IEICE The Institute of Electronics, Information and Communication Engineers

IEICE Proceeding Series

Synchronous Exponential Chaotic Tabu Search for Analog-Digital Hybrid Parallel Hardware Systems

Akihito Toyoda, Yoshihiko Horio, Kazuyuki Aihara

Vol. 2 pp. 370-373
Publication Date: 2014/03/18
Online ISSN: 2188-5079

2013 International Symposium on Nonlinear Theory and its Applications
NOLTA2013, Santa Fe, USA, September 8-11, 2013

Synchronous Exponential Chaotic Tabu Search for Analog-Digital Hybrid Parallel Hardware Systems

Akihito Toyoda†, Yoshihiko Horio† and Kazuyuki Aihara‡

†Tokyo Denki University
5 Senju Asahi-cho, Adachi-ku, Tokyo, Japan
Phone: +81-3-5284-5333
E-mail: 12kme35@ms.dendai.ac.jp
‡The University of Tokyo
4-6-1 Komaba, Shibuya-ku, Tokyo, Japan

Abstract—Exponential chaotic tabu search is an effective method for solving combinatorial optimization problems. Several modifications such as synchronous updating of the neuronal states has been introduced to try to take advantage of parallel processing in analog-digital hybrid hardware systems. Exponential chaotic tabu search often involves a 2-opt exchange. However, multiple neurons may fire during synchronous update, which prevents the unique determination of the target for 2-opt exchange. To overcome this problem, several neuron selection methods have been proposed, in which neurons are sorted according to the values of their internal states, but these are computationally intensive. These methods are also not suited to a quick and compact hardware implementation because of complicated parameter settings and physical restrictions in hardware devices.

In this paper, we propose a simplified synchronous exponential chaotic tabu search algorithm which is suitable for analog-digital hybrid parallel hardware implementation. First, we modify the chaotic neuron model. Second, we exclude global couplings between chaotic neurons, which impose a heavy hardware burden. Finally, we take into account all the possible restrictions and properties of hardware systems. Furthermore, we can optimize the performance simply by adjusting the external threshold value of the neurons. We confirm the efficiency of the proposed method for quadratic assignment problems through numerical simulations.

1. Introduction

Chaotic tabu search algorithms [1] implemented using chaotic neural networks are an effective method for solving quadratic assignment problems (QAPs), and hardware implementations are substantially faster than software implementations, allowing large-scale QAPs to be solved within reasonable time frames. A synchronous updating algorithm for a hardware implementation that simultaneously updates all neuronal states has been proposed [2]. However, multiple neurons may fire simultaneously during synchronous updates, preventing the 2-opt exchange from selecting a single neuron. To solve this problem, several neuron selection methods have been proposed, and the effectiveness of these methods has been confirmed by simulations [2–5]. The neuron selection methods proposed in [2–4] sort neurons according to their internal state values. For a size-N QAP, the total number of neurons is N^2, so sorting all neuronal states is time-consuming when N is large, which reduces the improvement brought by hardware speed. Moreover, high-precision (meaning large and expensive) hardware is needed to detect the infinitesimal differences among neuronal states. Therefore, in [5], a neuron selection method which does not rely on neuronal state sorting was proposed. However, this synchronous exponential chaotic tabu search algorithm is still not suitable for hardware implementation because of hardware restrictions.

In this paper, we propose an improved synchronous exponential chaotic tabu search algorithm for quadratic assignment problems which is suitable for hardware implementation. First, we modify the chaotic neuron model used in the synchronous exponential chaotic tabu search. Next, we remove global couplings among neurons, which prevent the realization of large-scale hardware-based tabu search systems. In addition, we can easily optimize the performance of the system by adjusting the threshold values applied externally to the neurons. Numerical simulations confirm the effectiveness of the proposed method, and show that performance is improved by increasing the number of iterations.

