2009 International Symposium on Nonlinear Theory and its Applications
NOLTA'09, Sapporo, Japan, October 18-21, 2009

A Method for Solving Asymmetric Traveling Salesman Problems Using Chaotic Neurodynamics and Block Shift Operations

Toshihiro TACHIBANA†* and Masaharu ADACHI†**

†Graduate School of Advanced Science and Technology, Tokyo Denki University
2-2 Kanda–Nishiki–cho, Chiyoda–ku, Tokyo 101–8457, Japan
Email: * 09ude01@ms.dendai.ac.jp, ** adachi@eee.dendai.ac.jp

Abstract—An asymmetric Traveling Salesman Problem is one in which the costs for travel between one city and another are not symmetric. We propose a method for solving such problems based on previous work by Hasegawa *et al.* It uses a chaotic neural network and tabu search; block shift operations and 2-opt exchange are used as exchange methods. The proposed method obtains better solutions with a shorter computational time than the method we previously proposed.

1. Introduction

The Traveling Salesman Problem (TSP) is a famous NP-hard problem. It is a combinatorial optimization problem which appears quite frequently in various fields such as transit and control [3]. The TSP suffers from combinatorial explosion: the number of feasible solutions increases as $\frac{(n-1)!}{2}$ as the number of cities, n, increases[3, 5]. In other words, if there are a large number of cities, it is difficult to obtain a good solution to this problem using simple techniques such as the round robin method[5].

Symmetric TSPs have been studied by many researchers because assuming symmetry reduces the computational cost of solving the problem. However, in everyday problems of this type, it is rare that the costs are symmetric. Therefore, there are demands for dealing with asymmetric cost problems [4]; however, thus far, asymmetric TSPs have not been extensively studied.

This paper is organized as follows. Section 2 introduces the asymmetric TSP. The method for solving TSPs by Hasegawa *et al.*, on which our proposed method is based, is described in Section 3. In Section 4, we describe the proposed method, and the results of some numerical experiments are given in Section 5. Lastly, we present some conclusions.

2. Asymmetric traveling salesman problems

The TSP is an optimization problem in which the goal is to determine the minimum-cost tour around n cities, visiting each city only once [4]. For an n-city problem, the 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 $f(\sigma)$ of a tour σ can be evaluated from the costs for travel from city j to city i, $d_{ij}(v_i, v_j \in V)$.

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

The goal of the TSP is to minimize the cost in Eq. (2). The nomenclature for TSPs used in this paper is summarized in Table 1.

Table 1: Nomenclature for TSP

d_{ij}	:	Cost to travel from city i to city j
$f_{TSP}(\sigma)$	:	Total cost of a tour
σ	:	The order of cities visited in a tour
V	:	Set of cities to visit
n	:	Number of the cities
u_{im}	:	Internal state of the neuron im
θ_{im}	:	Threshold of the neuron im
x_{im}	:	Output of the neuron im
$w_{im,jn}$	:	Synaptic weight from the neuron jn to the neuron im

A 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.

3. Solving the n-city TSP with n neurons [1]

The proposed method is based on the method of Hasegawa *et al.* which uses a chaotic neural network with tabu search. A feature of this method is that the number of neurons used is equal to the number of cities. We give an outline of the method of Hasegawa *et al.* in this section.

3.1. Coding of networks

The updating of the internal state of each neuron is done using the following set of equations.

$$f(x) = \frac{1}{1+e^{-\gamma x}} \tag{3}$$

$$\xi_i(t+1) = \max_j\{\zeta_j(t+1) + \beta\Delta_{ij}(t)\} \tag{4}$$

$$\eta_i(t+1) = -W\sum_{k=1}^{N} x_k(t) + W \tag{5}$$

$$\zeta_i(t+1) = -\alpha\sum_{d=0}^{s-1} k_r^d x_i(t-d) + \theta \tag{6}$$

$$x_i(t+1) = f\big(\xi_i(t+1) + \eta_i(t+1) + \zeta_i(t+1)\big) \tag{7}$$

If $x_i(t+1) > \frac{1}{2}$, city i is connected to the city j that gives the maximum in Eq. (4) using 2-opt exchanges. The nomenclature for neural networks used in this paper is summarized in Table 2.

Table 2: Nomenclature for networks

k_r	:	Decay parameter of the gain effect
α	:	Scaling parameter of the tabu effect
β	:	Scaling parameter of the gain effect
Δ_{ij}	:	Gain of the objective function value offered by the 2-opt exchange which links citied i and j.
θ, R	:	Positive biases
s	:	Tabu list size

3.2. Tabu search

Tabu search[6] is a technique to prevent the network state being trapped at local minimum and periodically repeating the same tour. It works by temporarily removing a city and its links from the list of candidates for spot exchange. Hasegawa's method has lists of tabus. One of these is a tabu applied to the city that is already selected for spot exchanges. This tabu list has a queue structure. That is, when the number of the cities in the tabu list becomes larger than the size of the list, the oldest city in the list returns to being a spot exchange candidate.

