© Copyright by Institute of Organization and Management in Industry „ORGMASZ”
Vol 6(4); p. 58 - 66 Year 2009
10.2478/v10061-009-XXX-Y

Application of a Modified Hopfield Network to the Traveling Salesman Problem

Zbigniew Nagórny

Abstract

This paper presents the results of the application of the modified Hopfield network to the travelling salesman problem (TSP). A cost function of the TSP consists of four components: two terms which ensure that the salesman's tour is valid, the term which forces neurons to have the output signal equal to 0 or 1, and the total length of the salesman's tour. For two 10-city benchmarks the average tour length of obtained solutions is equal to the optimal tour length. Other works did not report such results using the classical Hopfield network. For greater numbers of cities, the solution quality is significantly better in comparison with the quality of results achieved in other works. The method of auto-tuning of the constant in the fourth component of the cost function is presented. This method ensures very good quality results for randomly generated instances of the 10-city TSP. The presented network is destined for hardware implementation.

Keywords

combinatorial optimization, Hopfield network, NP-hard, parameter auto-tuning, travelling salesman problem (TSP)

Introduction

The travelling salesman problem (TSP) is a classical example of a combinatorial optimization problem, which has proved to be an NP-hard. In the TSP, the objective is to find the salesman's tour to visit all the N cities on his list once and only once, returning to the starting point after travelling the shortest possible distance. Additionally, we assume that the distance from city i to city j is the same as from city j to city i (symmetrical TSP). A tour can be represented as an ordered list of N cities. In this case, for $N > 2$ there is $N!/2N$ different tours (the same tour may be started from any city from among N cities and traversed either clockwise or anti-clockwise). Many methods are used for solving the TSP, e.g.: the Lin-Kernighan algorithm [14], simulated annealing [8,16,20], genetic algorithms [13,16], evolutionary algorithms [13,14], tabu search [14,16], ant colony optimization [2]. TSP can be solved using neural networks. A self-organizing neural network [3,5,14] and the Hopfield network [1,4-7,9-12,16,17,19-22] are able to solve the TSP. The advantages of using neural networks in hardware implementation are the possibility of parallel computation within many units working together called neurons and a decrease in computation time. Unfortunately, the quality of solutions for the TSP found by the network was worse than that reported in the Hopfield and Tank paper [1,7,21].

This paper presents the results of the application of the modified Hopfield network to the TSP. A brief description of the Hopfield network for the TSP is presented in Section 2. The simulation results achieved using the modified Hopfield network are given in Section 3. The method of auto-tuning of the constant in the fourth component of the TSP cost function is presented in Section 4. Finally, conclusions are drawn in Section 5.

Modified Hopfield Network for the TSP

The Hopfield network is able to solve optimization problems [6]. Very often the neuron activation function is given by [4,6,7,10-12,19,21]

$$V = f(U) = \frac{1}{2}\left[1 + \tanh\left(\frac{U}{U_0}\right)\right] \qquad (1)$$

where U is the neuron input signal, V is the output signal, and U_0 is a constant.

An energy function E is defined for this network, which is described by [6]

$$E = -\frac{1}{2}\sum_{i=1}^{X}\sum_{j=1}^{X}\omega_{ij} V_i V_j - \sum_{i=1}^{X} I_i V_i \qquad (2)$$

where X is the number of neurons, ω_{ij} is the weight of interconnect between the output of the neuron j and the input of the neuron i, and I_i is the external input signal of the neuron i. The Hopfield network continuously evolves in time to minimize the energy function [6].

The Hopfield network for the TSP is built of $X = NN$ neurons. The network consists of N rows, containing N neurons according to Fig. 1. All neurons have two subscripts. The first one defines the city number and the second one the position of the city in the tour. If a neuron in the stable state of the network, has the output signal $V_{xi} = 1$, it means that the city x should be visited in the stage i of the tour [6].

