2007 International Symposium on Nonlinear Theory and its Applications
NOLTA'07, Vancouver, Canada, September 16-19, 2007

A new parameter adjustment approach for solving vehicle routing problem with chaotic neurodynamics

Takashi Hoshino, Takayuki Kimura and Tohru Ikeguchi

Graduate School of Science and Engineering Saitama University
255 Shimo-ohkubo, Sakura-ku, Saitama, Saitama, 338-8570 JAPAN
Email: { hoshino, kimura, tohru } @nls.ics.saitama-u.ac.jp

Abstract—We have already proposed a solving method with two simple local searches driven by chaotic neurons for vehicle routing problem (VRP). The chaotic neuron qualitatively realizes refractoriness that makes it possible to memorize past searching history with an exponential decay. Therefore, the proposed method can escape from local minima effectively. We confirmed that the proposed method shows good performance for the VRP with time windows. However, in order to obtain good performance, it is inevitable to adjust an optimal parameter set. In this paper, we propose a new parameter adjusting approach for the proposed chaotic searching method. Using this approach, the performance of the proposed method is much improved without fine adjustment of parameters of the chaotic neurons.

1. Introduction

Creating an efficient delivery plan is a necessary issue in transportation systems, such as home-delivery service, transport of merchandise, school bus routing, and so on. To resolve the issue, the vehicle routing problem (VRP) is widely studied[1, 2], because the VRP shares fundamental property for various transportation systems. VRP is one of the NP-hard combinatorial optimization problems[3]. Then, it is very difficult to obtain an optimum solution in a realistic time. Thus, to develop an approximate method is a very important problem to obtain a good solution in reasonable time.

As for solving the combinatorial optimization problems such as traveling salesman problems (TSPs) or quadratic assignment problems (QAPs), a heuristic search with chaotic neurodynamics (chaotic search) is very effective[4, 5, 6, 7]. From this point of view, we have already proposed a chaotic search for VRP[1]. In the search[1], the CROSS-exchange[8] are introduced as a local search. The CROSS-exchange operates to exchange and insert customers. Although the proposed method[1] shows better performance than the tabu search[9], it take a relatively long time to obtain solutions because of complexity of the CROSS-exchange.

For the above reasons, we have already proposed a new simpler method with chaotic neurodynamics[10, 11]. In the proposed method, two simple local searches for exchange and insertion are driven by chaotic neurons. Using the simple local searches, the complexity is much reduced. In addition, the new proposed method can obtain a good solution faster than the method with the CROSS-exchange[1]. Although the performance is much improved, there still remains in these method[1, 10, 11]. The chaotic search [1, 4, 5, 6, 7, 10, 11] has many parameters, and it is necessary to find proper parameters for obtaining good results. Then, in this paper, we propose a new parameter adjusting approach to improve the proposed method[1, 10, 11]. Using the parameter adjusting approach, the performance of the proposed method becomes good without fine adjustment of the parameters.

2. Vehicle routing problem

VRP consists of a depot, vehicles and customers. The depot is an arrival and departure point of the vehicles. Each vehicle has a weight limit, and visits the customers to satisfy their demands. The customers are visited only once by one vehicle. In this paper, we treat VRP with time windows (VRPTW). In VRPTW, each customer has its own time window, and the vehicles have to visit the customers within the time window.

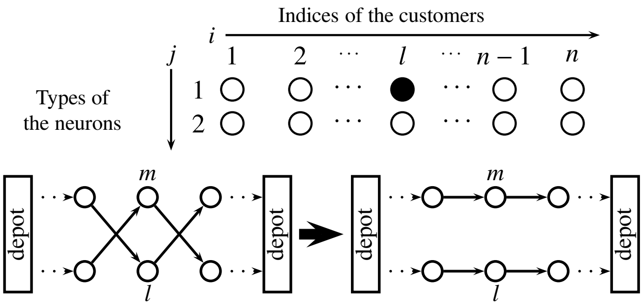
The object of VRP is to minimize the number of the vehicles, the total travel distance, and the total travel time. Generally, a primary object of VRP is to minimize of the number of the vehicles. Hence, we use the following objective function:

$$g(S) = \sum_{l=1}^{m} D_l + \gamma \times m, \qquad (1)$$

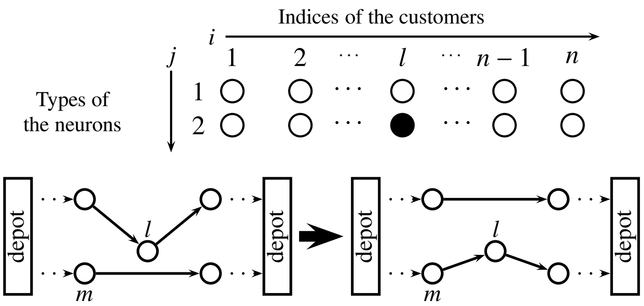
where S is a solution (all the tours of the vehicles), m is the number of vehicles, D_l is a total travel distance of the l-th vehicle, and γ is a scaling parameter. Because the first priority for VRP is to reduce the number of the vehicles in this paper, we set γ large.

3. Chaotic search with simple two local searches

We have already proposed a basic chaotic search for VRPTWs[1]. In the method[1], we used the CROSS-exchange[8] which operates to exchange and insert customers. Although the method is effective for VRPTWs,



(a) If a neuron in the first row ($j = 1$) fires, a customer is exchanged.



(b) If a neuron in the second row ($j = 2$) fires, a customer is inserted.

Figure 1: In this example, (a) the $(l, 1)$-th neuron fires, then the customer l is exchanged by the customer m whose gain is the maximum (Eq.(1)). (b) If the $(l, 2)$-th neuron fires, then the customer l is inserted into the position next to the customer m whose gain is the maximum (Eq.(1)).

it takes a long time to get a solution because of complexity of the CROSS-exchange. To reduce the complexity, we have proposed a new chaotic search [10, 11]. In the proposed method[10, 11], we use two simple local searches. The first one is to exchange the customer for another one, and the second one is to insert the customer into another place. Separating the fundamental functions of the CROSS-exchange, we could reduce the complexity of the CROSS-exchange.

In the proposed method[10, 11], chaotic neurons drive the two local searches: That is, if a neuron fires, a local search corresponding the firing neuron is performed. To realize this method, we use $2n$ chaotic neurons for an n-customer problem. Each neuron corresponds to each customer. Figure 1 shows how to code the firing of neurons for the local searches.

In the proposed method[10, 11], each neuron has a gain effect, a refractory effect and a mutual connection effect. These effects of the (i, j)-th neuron are defined as follows:

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

$$\zeta_{ij}(t+1) = -\alpha \sum_{d=0}^{t} k_r^d x_{ij}(t-d) + \theta, \tag{3}$$

$$\eta_{ij}(t+1) = -W \sum_{p=1}^{n} \sum_{q=1}^{2} x_{pq}(t) + W, \tag{4}$$

where $\xi_{ij}(t)$, $\zeta_{ij}(t)$ and $\eta_{ij}(t)$ represent the gain effect, the refractory effect, and the mutual connection effect, respectively. Then, an output of the (i, j)-th neuron is defined as follows:

$$x_{ij}(t+1) = f\{\xi_{ij}(t+1) + \zeta_{ij}(t+1) + \eta_{ij}(t+1)\}, \tag{5}$$

where $f(y) = 1/(1 + e^{-y/\epsilon})$ called sigmoid function and ϵ is a small positive parameter. If $x_{ij}(t) > 1/2$, the (i, j)-th neuron fires at the time t and the local search to which the neuron corresponds is performed.

In Eq.(2), β is a positive scaling parameter of the gain effect, and Δ_{ijm} is a gain value of the objective function (Eq.(1)) if the local searches are performed. $\Delta_{ijm} = g(S_B) - g(S_A)$, where S_B and S_A are solutions before and after the local searches are performed respectively. Here, m indicates a customer to be exchanged or inserted into their next order, and m is so selected that Δ_{ijm} takes the maximum gain. By the gain effect, the neuron corresponding a good operation become easy to fire.

