# On Chaotic Simulated Annealing

Lipo Wang and Kate Smith

***Abstract*— Chen and Aihara recently proposed a chaotic simulated annealing approach to solving optimization problems. By adding a *negative self-coupling* to a network model proposed earlier by Aihara *et al.* and gradually removing this negative self-coupling, they used the transient chaos for searching and self-organizing, thereby achieving remarkable improvement over other neural-network approaches to optimization problems with or without simulated annealing. In this paper we suggest a new approach to chaotic simulated annealing with guaranteed convergence and minimization of the energy function by gradually reducing the time step in the Euler approximation of the differential equations that describe the continuous Hopfield neural network. This approach eliminates the need to carefully select other system parameters. We also generalize the convergence theorems of Chen and Aihara to arbitrarily increasing neuronal input–output functions and to less restrictive and yet more compact forms.**

***Index Terms*— Annealing, chaos, energy function, Hopfield, neural network, optimization.**

## I. Introduction

Combinatorial optimization problems are ever present in science and technology. Since Hopfield and Tank's seminal work [6] on solving the travelling salesman problem with a Hopfield neural network (HNN) [5], the HNN's [4], [5] have been recognized as powerful tools for optimization (e.g., [10], [13]). The HNN's have an intriguing property that as each neuron in an HNN updates, an energy function is monotonously reduced until the network stabilizes. One can therefore map an optimization problem to a HNN such that the cost function of the problem corresponds to the energy function of the HNN and the final state of the HNN thus suggests a solution to the optimization problem with a low cost value. While some researchers have described HNN's as nothing more than naive gradient descent machines, the neural framework does brings about some important advantages over other gradient descent techniques: principally the inherent parallelism and hardware implementation which can potentially result in great speed-ups over conventional techniques for combinatorial optimization [12]. The employment of HNN's to solve problems of real-world significance has been hampered, however, by problems over the last decade with solution quality and slow development of suitable hardware to enable large sized problems to be solved.

Earlier attempts at solving various optimization problems with the HNN's suffered from the fact that a HNN can often be trapped at a local minimum in the complex energy terrain, which gives an optimization solution with an unacceptably high cost [15]. Several methods which allow for temporary energy increases, such as *simulated annealing* [7], have been proposed. Recent advances have now made modified HNN's competitive with the best heuristics for solving combinatorial optimization problems, and this has been demonstrated on a variety of real-world problems [10], [11]. The search still continues however, for further or alternative improvements to the standard neural algorithms to address the issue of solution quality: particularly improvements which are easily implementable in hardware.

Chen and Aihara [1] recently proposed a chaotic simulated annealing approach. By adding a *negative self-coupling* to a "transiently chaotic neural network" (TCNN) and gradually removing this negative self-coupling, they used the transient chaos generated by the fading negative self-coupling for searching and self-organizing, thereby achieving remarkable improvement over other neural-network methods, in terms of frequency of finding near-optimal solutions. However, a number of network parameters must be carefully chosen so as to guarantee the convergence of the TCNN and its minimization of the energy upon the removal of the transient chaos [2]. In addition, Chen and Aihara [2] used in proving their convergence theorems a particular sigmoidal function for all neurons. Hardware implementations may not easily ensure this form of input–output (I/O) function and may also need to allow for some variations in I/O among the neurons.

In this letter, we first suggest an alternative approach to chaotic simulated annealing with guaranteed convergence and minimization of the energy function, but *without* the need for choosing any other system parameters. We then generalize the convergence theorems to arbitrarily increasing I/O and to less restrictive and yet more compact forms.

Manuscript received November 9, 1997. This work was supported in part by the Australian Research Council and Deakin University.

L. Wang is with the School of Computing and Mathematics, Deakin University, Clayton, Victoria 3168, Australia.

K. Smith is with the Department of Business Systems, Monash University, Clayton, Victoria 3168, Australia.

Publisher Item Identifier S 1045-9227(98)04613-X.

