A New Method to Solve the Constraint Satisfaction Problem Using the Hopfield Neural Network

Wei-Dong SUN†, **Hiroki TAMURA**†, **Zheng TANG**†, *and* **Masahiro ISHII**†,

SUMMARY

The constraint satisfaction problem is constituted by several condition formulas, which makes it difficult to be solved. In this paper, using the Hopfield neural network, a new method is proposed to solve the constraint satisfaction problem by simplifying its condition formula. In this method, all restriction conditions of a constraint satisfaction problem are divided into two restrictions: restriction I and restriction II. In processing step, restriction II is satisfied by setting its value to be 0 and the value of restriction I is always made on the decreasing direction. The optimum solution could be obtained when the values of energy, restriction I and restriction II become 0 at the same time. To verify the validity of the proposed method, we apply it to two typical constraint satisfaction problems: N-queens problem and four-coloring problem. The simulation results show that the optimum solution can be obtained in high speed and high convergence rate. Moreover, compared with other methods, the proposed method is better than other methods.

key words: *Constraint satisfaction problem, Combinatorial optimization problem, Hopfield neural network, N-queens problem, Four-coloring problem*

1. Introduction

A constraint satisfaction problem is a problem to find a consistent assignment of values to variables. It is one kind of the combinatorial optimization problem. A number of commonly encountered problems in mathematics, computer science, molecular biology, management science, seismology, communications, and operation research belong to a class of combinatorial optimization problems [1]. The combinatorial optimization problem is a very difficult problem, it could take dozens of years to obtain one optimum solution even if the latest supercomputer is used [2][3].

The idea of using neural network to provide solution originated in 1985 when Hopfield and Tank demonstrated that the traveling salesman problem could be solved using the Hopfield neural network [4][5]. Since Hopfield and Tank's work [4][5], there has been growing interest in the Hopfield neural network because of its advantages over other approaches for solving optimization problems. The work by Wilson and Pawley [6] showed that the Hopfield neural network often failed to converge to valid solutions. Takefuji *et al.* [7][8] modified the motion equation in order to guarantee the local minimum convergence. However, with the Hopfield neural network, the state of system is forced to converge to a local minimum. In other words, the neural network cannot always find the optimum solution. Therefore, several neuron models and heuristics such as hysteresis binary neuron model [9], neuron filter [20], the hill-climbing term and omega function [10], Lagrange relaxation [11] and pots spin [12] have been proposed to improve the performance of the network. Despite the improvement of the performance of the Hopfield neural network over the past decade, this model still has some basic problems [13][14].

A constraint satisfaction problem has several constraint conditions, and this makes it difficult to be solved. In this paper we propose a new method to solve the constraint satisfaction problem using the Hopfield neural network. In this method, all the restriction conditions of a constraint satisfaction problem are divided into two restrictions: restriction I and restriction II. In processing step, restriction II is satisfied by setting its value to be 0 and the value of restriction I is always made on the decreasing direction. The optimum solution could be obtained when the values of energy, restriction I and restriction II become 0 at the same time. To verify the validity of the proposed method, we apply it to two typical constraint satisfaction problems: N-queens problem and four-coloring problem. The simulation results show that the optimum solution can be obtained in high speed and high convergence rate. Moreover, the comparison results show that the proposed method is better than other methods.

The rest of this paper is organized as follows. In section 2, we briefly introduce the Hopfield neural network and its relevant components for constraint satisfaction problems. Section 3 presents the details of the proposed method and its formulization method. The simulation results of testing the proposed method in real constraint satisfaction problems are described in section 4 for N-queens problem and in section 5 for four-coloring problem. Finally, the paper is concluded with general comments concerning the proposed method and its effectiveness to constraint satisfaction problems.

2. The Hopfield Neural Network for Constraint Satisfaction Problem

In this section, we briefly introduce the Hopfield neu-

†The authors are with the Faculty of Engineering, Toyama University; Gofuku 3190, Toyama city, JAPAN.
sun@hi.iis.toyama-u.ac.jp,
tang@iis.toyama-u.ac.jp

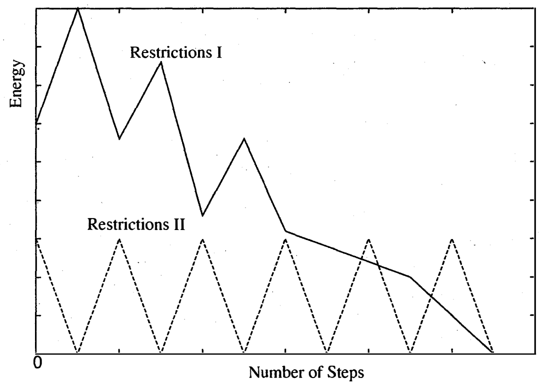

Fig. 1 The conceptual figure of the proposed method.

ral network and its relevant components for constraint satisfaction problems.

