An Efficient Graph Coloring Algorithm by Merging a Rapid Strategy into a Transiently Chaotic Neural Network with Hysteretic Output Function

Xiuhong Wang[1] and Qingli Qiao[2]

[1] College of Management and Economics, Tianjin University, Tianjin 300072, China
wangxh1965@tju.edu.cn
[2] Department of Biomedical Engineering, Tianjin Medical University, Tianjin 300070, China
Qlqiao@gmail.com

Abstract. In this paper an efficient graph coloring algorithm based on the transiently chaotic neural network (TCNN) is presented. This algorithm apply the TCNN with hysteretic output function instead of logistic output function, this make the model has higher ability of overcoming drawbacks that suffer from the local minimum. Meanwhile, a rapid strategy is merged in this model in order to avoid oscillation and offer a considerable acceleration of converging to the optimal solution. The numerical simulation results demonstrated that the proposed model has higher ability and more rapid speed to search for globally optimal solution of the graph coloring problem than the previous TCNN model with logistic output function and without the rapid strategy.

Keywords: graph coloring problem, hysteretic neural networks, transient chaos, rapid searching strategy.

1 Introduction

The graph coloring problems is one of the classical combinatorial optimization problems having widespread applications in areas such as frequency assignment problems and computer compiler optimization. The graph coloring problem is to color or label the vertices of a graph with the minimum number of colors, such that no two adjacent vertices are the same color. It is more difficult with constraint that the minimum number of colors is required for a given map or graph. In 1976, Appel and Haken [8] solved the four-color problem based on the sequential method whose computation time may be proportional to O(n^2) (where n is the number of regions to be colored) so that it took many hours to solve a large problem.

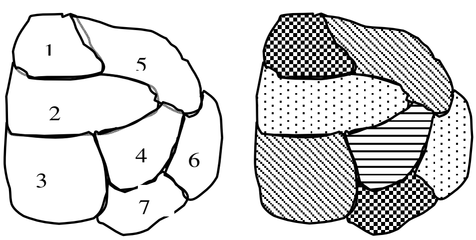
Solving combinatorial optimization problems has been one of the main motifs for the development of neural networks since Hopfield and Tank [1] proposed their recurrent network to traveling salesman problem (TSP). Takefuji [2] have used a discrete Hopfield-type network to solve the four-coloring map problem. While the classical Hopfield model may be trapped at local minimum and fail to reach global minimum of the objective functions, the transiently chaotic neural network (TCNN) was developed [3]. We have applied a TCNN to solve the four-coloring map problems, which have higher ability of searching for the globally optimal solution

H. Deng et al. (Eds.): AICI 2011, Part III, LNAI 7004, pp. 354–361, 2011.

because of its complicated chaotic dynamics [4]. Recently, non-monotonous output functions have been employed in various neural networks. The neuro-dynamics with a non-monotonous have been reported to possess an advantage of the memory capacity superior to the neural network models with a monotonous output function [5]. We have proposed a transiently chaotic neural network model with a hysteresic output function (HTCNN) to solve the graph coloring problems, which has higher ability of overcoming drawbacks that suffered from the local minimum [6]. In order to simulate continuous dynamics and accomplish fast calculation with parallel digital computers, our previous algorithm is operated in a synchronous and discrete way in which case much iteration is required before converging to optimal solution with small time difference. When time difference is large, however, the system becomes oscillatory and the search fails completely. This will make the speed of converging to the optimal solution slower. In this paper, a rapid searching algorithm is proposed and merged into the HTCNN, which offers a considerable acceleration of converging to the optimal solution when the TCNN is updated in a synchronous discrete manner.

2 TCNN with Hysteretic Output Function for the Graph Coloring Problems

In order to map the four-coloring graph problems to Hopfield network, a n×4 two-dimensional neural array is needed, where n is the number of regions to be colored, and a single region requires four neurons for the single-color assignment. Seven-region graph are colored by four colors as shown in Fig. 1 (a). 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.1 (b), where a 7×4 neural array is used. Fig. 1 (c) shows the 7×7 adjacency matrix d of the seven-region graph, which gives



(a)

Four-color

Region n

$$\begin{array}{c} 1\\2\\3\\4\\5\\6\\7 \end{array}\begin{bmatrix} 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0\\ 1 & 0 & 0 & 0 \end{bmatrix}$$

