A Learning Method based on Hopfield Neural Network and Its application in Point-feature Labeling Placement Problem

Zheng He,[†] *and Koichi Harada*[††],

Hiroshima University, Graduate School of Engineering, Japan

Summary

This paper proposes a new learning method based on Hopfield Neural (HN) network to optimize the Point-Feature Labeling Placement (PFLP) problem. The learning method attains a balance between penalty function and original objective function based on the principle of a physical weight balance, and can converge to a solution with better stability. This improved algorithm also allows HN network to be competitive among other traditional algorithms such as genetic algorithm and simulated annealing algorithm when solving the PFLP problem and other constrained problems..

Key words:

Constraint optimization, Hopfield neural network, Learning method, Point-feature labeling placement

1. Introduction

For last several decades, researchers have been trying to use Hopfield Neural (HN) network to solve combinatorial optimization problems because it could find an optimal solution faster than traditional methods. But it was questioned later because the HN network, in its original form, was unable to escape the local minima as a gradient descent technique and was limited by the fact that its penalty parameter approach to solve constrained optimization problems may result in some infeasible or poor quality solutions. Therefore, researchers tried to modify the energy function [1][2], or optimally tune the numerous involved parameters [3][4] in order to make the HN network converge to a feasible solution. Subsequent efforts to confine the HN network into the feasible constraint plane have obtained a method which can ensure the final solution to be feasible [5][6]. Despite the success, the HN network still cannot deal with the combinatorial optimization problems well; and the quality of solutions obtained from HN network is unlikely to be comparable to those optimized by traditional techniques [7].

From optimization viewpoint, the Hopfield neural network and the modified versions essentially belong to the penalty method for solving the constrained real optimization into which a combinatorial optimization is converted. In order for the penalty method to converge to a feasible solution, the weighting factors for the penalty terms must be sufficiently large. However, as the penalty terms become stronger, the role of the original objective function becomes relatively weaker. The solutions thus found are affected more by the penalty terms and hence less favourable in terms of the original objective. Worse still, as they become larger and larger, the problem becomes ill-conditioned. This is a typical problem with the penalty method and explains why it is difficult to obtain good quality solutions and good convergence simultaneously with Hopfield-type network.

This paper proposes a new method to solve combinatorial optimization problems by iterative computations based on Hopfield neural network optimum. In this method, the traditional Hopfield network optimization process is used as a basic calculation unit and is iterated according to certain learning method based on balance principle [8] which can trace the solution obtained from previous unit and regulate parameters of calculation unit for next iteration.. This method can overcome the problems with the Hopfield neural networks in solving combinatorial optimization..

In this paper we apply the proposed method in solving the Point-Feature Label Placement (PFLP) problem, one of typical combinatorial optimization problem. The obtained solution is compared with solutions optimized by Conventional HN Network (CHN), Genetic Algorithm (GA) and Simulated Annealing algorithm (SA). For convenience, we call the proposed method as Improved Hopfield Network (IHN).

2. Learning method in HN network

2.1 Main Principles of HN Network

The HN network was proposed by physicist John J. Hopfield in1982 [9]. It has a recurrent feature, and can be used to solve information retrieval or optimization problems.

The HN network consists of a set of fully interconnected discrete neurons i, which have two possible output states v_i: inactive (v_i=0) and active (v_i=1). Each of the neurons receives signal θ_i from external sources and signals v_j ($j\neq i$) from the rest of neurons of the network which are weighted by a synaptic interconnection ω_{ji} (a weight for the signal v_j received by the neuron i) respectively, see Fig.1.

Manuscript received March 25, 2006.
Manuscript revised March 30 , 2006.

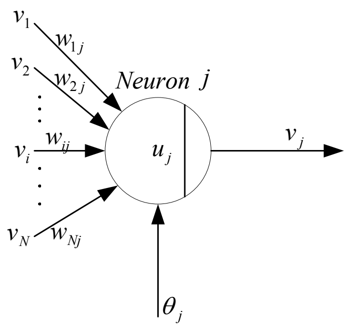


Fig.1. Inputs and output of a typical neuron of HN network

The input to neuron i, which is expressed by u_i, should be the sum of both above terms as the following equation:

$$u_i = \sum_{i \neq j} \omega_{ji} v_j + \theta_i \quad (1)$$

Neurons are selected randomly to update their outputs according to Eq(2), which are active if u_i is larger than the threshold of activation value U_i^{th}; inactive if u_i is smaller than U_i^{th}.. The state of the network is defined as the set of binary outputs of all the neurons.