3.3. 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. In effect, two links are cut and two new links are created. An example of 2-opt exchange is shown in Figure. 1.

4. Proposed method

4.1. Overview

We propose a method that is a modification of Hasegawa's method described in section 3. Although the proposed method uses the same network and the same algorithm, it differs through the use of two exchange methods

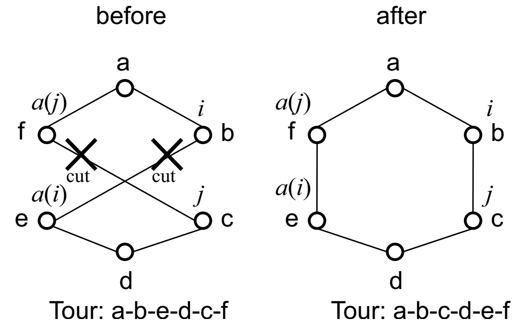

Figure 1: An example of 2-opt exchange: cities c and e are swapped.

selected with a fixed probability. The two exchange methods are block shift operations and 2-opt exchange. The proposed method also uses two search modes: a bidirectional search and an unidirectional one.

4.2. Block shift operations

The 2-opt exchange is the 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 shift operation has been proposed [2]. 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 Figure. 2. Although a 2-opt exchange could be viewed as a type of block shift operation, we use the term "block shift operation" in this paper only when one of the two sets of cities being exchanged has more than one element.

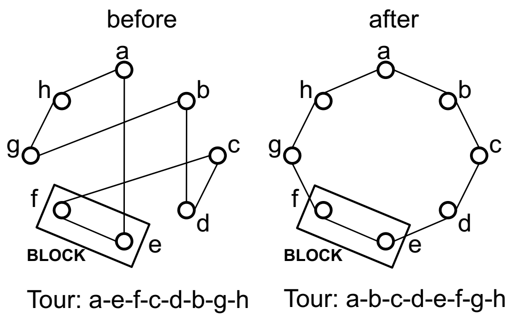

Figure 2: An example of a block shift operation: the BLOCK cities are e and f, and the exchange partner is city b.

Table 3: Experimental results of proposed and conventional methods for a 13-city asymmetric TSP.

Exact solution: 191.48				
Name	Proposed	previous work[2]	2-opt	Hasegawa[1]
Minimum tour	**191.48**	**191.48**	249.20	201.03
Maximum tour	263.66	249.35	321.91	272.96
Average tour	**212.58**	228.65	302.08	237.07

Table 4: Experimental results of proposed and conventional methods for a 70-city asymmetric TSP.

Exchange method	Proposed (Block)	2-point	or-opt	2-opt[1]
Minimum tour	**756.26**	1337.35	1053.52	823.90
Maximum tour	945.96	1895.92	1891.31	1053.52
Average tour	**853.00**	1589.86	1343.57	917.26
Computational time	40.8[s]	45.1[s]	**33.0**[s]	35.8[s]
Minimum relative frequency	**1.00**	1.77	1.39	1.09
Maximum relative frequency	**1.00**	2.00	2.00	1.11
Average relative frequency	**1.00**	1.86	1.58	1.08
Computational time relative frequency	1.00	1.11	**0.81**	0.88
Cumulative relative frequency	**1.00**	7.31	3.56	1.15

4.3. Combination of block shift operations and tabu search

We combine chaotic neural networks and block shift operations as follows. A BLOCK consists of two cities. If the i-th city in the tour is the first city of the BLOCK, the $(i+1)$-th city is the second city in the BLOCK. The partner for the exchange is determined by the procedure given in Section 3.1.

4.4. Bidirectional search and unidirectional search

In the proposed method, two search modes are used. The bidirectional search mode calculates both clockwise and counterclockwise tours. The unidirectional search mode calculates the tour in a specified direction. The bidirectional search mode uses block shift operations and 2-opt exchange, while the unidirectional search mode uses only block shift operations.

4.5. 3-opt exchanges

In this paper we use 2-opt and 3-opt exchanges throughout the paper. Moreover, a 3-opt exchange is made by executing two 2-opt exchanges (see Fig. 3).

The probability for the i-th neuron to be included the 3-opt exchange is given by $\frac{1}{i}$. This distribution is intended to make the constituent neurons have different probabilities for 3-opt exchange.

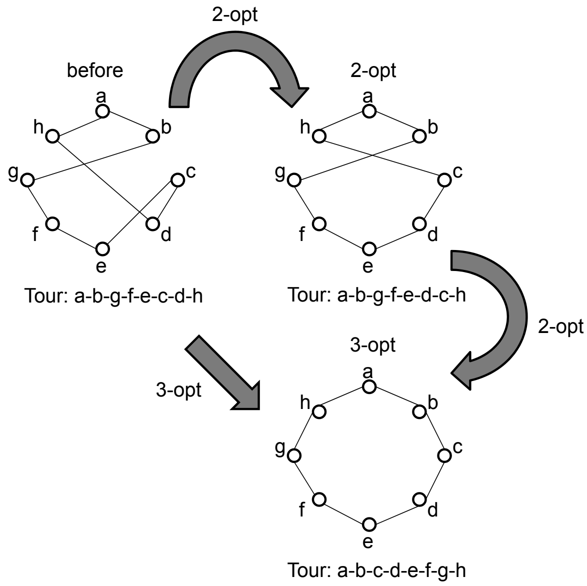

Figure 3: A sequence of two 2-opt exchanges (swapping c with d and then b with h) is equivalent to a 3-opt exchange.