In Eq.(3), α is a positive scaling parameter, $k_r (0 < k_r < 1)$ is a decay factor, and θ is a threshold value. Then, the refractory effect inhibits the firing of a neuron which has just been fired, which realizes an memory effect with a exponential decay. The strength of the refractory effect gradually decays depending on the value of k_r.

In Eq.(4), W is a positive scaling parameter. The mutual connection adjusts a firing ratio of all neurons. If many neurons fire, Eq.(4) becomes a small value, and in a reverse case, it becomes a large value.

In the proposed method[10, 11], the updating order of the neurons is deterministically defined so that the corresponding customer is clockwise for the depot. A single iteration of the proposed method is finished if all the neurons are updated, and the update is asynchronously conducted. Therefore, the multiple local searches are performed in a single iteration.

Moreover, to reduce the number of vehicles, the ejection chain[12] is executed at every iteration. If the number of routes can be reduced, the ejection chain is performed.

4. Computational results

To evaluate the performance of the proposed method, we solved 100-customer instances from Solomon's benchmark problems[13]. In these instances, there are six different types called R1, R2, C1, C2, RC1 and RC2. Here, R means a random allocation, C a clustered allocation, and RC their mixture. The time windows are narrow in the type 1 and wide in the type 2. A total number of the instances is 56. We compared the method with CROSS-exchange[1] and the method with two simple local searches (2TYPE) to the same initial solution produced by the Bräysy construction heuristic method[12].

In Eq.(1), we set γ to $1,000$, because reduction of the number of vehicles has the first priority. The parameters of the chaotic neurons in both methods are set as follows: $\beta = 0.04, \alpha = 0.5, k_r = 0.8, \theta = 0.4, W = 0.002$, and $\epsilon = 0.02$.

Table 1: Comparison of the two methods (2TYPE and CROSS).

method	R1	R2	C1	C2	RC1	RC2	Time
2TYPE[10, 11]	**12.33**	**3.09**	**10.00**	**3.00**	**12.00**	**3.38**	52min
	1223.30	970.88	891.30	589.86	1401.43	1223.37	
CROSS[1]	**12.55**	**3.07**	**10.00**	**3.00**	**12.06**	**3.39**	474min
	1225.89	985.25	854.07	596.51	1391.83	1178.10	

Because the neurons are randomly updated in the method with CROSS-exchange[1], we conducted 10 simulations, and averaged the results.

Table 1 shows computational results of the proposed method (2TYPE)[10, 11] and the previous method (CROSS)[1]. In Table 1, we show the total time for solving 56 instances. The ejection chain is introduced in the same way. The simulation is cut at $1,000$ iterations for both methods. The numbers indicated by boldfaces are average numbers of vehicles, and the numbers indicated by light faces are average total travel distances for each problem type.

In Table 1, both methods (2TYPE and CROSS) show similar performance, especially for the numbers of vehicles. As for the total travel distance, the 2TYPE has better performance than the CROSS for R1, R2 and C2. Although the 2TYPE shows similar results obtained by the CROSS, the proposed method takes shorter time than the previous method.

5. A parameter adjusting approach

In a chaotic search, many parameters exist. Then, in order to obtain a good result, it is strongly required to find an effective parameter set. If we use a wrong parameter set, neurons rarely fire. In such a case, a solution is hardly updated, because improvement of the solution is performed by the firing of the neurons. To avoid such undesirable situation, we propose a parameter adjusting approach for the chaotic search. In this approach, a value of θ in Eq.(3) is automatically adjusted using the firing rate of the neurons.

To realize this approach, we set a maximum firing rate $F_{\max}$ and a minimum firing rate $F_{\min}$. To keep the firing rate of the neurons between $F_{\max}$ and $F_{\min}$, we introduced the following two control methods: (i) the value of θ is decreased by 0.01, if the firing rate of the neurons is higher than $F_{\max}$, and (ii) the value of θ is increased by 0.01, if the firing rate of the neurons is less than $F_{\min}$. In addition, if the firing rate of the neurons is between $F_{\max}$ and $F_{\min}$, the value of θ is fixed. Using this approach, even if the neurons do not fire because of a wrong parameter set, the neurons will fire as increasing in the value of θ.