(b)

$$\begin{array}{c|ccccccc} & 1. & 2. & 3. & 4. & 5. & 6. & 7 \\ \hline 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0\\ 2 & 1 & 0 & 1 & 1 & 1 & 0 & 0\\ 3 & 0 & 1 & 0 & 1 & 0 & 0 & 1\\ 4 & 0 & 1 & 1 & 0 & 1 & 1 & 1\\ 5 & 1 & 1 & 0 & 1 & 0 & 1 & 0\\ 6 & 0 & 0 & 0 & 1 & 1 & 0 & 1\\ 7 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{array}$$

(c)

Fig. 1. (a) A 7-region map and four-colored map. (b) Neural representation for the map. (c) An adjacency matrix of the map

the boundary information between regions, where d_{XY}=1 if regions X and Y are adjacent to each other, and d_{XY}=0 otherwise.

In order to consider in such a way that no two adjacent regions are of the same color, the energy function is given by

$$E=\frac{A}{2}\sum_{X=1}^{n}\sum_{i=1}^{4}\sum_{\substack{j=1\\ j\neq i}}^{4}v_{Xi}v_{Xj}+\frac{B}{2}(\sum_{X=1}^{n}\sum_{i=1}^{4}v_{Xi}-n)^{2}+\frac{C}{2}\sum_{X=1}^{n}\sum_{\substack{Y=1\\ Y\neq X}}^{n}\sum_{i=1}^{4}d_{XY}v_{Xi}v_{Yi} \tag{1}$$

Where A, B and C are constant, d is the adjacency matrix. V_{Xi} is the output of the ith neuron in the X region. The first term corresponds to the row constraint in the neural array, which forces one region to be colored by only one color. The second term is the global inhibition to enforce the requirement that exactly "n" neurons are "1". The third term describes the boundary violation between regions; If X and Y regions have a common boundary (d_{XY}=1), X and Y region should not have the same color i. The minima of energy function E, that is, E=0 corresponds to the optimal solution of the four-coloring graph problem.

The connection weighting values $W_{Xi,Yj}$ of the neurons and threshold I_{Xi} of the neural network are

$$W_{Xi,Yj}=-A\delta_{XY}(1-\delta_{ij})-B-Cd_{XY}\delta_{ji} \tag{2}$$

$$I_{Xi}=nB \tag{3}$$

Where δ_{ij}=1,if $i=j$, otherwise δ_{ij}=0. The dynamic equation of the neurons is

$$\frac{du_{Xi}}{dt}=-\frac{u_{Xi}}{\tau}+\sum_{Y}\sum_{j}W_{XiYj}v_{Yj}+I_{Xi} \tag{4}$$

$$v_{Xi}(t)=\frac{1}{1+e^{-u_{Xi}(t)/\varepsilon}}=f(u_{Xi}) \tag{5}$$

Where τ is the time constant, ε is constant controlling the steepness of the sigmoid curve $f(u_{Xi})$. u_{Xi} is the internal state of neuron Xi.

A transiently chaotic neural network with hysteretic output function (HTCNN) for solving the graph coloring problems is created by introducing transient chaos into the system, and its output function is hysteresis [6]. The continuous dynamics of the HTCNN is

$$\frac{du_{Xi}}{dt}=-\frac{u_{Xi}}{\tau}+\sum_{Y}\sum_{j}W_{XiYj}v_{Yj}+I_{Xi}-z(v_{Xi}-I_{0}) \tag{6}$$

$$\frac{dz}{dt}=-\beta_{0}z \tag{7}$$

Where z (t) is the self-feedback connection weight, β_0 ($0<\beta_0<1$) is damping factor, and I_0 is a positive parameter. A value of z is used such that is strong enough to generate the chaotic dynamics for searching the global minima. It is then gradually decayed according to (7) such that the system becomes convergent to a stable fixed point. The transient chaos improved optimization ability is apparent in solving the graph coloring problems [4].