The cost function of the TSP proposed by Hopfield and Tank is the reason for difficulties in converging to valid tours [21]. The modified cost function of the TSP consists of four components: E_1, E_2, E_3, and E_4 [1]. E_1 ensures that all cities should be visited once (in the stable state of the network, in all rows one neuron has the output signal equal to 1). E_2 ensures that the salesman should visit one city in each stage of the tour (in all columns one neuron has the output signal equal to 1). E_3 forces neurons to have the output signal equal to 0 or 1 or near these values. E_4 is equal to the total length of the salesman's tour. These components are described by [1,9]

$$E_1 = \frac{A}{2}\sum_{x=1}^{N}\left(\sum_{i=1}^{N} V_{xi} - 1\right)^2 = = -A\sum_{x=1}^{N}\sum_{i=1}^{N} V_{xi} + \frac{A}{2}\sum_{x=1}^{N}\sum_{i=1}^{N}\sum_{x'=1}^{N}\sum_{i'=1}^{N}\delta_{xx'} V_{xi} V_{x'i'} + \frac{AN}{2} \qquad (3)$$

$$E_2 = \frac{B}{2}\sum_{i=1}^{N}\left(\sum_{x=1}^{N} V_{xi} - 1\right)^2 = = -B\sum_{x=1}^{N}\sum_{i=1}^{N} V_{xi} + \frac{B}{2}\sum_{x=1}^{N}\sum_{i=1}^{N}\sum_{x'=1}^{N}\sum_{i'=1}^{N}\delta_{ii'} V_{xi} V_{x'i'} + \frac{BN}{2} \qquad (4)$$

$$E_3 = \frac{C}{2}\sum_{x=1}^{N}\sum_{i=1}^{N} V_{xi}\left(1 - V_{xi}\right) = = \frac{C}{2}\sum_{x=1}^{N}\sum_{i=1}^{N} V_{xi} - \frac{C}{2}\sum_{x=1}^{N}\sum_{i=1}^{N}\sum_{x'=1}^{N}\sum_{i'=1}^{N}\delta_{xx'}\,\delta_{ii'} V_{xi} V_{x'i'} \qquad (5)$$

$$E_4 = \frac{D}{2}\sum_{x=1}^{N}\sum_{\substack{x'=1 \\ x' \neq x}}^{N}\sum_{i=1}^{N} d_{xx'} V_{xi}\left(V_{x',i+1} + V_{x',i-1}\right) =$$

$$= \frac{D}{2}\sum_{x=1}^{N}\sum_{i=1}^{N}\sum_{x'=1}^{N}\sum_{i'=1}^{N} d_{xx'} V_{xi} V_{x'i'}\left(\delta_{i',i+1} + \delta_{i',i-1}\right) \qquad (6)$$

where A, B, C, D are constants, $d_{xx'}$ is the distance between the cities x and x', subscripts $i + 1$ and $i - 1$ are defined modulo N, i.e., subscript $i + 1$ is equal to $i + 1$ (if $i < N$) or 1 (otherwise), subscript $i - 1$ is equal to $i - 1$ (if $i > 1$) or N (otherwise), $\delta_{ii'}$ is equal to 1 (if $i = i'$) or 0 (otherwise), similar $\delta_{xx'}$.

Comparing Eq. 2 and the TSP cost function we get equations for weight and external input signal values

$$\omega_{xi,x'i'} = -A\ \delta_{xx'} - B\ \delta_{ii'} + C\ \delta_{xx'}\delta_{ii'} - D\ d_{xx'}\left(\delta_{i',i+1} + \delta_{i',i-1}\right) \quad (7)$$

$$I_{xi} = A + B - \frac{C}{2} \quad (8)$$

where $\omega_{xi,x'i'}$ is the weight of interconnection between the neuron $x\,i$ and the neuron $x'i'$ and I_{xi} is the external input signal of the neuron $x\,i$. The Hopfield network with weights and external input signals described by Eqs. 7-8 can be used for solving the TSP.

The simulation of the Hopfield network can be done with the Euler method [4,7]

$$U_i^{t+\Delta t} = U_i^t + \Delta t \left[\left(\sum_{j=1}^{N} \omega_{ij} V_j \right) + I_i - \frac{U_i^t}{\tau} \right] \quad (9)$$

where τ is the time constant of the network, Δt is the time step of the Euler method, and U_i^t is the input signal of the neuron i in time moment t. A number of trials are performed and best results chosen.

Simulation Results

Salesman's tours for 4 benchmarks of the TSP using the modified Hopfield network were routed. These benchmarks are 10-, 10-, 16-, and 22-city problems. The first benchmark was used in the Hopfield and Tank paper [6], the second benchmark was used in [10,12]. Fig. 2 presents the position of the cities in the unit square for the first and second benchmarks. Table 1 contains the coordinates of the cities for these benchmarks [9,21]. The third and fourth benchmarks were the data sets "ulysses16.tsp" and "ulysses22.tsp" given in the TSP library [18]. Fig. 3 presents the position of the points (cities) on the earth for the third and fourth benchmarks. The geographical latitude and longitude of the points are given in the TSP library. A positive latitude means "North", positive longitude means "East", and negative longitude means "West". The distance between two points is their distance on the idealized sphere with radius 6378.388 kilometers. The optimal tours for these benchmarks are also given in the TSP library. In this work the same scaling of city distances as in [9] was used. After the scaling, the optimal tour lengths were 2.3632 and 2.4128, respectively.