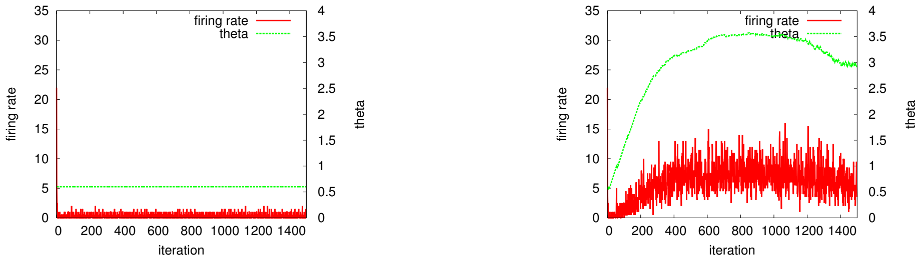
However, high firing rate of the neurons leads to a fatal situation that the neurons which have a small gain effect (Eq.(2)) frequently fire. It means that operations which deteriorate the solution are frequently performed. To avoid such a situation, we gradually decrease $F_{\max}$ to $F_{\min}$ by 0.01 at every iteration. By decreasing $F_{\max}$, the firing rate of the neurons gradually decreases. Then, we expect that the proposed adjustment approach has a similar effect as a simulated annealing.

To evaluate the parameter adjusting approach, we conducted computational simulation for wrong parameter sets and good parameter sets. In these simulations, we solved the Solomon's benchmark problems whose initial solutions are produced by the Bräysy construction method, and set γ to $1,000$ in Eq.(1) as well as Section 4. The simulations are cut at $1,500$ iterations. In the parameter adjusting approach, we set $F_{\max} = 20$, $F_{\min} = 5$.

First, as a wrong parameter set, we set the parameters of Eqs.(2)–(4) as follows: $\beta = 0.5, \alpha = 0.5, k_r = 0.94, \theta = 0.6, W = 0$, and $\epsilon = 0.002$. Figure 2 shows the firing rate of the neurons and the value of θ for R101, and Table 2 shows the results for all the problems. In Fig.2(a), the firing rate of the neurons is low. On the other hand, in Fig.2(b), the firing rate of the neurons rises as the value of θ increases. In addition, from the results of Table 2, solutions for all problems with the adjustment are better than the ones without an adjustment. As a result, we confirmed that the solution with the adjustment is better than the one without the adjustment.

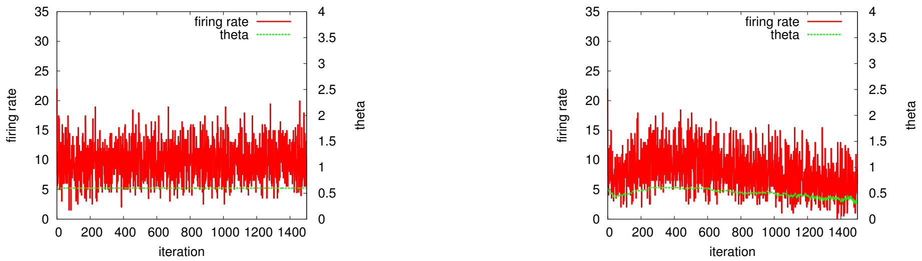
Next, as an effective parameter set, we set the parameters of Eqs.(2)–(4) as follows: $\beta = 0.02, \alpha = 0.2, k_r = 0.94, \theta = 0.6, W = 0$, and $\epsilon = 0.002$. Using the effective parameter set, as well as the case of the wrong parameter set, we evaluated the effectiveness of the parameter adjusting approach. Figure 3 and Table 3 show the results. In Fig.3(a), the firing rate of the neurons is kept between about 5 percent and 20 percent, and the result is better than that shown in Fig.2(b). In Fig.3(b), the firing rate gradually decreases by the parameter adjusting approach. As a result, a solution for R101 with the adjustment is better than the one without any adjustment. Moreover, in Table 3, solutions with the adjustment are better than solutions without the adjustment except RC2.