## II. An Alternative Approach to Chaotic Simulated Annealing

The dynamics of a $n$-neuron continuous Hopfield neural network (CHNN) [5] are described by

$$\frac{du_i}{dt} = -\frac{u_i}{\tau} + \sum_j T_{ij} V_j + I_i \tag{1}$$

where $i = 1, 2, \cdots n$, $I_i$ is the external input to neuron $i$ and is sometimes called "firing threshold" when replaced with $-I_i$. The internal state of neuron $i$, $u_i \in (-\infty, +\infty)$, determines the output of neuron $i$

$$V_i(t) = f_i[u_i(t)] \in [0, 1] \tag{2}$$

or $u_i(t) = f_i^{-1}[V_i(t)]$. Here $f_i$ is the neuronal I/O function for neuron $i$. $f_i$ may differ from neuron to neuron and does not need to have any symmetry properties; however, we assume that $f_i$ is monotonously increasing so that $f_i^{-1}$ exists. For example, the sigmoid function

$$f_i(x) = \tfrac{1}{2}[1 + \tanh(\lambda x)] \tag{3}$$

is often used, with $\lambda$ being the gain of the I/O function.

Hopfield introduced the following function [5] for the CHNN:

$$E(t) = -\tfrac{1}{2}\textstyle\sum_{i,j} T_{ij} V_i(t) V_j(t) - \sum_i I_i V_i(t) + \dfrac{1}{\tau} \displaystyle\sum_i \int_0^{V_i(t)} f^{-1}(\xi)\, d\xi. \tag{4}$$

Since $\partial E/\partial V_i = -\sum_j T_{ij} V_j - I_i + f_i^{-1}(V_i)/\tau = -du_i/dt$, provided that $T_{ij} = T_{ji}$ (regardless the sign of the self-coupling $T_{ii}$), then $dE/dt = \sum_i (\partial E/\partial V_i)(dV_i/dt) = -(du_i/dt) \cdot f_i' \cdot (du_i/dt) < 0$, if $du_i/dt \neq 0$ for at least one $i$, and $dE/dt = 0$ if and only if $du_i/dt = 0$ for all $i$. Hence $E$ is a Lyapunov (energy) function that monotonously decreases as the network updates until the network stabilizes. Based on the existence and the finiteness of such a Lyapunov function, Hopfield concluded [5] that the CHNN must stabilize itself: $\vec{u}(t) \to \vec{u}^o$, as $t \to +\infty$. Here $\vec{u}^o$ is independent of time $t$ and represents the stable fixed point of the network.

Now let us consider a different network which consists of $n$ neurons with the same I/O functions $f_i$, but obeys the following dynamic equations:

$$u_i(t+\Delta t) = \left(1 - \frac{\Delta t}{\tau}\right)u_i(t) + \Delta t\left[\sum_j T_{ij}V_j(t) + I_i\right]. \quad (5)$$

We note that in the limit $\Delta t \to 0$, this equation is the same as the CHNN (1). In fact, (5) is the Euler approximation of the CHNN (1).

The Euler approximation of the CHNN (5) is identical to the TCNN of Chen and Aihara [1], [2], [9]:

$$u_i(t+1) = ku_i(t) + \left[\sum_j T'_{ij}\, V_j(t) + I'_i\right] \quad (6)$$

if $\Delta t = 1$ and

$$k = 1 - \frac{\Delta t}{\tau}, \quad T'_{ij} = \Delta t T_{ij}, \quad I'_i = \Delta t I_i. \quad (7)$$

By adding a large negative self-coupling $T'_{ii}$ and gradually removing it in the TCNN (6), Chen and Aihara [1] markedly enhanced the probability of reaching an optimal or near-optimal solution. In this approach, other system parameters, such as $k$ and $\lambda$ [the gain in the sigmoidal function equation (3)], must be carefully selected with respect to the synaptic weight matrix according to their stability theorems [2], in order to assure network convergence and energy minimization.

We suggest an alternative approach to chaotic simulated annealing: starting from the Euler approximation of the CHNN (5) with a large time-step $\Delta t$, where the dynamics are chaotic [2], we gradually reduce the time-step $\Delta t$. The system is guaranteed to converge and to minimize the CHNN energy function (4), since in the limit of $\Delta t \to 0$, the system approaches the CHNN which is stable and minimizes the CHNN energy function. This approach does not require difficult choices of any system parameters to assure network convergence and energy conversion.

## III. Generalized Stability Theorems

Chen and Aihara [2] derived stability theorems for both the Euler approximation of the CHNN (5) and the TCNN (6) for both synchronous and asychronous updating, using a particular form of the neuronal I/O (3). We now generalize their theorems to arbitrarily increasing I/O functions (2) and to less restrictive but more compact forms.