The output function of above HTCNN is a hysteretic function which is depicted in Fig. 2 and is described as:

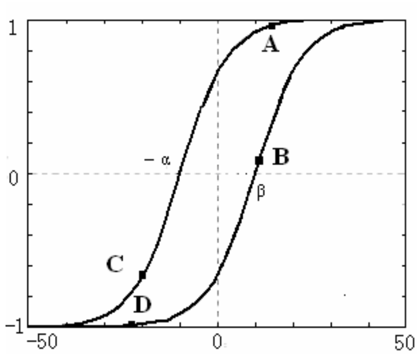


Fig. 2. Hysteretic output function

$$\mathrm{v}(\mathrm{u}/\dot{\mathrm{u}}) = \phi(\mathrm{u}-\lambda(\dot{\mathrm{u}})) = \tanh(\gamma(\dot{\mathrm{u}})(\mathrm{u}-\lambda(\dot{\mathrm{u}}))) \tag{8}$$

Where $\gamma(\dot{\mathrm{u}}) = \begin{cases} \gamma_\alpha, \dot{\mathrm{u}} \geq 0 \\ \gamma_\beta, \dot{\mathrm{u}} < 0 \end{cases}$, $\lambda(\dot{\mathrm{u}}) = \begin{cases} -\alpha, \dot{\mathrm{u}} \geq 0 \\ \beta, \dot{\mathrm{u}} < 0 \end{cases}$ and $(\gamma_\alpha, \gamma_\beta) > 0$, $\beta > -\alpha$.

HTCNN includes memory because of using hysteretic function as neuron's output function. And due to a change in the direction of the input, a system can pull itself out of a saturated region by jumping from one segment of the hysteretic output function to the other segment. This make the HTCNN has a tendency to overcome local minima. The HTCNN improve the optimization capacity in solving the graph coloring problems [6].

The calculation of the above differential equation must be converted to the difference equation by using the Eular discretization when a digital computer is used. Thus, the difference equation is written in the form:

$$u_{Xi}(t+1) = (1-\frac{\Delta t}{\tau})u_{Xi}(t) - z(t)(v_{Xi}(t) - I_0) + \Delta t(\sum_Y \sum_j W_{Xi,Yj} v_{Yj}(t) + I_{Xi}) \tag{9}$$

$$z(t+1) = z(t)(1-\beta) \tag{10}$$

Where let's set $k = (1\text{-}\Delta t/\varepsilon)$, and $\alpha_0 = \Delta t$.

If the time difference $\triangle t$ is small, the search can be carried out successfully, but it requires much iteration before reaching optimal solutions. It is expected that the number of iterations can be reduced by using larger $\triangle t$. When $\triangle t$ is too large, however, the network becomes oscillatory and the searching for the optimal solution fails completely. This has shown that the use of synchronous discrete computation cannot quickly converge to an optimal solution when analog HTCNN is a dopted to the searching. In the following section, we propose an algorithm that overcomes the above dilemma.

3 Merging a Rapid Strategy into HTCNN

Kindo and Kakeya [7] have proposed a geometrical method for analyzing the properties of associative memory model and provided the geometrical outline of the model's dynamics. Based on it, we give a short review of the geometrical explanation on neural dynamics. For simplicity, assume v=f(u)=sgn(u), I=0 (∀i and j), and z=0(∀i and j), Then $|v| = \sqrt{N}$ holds, for the state vector v has +1 or –1 as its components. Therefore v is always on the surface of the hypersphere SN-1 with radius $\sqrt{N}$, N is the number of mutually interconnected neurons. The neural dynamics are divided into two phases. In the first phase, the state vector v(t) is transferred to the vector u(t+1)=(1-△t)u(t)+△tWv(t) linearly (τ=1) with the weight matrix W. In the second phase, the vector is quantized to the nearest state vector that requires the least angle rotation. Therefore, from the hyperspherical viewpoint, linear transformation gives the major driving force of dynamics, while nonlinear transformation generates the terminal points of dynamics. That is to say linear transformation is more important than the nonlinear transformation when we discuss non-equilibrium dynamical properties of neural network. This suggests that the eigenspace analysis of the weight matrix gives major information to explain the global feature of the dynamics. Now we apply this approach to analyze the weight matrix of the neural network for solving the graph coloring problems.

As stated above, the good solutions of the graph coloring problems are located in the low energy area of the state space, and the low energy state of the network corresponds to the state that is composed mainly of the eigenvectors with large eigenvalues. Therefore good solutions have large components of eigenvectors with large eigenvalues and almost no components of eigenvector with negative eigenvalues.

While the synchronous discrete dynamics with large $\triangle t$ do not always realize state transition toward the low energy. Fig. 3 is used to illustrate the simple mechanism. Here the nonlinear transformation is neglected for simplicity, and the dynamics given by $u(t+1)=(1-\triangle t)u(t)+\triangle t$Wv(t) are illustrated. When $\triangle t$ is small, the state vector converges to the eigenvector of W with the largest positive eigenvalue, which spans the low energy states. When $\triangle t$ is large, however, the state vector is attracted to the eigenvector of *W* whose eigenvalue has larger absolute value. This means that the state vector stays in the higher energy states when a negative eigenvalue has larger absolute value than the maximum positive eigenvalue. In this case, $\triangle t$ has to be kept small to ensure convergence to a low energy state though larger $\triangle t$ leads to faster convergence when positive eigenvalues are dominant.

