Method for Solving Optimization Problems Using Algorithm Switching by Chaotic Neural Networks

Toshihiro Tachibana† and Masaharu Adachi‡

†Department of Applied Computer Sciences, Shonan Institute of Technology
1–1–25 Tsujido-Nishi-Kaigan, Fujisawa, Kanagawa 251–8511, Japan
‡Department of Electrical and Electronic Engineering, Tokyo Denki University
5 Senju-Asahi-Cho, Adachi-ku, Tokyo 120–8551, Japan
Email: tachibana@sc.shonan-it.ac.jp, adachi@eee.dendai.ac.jp

Abstract—We have proposed method for solving optimization problems using algorithm switching by chaotic neural networks. In this paper, we focus on asymmetric traveling salesman problems and multi-objective optimization problems. These problems have complex solution space to explore. For such problems, we assume that one can explore better solution by switching algorithms adaptively. To determine the switching, the proposed method utilizes tabu effects with exponential decays that are main feature of chaotic neurons in the network. The proposed method finds the same or better solutions than that by the previous method.

1. Introduction

Optimization Problems appears quite frequently in various fields. Design of DSP systems and networks, boring of circuit boards, Robot path planning are example of them.

Depending on the type of optimization problem, there are some cases when it cannot be solved by single algorithm. Therefore, in this paper, we have proposed a method for algorithm switching using chaotic neural networks. We will show performance of the proposed method in asymmetric traveling salesman problems (asymmetric TSPs) and multi-objective optimization problems (MOPs).

This paper is organized as follows. Section 2 introduces the asymmetric TSP and the MOP. In Section 3, we describe the proposed method of algorithm switching using chaotic neural networks. In Section 4, we apply the algorithm switching to asymmetric TSPs. Then results of numerical experiments are shown. In Section 5, we apply algorithm switching to MOPs. Then, results of numerical experiments are also shown. Lastly, we present conclusions.

2. Optimization Problems

2.1. Asymmetric traveling salesman problems

The traveling salesman problem (TSP) is an optimization problem whose goal is to determine the minimum-cost tour visiting n cities, under the conditions that one visits every city only at once [1]. For an n-city problem, a set of the cities is

$$V = \{v_1, v_2, \cdots, v_n\} . \tag{1}$$

A tour σ specifies the order in which these cities are visited. The total cost $f_{TSP}(\sigma)$ of a tour σ can be evaluated from the costs for traveling from city j to city i, $d_{ij}(v_i, v_j \in V)$ as follows.

$$f_{TSP}(\sigma) = \sum_{k=1}^{n-1} d_{\sigma(k),\sigma(k+1)} + d_{\sigma(n),\sigma(1)}. \tag{2}$$

Here, d_{ij} is the cost to travel from city i to city j, n is the number of the cities. The goal of the TSP is to minimize the cost in Eq. (2).

TSP is asymmetric when the cost of travel between two given cities is not symmetric, that is, $d_{ij} \neq d_{ji}$. In this paper, we do not include cases in which two cities are connected only in one direction.

2.2. Multi-objective optimization problems

The multi-objective optimization problems (MOP)[2] is an optimization problem when there are more than one objective functions. This problem is given as follows.

$$\text{minimize } f_i(\mathbf{x}) \ \ (i = 1, 2, \cdots, M), \tag{3}$$
$$\text{subject to. } g_j(\mathbf{x}) \leq 0 \ \ (j = 1, 2, \cdots, N). \tag{4}$$

3. Proposed Algorithm Switching method

We propose algorithm switching method which uses a chaotic neural network. A feature of this method is that the number of neurons is equal to the number of algorithms. The updating of the internal state of each neuron is represented by the following set of equations.

$$\xi_i(t+1) = -\beta_{sw}\Delta_{sw}(t), \tag{5}$$
$$\eta_i(t+1) = -W\sum_{k=1,k\neq i}^{N} x_k(t), \tag{6}$$
$$\zeta_i(t+1) = -\alpha_{sw}\sum_{d=0}^{t} k_r^d x_i(t-d) + \theta, \tag{7}$$
$$x_i(t+1) = f(\xi(t+1)+\eta_i(t+1)+\zeta_i(t+1)). \tag{8}$$

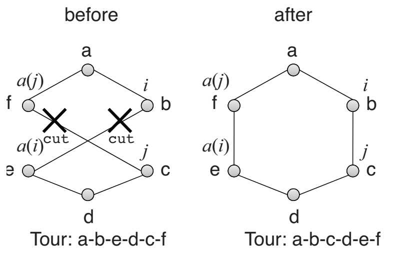

Figure 1: An example of 2-opt exchange: links b–e and f–c are swapped.