Let us calculate the change in energy between two time steps when the TCNN (6) is updated *synchronously*, according to (4)

$$\begin{aligned}\frac{\Delta E(t)}{\Delta t} \equiv \frac{E(t+1) - E(t)}{\Delta t} &= -\tfrac{1}{2}\sum_{i,j} T'_{ij}\,\Delta V_i(t)\,\Delta V_j(t) \\ &+ k\sum_i \Delta V_i(t)\, f_i^{-1}[V_i(t)] - \sum_i \Delta V_i(t)\, f_i^{-1}[V_i(t+1)] \\ &+ (1-k)\sum_i \{G_i[V_i(t+1)] - G_i[V_i(t)]\}. \quad (8)\end{aligned}$$

where $G_i(V_i) \equiv \int_0^{V_i} f_i^{-1}(\xi)\, d\xi$ and $\Delta V_i(t) \equiv V_i(t+1) - V_i(t)$. Expanding $G_i$ at $V_i(t+1)$ and using the fact that $G_i$ is concave-up [8], we obtain

$$\begin{aligned}G_i[V_i(t+1)] - G_i[V_i(t)] \le\; & G'_i[V_i(t+1)]\,\Delta V_i(t) \\ & - \tfrac{1}{2}\min\left\{\tfrac{d^2 G_i}{d^2 V_i}\right\}[\Delta V_i(t)]^2 \quad (9)\end{aligned}$$

where $\min\{d^2G_i/d^2V_i\} = \min\{1/f'_i\} \equiv 1/\beta_{\max}$ is the minimum curvature of $G_i$, $\beta_{\max}$ being the maximum slope of the I/O functions. Furthermore, if $k \ge 0$

$$-k\{f_i^{-1}[V_i(t+1)] - f_i^{-1}[V_i(t)]\}\,\Delta V_i(t) \le -\frac{k}{\beta_{\max}}[\Delta V_i(t)]^2. \quad (10)$$

Combining (8)–(10), in the case where $1 \ge k \ge 0$, we obtain

$$\frac{\Delta E(t)}{\Delta t} \le -\tfrac{1}{2}\sum_{i,j}\left[T'_{ij} + \tfrac{(1+k)}{\beta_{\max}}\,\delta_{ij}\right]\Delta V_i(t)\,\Delta V_j(t) \quad (11)$$

where $\delta_{ij} = 0$ if $i \ne j$ and $\delta_{ij} = 1$ if $i = j$. Hence $\Delta E(t) \le 0$, or the network is stable, if matrix $\{T'_{ij} + [(1+k)/\beta_{\max}]\delta_{ij}\}$ is positive-definite. Therefore a sufficient stability condition for a synchronous TCNN (6) is

$$1 \ge k \ge 0, \quad \text{and} \quad \frac{(1+k)}{\beta_{\max}} < -T'_{\min} \quad (12)$$

where $T'_{\min}$ is the minimum eigenvalue of matrix $T'_{ij}$. For $k > 1$, we expand $G_i$ at $V_i(t)$ and obtain, instead of (9)

$$\begin{aligned}&G_i[V_i(t+1)] - G_i[V_i(t)] \\ &\quad \ge G'_i[V_i(t)]\Delta V_i(t) + \frac{1}{2}\min\left\{\frac{d^2 G_i}{d^2 V_i}\right\}[\Delta V_i(t)]^2. \quad (13)\end{aligned}$$

We thus have an alternative stability condition for the TCNN in synchronous mode, similar to (12)

$$k > 1, \quad \text{and} \quad \frac{2k}{\beta_{\max}} > -T'_{\min}. \quad (14)$$

The stability condition for the Euler approximation of the CHNN (5) can be derived easily from (7) and (12), with the usual assumption that $\tau > 0$

$$\Delta t \le \tau, \quad \Delta t < \frac{2\tau}{1 - \beta_{\max}T_{\min}}, \quad \text{and} \quad T_{\min} \le 0. \quad (15)$$

Both the TCNN and the Euler approximation of the CHNN are stable if $T'_{\min} = \Delta t T_{\min} > 0$, since the matrix in (11) is automatically positive-definite. It is straightforward to derive that the stability conditions for both the TCNN and the Euler approximation of the CHNN for *asynchronous* updating are obtained from the conditions for *synchronous* updating (12), (14), and (15) with $T'_{\min}$ (or $T_{\min}$) replaced by $\min\{T'_{ii}\}$ (or $T_{ii}$).