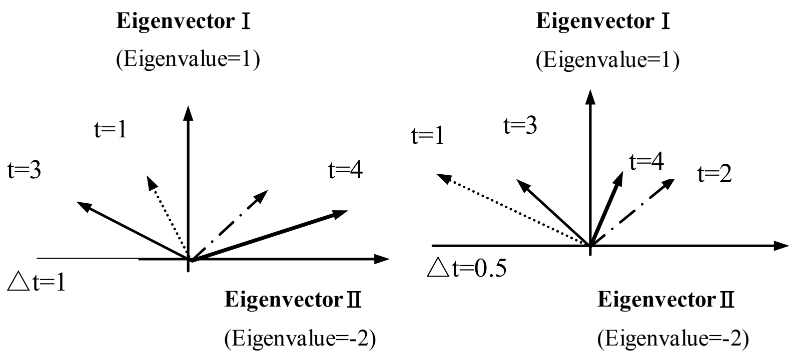


Fig. 3. Convergence of dynamics given by difference equation with small and large time differences △t

From this discussion, it is expected that synchronous and discrete state transition with large $\triangle t$ can proceed toward the low energy states if the effect of the minimal eigenvalue is canceled. The component of the eigenvector with the minimal eigenvalue is reduced from the weight matrix W of neural network for the graph coloring problem by calculating

$$\Psi_{XiYj} = W_{XiYj} - \rho\lambda_{\min} e_{Xi}^{(\min)} e_{Yj}^{(\min)} \tag{11}$$

Where $\lambda_{\min}$ is the minimal eigenvalue and $e_{ij}^{(\min)}$ is its normalized eigenvector, ρ is a positive constant, when ρ=1, the minimal eigenvalue component is eliminated from W completely. However, because reduction of small eigenvalues increases the firing rate of the network, the network converge to a solution which does not satisfy the constraints. To adjust the firing rate, the threshold should be raised in accordance with the increase of the average weight. Since the threshold is always active while the firing rate of neurons in the feasible solutions is $1/N$, the effect of the threshold is N times larger than that of the neurons. Therefore the threshold is

$$\Phi_{Xi} = I_{Xi} + \frac{1}{N}\rho\lambda_{\min}\sum_{Y,j} e_{Xi}^{(\min)} e_{Yj}^{(\min)} \tag{12}$$

This can keep the firing rate to the proper level.

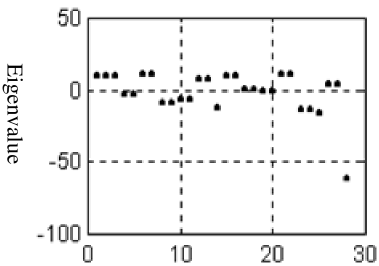
4 Solving Graph Coloring Problems by Merging the Rapid Strategy into HTCNN

In this section, we solve the graph coloring problem based on the transiently chaotic neural network with hysteretic output function which is merged the above rapid searching strategy (RHTCNN), where, the neuron output function is given by the hysteretic function as follows:

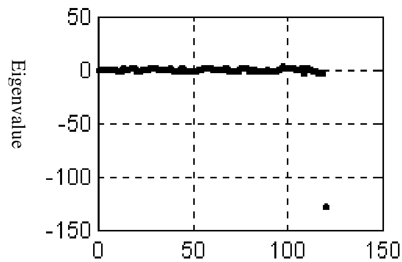
$$v_{Xi} = \begin{cases} 0.5\tanh(\gamma_{Xi}^{\alpha}(u_{Xi} + \alpha_{Xi})) + 0.5; \dot{u}_{Xi} \geq 0 \\ 0.5\tanh(\gamma_{Xi}^{\beta}(u_{Xi} - \beta_{Xi})) + 0.5; \dot{u}_{Xi} < 0 \end{cases} \tag{13}$$

Where, v_{Xi} and u_{Xi} are output value and internal input value of neuron X i.

