Control and Cybernetics

vol. **29** (2000) No. 3

Exponential and chaotic neurodynamical tabu searches for quadratic assignment problems

by

Mikio Hasegawa[1], Tohru Ikeguchi[2a] and Kazuyuki Aihara[3,4]

[1]Yokosuka Radio Communications Research Center
Communications Research Laboratory
Ministry of Posts and Telecommunications
3-4 Hikarino-oka, Yokosuka, Kanagawa 239-0847, Japan
E-mail:mikio@crl.go.jp

[2]Department of Information and Mathematical Science
Graduate school of Science and Technology
[a]Department of Information and Computer Sciences
Faculty of Engineering, Saitama University
255 Shimo-Ohkubo, Urawa, Saitama 338-8570, Japan
E-mail: tohru@nls.ics.saitama-u.ac.jp

[3]Department of Complexity Science and Engineering
Graduate School of Frontier Sciences
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan
E-mail: aihara@sat.t.u-tokyo.ac.jp
[4]CREST, Japan Science and Technology Corporation (JST)
4-1-8 Hon-cho, Kawaguchi, Saitama, 332-0012, Japan

Abstract: We propose a chaotic neurodynamical searching method for the Quadratic Assignment Problems (QAPs). First, we construct a neural network whose behavior is the same as that of the conventional tabu search. Using the dynamics of the tabu search neural network, we realize the exponential tabu search, whose tabu effect decreases exponentially with time, and we show the effectiveness of this type of exponential tabu search. Next, we extend this novel tabu search to a chaotic version. This chaotic method includes both effects of the chaotic dynamical search and the exponential tabu search, and exhibits better performance than the conventional and exponential tabu searches. Last, we propose an automatic parameter tuning method and show that the proposed method exhibits high performance even on large QAPs.

Keywords: neural networks, chaos, combinatorial optimization

1. Introduction

There are many combinatorial optimization problems in various fields. Although there are many effective methods for solving them, it takes a huge amount of computational time to obtain exactly optimum solutions for large problems. In particular, it is practically impossible to get globally optimum solutions to large NP-hard problems, for example, the Traveling Salesman Problem (TSP), the Quadratic Assignment Problem (QAP), and so on. Therefore, it is very important to develop heuristic methods for finding very good near optimum solutions in reasonable time.

In those heuristic algorithms, existence of undesirable local minima, at which gradient search algorithms stop, can be a very serious problem. In order to avoid stopping of search, various diversification algorithms have been proposed. The simplest diversification approach is based on a stochastic escape from local minima. For example, simulated annealing is a very famous stochastic method (Kirkpatrick, Galatt and Vecchi, 1983, Wilhelm and Ward, 1987). Although this diversification mechanism is very simple, it does not have any memory effects.

As one of deterministic approaches for avoiding trapping at undesirable local minima, tabu search has been developed and successfully applied to various combinatorial optimization problems (Glover, Taillard and de Werra, 1993, Glover and Laguna, 1997). In the case of tabu search, once the search dynamics gets trapped at undesirable local optimum, it escapes from there to other directions than those which have already been searched. This search mechanism may lead to better diversification than stochastic escaping, because the states which have already been visited become hard to be searched again.

Another diversification approach, applying chaotic neurodynamics (Aihara, Takabe and Toyoda, 1990, Aihara, 1990), has also been studied and its effectiveness has been shown (Nozawa, 1992, Yamada and Aihara, 1997, Chen and Aihara, 1995, Hasegawa, Ikeguchi, Matozaki and Aihara, 1995, Ishii and Satoh, 1997). The chaotic dynamical search has been considered effective because the chaotic dynamics searches solutions only along a fractal attractor whose Lebesgue measure is usually 0. However, since the conventional chaotic methods (Nozawa, 1992, Yamada and Aihara, 1997, Chen and Aihara, 1995, Hasegawa, Ikeguchi, Matozaki and Aihara, 1995, Ishii and Satoh, 1997) are based on the Hopfield–Tank (1985) neural network approach, it is hard to apply them to large size problems. This is because the number of mutual connections becomes huge in the case of large problems and heavy calculation is required. Moreover, satisfying the constraints can be another difficult problem in the search for good solutions. Then, those conventional chaotic methods often offer non-feasible solutions.

For more realistic applications of chaotic dynamics, we have already proposed a new approach (Hasegawa, Ikeguchi and Aihara, 1997), which combines chaotic dynamics and a heuristic method of the 2-opt algorithm that always produces

Tank neural network, it can be applied to much larger size of combinatorial optimization problems than the conventional chaotic neural network approach.

From the viewpoint of the "tabu effect" that forbids backward moves, there is a possibility that the chaotic neural network model (Aihara, Takabe and Toyoda, 1990, Aihara, 1990), which has been utilized in chaotic dynamical search approaches, may also have a similar effect to the tabu search. The chaotic neuron in the chaotic neural network model has the refractory effect, which is one of essential characteristics of real biological neurons; neurons become hard to fire just after previous firings. This refractory effect is similar to the tabu effect of the tabu search. The refractory effect inhibits firings of neurons which have recently fired. On the other hand, the tabu effect of the tabu search prohibits previously done moves. By using this inhibition of firings by the refractory effect, we can realize a kind of tabu effect. In other words, the tabu search can be realized with a neural network, in which neurons inhibited by refractory effects correspond to tabu moves. Furthermore, the chaotic neural network has also more complex dynamics than the tabu search, since such a "tabu effect" in the chaotic neural network takes an analogue value, that may lead to high performance for combinatorial optimization problems. Based on the above concept, we have already proposed (Hasegawa, Ikeguchi and Aihara, 2000) such a chaotic search including tabu search, and showed its effectiveness by mainly comparing it with the robust taboo search (Taillard, 1991) and the stochastic searches for real world problems.

