2010 International Symposium on Nonlinear Theory and its Applications
NOLTA2010, Krakow, Poland, September 5-8, 2010

A Method for Solving Very Large Scale TSPs by Chaotic Dynamics

Tohru Ikeguchi†, Shun Motohashi†, and Takafumi Matsuura†‡

†Department of Information and Computer Sciences, Saitama University
255 Shimo-ohkubo, Sakura-ku, Saitama 338-8570, Japan
‡Department of Management Science, Tokyo University of Science
1-3 Kagurazaka, Shinjuku-ku, Tokyo 162-8601, Japan
Email: 1tohru@mail.saitama-u.ac.jp, 2matsuura@ms.kagu.tus.ac.jp

Abstract—To solve traveling salesman problems (TSPs), we have already proposed two chaotic search methods, the Lin-Kernighan (LK) algorithm controlled by the chaotic dynamics (CS-LK) and the stem-and-cycle (S&C) ejection chain method controlled by the chaotic dynamics (CS-SC). The basic concept of the methods is that chaotic dynamics can effectively resolve a local minimum problem intrinsic to the heuristic algorithm. Although the CS-SC shows higher performance than the CS-LK, it is not so easy to apply the CS-SC to large scale instances because this method takes much calculation time to find good results. It is also important to develop an effective algorithm which finds good near-optimal solutions with less calculation time. In this paper, we propose a new method for solving very large scale TSPs with the order of 10^5 cities. In the method, we introduced a hybrid strategy by combining these two chaotic search methods. We used large scale instances form TSPLIB to evaluate the proposed method. We show that the proposed method exhibits very small gaps from the optimal solutions with less calculation time.

1. Introduction

We are often asked to solve various combinatorial optimization problems in our daily life; for example, scheduling, delivery planning, circuit design, computer wiring, and so on. Although these problems are ubiquitous and easy to describe, it is usually hard, if not impossible, to find their optimal solutions. These facts indicate that it is inevitable to design effective algorithms for solving combinatorial problems.

To develop an effective algorithm for the combinatorial optimization problems, the traveling salesman problem (TSP) is often used, because it is one of the most standard combinatorial optimization problems. For an N-city symmetric TSP, the number of all possible tours is $(N-1)!/2$. Thus, the number of tours exponentially diverges if the number of cities increases. It is widely acknowledged that the TSP belongs to a class of NP-hard. It means that it is almost impossible to obtain optimal solutions. Therefore, it is required to develop an effective approximate algorithm for finding near-optimal solutions in a reasonable time frame.

To discover approximate solutions, various local search algorithms have already been proposed, for example, the k-opt algorithm ($k = 1, 2, \ldots, N$), the Or-opt algorithm [1], the Lin-Kernighan (LK) algorithm [2], and the stem-and-cycle (S&C) ejection chain method [3, 4]. However, it is almost impossible to find optimal solutions only by the local search algorithms because of a local minimum problem. During the search, the state gets stuck at local minima.

To resolve the local minimum problem, various metaheuristic strategies have been proposed such as simulated annealing [5], genetic algorithm [6], tabu search [7], chaotic search [8], and so on. Among them, the chaotic search [8–12] shows good performance. In the chaotic search, to avoid local minima, execution of a local search algorithm is controlled by chaotic dynamics. In Refs. [8–12], to generate the chaotic dynamics, a chaotic neural network [13] is used. In the chaotic neural network, the basic element is a chaotic neuron proposed by Aihara et al. [13]. It can reproduce refractoriness, one of the important properties which real nerve cells have: When a neuron has just fired, the firing of this neuron is inhibited for a while by the refractoriness. In Refs. [8–12], execution of the local search algorithm is encoded by firings of the chaotic neuron. If the chaotic neuron fires, the corresponding local search algorithm is executed. Because the firing of the chaotic neuron is inhibited by the refractoriness, frequent firings of the chaotic neuron, or frequent execution of the local search is restricted. Thus, the chaotic search can escape from local minima efficiently.

We have already proposed two chaotic search methods [11, 14, 15]. In the first method, execution of the LK algorithm [2] is controlled by the chaotic dynamics. The LK algorithm [2] is one of the most famous variable depth search methods. As a result, the chaotic search method using the LK algorithm shows solving performance with less than 0.7% gaps from the optimal solution for instances with the order of 10^4 cities and can be applied to large scale instances with the order of 10^5 cities [11].

On the other hand, the S&C ejection chain method [3, 4] is also one of the most effective variable depth search methods. It is reported that the S&C ejection chain method leads to better solutions than the LK algorithm [4]. One of the reasons is that the S&C ejection chain method can explore more diversified solution space, because it introduces an

S&C structure, which is not a tour. Namely, the S&C ejection chain method can explore unfeasible solution space. However, the S&C ejection chain method also gets stuck at local minima because it is also a greedy algorithm.