The Hopfield neural network for constraint satisfaction problems consists of two elements named neuron unit and motion equation. The neuron unit is a collection of simple processing elements called neurons. Each neuron has an input potential U_i and an output potential V_i. The dynamic behavior of the network is described by the following motion equation with a partial derivation term of the energy function (E) and a decay term with a time constant τ [4][5].

$$\frac{dU_i(t)}{dt} = -\frac{\partial E(V_1, V_2, \cdots, V_N)}{\partial V_i} - \frac{U_i}{\tau} \quad (1)$$

Takefuji *et al.* showed that the decay term increases the energy function under some conditions [8]. They modified the motion equation in order to guarantee the local minimum convergence.

$$\frac{dU_i(t)}{dt} = -\frac{\partial E(V_1, V_2, \cdots, V_N)}{\partial V_i} \quad (2)$$

To compute the input potential of neurons, the time-idependent method is used in which the input potential of neurons at time $t+1$ depends on the value at time t [8].

$$U_i(t+1) = U_i(t) + \frac{dU_i(t)}{dt} \quad (3)$$

The output is updated from U_i using a non-linear function called neuron model. For example, according to the McCulloch-Pitts binary neuron model [30], it can be obtained:

$$V_i = \begin{cases} 1 & if \;\; U_i > 0 \\ 0 & otherwise \end{cases} \quad (4)$$

Each neuron updates its input potential according to the computation rule (Eq.(3)) and sends its output state in response to the input according to the input/output function (Eq.(4)).

3. The Proposed Method to Solve Constraint Satisfaction Problem Using the Hopfield Neural Network

In this section, we describe a new method to solve constraint satisfaction problem using the Hopfield neural network. Note that the process in the Hopfield neural network is sequential.

3.1 Simplification of Constraint Satisfaction Problem

A constraint satisfaction problem usually consists of several restriction condition formulas. In the proposed method, we classify these restriction condition formulas into two kinds: restrictions I and restrictions II. The conceptual figure of this proposed method is shown in Fig.1. Restriction I always is carried out in the decreasing direction; restriction II are satisfied by setting its value to be 0 in processing step, in other words, if a certain value of restriction II increases , only the same quantity will decrease, and it returns to 0 surely. The optimum solution can be calculated when the values of both restrictions become 0. State change is sequentially performed for every neuron. If each neuron is in the state of satisfying restrictions, it can be stabilized in the state, on the contrary, it becomes unstable if it is in the state where it does not satisfy the restrictions.

Although the constraint satisfaction problem is in the tendency to become complicated since it consists of several condition formulas, if some condition formulas are satisfied in the processing, the problem will be simplified and it will become easy to draw a solution. Moreover, the procedure at the time of drawing the optimum solution by becoming a unary formula decreases, and it is thought that it becomes possible to draw the optimum solution in a short time.

3.2 Formulization for Restrictions I and II

As discussed above, the formula of a constraint satisfaction problem consists of two restriction condition: restriction I and restriction II in the proposed method. Next, we give the mathematic description for the proposed method.

[1] For a certain neuron (i) in time (t), the value of restriction I is assumed to be $a_{ij}(t)$, of restriction II to be $b_{ik}(t)$. Therefore, the input of neuron (i) can be described as:

$$\begin{aligned} U_i(t+1) &= restriction\ I + restriction\ II \\ &= a_{ij}(t) + b_{ik}(t) \end{aligned} \quad (5)$$

[2] In time (t), the energy function is assumed to be $\sum_{j=1}^{M} P_j(t)$ for restriction I, $\sum_{k=1}^{M'} Q_k(t)$ for restriction II, and, $P_j(t) = \sum_{i=1}^{L} a_{ij}(t)$, $Q_k(t) =$

$\sum_{i=1}^{L} b_{ik}(t)$. So, the energy function of network can be given by

$$E = \sum_{j=1}^{M} P_j(t) + \sum_{k=1}^{M'} Q_k(t) \tag{6}$$

Here, L is the number of neurons, $(1, 2, \cdots, i, \cdots, L)$;$M$, neurons number of restriction I $(1, 2, \cdots, j, \cdots, M)$; M', neurons number of restriction II $(1, 2, \cdots, k, \cdots, M')$.

Thus, according to the definition above, the following conditions will be satisfied for restrictions I and II in the proposed method.

[Restriction I] Restrictions I is formulized as following.

$$\sum_{j=1}^{M} P_j(t+\alpha) - \sum_{j=1}^{M} P_j(t) \leq 0 \tag{7}$$

This formula makes the energy function always go in the reduction direction. However, $t + \alpha$ is the time when the processing in the Hopfield neural network is sufficient in time t.