In this paper, we discuss the effectiveness of the proposed algorithms with tabu effects. First, we realize the tabu search on a neural network with the refractory effect as a tabu effect. Inhibited states of neurons caused by refractory effects correspond to tabu moves. Using this tabu search neural network, we realize an exponential tabu search whose tabu effect decreases exponentially, and evaluate performance of this new tabu effect. Next, we extend this exponential tabu search to a chaotic version, by transforming the tabu search neural network to a chaotic neural network version (Aihara, Takabe and Toyoda, 1990, Aihara, 1990). We investigate the effectiveness of the exponential tabu search and the chaotic search, and the relation between those tabu effects and performances. Moreover, we also propose an automatic parameter tuning method of our chaotic neural network, for realizing better performance and easy applications of our method to various problems. We will apply these methods to the QAPs, since conventional tabu searches have been shown to be very effective for them, and compare performance of our method with the stochastic methods and tabu searches.

2. Quadratic Assignment Problems

The QAP is one of the NP-hard combinatorial optimization problems (Lawler, 1963, Pardalos, Rendl and Wolkowicz, 1994, Finke, Burkard and Rendl 1987).

permutation $\boldsymbol{p}$ which provides the minimum value of the objective function,

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

where a_{ij} and b_{ij} are the (i,j)th elements of A and B, $p(i)$ is the ith element of the permutation $\boldsymbol{p}$, and n is the size of the problem. It should be noted that the maximum size of the QAP whose exact optimum solutions are known is only 25 (Burkard, Karisch and Rendl, 2000).

There are many real applications formulated by the QAP, such as backboard wirings, facility layouts, scheduling problems, and so on (Lawler, 1963, Pardalos, Rendl and Wolkowicz, 1994, Finke, Burkard and Rendl 1987). Moreover, the QAP includes many other combinatorial optimization problems as special cases, for example the TSP.

3. Tabu Search for QAPs

In the conventional tabu searches for the QAP (Skorin-Kapov, 1990, Taillard, 1991, Taillard, 1995), a simple pairwise exchange procedure has been utilized for updating a state of the permutation $\boldsymbol{p}$. It is easy to decrease the objective function value $F(\boldsymbol{p})$ by exchanging two elements yielding a positive gain. However, this simple decreasing strategy makes algorithms stop at undesirable local minima. Then, in the case of tabu search, if there are no improving moves, the move that degrades the objective function the least is chosen. In order to avoid returning to the local optimum just visited, reverse moves are forbidden. This is realized by storing those moves in a data structure called tabu list. This list contains s elements which define forbidden moves, where s is the tabu list size. Once a move was stored in the tabu list, it is forbidden for s iterations, and becomes available s iterations later.

There are several types of tabu searches which are classified according to the construction of the tabu list. In our research, we construct a tabu list that prohibits moves which assign interchanged elements to the indices they had occupied in recent iterations. For example, if the element i is assigned to the jth index, $p(j)$ must be assigned to the $q(i)$th index for an actual exchange, where $p(j)$ is the element of the jth index and $q(i)$ is the index label at which i is located. In this case, both (i,j) and $(p(j),q(i))$ pairs are memorized in the tabu list, in the sequence of (element label, index label). In the case of updating the permutation, if either of two assignments caused by the interchange is in the tabu list, this move is tabu. This tabu list is almost the same as Taillard's (1991), but the tabu move in Taillard's tabu search requires that both of assignments, i to j and $p(j)$ to $q(i)$, be in the tabu list.

In the following, the above tabu search is implemented in a neural network.

4. Tabu search neural network and exponential tabu search

In the case of assigning an element i to some index j, there are $n \times n$ ways for selecting i and j, where n is the size of the problem. This means that $n \times n$ kinds of vectors could appear in the tabu list. In order to define all of these pairs, $n \times n$ neurons are prepared. Tabu effects are realized by these neurons. If a neuron has a large refractory effect, the corresponding move is tabu. Namely, this neural network behaves as follows: if the (i, j)th neuron fires, the element i is assigned to the jth index and $p(j)$ to $q(i)$, namely the elements i and $p(j)$ are exchanged. In this case, in order to realize the tabu search, both assignments, of i to j and $p(j)$ to $q(i)$, should be forbidden. Then, not only the (i, j)th neuron but also the $(p(j), q(i))$ th neuron are designed to rest for s iterations after the firing of the (i, j) th neuron, where s is the tabu list size. This tabu effect is realized by the refractory effect. For finding lower objective function values, the gain with a corresponding firing is applied to each neuron. Then, the tabu search can be realized by neural dynamics described by the following equations:

$$\xi_{ij}(t+1) = \beta\Delta_{ij}(t), \tag{2}$$

$$\gamma_{ij}(t+1) = -\alpha \sum_{d=0}^{s-1} k_r^d x_{p(j)q(i)}(t-d), \tag{3}$$

$$\zeta_{ij}(t+1) = -\alpha \sum_{d=0}^{s-1} k_r^d x_{ij}(t-d), \tag{4}$$

where β is the scaling parameter for the gain effect; k_r is the decay parameter of the tabu effect; α is the scaling parameter of the tabu effect; $\Delta_{ij}(t)$ is the gain of the objective function value and $\Delta_{ij}(t) = D_0(t) - D_{ij}(t)$; $D_0(t)$ is the present value of the objective function $F(\boldsymbol{p})$ at time t and $D_{ij}(t)$ is the value of $F(\boldsymbol{p}')$ which is the objective function value of $\boldsymbol{p}'$ that is made by exchanging elements i and $p(j)$ of $\boldsymbol{p}$ at time t which is produced by firing of this (i, j)th neuron; $x_{ij}(t)$ is the output of the (i, j)th neuron at time t; $\xi_{ij}(t), \gamma_{ij}(t)$ and $\zeta_{ij}(t)$ are internal states of the (i, j) th neuron at time t corresponding to the gain effect, the tabu effect of the assignment of $p(j)$ to $q(i)$, and that of i to j, respectively. If $\{\xi_{ij}(t+1) + \gamma_{ij}(t+1) + \zeta_{ij}(t+1)\}$ is the largest among all the neurons, the (i, j)th neuron fires and the element i is assigned to the jth index and $p(j)$ to $q(i)$, and $x_{ij}(t+1)$ and $x_{p(j)q(i)}(t+1)$ are set to 1 for memorizing both assignments. Outputs of all other neurons $x_{kl}(t+1)$ are set to 0. In fact, $\{\xi_{ij}(t+1) + \gamma_{ij}(t+1) + \zeta_{ij}(t+1)\}$ is equal to $\{\xi_{p(j)q(i)}(t+1) + \gamma_{p(j)q(i)}(t+1)$ $+ \zeta_{p(j)q(i)}(t+1)\}$, so there are two maximum firing neurons. However, firing of these two neurons means the same, because the outputs of both neurons are set to 1 and both assignments are done, if either neuron fires.

With this neural network, the same behavior as that of the conventional tabu search can be perfectly reproduced by setting $\alpha \to \infty$, $k_r = 1$ and the makes forbidden these assignments of i to j and $p(j)$ to $q(i)$ which have already been assigned in the last s iterations, because neurons corresponding to those assignments cannot fire by the tabu effect with infinite strength ($\alpha \to \infty$) in Eqs. (3) and (4).

On the other hand, it is also possible to decrease the tabu effect exponentially with $0 < k_r < 1$ and finite α, this situation being called exponential tabu search. In this tabu search, we use $s = t$. Then, the dynamics of Eqs. (2)–(4) can be reduced as follows:

$$\xi_{ij}(t+1) = \beta \Delta_{ij}(t), \tag{5}$$

$$\gamma_{ij}(t+1) = k_r \zeta_{p(j)q(i)}(t) - \alpha x_{p(j)q(i)}(t), \tag{6}$$

$$\zeta_{ij}(t+1) = k_r \zeta_{ij}(t) - \alpha x_{ij}(t). \tag{7}$$

In Table 1, results of the exponential tabu search (EX-TS) realized by the above tabu search neural network model are compared with the conventional tabu searches; (i) the ordinary tabu search (TS), which fixes the tabu list size and (ii) the random tabu search (RA-TS), which dynamically changes the tabu list size. Results are shown by percentages of average gaps between obtained solutions and the best known solutions for each problem. Each run is cut at $100n$ iterations, where n is the problem size. Since it has already been reported that a good tabu list size is between $0.9n$ and $1.1n$, the tabu list size s was set at n for the ordinary tabu search. For the random tabu search, tabu list sizes are set between ±10%. Parameter values for the exponential tabu search are set as follows: $k_r = 0.99$, $\alpha = 1$ and $\beta = 5$. Differences of the objective function value $\Delta_{ij}(t)$ in Eq. (2) are normalized by $a_M b_M$, where $a_M = \max_{ij}\{a_{ij}\}$ and $b_M = \max_{ij}\{b_{ij}\}$. In order to find better solutions by these methods, the following aspiration function is introduced for all the above methods: if the move leads to the best solution which could never be found in previous iterations, the move is executed even if the move is tabu.

Problem	size	TS	RA-TS	EX-TS
tai20a	20	0.872	*0.794*	**0.730**
tai35a	35	1.309	*1.256*	**1.180**
tai50a	50	1.672	*1.585*	**1.442**
tai60a	60	*1.671*	1.675	**1.540**

Table 1. Results of the conventional ordinary tabu search (TS), the random tabu search whose tabu list size is randomly changed (RA-TS) and the exponential tabu search (EX-TS) which decreases the tabu effect exponentially. Each result is shown by the percentage of the gap between the average of obtained solutions and the best known solution of each problem. Each run is cut at $100n$ iterations, where n is the size of the problem. The best result and the second best result

From Table 1, it can be clearly seen that the exponential tabu effect is more effective than conventional tabu searches. In contrast to the conventional tabu searches which completely forbid moves memorized in the tabu list, this novel tabu search reduces the tabu effect gradually. Even though the random tabu search changes the size of the tabu list dynamically, it also completely forbids tabu moves, namely, it does not lead to gradual decrease. It can be seen from Table 1 that, although the random tabu search is better than the ordinary tabu search, the exponential tabu search is even better than the former. Then, gradual decrease of the memory effect (tabu effect) in the exponential tabu search may be effective for various combinatorial optimization problems.

