Applied Intelligence 7, 215-225 (1997)

A Connectionist Approach for Solving Large Constraint Satisfaction Problems

A. Likas[1], G. Papageorgiou[2] and A. Stafylopatis[2]

[1] Department of Computer Science, University of Ioannina, 451 10 Ioannina, Greece.
[2] Department of Electrical and Computer Engineering, National Technical University of Athens, 157 73 Zographou, Athens, Greece.

Abstract

An efficient neural network technique is presented for the solution of binary constraint satisfaction problems. The method is based on the application of a double-update technique to the operation of the discrete Hopfield–type neural network that can be constructed for the solution of such problems. This operation scheme ensures that the network moves only between consistent states, such that each problem variable is assigned exactly one value, and leads to a fast and efficient search of the problem state space. Extensions of the proposed method are considered in order to include several optimisation criteria in the search. Experimental results concerning many real-size instances of the Radio Links Frequency Assignment Problem demonstrate very good performance.

1 Introduction

Constraint satisfaction problems (CSPs) [1] represent typical combinatorial problems that are usually solved with search techniques based on backtracking algorithms [2, 3]. In general, these techniques work well for small and medium problem sizes but encounter difficulties in dealing with large problem instances because of the huge state space that must be searched [4]. A main characteristic of such approaches is that they require a centralised control by the problem solver, thus making difficult a parallel implementation.

Distributed approaches to CSPs are based either on neural networks [5, 6, 7, 8, 9], or on stochastic learning automata [10, 11, 12]. Neural network approaches to solving CSPs, which are typically based on the Hopfield model [13, 14], have the property of not advocating a hierarchical control. As Hopfield networks suffer from the existence of local energy minima that do not represent feasible solutions, an annealing process is often introduced in the operation of the network and the resulting Boltzmann Machine model [15] generally succeeds in escaping from such equilibrium points. In the case of CSPs, however, it is not very efficient and cannot provide solutions to large problem instances in reasonable time.

The Guarded Discrete Stochastic network approach [5] constitutes an extension to the pure discrete Hopfield network. It manages to escape from local minima by adding an auxiliary network of 'guards' which push the network out of such states. This approach has the drawback that results concerning guaranteed convergence of Hopfield-type networks do not apply and oscillation phenomena may appear, since connections between the guard units and the network units are not symmetric.

The Extended Stochastic Automata method [10], which consitutes an extension of the stochastic learning automata approach to CSPs intoduced in [11], considers an underlying Hopfield-like network structure that improves the effectiveness of the learning algorithm. It is inherently parallel and asymptotically converges with probability one to a problem solution. It has proved quite effective in dealing with medium size problems.

Other neural network approaches to optimisation and constraint satisfaction problems are based on mean field approximation algorithms applied to binary or multistate (Potts) neurons [16, 17]. Also, simulated annealing has been applied to the solution of constraint satisfaction problems [18], as well as several techniques based on genetic algorithms [19, 20, 21].

The work presented in this paper concerns a method with good convergence properties that is based on the group update method applied to the operation of the discrete Hopfield network (or the corresponding Boltzmann Machine) [22]. First we provide a brief description of the group update technique in its general formulation, and next we present the proposed method for the solution of binary constraint satisfaction problems. Then we consider some extensions of the proposed basic model, which are used for incorporating optimisation requirements in the search. Finally performance results are presented concerning a real-world CSP, the Radio Links Frequency Assignment Problem, that illustrate the effectiveness of the method in dealing with very large problem instances.

2 Group Updates

Consider a binary Hopfield-type neural network with M units. Let w_{ij} denote the connection weights of the network (with $w_{ii} = 0$) and θ_i the threshold values of the units. The energy corresponding to state $\vec{y} = (y_1, \ldots, y_M)$ ($y_i \in \{0, 1\}$) is the following

$$E(\vec{y}) = -\frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M} y_i y_j w_{ij} - \sum_{i=1}^{M} \theta_i y_i \tag{1}$$

The operation of the network is *asynchronous*. At each time instant one unit is selected randomly and the change in the network's energy that will result if the selected unit i changes state is computed:

$$\delta E_i(\vec{y}) = (2y_i - 1)(\sum_{j=1}^{M} w_{ji} y_j + \theta_i) \tag{2}$$

In the case of the pure Hopfield network, if $\delta E_i(\vec{y}) < 0$ then the change is accepted, otherwise it is rejected. We shall use the term *trial* to denote the above operation. Changing the state of a unit will be referred to as *update*. Hence, a trial is eventually followed by an update. The above described scheme refers to the *single-update operation* of the network.

If the weights are symmetrical ($w_{ij} = w_{ji}$) it is ensured that the network will settle into a state corresponding to a local minimum of the energy function [13], where $\delta E_i(\vec{y}) \geq 0$ for all $i = 1, \ldots, M$. In solving optimisation problems, our objective is to reach the global energy minimum.

The Boltzmann Machine Optimiser [15] is based on the idea of applying an annealing schedule [23] to the operation of the Hopfield network from an initial high temperature down to a temperature value near zero. If the value of the temperature parameter T is not zero, the network does not reject all transitions that lead to an increase in energy. More specifically, in the case where $\delta E_i(\vec{y}) > 0$ the state of the unit with index i changes according to some probability function which in general depends on the quantity $\exp(\delta E_i(\vec{y})/T)$. As T tends to zero, the probability also tends to zero and the update rule becomes the same as in the pure Hopfield case.