$$v_i = \begin{cases} 0 & u_i \geq U_i^{th} \\ 1 & u_i < U_i^{th} \end{cases} \quad (2)$$

Assume that in such network the synaptic interconnections are symmetric (ω_{ij}=ω_{ji}) and there are no self feedback connections (ω_{ii}=0), the energy E for the whole network can be determined from energy function as the equation below [9]:

$$E = -\frac{1}{2}\sum_i \sum_j \omega_{ij} v_i v_j - \sum_i \theta_i v_i + \sum_i u_i U_i^{th} \quad (3)$$

Furthermore, if there is any change Δv_i happening in the inputs of a neuron i, the increment of the energy ΔE_i of this neuron is given by Eq(4).

$$\Delta E_i = -(\sum_j \omega_{ij} v_j + \theta_i - U_i^{th})\Delta v_i \quad (4)$$

From this expression, Δv_i is positive when the terms in brackets, defined as the updating condition in Eq(2), is positive; and Δv_i becomes negative in the other case. Therefore the energy increment for the whole network ΔE will always decrease however the input changes.

The ability to minimize the energy function in a very short convergence time makes the CHN described above be very useful in solving the problems with solutions obtained through minimizing a cost function. Therefore, this cost function can be rewritten into the form of the energy function as Eq(3) if the synaptic weights ω_{ij} and the external input θ_i can be determined in advance. And then the network develops from a random initial state and becomes stable until the whole network energy becomes minimum, which is defined as the solution of such problem.

2.2 Learning Method

2.2.1 Balance principle

A general constraint optimization problem can be converted into a non-constraint problem by adding a penalty function g to the original objective function f, as expressed by Eq(5). In order to get optimal and feasible solution, it is important to keep a balance between the original objective function f and the penalty function g.

$$F = \alpha \cdot f + \beta \cdot g \quad (5)$$

Therefore we assign two weighting factors α and β to the two functions respectively to control such balance condition, which is just like the commonly used balance in laboratory experiment. According to the weighting principle of common balance, the weights are usually added onto the lighter side of the balance from large to small in sequence until the balance scale reaches the critical balance position.

In our proposed problem, we treat α and β as the weights and try to obtain the balance between f and g by regulating the weights accordingly. Here we define the feasible optimization solution as the critical balance position.

2.2.2 Learning process

It was already known that HN network solved an optimization problem by decreasing the energy of an energy function corresponding to this optimization problem, as Eq(3) shows. Here, we suppose that the whole energy of the constraint optimization problem be composed by the energy related to objective function E_f and the energy related to penalty function E_g, then the energy function E can also be expressed as the following equation based on Eq(5):

$$E = \alpha \cdot E_f + \beta \cdot E_g \quad (6)$$

The previously introduced balance method will be used to protect the optimization problem from trapping into local minimum. The weighting factors α and β will be regulated based on the balance principle. Whenever α and β are changed, all the weights among all neurons in the whole network have to be updated once, which is obviously a time-consuming work. Therefore, we try to adopt some learning rules to control the balance process and save calculation time.

Fig.2 gives flow chart of this learning method. In this figure, it should be noted that:

a) Before the HN network reaches the stable state, each neuron may change its state for many times during the whole optimization process. So learning rules will be applied only after the whole network becomes stable in stead of when each neuron is updated.
b) HN network (except the first one) uses the result of previous iterate as the initial neuron state.
c) Checking the feasibility of solution using the value of E_g.. $E_{g>0}$ indicates infeasible solution; $E_{g=0}$ indicates infeasible solution.
d) The terminal condition is defined as: the number of times

that HN network needs to reach a stable state should be larger than a predefined value..

The detail about the learning rules will be described later.