5. Novel chaotic search including tabu effect

The tabu search neural network model (Eqs. (2)–(7)) has a similar architecture as the chaotic neural network model of Aihara, Takabe and Toyoda (1990). Both of them include refractoriness (tabu effect) with a temporal summation. However, there is a significant difference in the output function. The output of neurons $x_{ij}(t)$ of the tabu search neural network is defined exactly to be 0 or 1 due to detection of the maximum in all neurons. On the other hand, the chaotic neural network usually adopts an analog sigmoidal function. In the case of a single neuron model, chaotic behavior cannot be observed, if the output function is the step function whose outputs are only 0 or 1 (Aihara, Takabe and Toyoda, 1990, Aihara, 1990). Such an output function leads only to periodic dynamics or convergence to a fixed point. Although the tabu search neural network has a similar architecture to the chaotic neural network, the output of neurons $x_{ij}(t)$ does not take continuous values but only 0 or 1. Then, it is expected that the chaotic dynamics may be produced by introducing a continuous sigmoidal output function for the tabu search neural network. Here, the firing of a neuron is defined by the condition that $x_{ij}(t+1) > \frac{1}{2}$, where $x_{ij}(t+1) = f\{\xi_{ij}(t+1) + \eta_{ij}(t+1) + \gamma_{ij}(t+1) + \zeta_{ij}(t+1)\}$, $f(y) = 1/(1+e^{-y/\epsilon})$, and a similar role of detecting the maximum firing neuron is implemented by mutual connections with the internal state $\eta_{ij}(t)$ (Aihara, Takabe and Toyoda, 1990).

Then, the novel method, which includes both the tabu effect and the chaotic dynamics, is realized by the following equations:

$$\xi_{ij}(t+1) = \beta\Delta_{ij}(t), \tag{8}$$

$$\eta_{ij}(t+1) = -W \sum_{k=1}^{n} \sum_{l=1,\,(k\neq i \vee l\neq j)}^{n} x_{kl}(t) + W, \tag{9}$$

$$\gamma_{ij}(t+1) = -\alpha \sum^{s-1} k_r^d \{x_{p(j)q(i)}(t-d) + z_{p(j)q(i)}(t-d)\} + \theta, \tag{10}$$

$$\zeta_{ij}(t+1) = -\alpha \sum_{d=0}^{s-1} k_r^d \{x_{ij}(t-d) + z_{ij}(t-d)\} + \theta, \tag{11}$$

$$x_{ij}(t+1) = f\{\xi_{ij}(t+1) + \eta_{ij}(t+1) + \gamma_{ij}(t+1) + \zeta_{ij}(t+1)\}, \tag{12}$$

where W is the connection weight. If $x_{ij}(t+1) > \frac{1}{2}$, the (i,j)th neuron fires and the element i is assigned to the j th index, and $p(j)$ to $q(i)$, respectively. Because the tabu list consists of both assignments of (i,j) and $(p(j),q(i))$, $z_{p(j)q(i)}(t)$ should be prepared for memorizing the assignment of $(p(j),q(i))$ which does not correspond to the label of a firing neuron. In the case of updating the (i,j)th neuron, $z_{ij}(t+1)$ is reset to 0. For memorizing the assignment of $p(j)$ to $q(i)$ until the updating of the $(p(j),q(i))$th neuron, the output of the (i,j) th neuron $x_{ij}(t+1)$ is added to $z_{p(j)q(i)}(t+1)$, if the $(p(j),q(i))$th neuron is already updated on this iteration t, otherwise $x_{ij}(t+1)$ is added to $z_{p(j)q(i)}(t)$. Introduction of $z_{p(j)q(i)}(t)$ is essential, since the assignment of $p(j)$ to $q(i)$ will lead to its large accumulation at $z_{p(j)q(i)}(t)$, and the corresponding assignment can be avoided.