2. Quadratic Assignment Problem (QAP)

A QAP of size N consists of N locations and N units. An $N \times N$ distance matrix denotes distances between pairs of locations, and an $N \times N$ flow matrix describes mutual relations among units. A QAP is defined so that we need to find an assignment of the units to the locations that minimizes the cost function $F(\boldsymbol{p})$ given by

$$F(\boldsymbol{p}) = \sum_{i=1}^{N} \sum_{j=1}^{N} a_{ij} b_{p(i)p(j)}, \tag{1}$$

where $\boldsymbol{p}$ is a permutation of N elements given by Eq. (2) that expresses a feasible solution, a_{ij} is the distance between locations i and j, $p(i)$ represents the ith element of permutation $\boldsymbol{p}$, and $b_{p(i)p(j)}$ is the flow between units $p(i)$ and $p(j)$.

$$\begin{array}{rcl} \text{index} & : & 1\ ,\ 2\ ,\cdots,\ \ i\ ,\cdots,\ N \\ \boldsymbol{p} & : & \{p(1), p(2), \cdots, p(i), \cdots, p(N)\}. \end{array} \tag{2}$$

3. Synchronous exponential chaotic tabu search algorithm

Improved chaotic tabu search algorithms suitable for hardware implementation were proposed in [2–4]. These improved algorithms, referred to as synchronous updating algorithms in this paper, simultaneously update the states of all neurons.

The $(i,\ j)$th neuron in an $N\times N$ neural network used in the synchronous updating algorithm is described as follows.

$$\xi_{ij}(t+1) = \beta(F_1^{\boldsymbol{p}}(t) - F_{ij}^{\boldsymbol{p}}(t)), \tag{3}$$
$$\eta_{ij}(t+1) = k_f \eta_{ij}(t) - \alpha_\eta Y_{p(j)q(i)}(t) + R, \tag{4}$$
$$\zeta_{ij}(t+1) = k_r \zeta_{ij}(t) - \alpha_\zeta y_{ij}(t) + R, \tag{5}$$
$$x_{ij}(t+1) = \xi_{ij}(t+1) + \eta_{ij}(t+1) + \zeta_{ij}(t+1), \tag{6}$$
$$y_{ij}(t+1) = f(x_{ij}(t+1)), \tag{7}$$
$$f(x) = \frac{1}{1+\exp(-\frac{x}{\epsilon})}, \tag{8}$$

where $F_1^{\boldsymbol{p}}(t)$ is an initial objective function at time t, $F_{ij}^{\boldsymbol{p}}(t)$ is the objective function when element i is assigned to the jth index of $\boldsymbol{p}$, $\xi_{ij}(t)$ is the gain when the ith element is assigned to the jth index, $\eta_{ij}(t)$ is the tabu effect for assignment $(p(j), q(i))$, $\zeta_{ij}(t)$ is the tabu effect for assignment $(i,\ j)$, $x_{ij}(t)$ is the internal state of the $(i,\ j)$th neuron, $y_{ij}(t)$ is the output of the $(i,\ j)$th neuron, β is a scaling parameter for the gain, R is an external bias, k_r and k_f are decay parameters of the tabu effect, α is a scaling parameter for the tabu effect, $Y_{p(j)q(i)}(t)$ is 1 when $p(j)$ is assigned to index $q(i)$ and 0 otherwise, $f(\cdot)$ is a nonlinear output function, and ϵ is the gain parameter of $f(\cdot)$.

4. Neuron selection method for synchronous updates

A neuron selection method for determining the target of 2-opt exchange was proposed in [2]. The algorithm (abbreviated as MT-A) is defined as follows.

a) The internal states of all neurons in the network are updated simultaneously.
b) Neurons which have fired are sorted according to the values of $x_{ij}(t)$ in descending order. The first U neurons are selected, where U is a user-specified parameter.
c) The current permutation $\boldsymbol{p}_{\text{current}}$ is tentatively updated by using each of the selected neurons. For the kth neuron in the list we denote the resulting tentative permutation by $\boldsymbol{p}_k$. We also calculate the cost $F(\boldsymbol{p}_k)$.
d) In step c) we obtain U different costs $(F(\boldsymbol{p}_1), F(\boldsymbol{p}_2), \ldots, F(\boldsymbol{p}_U))$ corresponding to U tentative permutations $(\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_U$, respectively). The neuron that incurs the smallest cost is chosen and the current permutation $\boldsymbol{p}_{\text{current}}$ is definitively updated by using that neuron.