Our stability conditions consist of two parameter regions for the TCNN (12) and (14) and one parameter region for the Euler approximation of the CHNN (15), and are therefore more compact compared to those of Chen and Aihara [2] (three regions for each type of network). Furthermore, our conditions are less restrictive on the parameters involved. For instance, in the one-neuron TCNN considered by Chen and Aihara [2], the I/O function is given by (3) with a gain $\lambda = 125$. Hence $\beta_{\max} = \lambda/2 = 62.5$. With $k = 0.9$, Chen and Aihara's [2] theorems indicate that the network is stable if $-T'_{11} < 0.0288$, whereas our condition (12) gives $-T'_{11} < 0.0304$ and is closer to the bifurcation point $-T'_{11} = 0.0331$.

## IV. Conclusions

In summary, we have proposed an alternative approach to chaotic simulated annealing, in which the time-step $\Delta t$ in the Euler approximation of the CHNN starts from a large value, where chaos exists, and reduces to a small value so that the network stabilizes. This approach guarantees convergence and minimization of the energy function, and eliminates the need of choosing other system parameters. It should prove to be a powerful tool for efficiently obtain optimal or near-optimal solutions to a variety of optimization problems, which is currently under investigation. We have also generalized the Chen–Aihara convergence theorems for the transiently chaotic neural network and the Euler approximation of the CHNN to arbitrarily increasing neuronal I/O functions and to less restrictive but more compact forms. This should be useful for both hardware implementations and software simulations, as well as further theoretic analysis of the systems.

ACKNOWLEDGMENT

Many helpful comments and suggestions from the reviewers are gratefully appreciated.

REFERENCES

[1] L. Chen and K. Aihara, "Chaotic simulated annealing by a neural-network model with transient chaos," *Neural Networks,* vol. 8, no. 6, pp. 915–930, 1995.

[2] ——, "Chaos and asymptotical stability in discrete-time neural networks," *Physica,* vol. 104, pp. 286–325, June, 1997.

[3] F. Fogelman-Soulié, C. Mejia, E. Goles, and S. Martinez, "Energy function in neural networks with continuous local functions," *Complex Syst.,* vol. 3, pp. 269–293, 1989.

[4] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," in *Proc. Nat. Academy Sci. USA,* Apr. 1982, vol. 79, pp. 2554–2558.

[5] ——, "Neurons with graded response have collective computational properties like those of two-state neurons," in *Proc. Nat. Academy Sci. USA,* May 1984, vol. 81, pp. 3088–3092.

[6] J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," *Biol. Cybern.,* vol. 52, pp. 141–152, 1985.

[7] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science,* vol. 220, pp. 671–680, 1983.

[8] C. M. Marcus and R. M. Westervelt, "Dynamics of iterated-map neural networks," *Phys. Rev. A,* vol. 40, pp. 501–504, July 1989.

[9] H. Nozawa, "A neural-network model as a globally coupled map and applications based on chaos," *Chaos,* vol. 2, no. 3, p. 377–86, 1992.

[10] K. Smith and M. Palaniswami, "Static and dynamic channel assignment using neural networks," *IEEE J. Select. Areas Commun.,* vol. 15, no. 2, pp. 238–249, 1997.

[11] K. Smith, M. Palaniswami, and M. Krishnamoorthy, "Neural techniques for combinatorial optimization with applications," *IEEE Trans. Neural Networks,* accepted.

[12] M. Verleysen and P. Jespers, "An analog VLSI implementation of Hopfield's neural network," *IEEE Micro,* pp. 46–55, Dec. 1989.

[13] L. Wang, "Discrete-time convergence theory and updating rules for neural networks with energy functions," *IEEE Trans. Neural Networks,* vol. 8, pp. 445–447, Mar. 1997.

[14] ——, "On the dynamics of discrete-time, continuous-state Hopfield neural networks," *IEEE Trans. Circuits Syst. II,* to be published.

[15] 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, 1988.