5. Results of numerical experiments

In experiment 1, we take some benchmark problems of TSPLIB [7], which is a list of benchmark problems of symmetric TSPs, and modify them to be asymmetric TSPs. The 13-city problem is created by mixing problems eil76 and ftv47 from TSPLIB. We performed 30 trials for each problem. Performance comparisons of the various methods are shown in Table 3. The proposed method and our previous method[2] found the exact solution.

Table 5: Experimental results of proposed and conventional methods for p43.

Method	Proposed 2-opt (1:1)	Proposed 2-opt (7:3)	Proposed k-opt (1:1)	Proposed k-opt (7:3)	Hasegawa[1]	Local Search[8]	Hybrid GA[9]
Minimum tour	**5620**	**5620**	**5620**	5621	5622	N/A	N/A
Maximum tour	5630	5627	5628	5629	11105	N/A	N/A
Average tour	5623.2	5622.3	5622.5	5622.8	5857.8	N/A	N/A
Gap [%]	0.05	0.04	0.04	0.04	4.06	0.01	0

Table 6: Experimental results of proposed and conventional methods for rbg443.

Method	Proposed 2-opt (1:1)	Proposed 2-opt (7:3)	Proposed k-opt (1:1)	Proposed k-opt (7:3)	Hasegawa[1]	Local Search[8]	Hybrid GA[9]
Minimum tour	**2720**	**2720**	**2720**	**2720**	2891	N/A	N/A
Maximum tour	2727	2723	2729	2723	3058	N/A	N/A
Average tour	2720.6	2720.2	2721.2	2720.06	2935.1	N/A	N/A
Gap [%]	0.02	0.007	0.04	0.002	7.3	0.09	0

In experiment 2, we evaluated the relationship between the computational time and the tour. We use relative ratio to evaluate performance of the methods. In Table 4, we show an example of the relationship between the computational time and the tour for a 70-city asymmetric TSP modified from eil76 and ftv47 in TSPLIB. From the viewpoint of cumulative relative frequency the method proposed in this paper is the best. As a result, the proposed method provides the best balance from the point of view of the relative ratio.

In experiment 3, we evaluated some benchmark problems of TSPLIB [7], performing 100 trials for each problem. We used the 17-city asymmetric problem 'br17', the 43-city asymmetric problem 'p43', the 48-city asymmetric problem 'ry48p' and the 443-city asymmetric problem 'rbg443'. We show comparisons of the methods for 'p43' and 'rbg443' in Table 5 and Table 6. In these tables, the ratio of exchange methods (block shift : 2-opt) are 1 : 1 or 7 : 3. The results shown that a high ratio of block shift operations to 2-opt exchanges achieve the exact solution more frequently.

6. Conclusions

In this paper, we have proposed a method for solving an asymmetric TSP using chaotic neural dynamics. The proposed method gives better solutions than the method of Hasegawa *et al.*[1], although it is not superior to the existing method of using a hybrid genetic algorithm[9].

A future problem is to investigate a method to select good parameters automatically.

Acknowledgments

This work was supported by KAKENHI (20300085).

References

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

[2] T. Tachibana, M. Adachi, "A Method for Solving Asymmetric Traveling Salesman Problems Using Neural Networks and Block Shift Operations with Tabu Search", *Proc. of NOLTA 2008*, 69–72, 2008.

[3] RTRI Planning Systems, "Scheduling Algorithm for Railways", NTS, 2005 (in Japanese with English abstract).

[4] M. Kubo, "Engineering for Logistics", Asakura Shoten, 2001 (in Japanese with English abstract).

[5] M. Yagiura, T. Ibaraki, "Combinatorial Optimization", Asakura Shoten, 2001 (in Japanese with English abstract).

[6] F. Glover, "Tabu Search I", *ORSA Journal on Computing*, **1**, 190–260, 1989.

[7] TSPLIB (http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/)

[8] J. Cirasella, D. S. Johnson, L. A. McGeoch, W. Zhang, "The Asymmetric Traveling Salesman Problem: Algorithms, Instance Generators, and Tests", Algorithm Engineering and Experimentation, **2153**, 2001.

[9] L.N. Xing, Y.W. Chena, K.W. Yanga, F. Houa, X.S. Shena and H.P. Caia, "A Hybrid Approach Combining an Improved Genetic Algorithm and Optimization Strategies for the Asymmetric Traveling Salesman Problem", Engineering Applications of Artificial Intelligence, **21**, 1370–1380, 2008.