The notion of *group update* [22] constitutes a generalisation of the notion of single update in binary Hopfield-type networks. Instead of selecting a single unit to test for update, we select a group containing a number of units. In analogy with the single unit case, first a trial is performed by calculating the difference in the energy that would result if the state of *all* units in the group were altered simultaneously. Then a decision must be made as to whether a group update must take place, i.e. whether the states of *all* these units must change or not. Clearly, group operation differs from *synchronous* operation [15], which consists of simultaneously performing individual trials and updates and also from *block-sequential* operation [24]; according to the latter operation mode, the state vector is divided into subvectors which are sequentially updated, where within each subvector individual updates are performed simultaneously (synchronous operation within each subvector). It could be observed that the idea of group updating bears an analogy to the approach of *simultaneous perturbation* that has been developed and analyzed in the context of stochastic approximation [25].

Let $\vec{y} = (y_1, \ldots, y_M)$ be the state of the network at some time instant during operation; we consider a group G consisting of m units and perform a trial on this group. If the trial is successful, a group update takes place and the new state $\vec{z} = (z_1, \ldots, z_M)$ of the network will be such that

$$z_i = \begin{cases} y_i + u_i & \text{if } i \in G \\ y_i & \text{if } i \notin G \end{cases} \tag{3}$$

where $u_i = 1 - 2y_i$, $i \in G$. Equation (3) simply expresses the fact that units in G change state. It is proved in [22] that the following proposition holds:

Proposition 1. *The change in the energy of the network caused by a group update of group G from state $\vec{y}$ is given by*

$$\Delta E_G(\vec{y}) = \sum_{i \in G} \delta E_i(\vec{y}) - \frac{1}{2} \sum_{i \in G} \sum_{j \in G} u_i u_j w_{ij} \tag{4}$$

where $\delta E_i(\vec{y})$ is the change in the network's energy in case unit i is selected for a single *update from state $\vec{y}$.*

The energy of the new state $\vec{z}$ can be obtained from (1) with z_i in place of y_i. Using (3) and based on the symmetry of the weights we obtain the above result after performing some algebra and considering the expression for $\delta E_i(\vec{y})$ from (2).

It can be easily verified that the change in the network's energy due to a group update from state $\vec{y}$ is equal to the sum of the individual energy changes that would result if the units of the group sequentially changed state starting from $\vec{y}$ and following an arbitrary order. This expresses conservation of energy and provides an intuitive interpretation of the proposition.

In order for an operation scheme based on group updates to be successful, appropriate groups of units must be identified, which represents a highly problem dependent task. This topic, which is often referred to in the context of simulated annealing as the problem of choosing the correct move class, is very important. (Any problem could be solved instantly if the right move was chosen.) In many problem mappings there are constraints implying that certain groups of neurons should have the same state, or that some pairs of neurons should have opposite states. If we start the operation of the Hopfield-type network from an arbitrary initial state satisfying these requirements and proceed by performing the appropriate group updates, we always move between valid states. Thus, the search becomes more constrained and effective than if the whole state space were explored. In such a case, despite the fact that group updates may be computationally more expensive, there is a significant performance benefit, since the number of iterations required to reach an acceptable state is much smaller.

Grouping schemes can be *state-dependent* or *state-independent*. In the former case, the composition of the group that is considered for update at each time step is determined based on the current state of the network, while in the latter case the groups are specified from the beginning and remain fixed during the operation of the network. In the case of state independent grouping it is possible to reduce the state space and construct a coarse scale network having the property that single updates in that network simulate group updates in the original fine scale network [22]. In this paper we present an application of the group update technique in a case of state-dependent grouping and we show that this technique is very efficient in solving constraint satisfaction problems.

3 A Double-Update Scheme for Solving Binary CSPs

A Constraint Satisfaction Problem (CSP) consists of finding an assignment of values to a collection of variables that satisfies a set of constraints among them. In the following we shall assume that constraints are binary, i.e. each constraint specifies the values that cannot be taken on simultaneously by a given pair of variables.

We consider a general binary CSP involving a set of N variables $X_1, \ldots, X_N$ and a set of binary constraints C. Each variable X_k $(k = 1, \ldots, N)$ assumes values from a finite domain D_k. Let d_{kl} denote the l-th value in the domain D_k. The set of binary constraints C contains pairs of d_{kl} values that are incompatible with each other. For example, $(d_{kl}, d_{mn}) \in C$ means that the l-th value of D_k and the n-th value of D_m cannot be simultaneously assigned to variables X_k and X_m respectively. A solution to a given CSP is an assignment of values to all the variables so that none of the constraints is violated. It is important to note that we are dealing with the problem of finding one such solution and not all the existing solutions.

The representation of a CSP in terms of a binary Hopfield network (or Boltzmann Machine Optimiser) can be done easily [5]. The network units are arranged in a matrix structure with a matrix row corresponding to each of the variables. The number of units in each row k is equal to the number of values that can be assigned to variable X_k, i.e. it is equal to the size $|D_k|$ of the domain D_k. Under this representation, a unit (k, l) of the network corresponds to the value d_{kl} and the assignment of d_{kl} to X_k is represented by $y_{kl} = 1$, where y_{kl} is the state of unit (k, l).