As shown above, MT-A uniquely selects the target neuron by comparing the internal state values of U neurons. However, the computational load increases with the number of neurons to be sorted. In addition, high-precision circuits are required to detect small differences between internal state values.

5. Improved synchronous exponential chaotic tabu search algorithm

Neuron selection methods other than MT-A have been proposed [2–4]; however, the settings of many parameters are difficult to determine because of the complexity of the algorithms. In addition, all previously proposed neuron selection methods rely on sorting the neurons according to their internal state values, which imposes a heavy load on the hardware. With this in mind, we modified MT-A to eliminate the need for sorting. Instead the modified algorithm makes a random selection of U neurons from neurons whose internal state values exceed a certain threshold V_{ref} [5]. However, even with this modification, it is still difficult to find optimal values for the parameters.

To improve our modified version of MT-A, we make several further alterations. First, we introduce a threshold θ in place of the reference V_{ref}. By adjusting θ, we can easily control the firing rate of the neurons in the neuron selection method [5]. The resulting chaotic neuron model is given by

$$\xi_{ij}(t+1) = \beta(F_1^{\boldsymbol{p}}(t) - F_{ij}^{\boldsymbol{p}}(t)), \tag{9}$$
$$\eta_{ij}(t+1) = k_f \eta_{ij}(t) - \alpha_\eta Y_{p(j)q(i)}(t) + (1-k_f)\theta, \tag{10}$$
$$\zeta_{ij}(t+1) = k_r \zeta_{ij}(t) - \alpha_\zeta y_{ij}(t) + (1-k_r)\theta, \tag{11}$$
$$x_{ij}(t+1) = \xi_{ij}(t+1) + \eta_{ij}(t+1) + \zeta_{ij}(t+1), \tag{12}$$
$$y_{ij}(t+1) = f(x_{ij}(t+1)), \tag{13}$$
$$f(x) = \frac{1}{1+\exp(-\frac{x}{\epsilon})}, \tag{14}$$
$$Y_{p(j)q(i)} = \begin{cases} 1 & (y_{p(j)q(i)} > 0.5), \\ 0 & (y_{p(j)q(i)} \leq 0.5). \end{cases} \tag{15}$$

Since we can control the firing rate of neurons by adjusting θ, it is possible to optimize the system performance for many different parameter settings.

Equation (10) includes global couplings among neurons in the form of $Y_{p(j)q(i)}$, which impose a large load on the hardware. As the 2nd improvement, we completely omit the $\eta_{ij}(t)$ terms (Eq. (10)) to alleviate this load. The resulting chaotic neuron model is given by

$$\begin{aligned}
\xi_{ij}(t+1) &= \beta(F_1^{\boldsymbol{p}}(t) - F_{ij}^{\boldsymbol{p}}(t)), && (16)\\
\zeta_{ij}(t+1) &= k_r\zeta_{ij}(t) - \alpha_\zeta y_{ij}(t) + (1-k_r)\theta, && (17)\\
x_{ij}(t+1) &= \xi_{ij}(t+1) + \zeta_{ij}(t+1), && (18)\\
y_{ij}(t+1) &= f(x_{ij}(t+1)), && (19)\\
f(x) &= \frac{1}{1+\exp(-\frac{x}{\epsilon})}. && (20)
\end{aligned}$$

Third, we incorporate the hardware characteristics into the neuron model by introducing the following piecewise linear functions in the internal states of the neuron model.