We use RHTCNN to solve the 7-region and 30-region graph four-coloring problems. In the simulation of 7-region graph four-coloring problem, the parameters are chosen as A=1, B=1, C=1, k=0.985, α_0=0.015, I_0=0.65, β_0=0.01, z_0=0.08, ε=0.04, $\gamma_{Xi}^{\alpha} = \gamma_{Xi}^{\beta} = 50$, $\alpha_{Xi} = \beta_{Xi} = 0.02$, The eigenvalue distribution of the weight matrix W is shown in Fig. 4.(a). It has a extremely small eigenvalue –61.4267. Chose ρ=0.73 for calculating weight matrix Ψ and threshold Φ, time difference △t=0.05, the results with 100 different initial conditions in RHTCNN, HTCNN and TCNN are summarized in Table 1.



(a)



(b)

Fig. 4. Eigenvalue distribution of weight matrix for 7-region and 30-region map four-coloring problem

Part of Chinese map consists of 30 provinces or cites, so the adjacency matrix is given by 30×30 array, and 30×4 neural array is used. The parameters are chosen as A=1, B=1, C=1, k=0.998, α_0=0.012, I_0=0.65, β_0=0.03, z_0=0.1, ε=0.04, $\gamma_{Xi}^{\alpha} = \gamma_{Xi}^{\beta} = 50$, $\alpha_{Xi} = \beta_{Xi} = 0.02$, The eigenvalue distribution of the weight matrix W is shown in Fig. 4.(b). It has an extremely small eigenvalue –127.4.

Chose ρ=0.55 for calculating weight matrix Ψ and threshold Φ, time difference △t=0.02. The results with 100 different initial conditions in RHTCNN, HTCNN and TCNN are summarized in Table 2.

Table 1. Results of RHTCNN, HTCNN and TCNN for 7-region graph four-coloring problems

Neural network	RHTCNN	HTCNN	TCNN
minima of E	0.00227	0.0023	0.0056
Average iterations for convergence	106	137	280

Table 2. Results of rhtcnn, HTCNN and TCNN for 30-region graph four-coloring problems

Neural network	RHTCNN	HTCNN	TCNN
minima of E	1.29×10^{-7}	2.4×10^{-7}	0.0509
Average iterations for convergence	198	238	439

5 Conclusion

In this paper, we proposed an efficient graph coloring algorithm by merging a rapid strategy into a transiently chaotic neural network with hysteretic output function. By using hysteretic output function and transiently chaotic dynamics simultaneously, this algorithm has higher ability of searching global optimal solution. Meanwhile, by eliminating the components of the eigenvectors with eminent negative eigenvalues of the weight matrix, a rapid strategy is presented, which can avoid oscillation and converge to the optimal solution quickly and stably. Numerical simulations of 7-region and 30-region graph four-coloring problems show that the proposed algorithm can accelerate the speed of searching for optimal solution of the graph coloring problems under the synchronous discrete computation.

Acknowledgments. This work was partly supported by the National Nature Science Foundation of China (30870649), the National Basic Research Program (also called 973 Program) of China (Grant No. 2005CB724302) and Natural Science Funds of Tianjin (08JCYBJC03300).

References

1. Hopfield, J.J., Tank, D.W.: "Neural" Computation of Decisions in Optimization Problems. Biolog. Cybern. 52, 141–152 (1985)
2. Takefuji, Y., Lee, K.C.: Artificial neural networks for four-coloring map problems and K-colorability problems. IEEE Trans. Neural Networks 38(3), 326–333 (1991)
3. Chen, L., Aihara, K.: Chaotic simulated annealing by a neural network model with transient chaos. Neural Networks 8(6), 915–930 (1995)
4. Wang, X.H., Wang, Z.O., Qiao, Q.L.: Artificial neural network with transient chaos for four-coloring map problems and k-colorability problems. System Engineering–Theory and Practice 22(5), 92–96 (2002)
5. Nakagawa, M.: An Artificial Neuron Model with a Periodic Activation Function. Journal of the Physical Society of Japan 64, 1023 (1995)
6. Wang, X.H., Qiao, Q.L.: Solving Graph Coloring Problems Based on a Chaos Neural Network with Non-monotonous Activation Function. In: Proceedings of Fifth International Conference on Neural Computation, vol. 1, pp. 404–517 (2009)
7. Kindo, T., Kakeya, H.: A geometrical analysis of associative memory. Neural Networks 11, 39–51 (1998)
8. Appel, K., Haken, W.: The solution of the four-color-map problem. Scientific American (10), 108–121 (1977)