The connection weights $w_{kl,mn}$ and the threshold values θ_{kl} are chosen so that the network states corresponding to solutions of a given CSP are equilibium states of minimum energy:

$$\theta_{kl} = \beta \tag{5}$$

$$w_{kl,mn} = \begin{cases} -\alpha & \text{if } (d_{kl}, d_{mn}) \in C \\ -\eta & \text{if } k = m \text{ and } l \neq n \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where β, α and η are positive constants and $\alpha, \eta > \beta$. According to the above weight specification, a negative connection of strength α is imposed between every pair of units related to values that cannot be simultaneously assigned to the corresponding variables. This negative connection (with strength greater than the threshold value) does not allow that both units be active in equilibrium. In addition, in order to satisfy the requirement that at most one value can be assigned to each variable, a negative connection (which also has strength greater than the threshold) is imposed between any pair of units belonging to the same row. The above specification of weight and threshold values ensures that a state corresponding to a problem solution is an equilibrium point of minimum energy, which is equal to $-N\beta$. Indeed, since only inhibitory connections exist between units, there will be no two connected units such that both are active. Therefore, the first term of the energy function as expressed by equation (1) will be zero and only the second term involving the threshold values of the N active units will be present. Obviously, when the number of active units is less than N or some pair of connected units are active, the energy will be greater than the above value.

If the pure binary Hopfield network is used to solve this problem, it will be found that, in most of the runs, the network settles into equilibrium states corresponding to local minima of the energy function (having energy values $-k\beta$ with $k < N$). In these states, there exist some rows having exactly one active unit that is not connected to other active units in different rows. There also exist, however, rows having no active units at all, all units belonging to these rows being inhibited by the active units of the other rows. It is obvious that such local minimum states are not useful since no value is assigned to the variables corresponding to inactive rows.

The Boltzmann Machine Optimiser can be used for evading such local minima and searching for a global one. Although the Boltzmann Machine approach manages to lead the network out of local minimum traps and yields valid problem solutions for small and medium problem instances, it cannot be generally considered as highly effective for large-size problems.

We now consider a modification to the operation of the Hopfield network based on the group update technique, which leads to a much more effective search of the problem state space. We start from an initial network state having the property that there is exactly one active unit in each row. We shall refer to states having the above property as *consistent* states. Since this property constitutes a desirable feature of the final equilibrium network state, we perform state transitions so that the network always passes through consistent states. This is an idea that has been considered in other applications exhibiting analogous features, like for example in [26]. To achieve this, we consider that at every time instant a *double update* is performed, i.e. a group update concerning a group of two units.

Consider that, at a given time instant, the network is in a consistent state $\vec{y}$ and let $(i, i^\star)$ denote the index of the active unit in each row i ($i = 1, \ldots, N$). We randomly select a row k and a unit (k, l) in that row different from $(k, k^\star)$, i.e. such that $y_{kl} = 0$. The selected unit along with unit $(k, k^\star)$ constitute the group that will be tested for update. If this trial is successful, the network will move to a consistent state and we will set $k^\star = l$. The energy difference caused by the group update concerning the selected pair of units can be computed by direct application of equation (4):

$$\Delta E_G(\vec{y}) = \delta E_{kk^\star}(\vec{y}) + \delta E_{kl}(\vec{y}) + w_{kk^\star,kl}\ , \qquad G = \{(k, k^\star), (k, l)\} \tag{7}$$

The individual energy changes can be readily obtained:

$$\delta E_{kk^\star}(\vec{y}) = \beta + \sum_{m \neq k} w_{mm^\star,kk^\star} \tag{8}$$

$$\delta E_{kl}(\vec{y}) = -\sum_{m \neq k} w_{mm^\star,kl} + \eta - \beta \tag{9}$$

Since $w_{kk^\star,kl} = -\eta$, equation (7) yields for the energy difference caused by the double update:

$$\Delta E_G(\vec{y}) = \sum_{m \neq k} w_{mm^\star,kk^\star} - \sum_{m \neq k} w_{mm^\star,kl}\ , \qquad G = \{(k, k^\star), (k, l)\} \tag{10}$$

We define a *double-update equilibrium point* as a network state such that for each unit (k, l) with $l \neq k^\star$ it is valid that $\Delta E_G(\vec{y}) \geq 0$, where $G = \{(k, k^\star), (k, l)\}$. Using equation (10), the above equilibrium condition can also be written as

$$\sum_{m \neq k} w_{mm^\star,kk^\star} \geq \sum_{m \neq k} w_{mm^\star,kl}$$

The following proposition refers to the significance of double-update equilibrium points:

Proposition 2. *The states of the network that correspond to valid problem solutions are global minimum double-update equilibrium points.*

The proof of the proposition is given in the Appendix.

To avoid the possibility of getting trapped in local minima, the Boltzmann Machine approach has been adopted. At each time step a unit (k, l) (with $l \neq k^\star$) is selected and the energy difference $\Delta E_G(\vec{y})$ is computed for $G = \{(k, k^\star), (k, l)\}$. If $\Delta E_G(\vec{y})$ is negative, then units (k, l) and $(k, k^\star)$ change state. Otherwise, these changes are accepted with probability

$$p = \frac{1}{1 + \exp(\Delta E_G(\vec{y})/T)} \tag{11}$$

where T is the value of the temperature.

