuration $\vec{v}_{best}$.*

Fig. 1. The LMESA algorithm.

space and discover local minimum states of better quality. As the algorithm proceeds, T is decreased according to a cooling schedule, so as to allow the algorithm to settle to a promising region of the solution space.

It is interesting to note that the states considered during the annealing process are *all* stable states of the network since they are produced through relaxation of the discrete Hopfield model. Hence, at the higher level, we move only in the space of local minimum states, i.e. in a confined state–space. This allows us to consider as solution not the final stable state but the *best* stable state attained by the network during execution of the algorithm. The role of the annealing process in our case is to provide a second level of perturbations and, thus, enhance state–space exploration. The proposed LME algorithm augmented by Simulated Annealing (LMESA) is summarized in Fig. 1.

5. Experiments

5.1. Problem formulation

The effectiveness of the proposed approach has been tested on instances of the Weighted Maximum Independent Set (MIS) problem. The MIS constitutes an important discrete optimization problem and the solution of many other problems (for example Set Partitioning, Set Packing, Set Covering etc. [15]) can be reduced to the solution of this one.

The formulation of the MIS problem (weighted case) is the following: Consider an undirected graph $G = (V, E)$ where V (with $|V| = m$) is the set of vertices and E denotes the set of edges. Let also A denote the adjacency matrix of graph G, i.e., $a_{ij} = 1$ if $(i, j) \in E$, otherwise $a_{ij} = 0$. An *independent set* V' of this graph is a subset of V that contains vertices not connected to each other. If $c: V \to \mathbb{R}^+$ is a cost function assigning a cost to each vertex, the MIS problem is to find the independent set V' of maximum cost, where the cost of the set V' is defined as $f_c(V') = \sum_{k \in V'} c_k$.

A neural network architecture suitable for the MIS consists of n nodes with the following specification of weights w_{ij} and threshold values θ_i [6,15]:

$$\theta_i = c_i,$$

$$w_{ij} = \begin{cases} -\{\max\{\theta_i, \theta_j\} + \epsilon\} a_{ij} & \text{if } i \neq j, \\ 0 & \text{if } i = j, \end{cases}$$

where ϵ is a very small positive value (which is set equal to 0.5 in our experiments). This specification of weights and thresholds ensures that every one-change local minimum state corresponds to an independent set of the graph. Each such set is maximal in the sense that no other vertex can be added to it without violating the disjointness constraint. Moreover, the resulting energy function is *order preserving* [6] in the sense that the lower the final energy value, the better the cost of the final solution.

5.2. Performance evaluation

For our experiments we considered four graphs with 100, 200, 500 and 1000 vertices, respectively, that were constructed by deciding with probability 0.1 for each pair of vertices whether there would be an edge connecting the vertices of this pair. The cost of each vertex was an integer value specified through uniform selection in the range between 20 and 50.

Experiments were conducted for both the LME and LMESA models. We considered $\alpha^w = \alpha^\theta = 0.5$ and $\beta^w = \beta^\theta = 0.1$ for all tests. These values were empirically found to give the best results.

The annealing schedule that was used in the tests with the LMESA model has the following logarithmic form:

$$T_k = \frac{T_{k-1}}{1 + \log f(k)},$$

where $f(k) = f(k-1)(1+r)$ (with $f(0) = 1$) and $T_0 = 25$, $r = 0.0001$ denote the initial temperature and the reduction rate respectively. We considered that one trial is performed at each temperature step and the annealing terminates if no new state has been accepted for 50 consecutive iterations or if a maximum number of 1000 iterations has been reached.

For the LME algorithm we adopted at first exactly the same termination condition (algorithm

LME_{50}). Since the quality of the obtained solutions was not very good, we tried to increase the exploration time by increasing the number of allowed consecutive iterations without update to 100 (algorithm LME_{100}). This has resulted in improvement of the solution quality. It must also be noted that any further increase in the number of allowed iterations did not lead to solutions of significantly better quality, hence no further performance improvement of the LME method could thus be expected.

The results of our experiments are summarized in Tables 1 and 2. Table 1 displays average values obtained from 20 runs for each problem instance while the results in Table 2 are the best and worst cases obtained in these 20 runs. Table 1 depicts the improvement in solution quality provided by the proposed method, compared to the basic LME approach under both termination conditions. Table 2 shows that the LMESA algorithm exhibits better performance in terms of the worst and the best solutions found for each problem size. It must be noted that there is significant superiority of the LMESA algorithm in terms of the quality of the worst solution found, hence the method exhibits considerable robustness and reliability.

Finally, it is interesting to compare the behavior of the two methods when they start from exactly the same state. We carried out experiments for each method and for the graphs of 500 and 1000 vertices, recording the decrease of the network's energy. Figs. 2 and 3 clearly illustrate that by allowing the acceptance of transitions that lead to energy increase, solutions of better quality are finally obtained. In these figures LME stands for the LME_{100} variant.