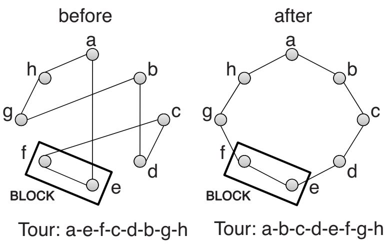

Figure 2: An example of a block–city exchange method: the block cities are e and f, and the exchange partner is city b.

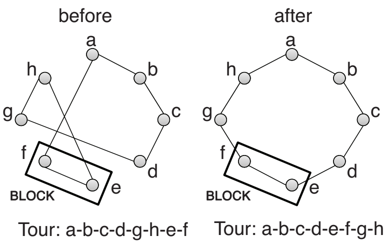

Figure 3: An example of a block insertion method: the block cities are e and f, and the block is inserted at the next to city d.

Here, $x_i(t)$ is the output of the i-th exchanging neuron at time t. $\xi_i(t)$ is the internal state for the gain effect of the exchanging neuron. $\eta_i(t)$ is the internal state for the feedback effect of the exchanging neuron. $\zeta_i(t)$ is the internal state for the tabu effect of the exchanging neuron. k_r is a decay parameter of the gain effect. $\Delta_{sw}(t)$ is a gain of the objective function value when the candidate exchange is executed. θ is a positive bias. α_{sw} is a scaling parameter of the internal effect. β_{sw} is a scaling parameter of the gain effect.

4. Application to Asymmetric TSP

4.1. Proposed method

In this section, we will apply the algorithm switching shown in Section 3 to Asymmetric TSPs. Here, the proposed method is based on the chaotic search[3]. This proposed method[4] uses three exchange methods by the switching. The exchange methods are shown in Section 4.2. In the proposed method, a switching among three exchange methods is determined by chaotic neurodynamics.

4.2. Exchange methods

4.2.1. 2-opt exchange

Two-opt exchange is a basic city exchange method for symmetric TSPs in which any two cities exchange their position in the tour. For example, a tour shown in left panel of Fig. 1 can be changed to right panel of the figure. Fig. 1 by cutting two links and creating two new links.

4.2.2. Block–city exchange method

The 2-opt exchange is a fundamental city exchange method for solving TSPs. When a 2-opt exchange is executed, the direction of a part of the tour is reversed by the city exchange. In symmetric TSPs, this reversal does not alter the cost of the tour. However, in an asymmetric TSP, the 2-opt exchange may cause the cost to rise. To overcome this disadvantage of the 2-opt exchange in asymmetric TSPs, the block–city exchange method has been proposed [5]. This method considers several cities as a BLOCK, and the whole BLOCK is exchanged with another city. During the exchange, the order of the cities in the BLOCK is maintained. We show an example of a block shift operation in Fig. 2.

4.2.3. Block insertion method

Block insertion method [5] considers several cities as a BLOCK, and the whole BLOCK is inserted another branch. During the exchange, the order of the cities in the BLOCK is maintained. We show an example of a block shift operation in Fig. 3.

4.3. Result of numerical experiments

We numerically evaluated the performance of the proposed method for some benchmark problems of TSPLIB [6] and DIMACS[7]. We used asymmetric problems. Asymmetric ratio AR indicates a degree of asymmetry. AR is computed by using the following set of equations.

$$
\begin{aligned}
A &= \{(v_i, v_j) \mid \text{with cost satisfies } d_{ij} \neq d_{ji}\}, \\
B &= \begin{pmatrix} n \\ 2 \end{pmatrix}, \\
AR &= \#(A)/B, \qquad (9)
\end{aligned}
$$

where $\#(A)$ denotes the number of elements in the set A. For example, $AR = 1.0$ shows that all branches are asymmetrical. On the other hand, $AR = 0.0$ shows that all branches are symmetrical. We show results of the performance of the proposed method and conventional methods in Table 1.

From Table 1, in rbg403 and rbg443, the proposed method and previous method has reached the exact solution in all trials. Also, in dc112, the gap of proposed method is smaller than that of previous method. Thus, the proposed method has improved performance.

This experiment yields that the proposed method has the same or better performance as compared to the previous and conventional methods.

Table 1: Experimental results of proposed and conventional methods for asymmetric TSPs.