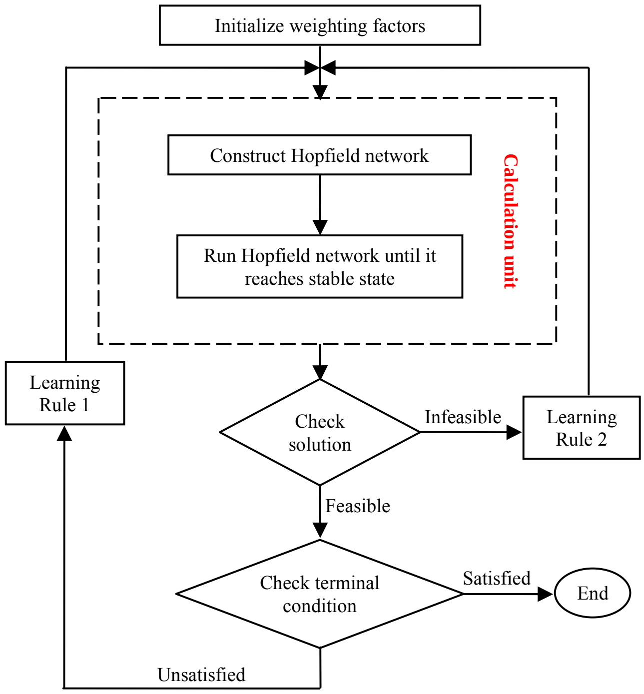


Fig.2. Learning method for constraint optimization

2.2.3 Learning rules

Learning rule 1

(1) Calculate $\Delta E_f = E_f^i - E_f^{i-1}$

(2) Update weighting factor α for objective function

If $\Delta E_f \neq 0$, $\alpha = 1 + |\Delta E_f / E_f^i| * \delta$;

Else, $\alpha = 1 + \delta$..

Here E_f^i is the energy of objective function E_f when the neural network reached stable at the i^{th} time and δ is the parameter to control the learning speed.

Learning rule 2

(1) Calculate $\Delta E_g = E_g^i - E_g^{i-1}$

(2) Update weighting factor β for penalty function

If $\Delta E_g \neq 0$, $\beta = 1 + |\Delta E_g / E_g^i| * \delta$;

Else, $\beta = 1.5 + \delta$..

Here E_g^i is the energy of penalty function E_g when the neural network reached stable at the i^{th} time.

The purpose of those learning rules is to adjust the weights of objective function and penalty function in Eq(6). In the other words, the learning rules change the value of α/β. Learning rule 1 increases α/β to increase the weight of objective function in Eq(6), Learning rule 2 decreases α/β to increase the weight of penalty function in Eq(6).

2.2.4 Analysis for learning method

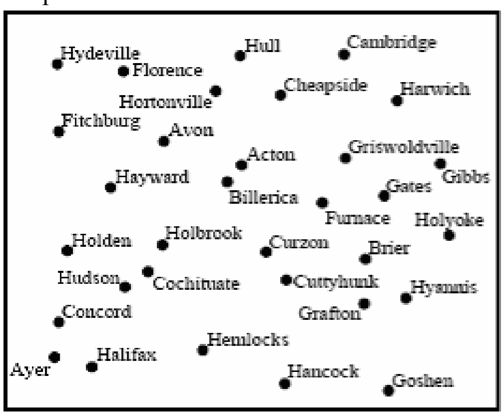
The proposed method is different from Lagrange method, which solves the problem by finding the proper Lagrange multiplier [10]..In our proposed method, tests show the proper ratio for α and β cannot always induce to fine solution. For example, after we solved a specified problem by IHN, we can get the values of α and β corresponding to a fine solution. Then we construct a CHN using those α and β, we cannot always obtain the same result after CHN running.

This is because the purpose of our learning method is not only find proper ratio for α and β but also changing whole energy function to help HN network escape from local minima.

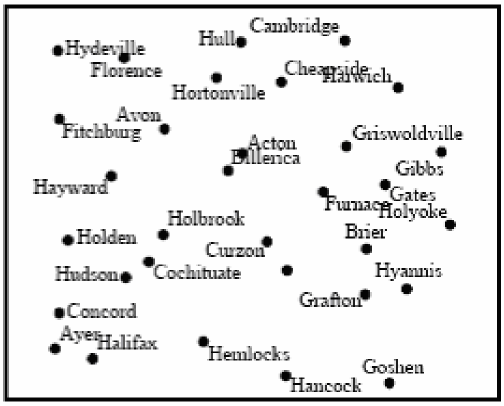
3. Optimization model for PFLP Problem

3.1 Introduction

In cartography, three different label-placement tasks are usually identified: area features labeling (such as oceans or countries), line features labeling (such as rivers or streets), and point features labeling (such as cities or mountain peaks) [11]. Determining the optimal placement of a label for an isolated point feature is a quite different task from the other two tasks; and the three placement tasks will share a common combinatorial aspect when the multiple features are present. Thus complexity arises because the placement of a label may cause global influences due to label-label overlaps. This combinatorial aspect of the label-placement task is independent on the nature of the features being labeled, and become the fundamental source of the difficulty in automating label placement. Fig. 3 shows examples of good labeling and bad labeling. Although this paper just concentrates on PFLP, the discussed optimization model can also be generalized to other label-placements.