Moreover, there is a significant difference of the update rule with respect to the tabu search neural network (Eqs. (2)–(4)). This neural network for chaotic search should be asynchronously updated. The reason is that every firing involves an exchange of elements of the permutation $\boldsymbol{p}$ except for the case that a neuron corresponds to an assignment of i to $p(i)$, when we define the condition for firing by $x_{ij}(t+1) > \frac{1}{2}$. While more than one neuron has the chance to fire in a single iteration, more than one exchange of elements cannot be done simultaneously. Then, in this paper, asynchronous update is used for avoiding such situations. If we want to update this neural network synchronously, other definition of firing is required that produces only a single firing in each iteration, for avoiding simultaneous exchanges. For example, selection of the maximum activity as in the tabu search neural network in the previous section generates only a single firing in each iteration. However, by such firing rules, tabu effects cannot be correctly preserved in the above neural network. The tabu effect is realized by refractory effects which depress firings when the value of the output of neurons, $x_{ij}(t)$, becomes large in Eqs. (10) and (11). Even when the output of more than one neuron becomes large, firing of all of those neurons is depressed in the later iterations. Then, if we use the asynchronous update, every interchange corresponding to every firing can be done and tabu effects can work correctly. However, in the synchronous update, we have to select only one of those neurons with large outputs for an actual interchange. As a result, incorrect tabu effects would occur in other neurons except for the selected one for this exchange, since those neurons produce tabu effects even though moves corresponding to them have not been done. Moreover, there is also inconsistency with the case that no neurons have large outputs. Namely, no tabu effect may work, if there are no neurons which have large outputs. In Hasegawa, Ikeguchi and Aihara and discussed the effectiveness by comparing it with stochastic searches. The basic part of this method was not a tabu search and does not have a correct tabu effect. In this paper, we introduce the asynchronous update for correctly preserving tabu effects for the tabu search.

By fixing $s - 1 = t$, which corresponds to memorizing of whole previous outputs $x_{ij}(t)$ decaying with k_r, Eqs. (10) and (11) can be reduced as follows:

$$\gamma_{ij}(t+1) = k_r\zeta_{p(j)q(i)}(t) - \alpha\{x_{p(j)q(i)}(t) + z_{p(j)q(i)}(t)\} + R, \qquad (13)$$

$$\zeta_{ij}(t+1) = k_r\zeta_{ij}(t) - \alpha\{x_{ij}(t) + z_{ij}(t)\} + R, \qquad (14)$$

where $R = \theta(1 - k_r)$.

Here, Eqs. (13) and (14) have almost the same formulation as in the original chaotic neural network model which produces chaotic dynamics (Aihara, Takabe and Toyoda, 1990). Accordingly, a chaotic search may be realized by this neural network model. Then, this novel search includes not only the tabu search but also the chaotic fluctuation, which is considered to be effective for combinatorial optimization (Hasegawa, Ikeguchi and Aihara, 1997).

Next, we evaluate the performance of the proposed chaotic search (Eqs. (8)–(14)), which is a modification of the tabu search and an extension to the chaotic neural network version. In Table 2, the results of the proposed chaotic search (CS) are compared with the conventional ordinary tabu search (TS), the random tabu search (RA-TS) which randomly changes the tabu list size s, and the exponential tabu search (EX-TS) proposed in the previous section. These results show percentages of gaps between the averages of solutions obtained and the best known solutions. For fair comparison, we fix the number of exchanges of elements of the permutation $\boldsymbol{p}$. In these experiments, the number of exchanges is fixed at $100n$ for each run, where n is the problem size. The reason why we fix the number of exchanges is that computation of the objective function value is heavy on the QAPs. Even though we use a simplified method for this calculation (Taillard, 1995), it still consumes the most of computational time on each method. In this experiment, two types of tabu effects are introduced, shorter ones and longer ones. For the ordinary tabu search, the tabu list size is set at n or $20n$, for shorter or longer tabu effects, respectively. Although it has already been reported that a good tabu list size is between $0.9n$ and $1.1n$, a larger tabu list size is also introduced in order to evaluate the performance of the tabu search on problems which may have very deep local minima. For avoiding traps in deep local minima, such a long tabu list size is required. The tabu list size of the random tabu search is set $\pm10\%$ of n or $20n$. For the exponential tabu search, $k_r = 0.99$ and $k_r = 0.999$ are introduced, for the shorter and longer tabu effects, respectively. For the novel chaotic search (Eqs. (8)–(14)), we use the following parameter values: $\beta = 5$, $R = 0.02$, $W = 20$, $\epsilon = 0.01$, $k_r = 0.99$ and $\alpha = 1$.

From the results of tabu searches in Table 2 it can be seen that the longer tabu effects are effective for problems introduced here. Namely, problems in-searches of TS, RA-TS and EX-TS, we see that the exponential tabu search is still best. The result suggests that this type of decay of the tabu effect should be effective for combinatorial optimization. The novel chaotic search method is a further extension of this exponential tabu search to the chaotic dynamic version. From Table 2 we can see that this novel chaotic search was most often the best. The chaotic dynamic search has been shown to be effective for combinatorial optimization by several researchers (Nozawa, 1992; Yamada and Aihara, 1997; Chen and Aihara, 1995; Hasegawa, Ikeguchi, Matozaki and Aihara, 1995; Ishii and Satoh, 1997; Hasegawa, Ikeguchi and Aihara, 1997), and an effect of this chaotic search is also included in this novel search method. In fact, the proposed method exhibits high performance as shown in Table 2.

Problem	TS		RA-TS		EX-TS		CS
	$s = n$	$s = 20n$	$s = n \pm 10\%$	$s = 20n \pm 10\%$	$k_r = 0.99$	$k_r = 0.999$	
tai20b	15.574	4.961	15.643	5.184	8.936	*1.288*	**1.180**
tai35b	7.976	4.966	7.691	5.175	5.603	*3.195*	**2.931**
tai50b	5.990	3.150	6.576	3.054	3.620	**1.163**	*1.218*
tai60b	7.388	3.203	7.512	3.945	4.348	*1.723*	**0.927**

Table 2. Results of the conventional ordinary tabu search (TS), the random tabu search whose tabu list size is dynamically changed (RA-TS), the exponential tabu search (EX-TS) which decrease tabu effects exponentially, and the novel chaotic search (CS) with tabu effects. The values are percentages of gaps between the average of obtained solutions and the best known solution for each problem. Each run is cut at $100n$ exchanges of permutation, where n is the size of the problem.