Therefore, to resolve the local minimum problem in the S&C ejection chain method, we proposed chaotic search method which controls the S&C ejection chain method by the chaotic dynamics. Although the introduction of the S&C ejection chain method leads to higher performance, it is not so easy to apply the chaotic search method using the S&C ejection chain method to large scale instances, because this method takes much calculation time to find good results. From the viewpoint of application of approximate algorithms to real life problems, it is also important to develop an effective algorithm which finds good near-optimal solutions with less calculation time.

Although there are several strategies of how to reduce the calculation cost, we introduce the following strategy: we combine two chaotic search methods, namely the LK-algorithm-base chaotic search and the S&C-ejection-chain-method-base chaotic search. In the proposed method, both the LK algorithm and S&C ejection chain method are controlled by the chaotic dynamics.

Numerical experiments show that by combining the LK algorithm with the S&C ejection chain method, although calculation costs of the proposed method are less than the chaotic search method using the S&C ejection chain method, its solving performance is improved than the conventional method. The proposed method has high solving ability for very large scale instances such as 10^5 order.

2. Chaotic search method using both the LK algorithm and S&C ejection chain method

The Lin-Kernighan (LK) algorithm [2] which is one of the most effective local search algorithms for solving the traveling salesman problem (TSP). On the other hand, the stem-and-cycle (S&C) ejection chain method [3, 4] is also one of the most effective local search algorithms, and it shows higher performance than the LK algorithm. Thus, if we apply the chaotic dynamics to the S&C ejection chain method, we can expect that the performance of the chaotic search method is much improved [15]. However, if we only introduce the chaotic search method using the S&C ejection chain method, it takes large amount of computational time to obtain high performance for solving the large scale TSPs.

To find good solutions with less calculation time, in this paper, we introduced a strategy of combination of two chaotic searches: the LK algorithm controlled by the chaotic dynamics [11, 14] and the S&C ejection chain method controlled by the chaotic dynamics [15].

To escape from local minima and to explore better solutions, the chaotic dynamics is introduced in both methods. To realize the chaotic dynamics, we use a chaotic neuron model proposed by Aihara et al. [13]. The chaotic neuron model can reproduce the refractoriness which is one of the important properties of real nerve cells. When a neuron has just fired, the firing of this neuron is inhibited for a while. By controlling executions of the LK algorithm (or S&C ejection chain method) by the firings of chaotic neuron, the chaotic search method can escape from local minima efficiently, because the same improvements are restricted for a while.

In the proposed method, two local search algorithms, the LK algorithm and S&C ejection chain method, are controled by chaotic dynamics. Namely, the LK algorithm controled by the chaotic dynamics and the S&C ejection chain method controled by the chaotic dynamics are stochastically selected to execute. To realize this stragety, we introduced a combination rate p. When a chaotic neuron is updated, either the LK algorithm or the S&C ejection chain method is controlled by dynamics of the chaotic neuron. In the proposed method, the LK algorithm is selected with probability $1 - p$, and the S&C ejection chain method is selected with probability p.

If $p = 0$, the proposed method is the same as the chaotic search method using the LK algorithm [11, 14]. If $p = 1$, the proposed method is the same as the chaotic search method using the S&C ejection chain method [15]. Then, the gain effect is described by the following equations:

$$\xi_i(t+1) = \max_j \{\beta(t)\Delta_{ij}(t) + \zeta_j(t)\} \tag{1}$$

$$\Delta_{ij}(t) = \begin{cases} \Delta_{ij}^{\mathrm{LK}}(t) & \text{with probability } 1-p \\ \Delta_{ij}^{\mathrm{SC}}(t) & \text{with probability } p. \end{cases} \tag{2}$$

$$\beta(t+1) = \beta(t) + \frac{q}{\frac{1}{N}\sum_{i=1}^{N}|\Delta_{ij}(t)|}, \tag{3}$$

where $\beta(t)$ is a scaling parameter of the gain effect at time t; $\Delta_{ij}^{\mathrm{LK}}(t)$ is a gain of the LK algorithm which connects cities i and j (Fig. 1(a)), namely $\Delta_{ij}^{\mathrm{LK}}(t) = D_0(t) - D_{ij}^{\mathrm{LK}}(t)$, where $D_0(t)$ is a length of a current tour at time t and $D_{ij}^{\mathrm{LK}}(t)$ is a length of a new tour if cities i and j are connected by the LK algorithm at time t. $\Delta_{ij}^{\mathrm{SC}}(t)$ is a gain of the S%C ejection chain method which connects cities i and j (Fig. 1(b)).