[Restriction II] Restrictions II is formulized as following.

$$b_{ik}(t+1) - b_{ik}(t) \leq 0 \tag{8}$$

When a step of processing is carried out about neuron (i), this formula presents that the value of restrictions II at time $t+1$ is the same as that at time t, or less than it. Even if $b_{ik}(t)$ of a neuron increases temporarily according to the conditions of restrictions I, or the conditions of the restrictions II of other neurons, while carrying out one loop, it returns to the state of $\sum_{k=1}^{M'} Q_k(t) = 0$ again. Therefore, it is not necessary to formulize like restriction I so that the energy function may be made to go in the reduction direction.

3.3 Algorithms

The following procedure describes the algorithms for solving constraint satisfaction problems based on the proposed method. Note that t is the step number and *t_limit* is maximum number of iteration step.

Step 1. Setteing Parameters.
(a) Set $t = 0$, $\triangle t = 1$, and set *t_limit* and other parameters.
(b) Randomize the initial value of $U_i(0)$ for $i = 1, 2, \cdots, N$.
(c) evaluate the values of $V_i(0)$ according to Equ.(4).

Step 2. Calculating Network Energy.
for $t = 1$ to *t_limit*, do:
(a) initialize $U_i = 0$, for $i = 1, 2, \cdots, N$.
(b) Update the $U_i(t+1)$ and $V_i(t+1)$ for $i = 1, 2, \cdots, N$.
(c) Calculate the energy E according to Equ.(6).
(d) Check system energy. If $E = 0$ (the optimum solution can be obtained), end the procedure.

4. Application to N-queens Problem

In this section, the proposed method is applied to one of the optimization problems: N-queens problem.

4.1 About N-queens Problem

In 1992, Takefuji presented a neural network for N-queens problem with the hysteresis binary neuron model [31]. Mandziuk and Macukow presented a neural network using the continuous sigmoid neuron model [28]. In 1995, Mandziuk improved their neural network by using the binary neuron model [29]. In 1994, Ohta *et al.* presented the neural network using the binary neuron model with the negative self-feedback [32]. In 1997, Takenaka *et al.* presented the neural network using the maximum neuron model with the competition resolution method [21].

N-queens problem is the problem to assign N queens with no collision in $N \times N$ chess board. Queen is the piece used in chess. Queen moves for vertical, horizontal and diagonal freely. One of the optimum solution of 5 queens problem is shown in Fig.2. To express the problem with neurons, we transform Fig.2 to expression with 5×5 neurons as shown in Fig.3. The output of the neuron corresponds to an existence of a queen. When a queen is placed, an output of the neuron is 1. An output where no queen is placed is 0.

4.2 The Motion Equation and Energy Function

The motion equation for the ijth neuron is given by:

$$\begin{aligned} U_{ij}(t+1) &= (\ restriction\ I + restriction\ II\ ) \\ &= -A\left(\sum_{k=1}^{N} V_{ik} - 1\right) - A\left(\sum_{k=1}^{N} V_{kj} - 1\right) \\ &\quad -B \sum_{1 \leq i-k, j-k \leq N (k \neq 0)} V_{i-k,j-k} \\ &\quad -B \sum_{1 \leq i-k, j+k \leq N (k \neq 0)} V_{i-k,j+k} \\ &\quad +V_{ij}(t) \end{aligned} \tag{9}$$

where, A, B are positive coefficients. In equation (9), the first term means the constraint that only on queen must be placed on row; the second term, on column; the third term, on lower right diagonal; the fourth term, on lower left diagonal. Among them, the first, second terms are corresponding to resticiton I, and the third, fourth terms are corresponding to restriction II. The

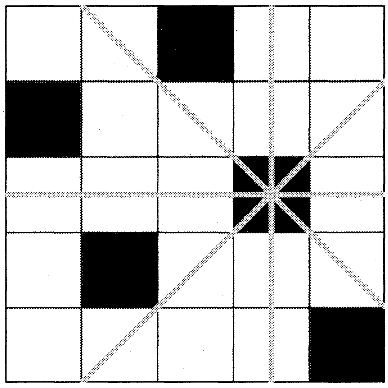

Fig. 2 An example of N-queens problem for N=5.

0	0	1	0	0
1	0	0	0	0
0	0	0	1	0
0	1	0	0	0
0	0	0	0	1

Fig. 3 Example of 5 × 5 neurons' configuration.

fifth term expresses whether there is a queen or not before updating the neuron state.