Several considerations are in place concerning the efficiency of the presented technique. A serious drawback of the single update scheme is that, if during network operation there is an active unit in some row, it is useless to select an inactive unit of the same row for update. Indeed, because of the strong inhibition applied to this unit by the active unit, no transition can take place. In order for a transition to occur, first the active unit must change state and then it is possible to select one of the inactive units for update. This sequence of transitions is essentially what is performed in one step under the double-update scheme. Another advantage of our technique is that we always move between consistent states. This property, in addition to favouring an efficient search, has an immediate impact on the computational complexity of the approach. In order to make a single update, an amount of computation must be performed which is proportional to the number of units in the network (in our case equal to $\sum_{k=1}^{N} |D_k|$). On the other hand, the double-update scheme ensures that there is exactly one active unit in each row. Therefore, the knowledge of the $k^\star$ index for each row k, reduces the computation required to perform a double update to the order of $2N - 1$ as can be observed from equation (10). This reduced complexity is of major significance for the overall performance of the proposed approach.

4 Extensions to the Basic Model

The above formulation of the double-update approach is sufficient for providing feasible solutions to constraint satisfaction problems. In general, there exist more than one feasible solutions to such problems. Therefore, satisfaction of constraints may be accompanied by additional requirements that limit the number of acceptable solutions. Two typical such requirements concern (a) minimisation of the number of different values used in the assignment and (b) minimisation of the largest value used.

The model presented in the previous section can be extended to provide optimisation of the above extra criteria, in addition to satisfying problem constraints. These extensions are based on the inclusion of an additional row with units that correspond to the elements of the union of the domains of the variables. Depending upon the applied optimisation criterion, we can choose appropriate (symmetric) connection weights and threshold values for these additional units, so that minimisation of the network energy leads to optimisation of the respective criterion. The units of the extra row follow the single-update mode of operation and are randomly selected for update. This extension preserves the convergence and stability properties of the network.

Consider that the network includes an extra row, which will be referred to as the 'zeroth row', representing the union of the domains of the problem variables. We assume that variables X_k $(k = 1, \ldots, N)$ refer to the same context and the corresponding domains D_k are subsets of a global domain D. A unit of the extra row will be denoted by $(0, j)$ $(1 \leq j \leq K)$, where K is the cardinality of D, and its state will be denoted by y_{0j}. We consider a common indexing of values in D in ascending order. The association of values in individual domains with the global indexing is performed by means of the function $r(k, l)$, which maps the lth value of D_k to its index in the global domain D. To distinguish the extra row from the original rows of the network, the latter will be referred to as actual rows and the corresponding units as actual units.

4.1 Minimisation of the Number of Different Values

To minimise the number of different values used in the assignment, the extra row is specified as follows. Each unit $(0, j)$ is connected to all actual units representing the same value, namely to all units (k, l) such that $r(k, l) = j$. The strength of all such connections is the same and equal to $\gamma > 0$. Also, each unit $(0, j)$ is characterised by a threshold value equal to $-\delta$, where $\delta > 0$ and $\gamma > \delta$ (Figure 1). Clearly, an extra unit can be activated as soon as at least one variable has been assigned the corresponding value. Since the existence of extra units affects the dynamics of actual units, we shall assume that $\alpha > \beta + \gamma$, so that the normal constraint satisfaction operation of the network is not disturbed. It should be noticed that the effect of extra units on the activation of actual units is moderated by the double-update mode of operation.

We are interested in states such that all problem constraints are satisfied and the

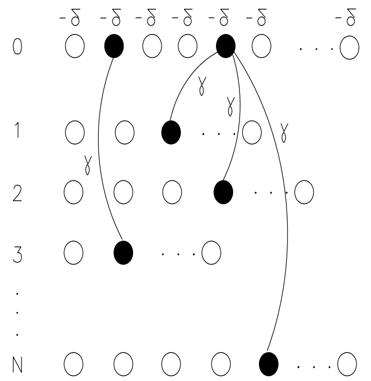

Figure 1: Minimisation of the number of different values

respective extra unit is active for every value used. It can be verified that under the above specification of weights and thresholds such a state will be an equilibrium point with energy equal to $-N\beta - N\gamma + Z\delta$, where Z is the number of active units of the extra row. The second term in the expression accounts for the N connections between active actual units and active extra units, whereas the third term involves the thresholds of active extra units. The notion of equilibrium refers here to both single-update (for all units) and double-update (for actual units). As explained in the case of the basic approach, any constraint violation would result in a higher energy value. Also, the energy would increase if any extra unit changed state. Therefore, Z attains its minimum possible value, which optimises the extra requirement, at an equilibrium point of minimum energy.

The expression giving the energy difference is now slightly modified as shown in the Appendix.

4.2 Minimisation of the Largest Value

In this case, each extra unit $(0, j)$ is connected to all actual units representing greater values, namely to all units (k, l) such that $r(k, l) > j$. The strength of these connections is equal to $-\gamma$ and the threshold of all extra units is taken equal to δ, where γ and δ are positive and $\gamma > \delta$ (Figure 2). We shall assume that $\gamma < \beta$. An extra unit is inhibited when at least one variable has been assigned a value larger than the value represented by

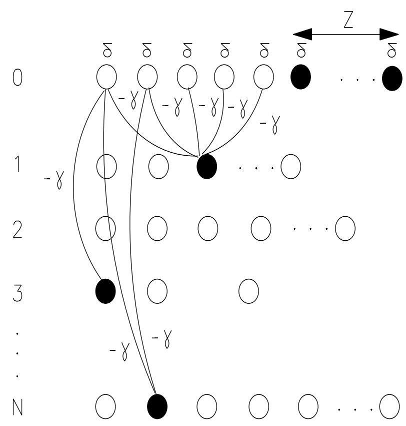

Figure 2: Minimisation of the largest value

that unit. In fact, actual units can put extra units into the inactive state, whereas extra units hinder actual units that tend to assume large values.