(a) Good labeling



(b) Bad labeling

Fig.3. examples of good labeling and bad labeling

The PFLP problem can be modeled as the combinatorial optimization which is stated as follows:
A set of n points is given; each of them must be labeled by assigning its label to one of m predefined positions. A complete label placement is represented by a vector $\vec{x} = (x_1, x_2, ..., x_n)$; each component $x_i \in \{1,2,...,m\} (i = 1,2,...n)$ identifies the assigned position of point i. The eight predefined positions for text labels most commonly used in cartography are shown in Fig.4..

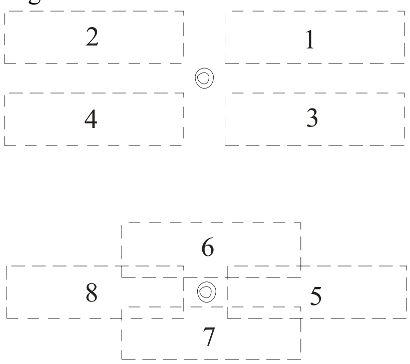


Fig.4. Possible label positions and their desirability relative to a given point

Two objects are of particular importance:
(i) Minimizing the degree to which labels overlap and obscure other features;
(ii) Maximizing the degree to which labels are unambiguously and clearly associated with the features they identify.

For a specific label placement $x_i \in \{1,2,...,m\} (i = 1,2,...n)$, there are three kinds of overlap:
a) Label overlaps the boundary of the map;
b) Label overlaps feature point ;
c) Label overlaps other labels.

The following nomenclatures express overlaps for each specific label based on the above explanations:

$Conb_i$: Use 0 and 1 to indicate whether the specific placement of the point i overlaps the map boundary or not. 1 indicates this label overlap the map boundary.
$Conp_i$: The number of feature points which the specific placement of the point i overlap.
$Conl_i$: The number of conflicting labels which the specific placement of the point i overlap..
In the three kinds of overlap, a) is the most serious one in labeling, and b) is more serious than c) [11]. Therefore the above mentioned relations of importance are considered into the first optimization object by three overlap penalties: A, B and C.. A scalar value $Conf(\vec{x})$ is used to express this object and the optimization object function is given by the equation below.

$$Conf(\vec{x}) = \sum_{i=1}^{n} (A \times Conb_i + B \times Conp_i + C \times Conl_i) \tag{7}$$

The second optimization object can approximately be evaluated through assigning certain value with relative desirability to each possible label positions; and then calculating the sum of the values for all label positions in the specific placement x_i. In cartography, the upper right position is the most preferred, and desirability values corresponding to the position numbers $j=1, 2,...,m$ are depicted in Fig.4 (smaller values indicate more desirable positions).
According to this optimization, the following objective function $f(\vec{x})$, which should be minimized, is used in this work for evaluating a label placement [12][13] :

$$\begin{aligned} f(\vec{x}) &= Conf(\vec{x}) + D\sum_{i=1}^{n} \frac{x_i - 1}{m} \\ &= \sum_{i=1}^{n} (A \times Conb_i + B \times Conp_i + C \times Conl_i) + D\sum_{i=1}^{n} \frac{x_i - 1}{m} \end{aligned} \tag{8}$$

The second term adds a position penalty D which is related to desirability rank of each actual label position.
Marks and Shieber had already shown that the PFLP problem and its various variants were belonged to the NP-completeness [14]. Even though the position desirability is ignored, minimization of label conflicts is also NP-completeness.