According to the U_{ij} from the equation (9), the value of V_{ij} is defined by

$$V_{ij} = \begin{cases} 1 & if\ \ U_{ij} > 0 \\ 0 & otherwise \end{cases} \tag{10}$$

The energy function is given by

$$\begin{aligned} E &= \left(\sum_{j=1}^{M} P_j(t) + \sum_{k=1}^{M'} Q_k(t) \right) \\ &= \frac{1}{2} A \sum_{i=1}^{N} \left(\sum_{j=1}^{N} V_{ij} - 1 \right)^2 \\ &\quad + \frac{1}{2} A \sum_{j=1}^{N} \left(\sum_{i=1}^{N} V_{ij} - 1 \right)^2 \\ &\quad + B \sum_{i=1}^{N} \sum_{j=1}^{N} \left\{ V_{ij} \left(\sum_{1 \le i-k, j-k \le N (k \ne 0)} V_{i-k,j-k} \right) \right\} \\ &\quad + B \sum_{i=1}^{N} \sum_{j=1}^{N} \left\{ V_{ij} \left(\sum_{1 \le i-k, j+k \le N (k \ne 0)} V_{i-k,j+k} \right) \right\} \end{aligned} \tag{11}$$

According to equation (11), the first term becomes zero if one queen is placed in every row. The second term becomes zero if one queen is placed in every column and the third, fourth term become zero if no more than one queen is placed on any diagonal line. In overall, the values of the energy function always become positive or 0, and it tends to increase if all constraints are not fulfilled.Therefore, if the neuron state is changed along the decreasing direction of energy function, it is possible that the energy becomes zero. This is the optimization solution when the energy becomes zero.

4.3 Simulation Results

4.3.1 The Change Situation of Energy

This simulation aims to observe the change situations of system energy, restriction I and restriction II until a optimum solution is obtained. The parameters are set to be: $A = 1, B = 1, t\_limit = 1000$. In the initial state we let $V_{ij} = 0$ for all ij, and the experiments for 10-500 queens are carried out. Here, the initial state represents whether a neuron is firing or not before updating in the network, and whether a queen is placed or not in the chess. On a mathematic expression, it means to initialize the V_{ij} by 0 or 1. A change situation of the energy when carrying out a simulation with such an initial state is shown in Fig.4. It illustrates the changes of energy, restriction I and restriction II in one step (all neurons are sequentially processed by a unit of 1 time). It can be seen easily that Fig.4 illustrates the same result as discripted in the conceptual figure (Fig.1). It turns out that restrictions II is satisfied by setting its value to be 0 and restrictions I is brought in the reduction direction. The optimum solution can be obtainted when the energy value of restriction I and restriction II become 0 at the same time.

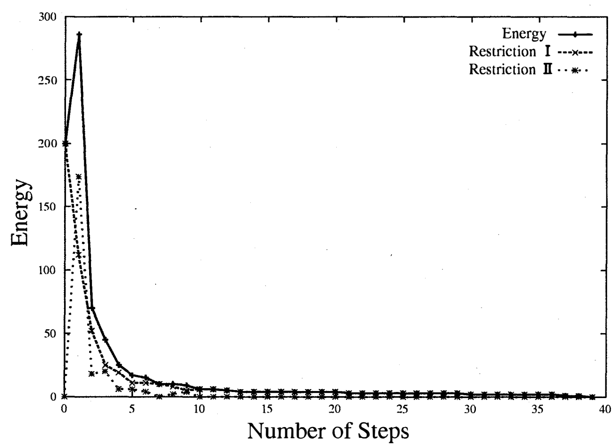

Fig. 4 The change situation of energy.

Table 1 Simulation results of N-queen problems.

Queens	Neuron filter[20]		Takefuji[7]		Maximum NN[21]		The proposed method	
N	Conv.(%)	Ave.step	Conv.(%)	Ave.step	Conv.(%)	Ave.step	Conv.(%)	Ave.step
10	31	159.3	31	162.8	26	71.2	100	16.4
20	51	286.2	51	290.6	47	142.0	100	23.5
30	52	246.7	52	253.9	53	148.3	100	22.8
50	86	301.2	86	308.4	78	176.6	100	31.6
100	98	288.7	98	300.9	99	174.2	100	43.0
150	96	400.6	96	411.0	95	151.8	100	62.1
200	94	508.8	93	517.6	95	152.7	100	87.6
300	86	597.1	85	616.8	95	152.8	100	104.6
400	70	659.2	69	677.8	87	152.6	100	165.8
500	69	748.4	67	756.8	86	139.4	100	251.5

4.3.2 The Result of Comparison with Other Methods

This simulation aims at evaluating the validity and effectivity of the proposed method by comparing with the other methods. The other methods include neuron filter [20], Takefuji method [7] and Maximum NN method [21].

In this simulation, the parameters are set to be: $A = 1, B = 1, t\_limit = 1000$. 100 simulations runs with different initial states are preformed for 10-500 queens. We use the convergence rate and the number of average steps of each solution method for comparison. Here, the number of average steps is the average value of the number of steps required for the convergence. The convergence rate expresses the average convergence times on the optimum solution during 100 trial. The simulation results are shown in Table 1, where the convergence rate and the average number of steps required for the convergence are summarized.