Instance	Asymmetric Ratio AR	Gap [%] by Proposed	Gap [%] by Previous method[5]	Gap [%] by Quick ACO[8]	Gap [%] by RAI[9]
p43	0.442	0.01 ± 0.02	0.01 ± 0.007	0.02	0.01
dc112	0.151	$\mathbf{0.3} \pm 0.07$	0.5 ± 0.08	6.3	**0.02**
dc126	0.678	0.1 ± 0.06	0.2 ± 0.2	N/A	N/A
rbg403	0.923	$\mathbf{0} \pm 0$	$\mathbf{0} \pm 0$	N/A	0.03
rbg443	0.925	$\mathbf{0} \pm 0$	$\mathbf{0} \pm 0$	6.3	0.02

5. Application to MOP

5.1. Proposed method

In this section, we will apply the algorithm switching shown in Section 3 to MOPs. This proposed method[10] uses switching three particle swarm optimization (PSO) methods[11]. These PSO methods are shown in below.

The first method is multi-objective optimization PSO (MOPSO) [12]. The second method is optimized multi-objective PSO (OMOPSO) [13]. This method is added mutation such as in the genetic algorithm to MOPSO. The third method is speed-constrained multi-objective PSO (SMPSO) [14]. This method is added limit to speed of particle motion to OMOPSO.

In the proposed method, a switching among three PSO methods is determined by chaotic neurodynamics.

5.2. Result of numerical experiments

We numerically evaluated the performance of the proposed method for some benchmark problems of ZDT test suites[15], by performing 21 trials for each problem.

Generational distance GD is calculated by using the following equation.

$$GD = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n}. \tag{10}$$

Here, d_i is the Euclidean distance of Pareto optimal solution closest to the solution and the i-th solution. We show results of the average of GD and the standard deviation of the proposed method and that of the conventional methods in Table 2. Also, we show results of Non-dominated solution set in Figs. 4-7.

Table 2: Average of GD and standard deviation.

	ZDT3	ZDT4
Proposed	0.00145 ± 0.00060	0.00140 ± 0.00091
MOPSO	1.62514	0.12759
OMOPSO	0.02278	0.03348
SMPSO	0.01167	0.17198

From Table 2 and Figs. 4-7, the proposed method has reached closer to the Pareto optimum solution in comparison with conventional methods.

6. Conclusion

In this paper, we have proposed a method for solving optimization problems using algorithm switching by chaotic neural networks. We have proposed methods of algorithm switching for asymmetric TSP and MOP as application examples of them. As a result, the proposed method showed the same or better performance as compared with conventional methods.

References

[1] M. Kubo, “Engineering for Logistics”, Asakura Shoten, 2001 (in Japanese with English abstract).

[2] C. A. Coello Coello, G. B. Lamont, D. A. Van Veldhuizen, “Evolutionary Algorithms for Solving Multi-Objective Problems”, Springer, 2007.

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

[4] T. Tachibana, I. Furushima, M. Adachi, “Solving Asymmetric Traveling Salesman Problems Using Switching of Exchange Methods”, *Proc. of 2014 IEICE General Conf.*, , 2014 (in Japanese).

[5] T. Tachibana, M. Adachi, “A Method for Solving Asymmetric Traveling Salesman Problems Using Block Shift Operation and Chaotic Neurodynamics”, *IEEJ Trans. on Electronics, Information and Systems*, **132**(5), 774–781, 2012 (in Japanese).

[6] G. Reinelt: “TSPLIB — a Traveling Salesman Problem Library”, *ORSA J. on Computing*, **3**, 376–384, 1991.

[7] D. S. Johnson, L. A. McGeoch, F. Glover, C. Rego, ”8th DIMACS Implementation Challenge: The Traveling Salesman Problem”, (http://www.research.att.com/ dsj/chtsp/), 2000.

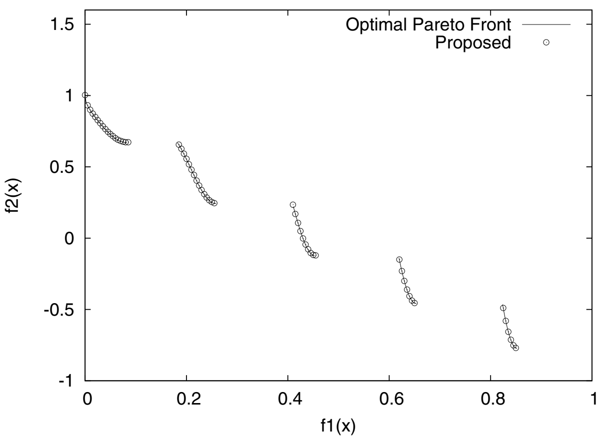

Figure 4: Non-dominated solution set by proposed method.