We are interested in states representing problem solutions, in which all extra units corresponding to values greater than or equal to the largest value assigned are active, whereas the remaining extra units are inactive. It can be verified that such states constitute equilibrium points with energy equal to $-N\beta - Z\delta$, where Z is the number of values of the domain D which are greater than or equal to the largest value used in the assignment. Only the term involving the threshold values of the extra units is present in this case, since active extra units are not connected to active actual units. Therefore, Z attains its maximum possible value, which minimises the largest assigned value, at an equilibrium point of minimum energy.

Details concerning computation of the energy difference are given in the Appendix.

5 Performance Evaluation of the Double Update Scheme

The Radio Links Frequency Assignment Problem (RLFAP) is a problem that occurs in many civil and military applications [27, 28]. It consists of assigning radio frequencies to communication links, subject to a set of constraints, so that no interference occurs. However, it may be impossible to satisfy all constraints, in which case trying to minimise the number of violated constraints is a more realistic goal. The RLFAP is a variant of the k-Graph Colouring Problem and, thus, is NP-complete [29]. In the attempt to solve the RLFAP, several techniques have been applied so far [8, 9, 18, 20, 21], which generally address small-size instances of the problem. In this paper, we consider application of the double-update approach to the solution of the CELAR problems, which constitute a collection of hard large-size instances of the RLFAP.

5.1 The CELAR Data Set of Problems

In the CELAR problems[3] a set of possible frequencies is given to operate the radio links. Each of the radio links has to be assigned one of the frequencies, while satisfying a large number of constraints. For any two 'neighbouring' links i and j, if f_i (f_j) is the frequency assigned to link i (link j), two types of constraints occur:

- $|f_i - f_j| > c_{ij}$ Inequality constraints
- $|f_i - fj| = c_{ij}$ Equality constraints

where c_{ij} is a given number.

The concept of 'neighbouring' links, here, not only depends on the fact that the links lie geographically close to each other, but also on various electromagnetic characteristics (such as propagation, transmission power, etc), which may result in electromagnetic incompatibility (at certain frequencies) between the links.

¿From the CELAR data set that contains eleven problems (scen01–scen11) we will present results on six of the problems (scen01–scen05, scen11), which all have feasible solutions (all constraints can be satisfied). For problems scen06–scen10 there exists no feasible solution, so these are constraint *optimisation* problems and are not considered in the present paper.

Each variable (link) can be assigned a frequency from its respective allowed domain of frequencies. There are seven available frequency domains for these problems. The following data set is given:

scen01: 916 variables (links), 5548 constraints, no variable pre-assigned.
scen02: 200 variables, 1235 constraints, no variable pre-assigned.
scen03: 400 variables, 2760 constraints, no variable pre-assigned.
scen04: 680 variables, 3967 constraints, with 280 variables pre-assigned.

[3] The data was provided by the French "Centre d' ELectronique de l' ARmement" (CELAR).

scen05: 400 variables, 2598 constraints, no variable pre-assigned.
scen11: 680 variables, 4103 constraints, no variable pre-assigned.
For the above problems we have to find solutions that satisfy all the constraints. Generally, there exist more than one feasible solutions. In addition, we have to minimise the number of different frequencies used in the assignment (for scen01–scen04) or to minimise the largest frequency value used (for scen05). No such extra requirement holds for scen11.

5.2 Experimental Results

It is easy to represent a Radio Links Frequency Assignment Problem as a binary CSP. Each radio link is considered as a variable, so, if N radio links have to be given a frequency, then N rows are used. The domain of each variable can be mapped to the corresponding row, that is, if the i-th radio link can take values from a set of k frequencies, then the i-th row will have k nodes, from which only one can be 'on' at a time. One extra row (zeroth row) represents the union of all domains and is used for the minimisation of either the number of different frequencies or the largest frequency assigned.

In what concerns the characteristics of the annealing schedule, the following logarithmic scheme has been adopted [15]:

$$T_n = \frac{T_{n-1}}{1 + \log f(n)}$$

where T_n denotes the temperature at the n-th step of the schedule, and $f(n) = f(n-1)(1+r)$ with $f(0) = 1$, r being the temperature reduction rate. The starting temperature T_0 and the number of double-update trials per temperature were different for each problem instance. Table 1 shows the various configurations used, which were determined through systematic experimentation. The frequency of trials on the zeroth row was one per temperature step. The reduction rate was the same for all instances, $r = 10^{-4}$. The annealing procedure terminated if no double update had taken place for 4 consecutive temperature steps.

The value of the parameters α, γ and δ was fixed to 2.0, 1.9 and 1.8 respectively for all the experiments, whereas β was properly selected following the case. As can be seen from the equations concerning the computation of energy differences, however, the values of β and η need not be specified in our implementation.