3.2 Optimization model

To map the PFLP model onto Hopfield neural network, Eq(2) must be rewritten into 0-1 Integer Programming Problems[15]:
N: Number of point-features in the map;
M: Number of available positions for each feature point;
$Conb_{ij}$: Flags to indicate whether the label of point i in position j overlap map boundary or not, 0 is no overlap and 1 is overlap;
$Conb_{ijmn}$: Flags to indicate whether the label of point m in position n overlaps the label of point i in position j or not, 0 is no overlap and 1 is overlap;
$Conp_{ij}$: Number of feature points overlapped by point i in position j;

x_{ij} ($i=1,2,...,N$; $j=1,2,...,M$): Decision variables to indicate weather the label of point i uses position j or not, 1 is use and 0 is no use.

Based on the above explanation, the objective function of the proposed combinational optimization problem of PFLP has been given by Eq(8) for minimizing the degree to which labels overlap and obscure other features.

$$f = A\sum_{i=0}^{N-1}\sum_{j=0}^{M-1} Conb_{ij}x_{ij} + B\sum_{i=0}^{N-1}\sum_{j=0}^{M-1} Conp_{ij}x_{ij} + C\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}\sum_{k=0}^{N-1}\sum_{l=0}^{M-1} Conl_{ijkl}x_{ij}x_{kl} + D\sum_{i=0}^{N-1}\sum_{j=0}^{M-1} jx_{ij} \tag{9}$$

In addition, the constraint conditions which limit each point to own and only own one label placement can be expressed as a penalty term [16]..

$$g = \sum_{i=0}^{N-1}\sum_{k=0}^{N-1}\sum_{j=0}^{M-1}\sum_{l=0}^{M-1} \delta_{ik}(1-\delta_{jl})x_{ij}x_{kl} - \sum_{i=0}^{M-1}\sum_{j=0}^{M-1} x_{ij} + \sum_{i=0}^{N-1} 1 \tag{10}$$

Here: $\delta_{mn} = \begin{cases} 1 & m = n \\ 0 & m \neq n \end{cases}$

Substitute Eq(9) and Eq(10) into Eq(3), optimization model for PFLP problem can be obtained as Eq(11)..

$$\begin{aligned} F = \alpha\cdot(&A\sum_{i=0}^{N-1}\sum_{j=0}^{M-1} Conb_{ij}x_{ij} + B\sum_{i=0}^{N-1}\sum_{j=0}^{M-1} Conp_{ij}x_{ij} \\ &+ C\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}\sum_{k=0}^{N-1}\sum_{l=0}^{M-1} Conl_{ijkl}x_{ij}x_{kl} + D\sum_{i=0}^{N-1}\sum_{j=0}^{M-1} jx_{ij}) \\ &+ \beta\cdot(\sum_{i=0}^{N-1}\sum_{k=0}^{N-1}\sum_{j=0}^{M-1}\sum_{l=0}^{M-1} \delta_{ik}(1-\delta_{jl})x_{ij}x_{kl} - \sum_{i=0}^{M-1}\sum_{j=0}^{M-1} x_{ij} + \sum_{i=0}^{N-1} 1) \end{aligned} \tag{11}$$

Then comparing with the standard energy function of HN network, PFLP can be mapped onto CHN [9][16].. The weight ω and bias θ of Hopfield neural network is given by Eq(12) accordingly:

$$\begin{aligned} \omega_{ijkl} &= -C\cdot\alpha\cdot Conl_{ijkl} - \beta\cdot\delta_{ik}\cdot(1-\delta_{jl}) \\ \theta_{ij} &= -\frac{\alpha}{2}\cdot(A\cdot Conb_{ij} + B\cdot Conp_{ij} + D\cdot j) + \frac{\beta}{2} \end{aligned} \tag{12}$$

4. Test verifications

4.1 Test Setup

Tests on the PFLP problem were carried out in order to verify the proposed IHN network in solving PFLP problem. The parameters involved in our PFLP model are described as the following:

N point-features with labels of 30×7 units are randomly placed on a map of 792×612 units. Optimization will carried out for ten maps with N=100, 200… 1000 respectively.. Labels were allowed to be placed in the eight positions around the point feature as Fig.4 shows: M=8. Penalty factors are defined as: A=100, B=200, C=16, D=1.

This combinatorial problem was solved by the proposed HN network with and without learning process, as well as two traditional methods, GA and SA, for comparison.

4.1.1 by learning method

Based on Eq(12), parameters α and β should be dynamically adjusted during the learning process (see Fig.2) to obtain valid solutions. Parameter to control the learning speed was setup as δ=0.2. Final solutions this algorithm was unified by using Eq(9) for a convenient comparison..