$$\xi_{ij}(t) = \begin{cases} 1 & (1 \le \xi_{ij}(t)), \\ \xi_{ij}(t) & (-1 < \xi_{ij}(t) < 1), \\ -1 & (-1 \ge \xi_{ij}(t)), \end{cases} \quad (21)$$

$$\zeta_{ij}(t) = \begin{cases} 1 & (1 \le \zeta_{ij}(t)), \\ \zeta_{ij}(t) & (-1 < \zeta_{ij}(t) < 1), \\ -1 & (-1 \ge \zeta_{ij}(t)), \end{cases} \quad (22)$$

$$x_{ij}(t) = \begin{cases} 1 & (1 \le x_{ij}(t)), \\ x_{ij}(t) & (-1 < x_{ij}(t) < 1), \\ -1 & (-1 \ge x_{ij}(t)). \end{cases} \quad (23)$$

Fourth, to eliminate the need for sorting, we randomly reassign neuron numbers in the neural network. Then, we select U neurons that have fired from a block of 50 successive neurons in the randomized chaotic neural network. For the 2-opt exchange, we select from among these U neurons the neuron that gives the minimum cost.

Finally, we reselect the neuron block in each iteration by sequentially sliding the block by W neurons, where W is a parameter.

6. Numerical simulations

We conducted simulations using the neuron model given in Eqs. (16) to (23) for the Kra30a/b and the Lipa50a/b benchmark problems [6] with $U = 5$. In the simulations, we used the GAP performance measure:

$$GAP = \frac{\sum_{n=1}^{TR} EM_n}{TR} \times 100\ [\%], \quad (24)$$

where TR is the total number of trials, and EM_n is the gap between the optimal solution and the best solution obtained in the nth trial.

We used a fixed parameter set; $\alpha_\zeta = 0.3$, $\beta = 0.5$, and $k_r = 0.3$. In addition, we adjusted the value of θ so that the average firing rate of the chaotic neural network during one trial was about 10%. Random initial permutations were used for each trial and we varied the number of iterations in a trial from 1000 to 100000 for each problem. The performance measure, GAP, was calculated with $TR = 100$ and $U = 5$. We compared the results of our proposed method with the results obtained using the existing MT-A approach.

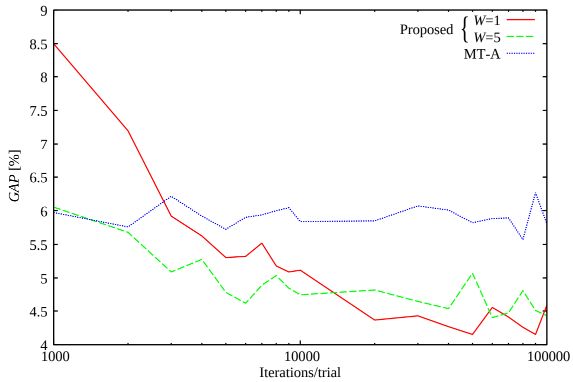

Figure 1: GAP for various iterations/trial for Kra30a.

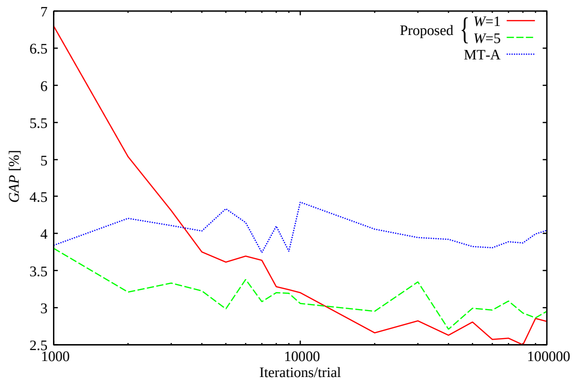

Figure 2: GAP for various iterations/trial for Kra30b.

7. Simulation results

Figures 1 to 4 show that the value of GAP for the proposed method decreases as the number of iterations per trial is increased. In Fig. 1, 2 and 4, the proposed method surpasses MT-A for a large number of iterations per trial. In Fig. 3, MT-A is still better than the proposed method when the number of iterations per trial is less than 100000.

However, we could expect that the proposed method will finally exceed MT-A if we further increase the number of iterations per trial, which would be feasible with a fast hardware system. In addition, we can confirm that the proposed method gives good performance without parameter tuning. From these results, we can conclude that the proposed method is an effective method for solving QAPs using an analog-digital hybrid parallel hardware system.