In order to estimate the effectiveness of the double-update approach in dealing with the six instances of the RLFAP, a series of 20 experiments have been conducted for each instance. All experiments were performed on a Silicon Graphics machine with R4000 processors running at 100MHz. It must be noted that in order to overcome the memory

Annealing Schedule		
CELAR	T_0	trials/temperature step
scen01	0.20	64N
scen02	0.15	$N^2/2$
scen03	0.2	$N^2/10$
scen04	1.0	24N
scen05	2.5	$N^2/2$
scen11	5.0	32N

Table 1: Annealing Schedule for the CELAR problems

Double-update Boltzmann Machine				
CELAR	optimal	best found	average CPU time	standard deviation
scen01	16	20	4hrs	17min
scen02	14	14	5min	1min
scen03	14	16	53min	9min
scen04	46	46	61min	7min
scen05	792	792	90min	12min
scen11	0	0	2.4hrs	20min

Table 2: Results for the CELAR problems using Double-Update Boltzmann Machine

Tabu Search		
CELAR	best found	average CPU time
scen01	18	3hrs
scen02	14	4min
scen03	14	34min
scen04	n.a.	n.a.
scen05	n.a.	n.a.
scen11	0	54min

Table 3: Results for the CELAR problems using Tabu Search

limitations concerning the storage of the weight matrix in the case of large problem instances, we have used a version with on-line computation of weights, similar to the one proposed in [5]. According to this approach, the connection weights are not computed and stored before network operation, but the value of each weight is determined only when it is used in a computation. Table 2 summarizes the obtained results, which concern the quality of the solution, as well as mean and standard deviation of execution time. For the first four problems the displayed solution figure concerns the number of different frequencies used, whereas the largest frequency used is shown for scen05. In what concerns scen11, a cost function is defined which takes its optimal value 0 when all constraints are satisfied. In all experiments, a feasible solution was obtained. In the cases where the optimal solution has been attained, this solution was obtained in more than 50% of the experiments. In the cases where a near-optimal solution has been reached, this solution was obtained for more than 60% of the experiments.

As a measure of comparison for effectiveness and speed, we chose the Tabu Search method [30, 31], which is considered an efficient way of finding high quality solutions to hard combinatorial optimisation problems. Although the double-update approach constitutes an improvement on the regular Hopfield and Boltzmann Machine models, the latter techniques cannot be used for comparison, since, as already discussed, they are not capable of tackling hard problems of the type considered here. Table 3 shows the results reported in [32], where Tabu Search is applied to the same problems. In this case 20 experiments were carried out for each problem instance on a 130MHz DEC Alpha machine. We should note here that Table 3 refers to results obtained without any kind of pre-processing.

¿From the comparison of Tables 2 and 3 we can see that solutions are obtained in about the same time using both methods (as the experiments were not carried out on the same machine, we were primarily interested in the quality of the solution and a rough time comparison rather than in rigorous statistics). As far as the quality of the solutions is concerned, the double-update method provides optimal solutions for four problems (02, 04, 05, 11) and near-optimal for two (01, 03), whereas Tabu Search provides optimal solutions for three problems (02, 03, 11), near-optimal for one (01) and no solution for two (04, 05). For problems 04 and 05 Tabu Search yields a solution only if a pre-processing technique (arc-consistency) is applied [32]. It is obvious that the proposed method exhibits good performance in solving this set of problems and is more flexible and effective than Tabu Search.

In addition to the above experiments, we tried to solve the CELAR problems by implementing two other methods, specialized on the Frequency Assignment Problem. The first one is based on genetic search [20, 21], while the other is based on a neural network model with synchronously operating units [9]. The genetic search method introduces two specialized genetic operators, the conflict-based crossover operator and the conflict-based mutation operator, while it uses integer string encoding. The constraints are represented as penalty terms in the fitness function of the model. On the other hand, the synchronous neural network model is composed of $n \times m$ McCulloch-Pitts processing elements, for an n-cell-m-frequency problem, with continuous input and binary output. The motion equation of each processing element includes penalty terms for contraint violations. Both methods failed to provide remarkable results, as they only managed to solve small subproblems of the actual problems of the CELAR data set, altough we experimented with numerous configurations and parameters settings.

To return to the proposed method, an important remark that must be made concerns the fact that, although in all examined problems the annealing starts from a relatively low temperature value T_0, it does not constitute a negative factor for the speed of the approach as is the case with the pure Boltzmann Machine. In our technique, the incorporation of annealing provides the network with the required flexibility to escape from local minima traps and effectively explore the energy landscape. This kind of behaviour can be explained by considering equation (10), which shows that the variation of energy when the double-update scheme is employed is much smoother than in the case of single updates. Experiments have shown that if the above annealing schedule is employed by the pure Boltzmann Machine, the network will settle into local minimum states in most of the cases.

6 Conclusions

We have developed an effective method for solving general binary constraint satisfaction problems that is based on the discrete Hopfield neural network (Boltzmann Machine) operating under the double-update mode. The proposed method constitutes a major improvement on Hopfield-type models, since it is capable of solving hard constraint satisfaction problems yielding very satisfactory solutions to large realistic instances. The generality of the method stems from the fact that it assumes no knowledge about the problem domain other than value and value-pair inhibitions. Therefore, it can be easily employed for any problem that assumes finite domains and binary constraints between pairs of domain values. Moreover, in problems where required or preferred conditions on solutions can be specified, such preferences can be easily incorporated by adjusting the threshold values of the network units, as well as the strengths of negative connections, which in our formulation were considered to be fixed for all units. Such extensions constitute a topic of further research, in an attempt to develop variants of the proposed method suitable for problems involving soft constraints (constraint optimisation problems).

A considerable advantage of the approach is that it can be implemented on a parallel machine in order to further increase the speed of convergence. Parallelism can be either at the unit level, i.e. in computing the energy difference ΔE_G caused by the double update, or at the row level, where one processor is responsible for simulating the operation of a whole row. Alternatively, the method can be implemented on special purpose hardware devices [33, 34, 35, 36] devoted to high-speed execution of neural networks. Such devices can also be embedded into larger systems to construct high performance intelligent machines.