4.1.2 by SA

SA is one of the best methods to solve a widely complex problems and a lot of solution combinations [17].

The cost function using by SA was Eq(7). The annealing temperature T is defined as Eq(13):

$$T = \frac{k}{\log(t+20)} \tag{13}$$

Where k is a constant value and $k = 20$; t is the time step. The probability to accept bad solution is defined as Eq(14):

$$p = e^{-\frac{\Delta energy}{T}} \tag{14}$$

Where $\Delta energy$ is the change of cost function.

4.1.3 by GA

GA has been considered as a powerful heuristic search method to solve combinational optimization problems. However its execution times are significantly high, although they provided good quality of solutions [18].

The implementation was carried out by a software package named Galib (version 2.4.5), which was instantiated to implement a steady-state genetic algorithm, with 1% of the population replaced each generation [19]. The cost function was Eq(7). An ordered list of point positions was used as the genome, and the genetic operator was an edge recombination crossover operator (partial match crossover was also tried, but performed poorly). The population size was specified as 100. The optimal solution was obtained when the difference between two best solutions from consecutive populations was smaller than a predefined error tolerance of *0.01*.

4.2 Results and Comparisons

4.2.1 Comparison 1: stability

The optimization is carried out for 100 times for a map with *N=100*. Table1 compares the test results in stability and convergence time by the proposed Improved HN network (IHN) and Conventional HN network (CHN). It is obviously that solutions of IHN are valid, and more stable and better than CHN. But the calculation time of IHN is much longer than CHN since the IHN has to call CHN for many times during optimization.

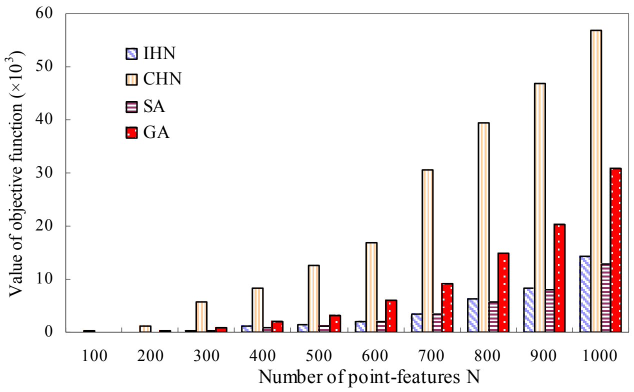
Table 1 Comparison Stability and convergence time

	VALID TIMES	BEST RESULT	AVERAGE SOLUTION	AVERAGE TIME (MS)
CHN	77%	334	367	15
IHN	100%	18	20	397

Some points should be noticed about the HN network here. Although the CHN can be convergent to valid solutions by setting larger value to the ratio of β/α, the solution must be worse since the constraint condition is too much emphasized, so the stability comparison would be of no significance. On the contrary, this ratio should not be too small for similar reason. The values of α and β can be determined through repeated tests, from which it is deduced that when α=1 and β=5, most of the produced solutions by CHN is valid and of good quality.

4.2.2 Comparison 2: solution quality and calculation time

Furthermore, 10 maps including different numbers of point-feature N (N=100,200,…,1000) are randomly created. Optimization process is carried out 20 times for each map respectively, the value of objective function and calculation time for each time are obtained by using IHN, CHN, SA and GA. Fig..5 lists average results of 20 times.



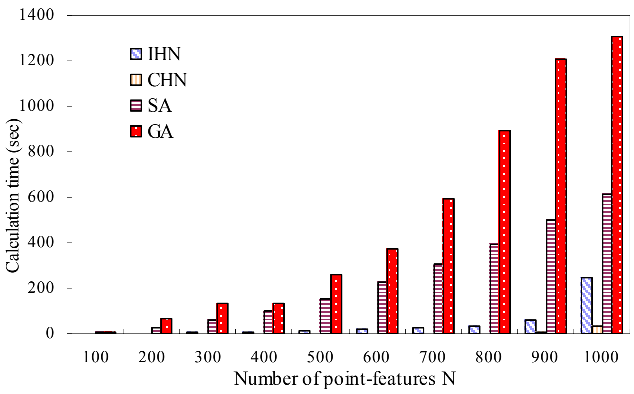


Fig.5. Average solution and calculation time

As the figures show, the IHN performed surprisingly well. The solution quality of IHN is much better than CHN and GA; and is as good as that of SA, and eventually better than SA. Moreover, its calculation time is shorter than that of GA and SA.