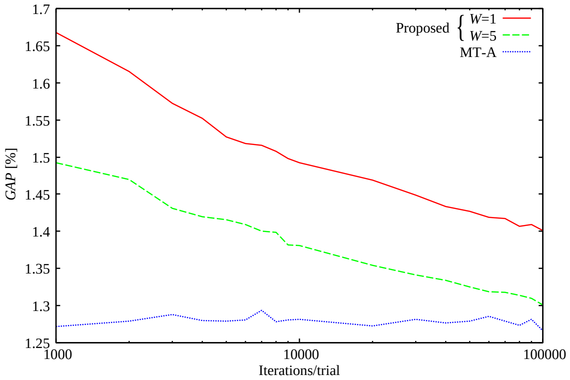

Figure 3: *GAP* for various iterations/trial for Lipa50a.

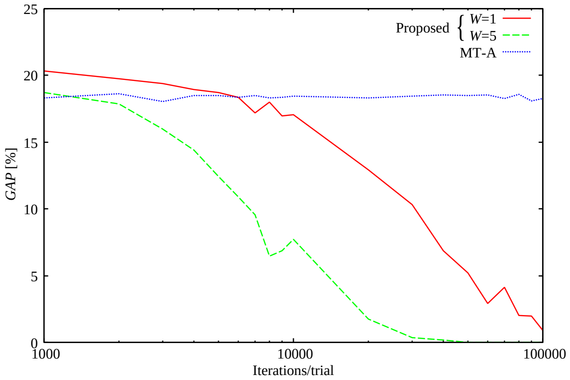

Figure 4: *GAP* for various iterations/trial for Lipa50b.

8. Conclusion

In this paper, we have simplified the algorithm for synchronous exponential chaotic tabu search for use with an analog-digital hybrid parallel hardware system. By numerical simulation, we confirmed that the proposed method exhibits good performance which is comparable with that of the conventional method. We have also proposed a way of incorporating features of the hardware into the algorithm to speed up the implementation. In the future, we plan to build a prototype analog-digital hybrid hardware system to implement the proposed synchronous exponential chaotic tabu search.

Acknowledgment

This research was supported by Aihara Innovative Mathematical Modelling Project, the Japan Society for the Promotion of Science (JSPS) through the "Funding Program for World-Leading Innovative R&D on Science and Technology (FIRST Program)," initiated by the Council for Science and Technology Policy (CSTP).

References

[1] M. Hasegawa, T. Ikeguchi and K. Aihara, "A novel chaotic search for quadratic assignment problems," *European Journal of Operational Research*, vol. 139, pp. 543–556, 2002.

[2] N. Yokota, Y. Horio, K. Aihara and M. Hasegawa, "A modified synchronous exponential chaotic tabu search for quadratic assignment problems," *Tech. Rep. IEICE*, vol. NLP107, no. 561, pp. 49–54, 2008.

[3] A. Shirakuma, Y. Horio and K. Aihara, "An improved neuron selection technique in an exponential chaotic tabu search hardware system for large quadratic assignment problems," in *Proc. Int. Workshop on Nonlinear Circuit and Signal Processing,* CD-ROM 2009.

[4] T. Kawamura, Y. Horio and M. Hasegawa, "Mutual information analyses of chaotic neurodynamics driven by neuron selection methods in synchronous exponential chaotic tabu search for quadratic assignment problems," *ICONIP 2010*, Part I, LNCS 6443, pp. 49–57, 2010.

[5] Akihito Toyoda, Yoshihiko Horio and Kazuyuki Aihara, "An Improved Exponential Chaotic Tabu Search in Quadratic Assignment Problems for Parallel Hardware Implementation, " *Tech. Rep. IEICE*, vol. 112, no. 6, NLP2012-3, pp. 13–18, 2012.

[6] R. E. Burkard, S. E. Karisch and F. Rendl, "QAPLIB - A quadratic assignment problem library,"
http://www.opt.math.tu-graz.ac.at/qaplib/