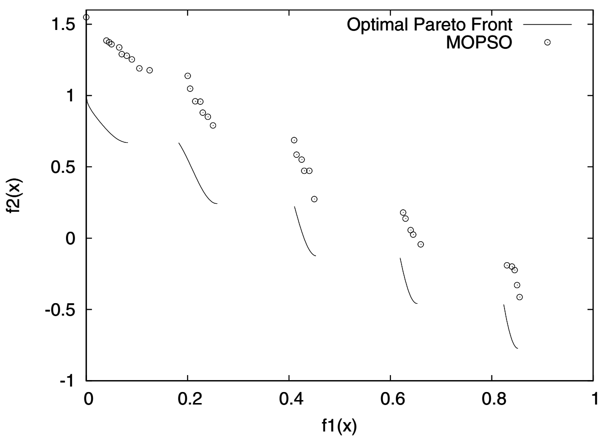

Figure 5: Non-dominated solution set by MOPSO.

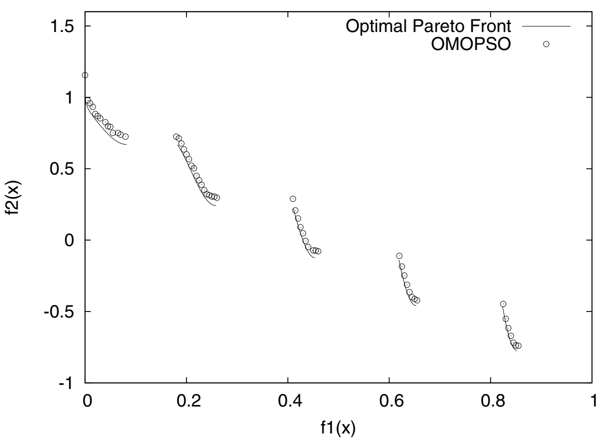

Figure 6: Non-dominated solution set by OMOPSO.

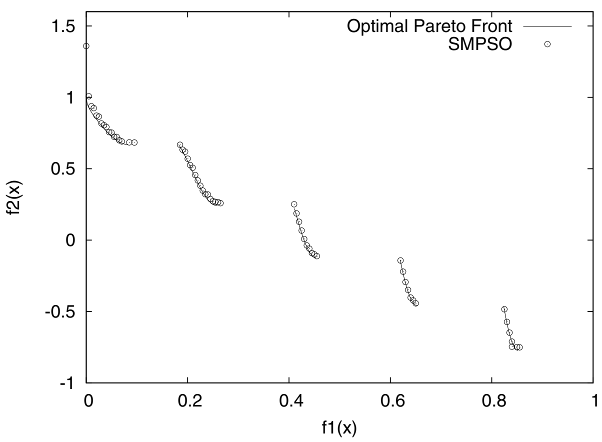

Figure 7: Non-dominated solution set by SMPSO.

[8] W. Cheng, B. Scheuermann, M. Middendorf, "Quick-ACO: Accelerating Ant Decisions and Pheromone Updates in ACO", *Evolutionary Computation in Combinatorial Optimization*, **6622/2011**, 238–249, 2011.

[9] J. Brest, J. Žerovnik, "A Heuristic for the Asymmetric Traveling Salesman Problem", *Proc. of MIC 2005*, 45–150, 2005.

[10] T. Tachibana, M. Adachi, "Solving Multi-objective Optimization Problems Using Particle Swarm Optimization Methods with Switching by Chaotic Neurodynamics", *Technical Report of IEICE*, **111**(498), 51–56, 2012 (in Japanese).

[11] J. Kennedy, R. Eberhart, "Particle Swarm Optimization". *Proc. of IEEE Neural Networks*, 1942–1948, 1995.

[12] C. A. Coello Coello, M. S. Lechuga,"MOPSO: a proposal for multiple objective particle swarm optimization" ,*Proc. of IEEE CEC 2002*, 1051–1056, 2002.

[13] M. R. Sierra, C. A. Coello Coello, "Improving PSO-Based Multi-objective Optimization Using Crowding, Mutation and ϵ-Dominance", *Proc. of EMO 2005*, 505–519, 2005.

[14] A. J. Nebro, J. J. Durillo, J. García-Nieto, "SMPSO: A new PSO-based metaheuristic for multi-objective optimization", *Proc. of MCDM 2009*, 66–73, 2009.

[15] E. Zitzler, K. Deb, and L. Thiele, "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results", *Evolutionary Computation*, **8**(2), 173–195, 2000.