The double-update approach can also be effective in the case of more general optimisation problems that involve, in addition to constraint satisfaction, optimisation of a cost function. There are many combinatorial optimisation problems that can be mapped onto a discrete Hopfield network characterised by a matrix structure and the constraint of having exactly one active unit in each row in a valid equilibrium state. In the case of such problems, appropriate extensions to the basic model can be developed.

Appendix

Proof of Proposition 2

Consider a state that corresponds to a solution of the given problem instance. In such a state no constraints are violated and, hence, there are no active units connected through inhibitory connections. Therefore, for each active unit $(k, k^\star)$ it is valid that $\sum_{m\neq k} w_{mm^\star,kk^\star} = 0$ and for each unit (k,l) (with $l \neq k^\star$) $\sum_{m\neq k} w_{mm^\star,kl} \leq 0$. Consequently, we have from equation (10) that $\Delta E_G(\vec{y}) \geq 0$ for every possible group, i.e. the corresponding state is a double-update equilibrium point as defined above. This state corresponds to a global energy minimum with value $-N\beta$ as explained earlier. ∎

Minimisation of the Number of Different Values

Equations (8)–(9), which prescribe the individual energy changes of actual units, are now modified as follows to take into account the extra units:

$$\delta E_{kk^\star}(\vec{y}) \;=\; \beta + \sum_{m\neq k} w_{mm^\star,kk^\star} + \gamma y_{0\,r(k,k^\star)} \tag{12}$$

$$\delta E_{kl}(\vec{y}) \;=\; -\sum_{m\neq k} w_{mm^\star,kl} + \eta - \beta - \gamma y_{0\,r(k,l)} \tag{13}$$

The energy difference caused by the double update, which is still computed using equation (7), takes the form:

$$\Delta E_G(\vec{y}) = \sum_{m\neq k} w_{mm^\star,kk^\star} - \sum_{m\neq k} w_{mm^\star,kl} + \gamma y_{0\,r(k,k^\star)} - \gamma y_{0\,r(k,l)}\;, \qquad G = \{(k,k^\star),(k,l)\} \tag{14}$$

As already mentioned, the extra units follow the single-update mode of operation. At each time step, row 0 can be selected according to some probabilistic scheme and then an extra unit will be randomly selected and tested for update. The energy difference due to a change in the state of unit $(0, j)$ can be computed as follows:

$$\delta E_{0j}(\vec{y}) = (2y_{0j} - 1)\Big(\sum_{m=1}^{N} \gamma\, I[r(m,m^\star) = j] - \delta\Big) \tag{15}$$

where $I[A]$ denotes the indicator function of the event A.

Minimisation of the Largest Value

The individual energy changes of actual units now take the form:

$$\delta E_{kk^\star}(\vec{y}) \;=\; \beta + \sum_{m\neq k} w_{mm^\star,kk^\star} - \sum_{j<r(k,k^\star)} \gamma y_{0j} \tag{16}$$

$$\delta E_{kl}(\vec{y}) \;\; = \;\; -\sum_{m\neq k} w_{mm^\star,kl} + \eta - \beta + \sum_{j<r(k,l)} \gamma y_{0j} \tag{17}$$

and the energy difference caused by the double update is given by:

$$\Delta E_G(\vec{y}) = \sum_{m\neq k} w_{mm^\star,kk^\star} - \sum_{m\neq k} w_{mm^\star,kl} - \sum_{j<r(k,k^\star)} \gamma y_{0j} + \sum_{j<r(k,l)} \gamma y_{0j} \; , \qquad G = \{(k,k^\star),(k,l)\} \tag{18}$$

Finally, following the single-update mode of operation for extra units, the energy difference due to a change in the state of unit $(0,j)$ is computed as follows:

$$\delta E_{0j}(\vec{y}) = (2y_{0j} - 1)(\sum_{m=1}^{N} \gamma \, I[r(m,m^\star) > j] - \delta) \tag{19}$$

References

[1] E. Tsang, *Foundations of Constraint Satisfaction*, Academic Press, 1993.

[2] J. Bitner and E. Reingold, "Backtracking Programming Techniques", *Communications of the ACM* **18** (1975) 651–656.

[3] V. Kumar, "Algorithms for Constraint Satisfaction Problems: A Survey", *The AI Magazine* **13** (1992) 32–44.

[4] H. A. Stone and J. M. Stone, "Efficient Search Techniques - An Empirical Study of the N-Queens Problem", *IBM Journal on Research and Development* **31** (1987) 464–474.

[5] H. Adorf and M. Johnston, "A Discrete Stochastic Neural Network Algorithm for Constraint Satisfaction Problems", *Proc. Int. Joint Conf. on Neural Networks*, San Diego, CA, June 1990, Vol. 3, 917–924,

[6] G. Tagliarini and E. Page, "Solving Constraint Satisfaction Problems with Neural Networks", *Proc. Int. Conf. on Neural Networks*, San Diego, CA, June 1987, Vol. 3, 741–747.

[7] Y. Takefuji and H. Szu, "Design of Parallel Distributed Cauchy Machines", *Proc. Int. Joint Conf. on Neural Networks*, Washington DC, June 1989, Vol. 1, 529–532.