5.3. Parallelization issues

The execution of the above experiments showed that both methods (as any other method treating problems of this size) require a rather large amount of computational time. The usual remedy to excessive computational time has been the parallelization of the slow algorithms. This has been successful for SA and related techniques [7–12]. Some of them [11,12] can be directly applied to the proposed LMESA algorithm, as they are general methods for the parallelization of SA.

In particular, in [11] two techniques are presented. According to the first one, each processor is al-

Table 1
Average comparative results for the LME and LMESA methods

Method	Graphs							
	100 vertices		200 vertices		500 vertices		1000 vertices	
	Iter.	Cost	Iter.	Cost	Iter.	Cost	Iter.	Cost
LME_{50}	115	1053	120	1428	128	1792	170	2081
LME_{100}	204	1066	270	1442	290	1918	298	2135
LMESA	462	1077	488	1497	361	1983	238	2194

Table 2
Best and worst case cost comparative results for the LME and LMESA methods

Method	Graphs							
	100 vertices		200 vertices		500 vertices		1000 vertices	
	Best	Worst	Best	Worst	Best	Worst	Best	Worst
LME_{50}	1085	907	1534	1248	1969	1593	2221	1958
LME_{100}	1085	926	1544	1308	2036	1789	2331	2016
LMESA	1085	994	1544	1411	2042	1894	2371	2029

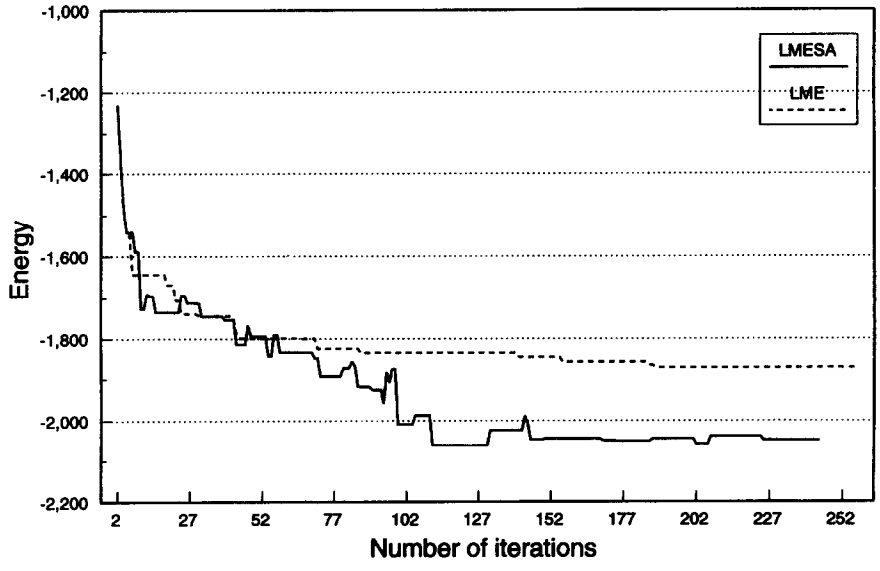

Fig. 2. Comparison of energy evolution for graph with 500 vertices.

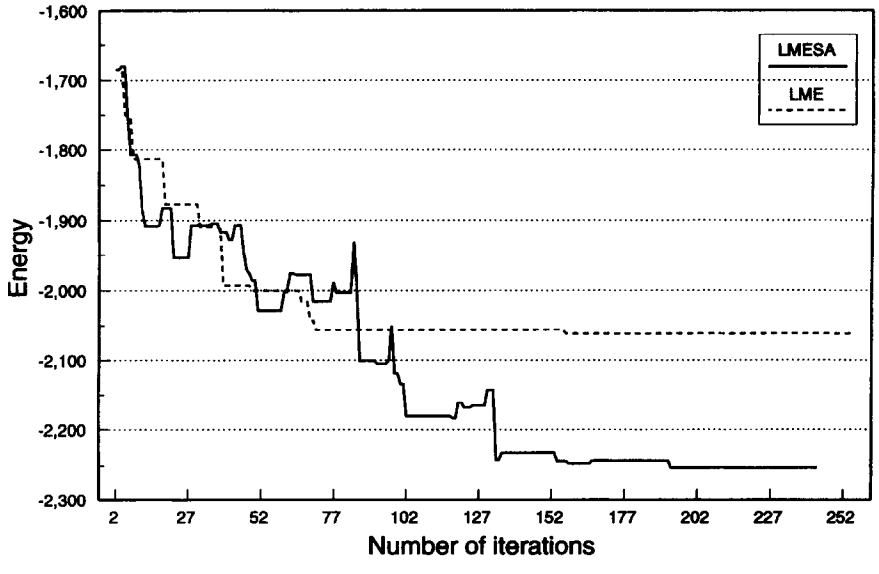

Fig. 3. Comparison of energy evolution for graph with 1000 vertices.