Four optimized maps contented 700 points as examples show the same conclusion. Fig.6 indicates the map with optimized label-placements obtained from each algorithm. Labels printed in solid marks overlap other labels, points or boundary of the map. Labels printed in open marks are free of overlaps.

5. Conclusion

This paper proposes a new learning method which applies balance principle based on Hopfield neural networks to solve the constraint optimization problems. An Improved HN network (IHN) was set up and applied to deal with a typical combinatorial optimization problem: Point-feature labeling placement. Results show that IHN successfully overcome the disadvantages of conventional HN network such as: unstable, invalid or bad solution. In addition, this HN network based on learning method demonstrates surprisingly good performance in solution quality and convergence time compared with conventional HN network or traditional optimization algorithms such as GA and SA.

References

[1] R. D. Brandt, Y. Wang, A. J. Laub and S. K. Mitra, “Alternative networks for solving the traveling salesman problem and the list—matching problem,” Int. Conf. Neural Networks, 1998, vol. 2, pp: 333-340.

[2] D. E. VandenBout and T. K. Miller, “A traveling salesman objective function that works,” Int. Conf. Neural Networks1998, vol. 2, pp: 299-303.

[3] B. K. Parsi, “Dynamical stability and parameter selection in neural optimization,” Int. Joint Conf. Neural Networks, 1992, vol. 4, pp: 566-577.

[4] W. K. Lai and G. G. Coghill, “Genetic breeding of control parameters for the Hopfield/Tank neural net,” Int. Joint Conf. Neural Networks, 1992, vol. 4, pp: 618-623.

[5] P. Chu, “A neural network for solving optimization problems with linear equality constraints,” Int. Joint Conf. Neural Networks, 1992, vol.2, pp: 272-277.

[6] A. H. Gee, “Problem solving with optimization networks,” Ph.D. dissertation, Queen’s College, Cambridge, U.K., 1993.

[7] A. H. Gee and R. W. Prager, “Limitations of neural networks for solving traveling salesman problems,” IEEE Trans. Neural Networks, vol. 6, pp: 280-282, 1995.

[8] Z. He and K. Harada, “Balance algorithm for point-feature label placement problem,” lecture notes in computer science, 2005, 3696, pp: 179~184.

[9] J. J. Hopfield and D. W. Tank, “Neural Computation of Decisions in Optimization Problems,” Biological Cybernetics, 1985, vol. 52, pp: 141-152.

[10] Bertsekas, D. P., "Constrained Optimization and Lagrange Multiplier Methods", Academic Press, 1982.

[11] E. Imhof, “Positioning names on maps,” The American Cartographer, 1975, vol. 2(2), pp: 128-144.

[12] J. Christensen, J. Marks and S. Shieber, “An empirical study of algorithms for point-feature label placement,” ACM Trans. Graphics, vol. 14(3), pp: 203-232.

[13] J. Christensen, J. Marks and S. Shieber, “Placing text labels on maps and diagrams,” Graphic Gems IV, 1994, pp: 497-504.

[14] Marks J, Shieber S (1991) The Computational Complexity of Cartographic Label Placement, Technical Report TR-05-91, Harvard University.