In order to explain the relation among these searching mechanisms, Table 3 summarizes characteristics of these methods from the viewpoint of the tabu effect. The fifth column of this table shows actual dynamics of the internal state $\zeta(t)$ of a neuron in each method. In the case of the conventional ordinary tabu search, the tabu list size s is fixed and the infinite strength of tabu does not change for the period. In the random tabu search, the tabu list size is randomly changed, keeping the infinite strength of tabu. These two conventional tabu searches have only two values on tabu effects $\zeta_{ij}(t)$, 0 or $-\infty$. On the other hand, the exponential tabu search exponentially decreases the strength of this effect. Results in Tables 1 and 2 indicate that it is better to use this exponentially decreased tabu effect than the fixed and infinite strength tabu effect which has been utilized in conventional tabu searches. By extending the tabu search neural network to the chaotic version, chaotic fluctuation is also incorporated in the tabu effect. This chaotic dynamical method exhibits the best performance in

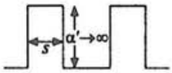
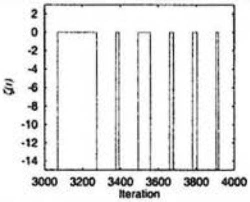
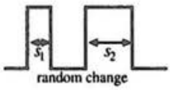
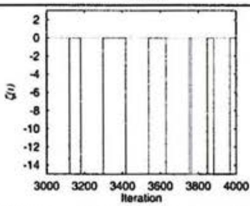
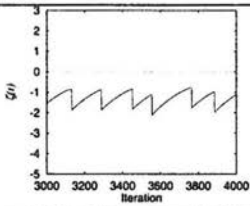
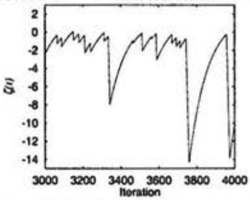
	Parameters	Output Function	Memory Effect (Tabu Effect)	Dynamics of $\zeta(t)$
Ordinary Tabu Search	$k_r = 1$ $\alpha \to \infty$ s: fixed			
Random Tabu Search	$k_r = 1$ $\alpha \to \infty$ s: random	Detecting Maximum Firing		
Exponential Tabu Search	$0 < k_r < 1$ α: finite $s - 1 = t$			
Chaotic Neural Networks	$0 < k_r < 1$ α: finite $s - 1 = t$			

Table 3. Relationships between the former tabu searches and the novel chaotic dynamical search using chaotic neural networks.

6. Controlling and annealing

Although the effectiveness of chaotic search for combinatorial optimization problems has been shown by many experimental results (Nozawa, 1992; Yamada and Aihara, 1997; Chen and Aihara, 1995; Hasegawa, Ikeguchi, Matozaki and Aihara, 1995; Ishii and Satoh, 1997; Hasegawa, Ikeguchi and Aihara, 1997), the performance of these methods depends on parameter values of the chaotic neural networks. If we could select appropriate values of the parameters for efficient searches, higher performance might be realized as shown in previous studies. However, when we fail to find such suitable parameter values, these methods on problems. Therefore we have to find, or at least try to set, the best parameter values manually, for each different problem, but this process is not easy when we want to solve many different problems.

In order to overcome the above disadvantage of the proposed method, we propose a simple automatic parameter tuning method for our chaotic neural network. For robust applications to various problems, we have to consider differences of the influences between problems, that is, the distribution of gain input $\Delta_{ij}(t)$ in Eq. (8). Then, we consider controlling of the externally applied input term $\xi_{ij}(t)$ which includes the gain effect $\Delta_{ij}(t)$.

In this paper, the mean and the variance of the input $\xi_{ij}(\mathrm{t})$ are controlled to be the same even if various different problems are applied, which have different distribution for this input. Then, the dynamics of $\xi_{ij}(t)$ is redefined by introducing two variables, $F_r(t)$ and $\beta(t)$, for controlling the mean and the variance of the input $\xi_{ij}(t)$ as follows:

$$\xi_{ij}(t+1) = \beta(t)(\Delta_{ij}(t) - F_r(t)). \tag{15}$$

We also control the connection weight W in Eq. (9) for obtaining an appropriate connection effect, with introduction of a new variable $W(t)$. $W(t)$ is controlled according to the variance of the input $\xi_{ij}(t)$. Then, controlled variables in our algorithm are these three, $F_r(t)$, $\beta(t)$, and $W(t)$, and other parameter values are fixed.