lowed to evaluate only one move and waits until all the other processors complete their evaluation. Then one of the accepted moves is chosen randomly, the processors' memories are updated with the new configuration and the next evaluation step takes place. According to the second one, which is especially effective at low temperatures, the processors perform trials in parallel, until one of the

processors in use accepts a move. When an accepted move is adopted, the processors are synchronized, their memories are updated with the new configuration and the next evaluation step takes place. An alternative to the above approaches could be the selection of the best new state among the ones generated by the processors in use.

In [12] the technique of *generalized speculative computation* is introduced. The method is based on the concurrency technique of speculative computation, in which work is performed before it is known whether or not it is needed. Each processor receives a loop index at runtime and performs three steps: generate next state, evaluate it and decide to accept it or not, according to the state that is *presumed* to be current. At the end of the three steps it raises a flag to indicate its decision result. The decision results are then collected and the decision result of the lowest numbered loop index is selected to initiate the next level. To ensure that the parallel version generates the same decision sequence as sequential SA, the same sequence of seeds is used to generate pseudo-random numbers.

Preliminary experiments on the application of the techniques presented in [11] (and some variants of them) suggest that it is possible to obtain solutions of good quality in a much smaller number of iterations, therefore significant benefits in terms of execution time can be expected.

6. Conclusions

A discrete optimization approach has been presented, which is based on the Hopfield neural network. The parameter perturbation mechanism introduced by the LME algorithm was adopted and augmented through the integration with the SA methodology. The new method LMESA constitutes a more flexible search procedure and is able to explore the state-space more adequately than the original LME model, as it is shown by the experimental results on instances of the MIS problem. In particular, these results indicate that the proposed method leads to significant improvement over the conventional LME approach in terms of quality of the obtained solutions, not only on average but also in the best and worst case.

Future research concerning this method can follow various directions. Modified versions of the method can be implemented on parallel machines and speed up the LME and LMESA algorithms. Alternative functions could be investigated to allow for a more intelligent perturbation of the Hopfield network. Of interest is also a theoretical analysis of the method as far as convergence to the global optimum is concerned. Finally, consideration must be given to the parameters that control perturbation, which might improve exploration ability if they are not constant. Work on the above topics is currently in progress.

References

[1] M. Garey, D. Johnson, Computers and Intractability: A Guide to The Theory of NP-Completeness, Freeman, New York, 1979.

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

[3] J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proc. Nat. Acad. Sci. USA 79 (1982) 2554–2558.

[4] J.J. Hopfield, D.W. Tank, Neural computation of decisions in optimization problems, Biol. Cybernet. 52 (1985) 141–152.

[5] S. Kirkpatrick, S. Gellat, M. Vecchi, Optimization by simulated annealing, Science 220 (1983) 671–680.

[6] E. Aarts, J. Korst, Simulated annealing and Boltzmann machines, A Stochastic Approach to Combinatorial Optimization and Neural Computing, Wiley, New York, 1989.

[7] R. Azencott, (Ed.), Simulated Annealing: Parallelization Techniques, Wiley, Chichester, UK, 1992.

[8] D.R. Greening, Parallel simulated annealing techniques, Phys. D 42 (1990) 293–306.

[9] A. Likas, G. Papageorgiou, A. Stafylopatis, A parallelizable operation scheme of the boltzmann machine optimizer based on group updates, Neural, Parallel & Scientific Computations 3 (4) (1995) 451–465.

[10] A. Likas, A. Stafylopatis, Group updates and multiscaling: An efficient neural network approach to combinatorial optimization, IEEE Trans. Systems Man Cybernet. 26 (1996) 222–232.

[11] P. Roussel-Ragot, G. Dreyfus, A problem independent parallel implementation of simulated annealing: Models and experiments, IEEE Trans. Computer-Aided Design 9 (8) (1990) 827–835.

[12] A. Sohn, Parallel N-ary speculative computation of simulated annealing, IEEE Trans. Parallel Distrib. Syst. 6 (10) (1995) 997–1005.
[13] M. Peng, N.K. Gupta, A.F. Armitage, An investigation into the improvement of local minima of the hopfield network, Neural Networks 9 (7) (1996) 1241–1253.
[14] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, Equation of state calculations by fast computing machines, J. Chem. Phys. 21 (6) (1953) 1087–1092.
[15] V. Zissimopoulos, V. Paschos, F. Pekergin, On the approximation of NP-complete problems by using the boltzmann machine method. The Cases of Some Covering and Packing Problems, IEEE Trans. Comput. 40 (12) (1991) 1413–1419.