As shown in Table 1, the convergence rate in the proposed method is all 100%, but, in the other methods they are not so well. Furthermore, the required average steps for the proposed method are much less than the other methods. For example when N=100, the average steps are 288.7, 300.9 and 174.2 for Neuron filter [20], Takefuji [7] and Maximum NN [21], respectively, but the proposed needs only 43 steps for the 100% convergence rate. It demonstrates that the proposed method increases the convergence rate and reduces the average steps compared with the other methods. That is to say, the proposed method performs better than the other methods.

5. Application to Four-Coloring Problem

5.1 About Four-Coloring Problem

A mapmaker colors adjacent countries with different colors so that they may be easily distinguished. This is not a problem as long as one has a large number of colors. However, it is more difficult with a constraint that one must use the minimum number of colors required for a given map. It is still easy to color a map with a small number of regions. In the early 1850's, Francis Guthrie was interested in this problem, and he brought it to the attention of Augustus De Morgan. Since then many mathematicians, including Arthur Kempe, Peter Tait, Percy Heawood, and others tried to prove the problem that any planar simple graph can be colored with four colors. A four-coloring problem is defined that one wants to color the regions of a map in such a way that no two adjacent regions (that is, regions sharing some common boundary) are of the same color. In August 1976, Appel and Haken presented their work to members of the American Mathematical Society [24]. They showed a computer-aided proof of the four-coloring problem. However, their coloring was based on the sequential method so that it took many hours to solve a large problem. Their computation time may be proportional to $O(x^2)$ where x is the number of the regions. Moreover, few parallel algorithms have been reported. Dahl [25], Moopenn et al. [26], and Thakoor et al. [27] have presented the first neural network for map K-colorability problems.

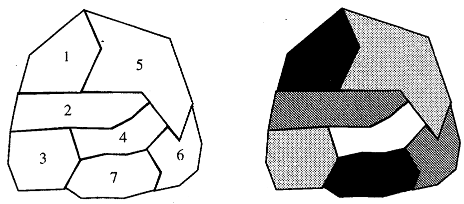

Fig. 5 An example of 7-region map.

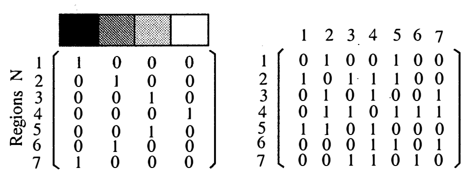

Fig. 6 Neural representation of the 7-region map.

In order to map the four-coloring problem to the Hopfield neural network, a $x \times 4$ two dimensional neural array is needed, where x is the number of regions to be colored, and a single region requires four neurons for the single-color assignment. 7-region map are colored by four colors as shown in Fig.5. If red, yellow, blue and green are represented respectively by 1000, 0100, 0010 and 0001, the neural representation for the problem is given in Fig.6, where a 7×4 neural array is used. Fig.6 also shows the 7×7 adjacency matrix D of the seven-region map, which gives the boundary information between regions, where $D_{xy} = 1$.

5.2 The Motion Equation and Energy Function

According to the four-coloring problem constraint conditions, we can obtain the motion equation for c neuron as

$$\begin{aligned} U_{xc}(t+1) &= (\ \textit{restriction } I + \textit{restriction } II\ ) \\ &= -A(\sum_{k=1}^{4} V_{xk} - 1) - B\sum_{y=1}^{N} D_{xy}V_{yc} \\ &\quad + V_{xc}(t) + dA_{xc} \end{aligned} \quad (12)$$

where, A and B are positive constants, D is the adjacency matrix, and V_{xk} is the output of kth neuron in the x region.

And depending upon the U_{xc}, the V_{xc} can be determined by

$$V_{xc} = \begin{cases} 1 & (U_{xc} > 0) \\ 0 & (U_{xc} \le 0) \end{cases} \quad (13)$$

In the equation (12), the first term represents the row constraint in the neural array, which forces one region to be colored by one and only one color. It corresponds to restriction I in this paper. In the case that no color neuron is firing, $\sum_{k=1}^{4} V_{xk} = 0$, then $-(\sum_{k=1}^{4} V_{xk} - 1) = +1$. It suggests that the value of U is changed on the positive direction. That is, V is drawn towards firing direction. In the case that only one color neuron is firing, $\sum_{k=1}^{4} V_{xk} = 1$, then $-(\sum_{k=1}^{4} V_{xk} - 1) = 0$. It suggests there is no change in U. Similarly, in the case that two or over two color neurons are firing, the value of U is changed in the negative direction. That is, V is drawn towards non-firing direction.

The second term represents the same color neuron cannot be arranged in the adjacent regions. It corresponds the restriction II in this paper. $D_{xy}V_{yc}$ becomes $+1$ only in the case that the same color neuron is firing in the two adjacent regions x and y. This is because $D_{xy} = 0$ if region x, y are not adjacent regions and V_{yc}=0 if the same color neurons are colored. Thus, it can be said that the second term is the sum of firing c neuron in the region adjoined with x region. Since $-B$ is multiplied to this sum, the more this sum is, the more V is drawn towards non-firing direction eventually.