First, in order to reduce the differences of the mean of the inputs, the variable $F_r(t)$ is tuned according to this mean value of $\Delta_{ij}(t)$, which is defined as $\bar{\Delta}(t)$. If the number of firing neurons in a single iteration, $N_f(t)$, is small (i.e. $N_f(t) < n/8$), then $F_r(t)$ is reduced with $F_r(t+1) = F_r(t)+C(\bar{\Delta}(t)-F_r(t))$, for increasing the firing rate, where C is the controlling rate and fixed at 0.01 in this paper. Otherwise, $F_r(t+1) = (1-C)F_r(t)$. In fact, for making the mean of the inputs be the same for every problem, we should control this mean to be 0, by tuning $F_r(t)$ to be $\bar{\Delta}(t)$. However, this is useless for selecting the positive gain moves, because many of negative gain moves become positive inputs with Eq. (15) (in most cases, $F_r(t) < 0$), which would make it impossible for the neural network to select positive gain moves. Then, $F_r(t)$ is reduced to 0 when the network firing rate is high enough for searching.

Second, for controlling the variance of the inputs $\xi_{ij}(t)$, the variable $\beta(t)$ is automatically tuned according to a standard range $S_R(t)$. In this paper, we use the standard deviation of $\Delta_{ij}(t)$ for this standard range $S_R(t)$. Then, $\beta(t)$ is controlled for realizing that standard deviation of actual input values, which is $S_R\beta(t)$, becomes B, where B is defined for deciding the strength of the effect of gradient dynamics. Namely, this control keeps the standard deviation of inputs to be B even if the proposed algorithm is applied to various problems. Then, control of $\beta(t)$ is done as follows: $\beta(t+1) = \beta(t) + C(B/S_R - \beta(t))$.

Finally, for making appropriate connection effects and keeping effective firing rate, $W(t)$ is controlled. Since the gain effect and chaotic fluctuation are very the gain effect or chaotic fluctuation. On the other hand, the connection effects are still required for keeping appropriate firing rates of the neural network. Although control of firing rates is included in tuning of $F_r(t)$ as described above, it is better to make less change of the variable $F_r(t)$ for selecting moves with better gains, and the connection effect should become the main effect for making an appropriate firing rate of the neural network. Then, $W(t)$ is controlled according to the variance of the input which is $S_R\beta(t)$. Here, a parameter W_B is defined for the basic strength of the connection effect. Then, $W(t)$ is controlled to be $W_B S_R\beta(t)$. Namely, $W(t+1) = W(t) + C(W_B S_R\beta(t) - W(t))$.

In order to realize even better performance, we also introduce chaotic simulated annealing. The strength of the gradient dynamics is gradually strengthened, compared with chaotic fluctuation. In order to do this, we gradually increase the parameter B which is the target value of the standard deviation of the input, and make large the strength of the effect of the gain, $\beta(t)$. It should be noted that this annealing method is also good for exploring parameter values of B. Although we use the parameter tuning method, it does not always work perfectly. In the above method, the standard deviation of the input is controlled to be the same for various problems, by tuning the parameter $\beta(t)$ which is a weight of the input. However, since the distributions of gain values $\Delta_{ij}(t)$ differ significantly for various problems, adjusting only the standard deviation is not enough for each problem. Since parameter values change through the better performance ranges by this annealing, finding of better solutions becomes more easily.

Results of the novel chaotic search with above improvements are shown in Table 4. In these experiments, the novel method is applied to larger problems. Although only a single parameter set was enough for the experiments in Table 2, it becomes difficult to keep high performance on larger problems, since the best parameter set is different from problem to problem. However, manual parameter finding is quite difficult when we want to solve various problems, because the chaotic neural network method proposed in this paper has many

	size	TS	RA-TS	EX-TS	CSPT
tai60b	60	2.995	3.162	*1.817*	**1.469**
tai64c	64	0.139	0.264	*0.0965*	**0.0275**
tai80b	80	3.251	4.208	*1.700*	**1.343**
tai100b	100	3.167	3.269	*2.349*	**1.362**
tai150b	150	1.932	1.929	*1.495*	**1.365**
tai256c	256	0.411	0.444	*0.331*	**0.299**

Table 4. Results of the conventional ordinary tabu search(TS), the random tabu search whose tabu list size dynamically changes (RA-TS), the exponential tabu search (EX-TS) and the chaotic dynamical search with the automatic parameter tuning method (CSPT). Each run is cut at $100n$ exchanges of permutation,

parameters. Then, we apply the automatic parameter tuning method for larger problems. From Table 4, the effectiveness of the novel chaotic search can be seen. The second best was the exponential tabu search which also exhibits high performance. Namely, gradual decay of tabu effects, which is included in the exponential tabu search and the chaotic search, may be effective for combinatorial optimization. Moreover, by extending such tabu effects to chaotic version, the best performance could be realized.

7. Conclusions

In this paper, we propose a novel search method for solving QAP. Since our novel method is constructed by transforming the tabu search to a chaotic version, it has both effects of the chaotic search and the tabu search. The tabu effect utilized in the novel search decreases exponentially with time. First, we show the effectiveness of the exponential tabu search which utilizes such tabu effects. This was realized on a neural network model which also includes algorithms of the conventional tabu search. Then, we extended it to a chaotic version which has even better performance. Moreover, we also propose an automatic parameter tuning method for our novel search algorithm. By this method, the novel chaotic search can be easily applied to various problems without laborious manual parameter setting.