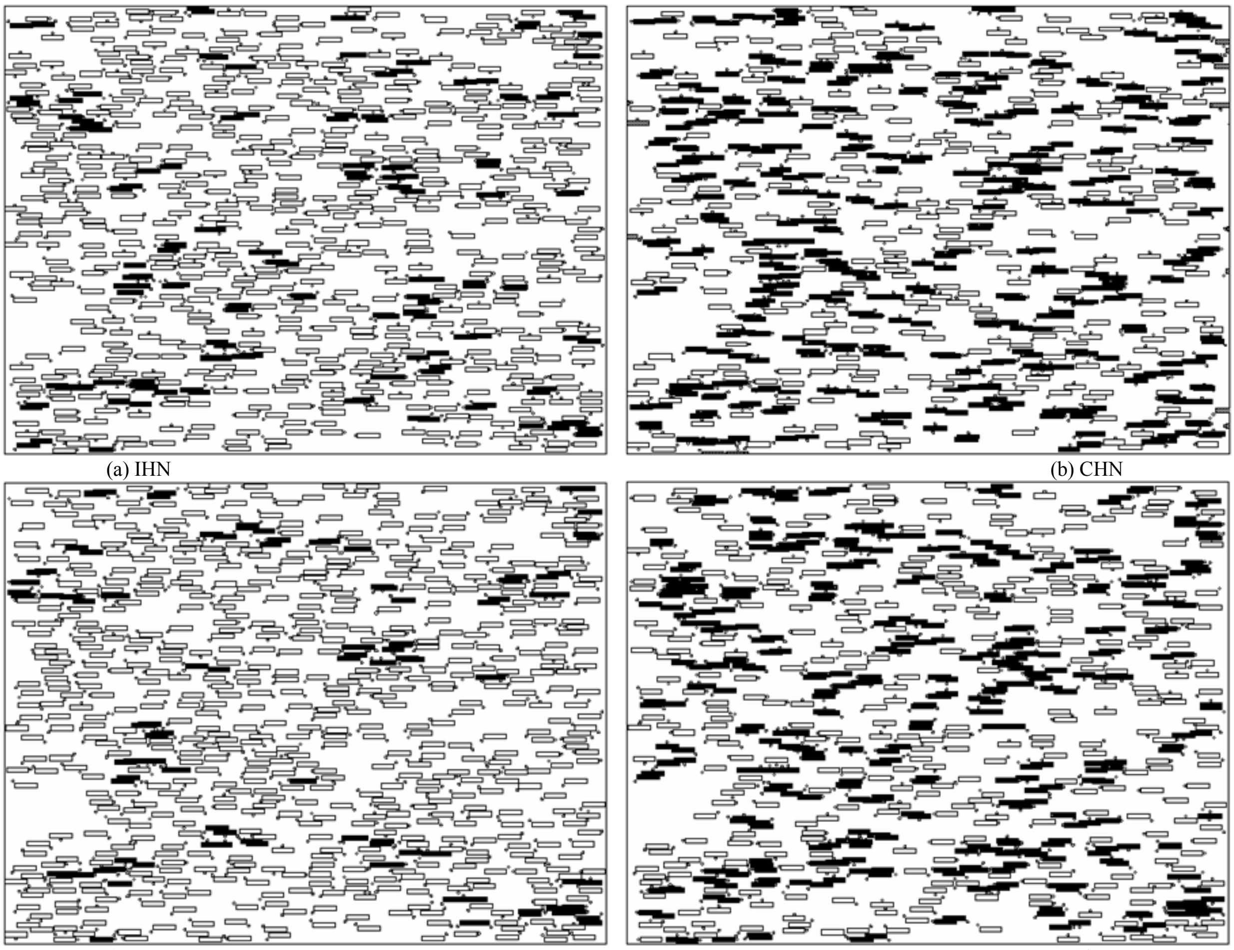
[15] Zoraster S (1990) The Solution of Large 0-1 Integer Programming Problems Encountered in Automated Cartography. Operations Research, 38(5):752-759.

[16] Ramanujam J, Sadayappan P (1995) Mapping Combinatorial Optimization Problems onto Neural Networks. Information Sciences 82:239 -255.

[17] Kirkpatrick S, Gelatt CD, Jr., Vecchi MP (1983) Optimization by simulated annealing. Science, 220:671-680.

[18] Michalewicz Z (1992): Genetic Algorithms+Data Structures=Evolution Programs. Springer, Berlin.

[19] Wall M, Genetic Algorithm Software Library Galib. http://lancet.mit.edu/ga/..



(a) IHN (b) CHN

(c) SA (d) GA

Fig.6. A sample map of 700 point-features with labels placed by four different algorithms.

Zheng He graduated Wuhan University of Technology (China) in 1999, and received ME in 2002 from the same University. After finishing MS, he joined OPENET Co., Ltd. working for developing wireless communication projections. His main research field is computer graphics. He is working for his Ph.D in Hiroshima University.

Koichi Harada received the BE in 1973 from Hiroshima University, and MS and Ph.D in 1975 and 1978, respectively, from Tokyo Institute of Technology. His current research is mainly in the area of computer graphics. Special interests include human computer interface through graphics; 3D data input techniques, data conversion between 2D and 3D geometry, effective interactive usage of curved surfaces. He is a member of ACM, IPS of Japan, and IEICE of Japan.