The third term represents the color neuron state before updating. In addition, since state change in the proposed method is sequentially performed for every neuron, the value of V_{xc} at time (t) and time $(t+1)$ is intermingled.

The fourth clause dA_{xc}, which is a special term, has the motion to fire a neuron of some region (the round region of middle in Fig.7) forcibly when surrounding of this region is colored by all four colors as shown in Fig.7, and it is impossible to also place all the color. That is, when there is no color in the region in the convergent state, the neural network gives a positive big value to dA_{xc}, and make a neuron to fire forcibly.

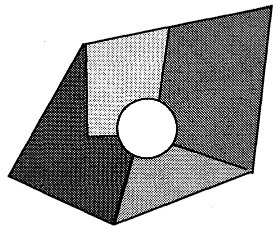

Fig. 7 An example for a colorless state.

The energy function which arranged in the four color problem is given by the following formula using the energy function of the Hopfield neural network.

$$\begin{aligned} E &= \left(\sum_{j=1}^{M} P_j(t) + \sum_{k=1}^{M'} Q_k(t) \right) \\ &= \frac{1}{2}A\sum_{x=1}^{N}(\sum_{c=1}^{4} V_{xc} - 1)^2 \\ &\quad + \frac{1}{2}B\sum_{x=1}^{N}\sum_{y=1}^{N}\sum_{c=1}^{4} D_{xy}V_{xc}V_{yc} \end{aligned} \quad (14)$$

The first term is the constraint that a region is colored by one and only one color. If the constraint is fulfilled, the value of the first term becomes 0, otherwise, positive value. The second term is the constraint that adjoined region cannot be colored by the same color. If the constraint is satisfied, the value of the first term becomes 0, otherwise, positive value. On the whole, the energy function takes only positive value or 0, and its value can increase if restrictions are not satisfied. Therefore, the energy function of the Hopfield neural network can be changed on the decreasing direction if restrictions are satisfied. When the values of two restrictions become 0, the value of the energy becomes 0, too. Thus, the optimum solution can be obtained.

5.3 Simulation Results

In this section, we apply the proposed method to the four-coloring problem. Simulations are performed on three kinds of maps: 48 regions (American Map), 110 regions, and 210 regions. The parameters are set to be: $A = 1, B = 1, t\_limit = 1000$.

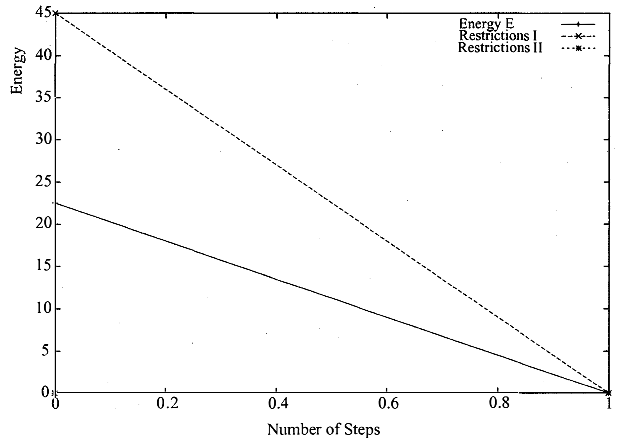

Fig. 8 The change situation of energy (48 Regions).

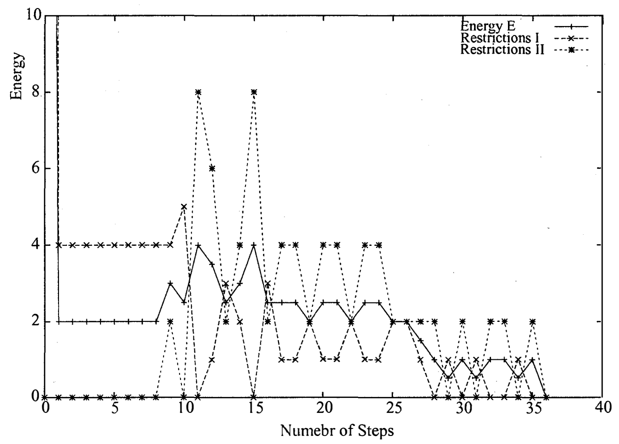

Fig. 9 The change situation of energy (110 Regions).