[8] D. Kunz, "Channel Assignment for Cellular Radio Using Neural Networks," *IEEE Trans. on Vehicular Technology* **40** (1991) 188–193.

[9] N. Funabiki and Y. Takefuji, "A Neural Network Parallel Algorithm for Channel Assignment Problems in Cellular Radio Networks," *IEEE Trans. on Vehicular Technology* **41** (1992) 430–437.

[10] D. Kontoravdis, A. Likas and A. Stafylopatis, "Embedding Knowledge into Stochastic Learning Automata for Fast Solution of Binary Constraint Satisfaction Problems", *Proc. Europ. Symp. on Artificial Neural Networks (ESANN'93)*, Brussels, April 1993, 21–26.

[11] F. Ricci, "Constraint Reasoning with Learning Automata", *Proc. 3rd Workshop of AI*IA Interest Group in Automatic Learning*, Rome, May 1992.

[12] M. A. Thathachar and P. S. Sastry, "Relaxation Labeling with Learning Automata", *IEEE Trans. Pattern Analysis and Machine Intelligence* **8** (1986) 256–268.

[13] J. Hopfield, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", *Proc. Nat. Academy of Sciences USA* **79** (1982) 2554–2558.

[14] J.J. Hopfield and D.W. Tank, "Neural Computation of Decisions in Optimization Problems", *Biological Cybernetics* **52** (1985) 141–152.

[15] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines, A Stochastic Approach to Combinatorial Optimization and Neural Computing*, (John Wiley & Sons, Chichester, 1989).

[16] L. Gislén, C. Peterson and B. Söderberg, "Complex Scheduling with Potts Neural Networks", *Neural Computation* **4** (1992) 805–831.

[17] M. Ohlsson, C. Peterson and B. Söderberg, "Neural Networks for Optimization Problems with Inequality Constraints: The Knapsack Problem", *Neural Computation* **5** (1993) 331–339.

[18] M. Duque-Antón, D. Kunz and B. Rüber "Channel Assignment for Cellular Radio Using Simulated Annealing," *IEEE Trans. on Vehicular Technology* **42** (1993) 14–21.

[19] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, (Springer-Verlag, 1992).

[20] R. Dorne and J.K. Hao, "An Evolutionaty Approach for Frequency Assignment in Cellular Radio Networks," *Proc. IEEE Intl. Conf. on Evolutionary Computation (ICEC'95)*, Perth, Australia, 1995.

[21] J.K. Hao and R. Dorne, "Study of Genetic Search for the Frequency Assignment Problem," *Proc French Nat. Conf. on Artificial Evolution (EA'95)*, Brest, France, Sept. 1995, (Lecture Notes in Computer Science, Springer-Verlag).

[22] A. Likas and A. Stafylopatis, "Group Updates and Multiscaling: An Efficient Neural Network Approach to Combinatorial Optimization", *IEEE Transactions on Systems, Man and Cybernetics* (1996) to appear.

[23] S. Kirkpatrick, C.D. Gellat Jr. and M.P. Vecchi, "Optimization by Simulated Annealing", *Science* **220** (1983) 671–689.

[24] Y. Kamp and M. Hasler, *Recursive Neural Networks for Associative Memory*, (John Wiley & Sons, Chichester, 1990).

[25] J.C. Spall, "Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation," *IEEE Trans. Automatic Control* **37** (1992) 332–341.

[26] G.V. Wilson and G.S. Pawley, "On the Stability of the TSP Algorithm of Hopfield and Tank", *Biological Cybernetics* **58** (1988) 63–70.

[27] K. Chiba, F. Takahata and M. Nohara, "Theory and Performance of Frequency Assignment Schemes for Carriers with Different Bandwidths under Demand Assignment SCPC/FDMA Operation", *IEICE Trans. Commun.* **E75-B** (1992) 476–486.

[28] A. Raychaudhuri, "Optimal Multiple Interval Assignments in Frequency Assignment and Traffic Phasing", *Discrete Applied Mathematics* **40** (1992) 319–332.

[29] M. Garey and D. Johnson, *Computers and Intractability: a Guide to the Theory of NP-completeness*, (W.H. Freeman and Company, San Francisco, 1979).

[30] F. Glover, "Tabu Search Part I", *ORSA J. Comput.* **1** (1989) 190–206.

[31] F. Glover, "Tabu Search Part II", *ORSA J. Comput.* **2** (1990) 4–32.

[32] A. Bouju, J.F. Boyce, C.H.D. Dimitropoulos, G. vom Scheidt, J.G. Taylor, A. Likas, G. Papageorgiou, A. Stafylopatis, "Intelligent Search for the Radio Links Frequency

Assignment Problem", Proc. Int. Conf. on Digital Signal Processing (DSP'95), Vol. 1, 408–414, Limassol, Cyprus, June 1995.

[33] D. Abramson, "A Very High Speed Architecture for Simulated Annealing", *Computer* **25** (1992) 27–36.

[34] A. de Gloria, P. Faraboschi and S. Ribella, "A Dedicated Massively Parallel Architecture for the Boltzmann Machine", *Parallel Computing* **18** (1992) 57–73.

[35] M. Skubiszewski, "A Hardware Emulator for Binary Neural Networks", *Proc. Int. Neural Network Conf.*, Paris, July 1990, Vol. 2, 555–558.

[36] M. Skubiszewski, "An Exact Hardware Implementation of the Boltzmann Machine", Research Report, DEC Paris Research Laboratory, 1992.