This type of search is also applicable to other combinatorial optimization problems. We have already shown that the novel chaotic search is also effective for very large TSPs (Hasegawa, Ikeguchi and Aihara, 1998). Since the proposed chaotic search uses asynchronous update, it is more suitable for problems for which a gain of the objective function value can be easily calculated. Namely, for this novel chaotic searching method, the TSPs and other problems whose gaps can be easily calculated, are more suitable than the QAP. Even though the QAP is not specially fit for this novel algorithm, the method exhibits high performance because its searching ability is high enough.

The proposed method has the tabu effects different from those of the conventional tabu searches. The method utilizes the tabu effects which are reduced exponentially. Our results show that this type of memory effect may be effective for combinatorial optimization problems. Moreover, the novel chaotic search also includes chaotic dynamics which has been shown to be effective for combinatorial optimization. Since the proposed chaotic search includes both effects of the chaotic dynamical search and the tabu search, high performance can be realized.

Acknowledgement. The authors would like to thank Professor Kohji Itoh (Science University of Tokyo) for valuable comments and encouragements. This research was partially supported by the grant on Research for the Future Pro-

References

AIHARA, K. (1990) Chaotic Neural Networks. In: H. Kawakami, ed., *Bifurcation Phenomena in Nonlinear Systems and Theory of Dynamical Systems.* World Scientific, 143–161.

AIHARA, K., TAKABE, T. and TOYODA, M. (1990) Chaotic Neural Networks. *Physics Letters A*, **144**, 333–340.

BURKARD, R.E., KARISCH, S.E. and RENDL, F. (2000) QAPLIB — A Quadratic Assignment Problem Library. Available via world wide web at http://www.imm.dtu.dk/~sk/qaplib/.

CHEN, L. and AIHARA, K. (1995) Chaotic Simulated Annealing by a Neural Network Model with Transient Chaos. *Neural Networks*, **8**, 915–930.

FINKE, G., BURKARD, S.C. and RENDL, F. (1987) Quadratic assignment problems. *Annals Discrete Math.*, **31**, 61–82.

GLOVER, F. and LAGUNA, M. (1997) *Tabu Search.* Kluwer Academic Publishers.

GLOVER, F., TAILLARD, E. and DE WERRA, D. (1993) A User's Guide to Tabu Search. *Annals of Operations Research*, **41**, 3–28.

HASEGAWA, M., IKEGUCHI, T. and AIHARA, K. (1997) Combination of Chaotic Neurodynamics with the 2-opt algorithm to solve Traveling Salesman Problems. *Physical Review Letters*, **79**, 2344–2347.

HASEGAWA, M., IKEGUCHI, T. and AIHARA, K. (1997) A New Modern Heuristic Approach using Chaotic Dynamics for Quadratic Assignment Problems. *Tech. Rept. of IEICE*, **97**, 52, 53–60.

HASEGAWA, M., IKEGUCHI, T. and AIHARA, K. (1998) A Novel Approach for solving Large Scale Traveling Salesman Problems by Chaotic Neural Networks. *Proc. of International Symposium on Nonlinear Theory and its Applications*, **2**, 571–574.

HASEGAWA, M., IKEGUCHI, T. and AIHARA, K. (2000) A Novel Chaotic Search for Quadratic Assignment Problems. Submitted to *EJOR.*

HASEGAWA, M., IKEGUCHI, T., MATOZAKI, T. and AIHARA, K. (1995) Solving Combinatorial Optimization Problems by Nonlinear Neural Dynamics. *Proc. of International Conference on Neural Networks, Perth*, **6**, 3140–3145.

HOPFIELD, J.J. and TANK, D.W. (1985) Neural Computation of Decisions in Optimization Problems. *Biol. Cybern.*, **52**, 144–152.

ISHII, S. and SATOH, M. Chaotic Potts Spin Model for Combinatorial Optimization Problems. *Neural Networks* **10**, 941–963.

KIRKPATRICK, S., GALATT, C. and VECCHI, M. (1983) Optimization by Simulated Annealing. *Science*, **220**, 671–680.

LAWLER, E. (1963) The quadratic assignment problem. *Management Science*, **9**, 586–599.

NOZAWA, H. (1992) A neural network model as a globally coupled map and applications based on chaos. *Chaos*, **2**, 377–386.

PARDALOS, P.M., RENDL, F. and WOLKOWICZ, H. (1994) The quadratic as-

H. Wolkowicz, eds., *Quadratic assignment and related problems. DIMACS*, **16**, 1–42.

SKORIN-KAPOV, J. (1990) Tabu search applied to the quadratic assignment problem. *ORSA Journal on Computing*, **2**, 33–45.

TAILLARD, E.D. (1991) Robust taboo search for the quadratic assignment problem. *Parallel Computing*, **17**, 443–455.

TAILLARD, E.D. (1995) Comparison of Iterative Searches. *Location Science*, **3**, 87–105.

WILHELM, M. and WARD, T. (1987) Solving Quadratic Assignment Problems by Simulated Annealing. *IIE Transactions*, **19**, 107–119.

YAMADA, T. and AIHARA, K. (1997) Nonlinear Neurodynamics and Combinatorial Optimization in Chaotic Neural Networks. *Journal of Intelligent and Fuzzy System*, **5**, 53–68.