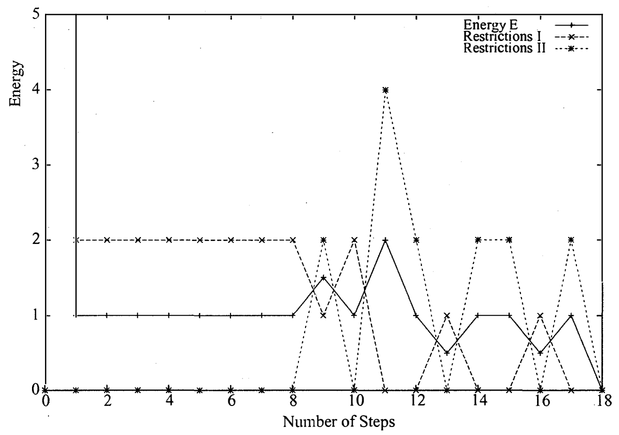

Fig. 10 The change situation of energy (210 Regions).

5.3.1 The Change Situation of the Energy

This simulation aims to observe the change situations of system energy, restriction I and restriction II until a optimum solution is obtained. First, in the initial state we let V_{xc} =0, the change state of energy, restriction I and restriction II are illustrated in Fig.8, Fig.9 and Fig.10, respectively. The initial value of dA_{xc} is set to be 0, and after 10 steps (It is judged as the convergence state of the Hopfield neural network.), it is set to be 100. As shown in Fig.8, the minimum value can be obtained in 1 step for the 48 regions. Fig.8, Fig.9 and Fig.10 illustrate that restriction I is drawn on the decreasing direction and the restriction II is always satisfied until the convergence state of the Hopfield neural network is obtained. It turns out that this is in agreement with that explained in Section 3 (see Fig.1).

5.3.2 The Comparison with Other Methods

Next, in order to compare the proposed method with the other methods, the simulations are performed 100 times from different initial state. As examples, the other methods include Takefuji method [8] and Yamada method [23]. The datas of Takefuji method and Yamada method used in this paper are from the data carried by the paper [23]. The regions that can be compared are the map of 48 regions and 210 regions, and we use the convergence rate and the number of average steps of each solution method for comparison. Here, the number of average steps is the average value of the number of steps required for the convergence. The convergence rate expresses the average convergence times on the optimum solution during 100 trial. The simulation results are shown in Table 2.

Table 2 shows that the 100% convergence rate can be obtained in all solution methods. It turns out that the minimum value can be calculated with the fewest number of steps for the 48 regions map. However, as shown in Fig.8, the minimum value can also be calculated at only one step by the proposed method depending on the initial state of neurons. By the simulation result of 210 regions, the minimum value can be calculated with the fewest number of steps. It depicts that the proposed method is better than the other methods.

Moreover, the average CPU time at the time of convergence obtained by the proposed method is shown in Table 2, too. It suggests that the four-coloring problem with which it dealt in this paper can be solved in several seconds.

In addition, the computer to perform the simulations in this paper is CPU: Pentium III 800Hz; OS: Winodws 2000; and the compiler is performed in the environment of VC++6.0.

6. Conclutions

In this paper, we proposed a new method to solve the constraint satisfaction problems using the Hopfield neural network. In this method, all the restriction conditions of a constraint satisfaction problem are divided into two restrictions: restriction I and restriction II.

Table 2 Simulation results of four-coloring problems.

Regions	Takefuji's method [8]		Yamada's method [23]		Proposed method		
N	Conv.(%)	Ave.step	Conv.(%)	Ave.step	Conv.(%)	Ave.step	CPU time (sec.)
48	100	89	100	6	100	12.91	0.0176
110	-	-	-	-	100	54.17	0.1827
210	100	769	100	49	100	11.22	0.1679

In processing step, restriction II is satisfied by setting its value to be 0 and the value of restriction I is always made on the decreasing direction. The optimum solution could be obtained when the values of energy, restriction I and restriction II become 0 at the same time. As two typical examples of constraint satisfaction problems, the proposed method was demonstrated by simulating the N-queens problem and four-coloring problem. The simulation results showed that the proposed method could increase the convergence rate and reduce the average steps and perform better than the other methods.

References

[1] R. Garey and S. Johnson, "Computers and Intractability: A guide to theory of NP-completeness," Freeman and Company, 1991.

[2] G.Brassard, P.Bratley, "Algorithmic theory and practice," Prentice-Hall Inc., 1988.

[3] N. Wirth, "Algorithms + Data Structures = Programs," Prentice-Hall Inc., 1976.

[4] J.J.Hopfield and D.W.Tank, ""Neural" computation of decisions in optimizatiton problems," Biol.Cybern., No.52, pp.141-152, 1985.

[5] J.J.Hopfield and D.W. Tank, "Computing with neural circuits: A model," Science, No.233, pp.625-633, 1986.

[6] G.V. Wilson and G.S. Pawley, "On the stability of the traveling salesman problem algorithm of Hopfield and Tank," Biol. Cybern., Vol.58, pp.63-70, 1988.

[7] Y.Takefuji, "Neural Network Parallel Computing," Kluwer Publishers, 1992.