Although the performance with the adjustment is almost the same as the one without adjustment in the case of the effective parameter set, the parameter adjusting approach is effective for wrong parameter sets. Thus, using the parameter adjusting approach, we can obtain good performance even if we set the values of the parameters roughly. Simulation time is almost the same if we use the parameter adjusting approach. In this simulation, difference of the simulation time between the chaotic search with adjusting approach and without one was about one minute.



(a) R101 without an adjustment (Solution: 19 / 1717.88) (b) R101 with the adjustment (Solution: 19 / 1665.34)
(Solution: the number of vehicles / total travel distances)

Figure 2: Comparison of the firing rate of the neurons and the value of θ without a parameter adjustment (a), and with the parameter adjustment (b) for R101 using a wrong parameter set.



(a) R101 without an adjustment (Solution: 19 / 1664.50) (b) R101 with the adjustment (Solution: 19 / 1659.42)

Figure 3: Comparison of the firing rate of the neurons and the value of θ without a parameter adjustment (a), and with the parameter adjustment (b) for R101 using an effective parameter set.

Table 2: Results of the case without and with adjustments for all the problems in a wrong parameter set.

adjustment	R1	R2	C1	C2	RC1	RC2
without	**12.83**	**3.18**	**10.22**	**3.00**	**13.00**	**3.62**
	1285.21	1104.92	1010.93	602.65	1525.10	1318.37
with	**12.42**	**3.09**	**10.00**	**3.00**	**12.38**	**3.38**
	1219.08	1002.67	873.21	589.93	1426.27	1238.50

Table 3: Results of the case without and with adjustments for all the problems in a good parameter set.

adjustment	R1	R2	C1	C2	RC1	RC2
without	**12.50**	**3.09**	**10.00**	**3.00**	**12.12**	**3.38**
	1225.40	958.13	849.37	592.63	1380.86	1168.67
with	**12.50**	**3.09**	**10.00**	**3.00**	**12.00**	**3.38**
	1206.07	950.22	834.37	590.25	1369.12	1186.27

6. Conclusion

To solve VRPTWs, we have already proposed a method using two types of local searches driven by chaotic neurons. From the computational simulations, the method shows similar performance in reasonable time comparing with the method using the CROSS exchange[1]. To improve the performance, we proposed a new parameter adjustment approach for the chaotic search[1, 10, 11]. The computational simulations show that the parameter adjustment approach is very effective when we set a wrong parameter set. On the other hand, if the parameter set is good, the performance with the adjustment is almost same as the one without adjustment. However, as shown in Section 5, the performance for many problems can be improved using the parameter adjustment approach. Thus, it is an important future work to analyze the effectiveness of the parameter adjusting approach and propose a better parameter adjusting approach for improvement the performance of the chaotic search.

The research of TI is partially supported by Grant-in-Aids for Scientific Research (B) (No.16300072) and (C) (No.17500136) from JSPS, and a research grant from The Mazda Foundation.

References

[1] T. Hoshino et al., *IEICE Trans.*, **J91-A**(5): 431–441, 2007, in Japanese.

[2] O. Bräysy et al., *Internal Report STF42 A01022*, 2001.

[3] J. K. Lenstra et al., *Networks*, **11**: 221–227, 1981.

[4] M. Hasegawa et al., *Phys. Rev. Lett.*, **79**: 2344–2347, 1997.

[5] M. Hasegawa et al., *Neural Networks*, **15**: 271–283, 2002.

[6] M. Hasegawa et al., *Control and Cybernetics*, **29**: 773–788, 2000.

[7] M. Hasegawa et al., *EJOR*, **139**: 543–556, Jun. 2002.

[8] E. Taillard et al., *Transportation Science*, **31**: 170–186, 1997.

[9] F. Glover, *ORSA J. on Comp.*, **1**: 190–206, 1989.

[10] T. Hoshino et al., *Tech. Rep. IEICE*, **106**(452): 23–28, 2007.

[11] T. Hoshino et al., to appear in *Proc. of the 7th Metaheuristics Int. Conf.*, 2007.

[12] O. Bräysy, *INFORMS J. on Comp.*, **15**(4): 347–368, 2003.

[13] http://web.cba.neu.edu/~msolomon/problems.htm.