# Comments on "The Effects of Quantization on Multilayer Neural Networks"[1]

Oh-Jun Kwon and Sung-Yang Bang

In this letter we point out and correct the errors in the above paper[1] in the derivations of the following equations:

1) in the left column of p. 1147

$$\sigma^2_{y_i^0} = \frac{(\Delta_0 \cdot 2^N)^2}{144} \cdot K_1 \tag{1}$$

2) in the right column of p. 1147

$$\sigma^2_{y_i^1} = \frac{(\Delta_1 \cdot 2^N)^2}{12} \cdot \frac{K_2}{2A}\left[\frac{A}{2} - \tanh\left(\frac{A}{2}\right)\right]. \tag{2}$$

We think the derivations should be as follows.

a) Since the products of the weight and the input are independent, we can rewrite the variance of $y_i^0$ as follows:

$$\begin{aligned}\sigma^2_{y_i^0} &= \mathrm{var}\left[\sum_{k=0}^{K_1-1} w_{ik}^0 x_k^0\right] \\ &= \sum_{k=0}^{K_1-1} \mathrm{var}[w_{ik}^0 x_k^0]. \end{aligned}\tag{3}$$

Also $w_{ik}^0$ and $x_k^0$ are independent and $E[w_{ik}^0] = 0$ by the assumption of the uniform distribution of the weights given by [1]. And, by the assumption that the inputs are uniformly distributed between zero and one, the expectation of an input is

$$E[x_k^0] = \frac{1}{2}. \tag{4}$$

Therefore

$$\begin{aligned}\mathrm{var}[w_{ik}^0 x_k^0] &= E[(w_{ik}^0 x_k^0)^2] - E^2[w_{ik}^0 x_k^0] \\ &= E[(w_{ik}^0 x_k^0)^2] \\ &= E[(w_{ik}^0)^2] \cdot E[(x_k^0)^2] \\ &= \mathrm{var}[w_{ik}^0] \cdot (\mathrm{var}[x_k^0] + E^2[x_k^0]) \\ &= \frac{(\Delta_0 \cdot 2^N)^2}{12} \cdot \left(\frac{1}{12} + \frac{1}{4}\right) \\ &= \frac{(\Delta_0 \cdot 2^N)^2}{36}. \end{aligned}\tag{5}$$

Finally, substituting (5) into (3), we obtain

$$\sigma^2_{y_i^0} = \frac{(\Delta_0 \cdot 2^N)^2}{36} \cdot K_1. \tag{6}$$

b) The output of the hidden neurons is between zero and one since the following sigmoidal function is used in neurons:

$$f(u) = (1 + e^{-u})^{-1}. \tag{7}$$

By the assumption given by [1] that the outputs of the hidden neurons are uniformly distributed, the expectation of the output of a hidden neuron is

$$E[x_l^1] = \tfrac{1}{2}. \tag{8}$$

Therefore, as in (a), we can derive the variance of $y_i^1$ as follows:

$$\begin{aligned}\sigma^2_{y_i^1} &= \mathrm{var}\left[\sum_{l=0}^{K_2-1} w_{il}^1 x_l^1\right] \\ &= \sum_{l=0}^{K_2-1} \mathrm{var}[w_{il}^0 x_l^1] \\ &= \sum_{l=0}^{K_2-1} \mathrm{var}[w_{il}] \cdot (\mathrm{var}[x_l^1] + E^2[x_l^1]) \\ &= K_2 \cdot \frac{(\Delta_1 \cdot 2^N)^2}{12}\left\{\frac{1}{2A}\left[\frac{A}{2} - \tanh\left(\frac{A}{2}\right)\right] + \frac{1}{4}\right\} \\ &= \frac{(\Delta_1 \cdot 2^N)^2}{12} \cdot \frac{K_2}{2A}\left[A - \tanh\left(\frac{A}{2}\right)\right]. \end{aligned}\tag{9}$$

Manuscript received November 9, 1997.

The authors are with the Department of Computer Science and Engineering, Pohang University of Science and Technology, Pohang, 790-784, Korea.

Publisher Item Identifier S 1045-9227(98)04613-X.

[1] G. Dündar and K. Rose, *IEEE Trans. Neural Networks*, vol. 6, pp. 1446–1451, Nov. 1995.