[8] Y.Takefuji and K.C.Lee, "Artificial neural networks for four-coloring map problem and K-Colorability problems," IEEE Trans. Circuits & Syst., 38, 3, pp.326-333, 1991.

[9] Y. Takefuji and K. C. Lee, "An artificial hysteresis binary neuron: A model suppressing the oscillatory behaviors of neural dynamics," Biol. Cybern., Vol.64, pp.353-356, 1991.

[10] Y. Takefuji and K. C. Lee, "A parallel algorithm for tiling problems," IEEE Trans., Neural Networks, Vol.1, No.1, pp.143-145, March 1990.

[11] Z. L. Stan, "Improving convergence and solution quality of Hopfield-type neural networks with augmented Lagrange multipliers," IEEE Trans., Neural Networks, Vol.7, No.6, pp.1507-1516, 1996.

[12] S. Ishii and S. Masaaki, "Chaotic potts spin model for combinatorial optimization problems," Neural Networks, Vol.10, pp.941-963, 1997.

[13] B. S. Cooper, "Higher order neural networks-Can they help us optimize?" Proc. Sixth Australian Conference on Neural Networks (ACNN '95), pp.29-32, 1995.

[14] D. E. Bout and T. K. Miller, "Improving the performance of the Hopfield-Tank neural network through normalization and annealing," Biol. Cybern., Vol.62, pp.129-139, 1989.

[15] Y. Takefuji, S. Oka, "Neural Computing for Solving Intractable Problems, "The journal of the Institute of Electronics, Information, and Communication Engineers, Volume 79, Number 9, 1996.

[16] H.N.Schaller, "Constraint satisfaction problem," Optimization Techniques, edited by C.T.Leondes, San Diego: Academic Press, pp. 209-248, 1998.

[17] E.Aarts and J.Korst, "Simulated Annealing and Boltzmann Machines," Chichester: John Wiley and Sons, 1989.

[18] Y.Takefuji, "Neural Network Parallel Computing," Boston: Kluwer Academic, 1992.

[19] K.Smith, M.Palaniswami and M.Krishnamoorthy, "Neural techniques for combinatorial optimization with application," IDDD Trans, Neural Networks, Vol.9, pp.1301-1318,1998.

[20] Y.Takenaka, N.Funabiki, and T.Higashino, "A proposal of neuron filter: A constraint resolution scheme of neural networks for combinational optimization problem," IEICE Trans. Fundamentals, Vol.E83-A, No.9, pp.1815-1823, 2000.

[21] Y.Takenaka, N.Funabiki, and S.Nishikawa, "A proposal of competition resolution methods on the maximum neuron model thorough N-Queen problems," J.IPSJ, Vol.38, No.11, pp.2142-2148, 1997.

[22] Kedem Z.M., Palem K.V.,Pantziou G.E., Spirakis P.G. and Zaroliagis C.D., "Fast Parallel Algorithms for Coloring Random Graphs," Lecture Notes in Computer Science 570, Graph-Theoretic Concepts in Computer Science, pp.135-147, Spinger-Verlag , 1992.

[23] Yamada et al, "Four-coloring probelm algorithm using the maximum neural network," IEICE Technology report, NLP97-144,NC97-96,pp.59-66, 1998.

[24] K. Appel and W. Haken, "The solution of the four-color-map problem," Scientific American, pp.108-121, Oct.1977

[25] E.D. Dahl, "Neural network algorithm for an NP-complete problem: Map and graph coloring," in Proc. First Int. Conf. On Neural Networks, Vol.III, pp.113-120, 1987.

[26] A. Moopenn et al., "A neurocomputer based on an analog-digital hybrid architecture," in Proc. First Int, Conf. Neural Networks, Vol. III, pp.479-486, 1987.

[27] A. P. Thakoor et al., "Electronic hardware implementations of neural networks," Appl. Opt. Vol. 26, pp. 5085-5092, 1987.

[28] J. Mandziuk and B. Macukow: "A neural network designed to solve the N-queen problem", Boil. Cybern, Vol.66, pp.375-379, 1992.

[29] J. Mandziuk, "Solving the N-queen problem with a binary Hopfield-type network," Boil. Cybern, Vol.72, pp.439-445, 1995.

[30] W. S. McCulloch and W. H. Pitts: "A logical cauculus of ideas immanent in nervous activity," Bull. Math. Biophys, Vol.5, pp.115-133, 1943.

[31] Y. Takefuji and K. C. Lee, "An artificial hysteresis binay neuron: A model suppressing the oscillatory behaviors of dynamics," Biol. Cybern., Vol. 67, pp.243-251, 1991.

[32] M. Ohta, A. Ogihara, and K. Fukunaga, "Binary neural network with negative self-feedback and its application to N-queens problem," IEICE Trans. Inf. & Syst., Vol.E77-D, No.4, pp.459-465, 1994