Next, the refractory effect is described as follows:

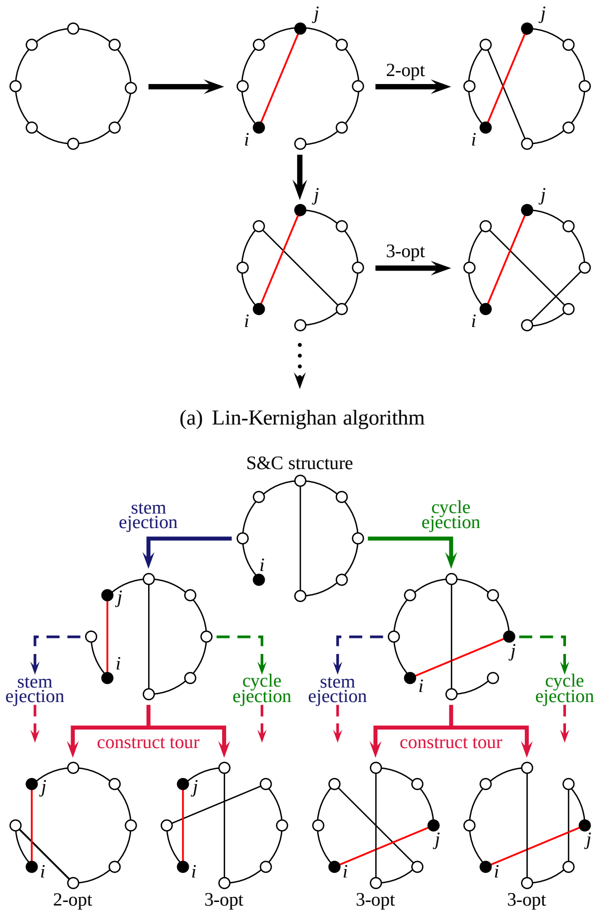
$$\zeta_i(t+1) = -\alpha \sum_{d=0}^{t} k_r^d x_i(t-d) + \theta, \tag{4}$$

where α is a scaling parameter of the refractory effect; k_r is a decay parameter; $x_i(t)$ is an output of the ith chaotic neuron at time t; and θ is a threshold value. If a neuron has fired in the past, Eq. (4) becomes negative. Therefore, the refractory effect inhibits the firing of the neuron in response to the past firing history.

Finally, the output of the ith chaotic neuron at time $t + 1$ is described as follows:

$$x_i(t+1) = f\left(\xi_i(t+1) + \zeta_i(t+1)\right), \tag{5}$$

where $f(y) = 1/(1 + e^{-y/\epsilon})$. If $x_i(t) \geq 1/2$, the ith chaotic neuron fires at time t, and the LK algorithm (or S&C ejection chain method) which connects cities i and j is executed. All neurons are updated asynchronously and randomly. A single iteration is defined to be updates of all neurons.



(a) Lin-Kernighan algorithm

(b) S&C ejection chain method

Figure 1: Procedures of (a) the LK algorithm and (b) the S&C ejection chain method.

3. Results

To evaluate performance of the proposed method, we used benchmark instances from TSPLIB [16]. We conducted numerical simulations using the gcc compiler on 2.66GHz Intel Core 2 Duo processor with 4GB memory running on MAC OS X 10.5.8.

Initial solutions are constructed by the nearest neighbor method. Then, to eliminate redundant searches, we use a candidate list: quadrant neighbors.

Then, we discuss performance of the three methods, namely CS-LK, CS-SC, and CS-LKSC with $p = 0.1$ for various instances. In the experiment, we use large scale instances from TSPLIB [16]. Parameters except q of the CS-LK, CS-SC, and CS-LKSC are set to the same values: $\beta(0) = 0, \alpha = 1.0, k_r = 0.5, \theta = 1.0$, and $\epsilon = 0.002$. Values of q should be properly decided depending on which local search algorithms we use. For the CS-LK, $q = 0.045$; for the CS-SC, $q = 0.060$; and for the CS-LKSC, $q = 0.045$. Each method is applied for 200 iterations. Finally, the best solutions obtained for 200 iterations are improved by its local search algorithm until no further improvements are found.

Table 1 shows the results of the proposed methods for the instances with more than 10^4 cities of TSPLIB. We fix the number of trials to 10 for all instances.

From the results, the proposed chaotic search methods obtain solutions within less than 1% for all instances. Although the CS-SC and CS-LKSC show higher performance than the CS-LK, the CS-SC takes more running time, particularly for large scale instances. On the other hand, the CS-LKSC can obtain better solutions in shorter time than the CS-SC for all instances. Although running time of the CS-LKSC is almost the same as that of the CS-LK for instances with the order of 10^5 cities, for larger instances, the CS-LKSC takes much calculation time than the CS-LK. Thus, if we apply the CS-LKSC to larger instances, we need to adjust the combination rate p to a smaller value.

4. Conclusions

In this paper, we proposed a novel chaotic search method which combines the chaotic search method using the Lin-Kernighan algorithm with that using the stem-and-cycle ejection chain method. We used large scale instances from TSPLIB. From the results, it is revealed that the proposed method has higher performance than the conventional chaotic search method. Then, performance of the proposed method does not depend on the size of the instance.

Acknowledgments

The authors wish to thank K. Aihara, Y. Horio, M. Adachi, and M. Hasegawa for their fruitful comments and discussions. The research of T.I. is partially supported by Grant-in-Aid for Scientific Research (B) (No.20300085) from the JSPS.

References

[1] I Or. Traveling salesman-type combinatorial problems and their relation to the logistics of regional blood banking. *Ph.D thesis. Department of Industrial Engineering and Management Science, Northwestern University, Evanston, Illinois*, 1967.

[2] S. Lin and B. Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, 21:498–516, 1973.

Table 1: Average gaps of the chaotic search methods. CS-LK stands for the chaotic search method using the LK algorithm [11, 14], CS-SC stands for the chaotic search method using the S&C ejection chain method [15] and CS-LKSC stands for the chaotic search method using the LK algorithm and S&C ejection chain method. Bold faces represent the best result.

Instance	Type	Gap [%]			Time [s]		
		CS-LK	CS-SC	CS-LKSC	CS-LK	CS-SC	CS-LKSC
brd14051	EUC_2D	0.775	0.726	**0.656**	363.57	1431.34	474.55
d15112	EUC_2D	0.701	0.669	**0.612**	395.83	1634.78	525.05
d18512	EUC_2D	0.770	0.708	**0.654**	587.93	2458.86	786.24
pla33810	CEIL_2D	0.672	0.682	**0.531**	852.30	9283.02	1782.05
pla85900	CEIL_2D	0.557	0.580	**0.461**	5438.32	63267.95	11751.24
rl11849	EUC_2D	0.717	0.657	**0.566**	177.85	1050.42	274.58
usa13509	EUC_2D	0.668	0.621	**0.612**	264.00	1327.78	380.31

[3] F.Glover. New ejection chain and alternating path methods for traveling salesman problems. *Computer Science and Operations Research*, pages 449–509, 1992.

[4] C. Rego. Relaxed tours and path ejections for the traveling salesman problem. *European Journal of Operational Research*, 106:522–538, 1998.

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

[6] J.H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. The University of Michigan Press, 1975.

[7] F. Glover. Tabu search–part I. *ORSA J. Computing*, 1:190–206, 1989.

[8] M. Hasegawa, T. Ikeguchi, and K. Aihara. Combination of chaotic neurodynamics with the 2-opt algorithm to solve traveling salesman problems. *Physical Review Letters*, 79:2344–2347, 1997.

[9] M. Hasegawa, T. Ikeguchi, and K. Aihara. Solving large scale traveling salesman problems by chaotic neurodynamics. *Neural Networks*, 15:271–283, 2002.

[10] M. Hasegawa, T. Ikeguchi, and K. Aihara. Tabu search for the traveling salesman problem and its extension to chaotic neurodynamical search. *Proceedings of International Symposium of Artificial Life and Robotics*, 1:120–123, 2002.

[11] S. Motohashi, T. Matsuura, T. Ikeguchi, and K. Aihara. The lin-kernighan algorithm driven by chaotic neurodynamics for large scale traveling salesman problems. *Lecture Notes in Computer Science*, 5769:563–572, 2009.

[12] T. Matsuura and T. Ikeguchi. Chaotic search for traveling salesman problems by using 2-opt and oropt algorithms. *Lecture Notes in Computer Science*, 5164:587–596, 2008.

[13] K. Aihara, T. Takabe, and M. Toyoda. Chaotic neural networks. *Physics Letters A*, 144:333–340, 1990.

[14] S. Motohashi, T. Matsuura, and T. Ikeguchi. Chaotic search method using the Lin-Kernighan algorithm for traveling salesman problems. *Proceedings of International Symposium on Nonlinear Theory and its Applications (NOLTA 2008)*, pages 144–147, 2008.

[15] S. Motohashi, T. Matsuura, and T. Ikeguchi. Chaotic search method based on the ejection chain method for traveling salesman problems. *Proceedings of International Symposium on Nonlinear Theory and its Applications (NOLTA2009)*, pages 304–307, 2009.

[16] TSPLIB. available: http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/.