The modified Hopfield network was simulated in software using a program written in C++. The simulations were performed according to Eqs. 1 and 9, they had been put $U_0 = 0.1$, $\tau = 1$, and $\Delta t = 0.01$. The initial input signals of neurons were independently randomly chosen in the interval between $-0.1U_0$ and $0.1U_0$. The following motivation of settings of the interval was used. For the neuron activation function according to Eq. 1 and $U_0 = 0.1$, neuron output signals were near 0.5 and none of the neurons was privileged. The random perturbation $\pm 0.1U_0$ was used in the same way as in other works [6,7]. The neurons were updated synchronously, i.e., all neuron input and output signals were calculated and updated at the same time. The termination criterion was that, the network was regarded as being in a stable state when all $|V_{xi}^{t+\Delta t} - V_{xi}^{t}| \leq 10^{-6}$. In the stable state all neurons with $V_{xi} \geq 0.5$ were set $V_{xi} = 1$, otherwise set $V_{xi} = 0$. A similar solution was used or reported in

[7,21]. The neuron weights and external input signals were calculated according to Eqs. 7-8. The constants in Eqs. 7-8 were set $A = 5$, $B = 5$, and $C = 0.5$ [15]. The constant D in Eq. 7 was set to a value, for which almost all the solutions were valid, i.e., 90% to 100% of solutions. The solutions are valid if the components E_1 and E_2 in Eqs. 3-4 are equal to 0. For all benchmarks 100 trials were performed using the modified Hopfield network.

Table 2 presents the results achieved for each benchmark. Table 2 contains the average tour length for all the trials, tour length for the best achieved solution, optimal tour length, ratio of the average tour length to the optimal tour length, and the value of the constant D. It can be seen in Table 2, for 10-city problems the modified Hopfield network found the optimal solution with 100% success rate. Other works did not report such results using the classical Hopfield network. For greater city number problems, the solution quality is significantly better in comparison with the quality of results achieved using the classical Hopfield network in other works. For the first benchmark, the Hopfield network gave valid solutions for 15% [21], 3.75% [7], 36% [10,11] or near 100% [9,12] of trials. The network found the optimal solution for 0.5% [7], about 6% [9] of trials, did not find this solution [21] or the optimal solution was obtained [10-12]. The average tour length was about 25% [10,11], 15% [9] or about 10% to 20% [12] greater than the optimal solution length. For the second benchmark, the Hopfield network gave valid solutions for 45% [10] or near 100% [12] of trials, the average tour length was about 16% [10] or about 20% to 25% [12] greater than the optimal solution length, the optimal solution was not obtained [12]. For the third and fourth benchmarks, the average tour lengths of solutions obtained by using the Hopfield network were about 27% and 37% greater than the optimal solution length, respectively [9]. The optimal tours were not obtained [9]. The best tour lengths were equal to about 2.5 and 2.8, respectively [9].

It should be noted that a small value of the constant C in comparison with values of constants A and B in Eqs. 7-8 and the threshold equal to 0.5, according to which neurons are set to 0 or 1 in the stable state of the network, contribute significantly to better results. Table 3 presents results obtained for the constant $C = 5$ and the threshold equal to 0.9 (such values were used in many other works). The constant D in Eq. 7 was set to a value, for which almost all the solutions were valid, i.e., 90% to 100% of solutions, as before. It can be seen in Table 3 that the results for the constant $C = 5$ and the threshold equal to 0.9 are significantly worse than those, presented in Table 2.

In other works similar quality results to those achieved in this work were obtained only as a consequence of using much more complicated solutions in the Hopfield network [9-11,17].

Simulation Results With Auto-Tuning of the Constant D

In this section results obtained using the modified Hopfield network with auto-tuning of the constant D are presented. The simulations were performed for 100 10-city problems. For each problem 100 trials were executed. The coordinates of the cities used in these simulations were randomly generated in the unit square.

In Section 3 in the network stable state, all neurons with $V_{xi} \geq 0.5$ were set $V_{xi} = 1$, otherwise set $V_{xi} = 0$. If none of the neurons which were assigned to city x, had an output signal equal to 1, the solution was not valid because the position of city x in the tour was not determined. In this section the following new solutions are used. None of the neuron output signals is set to 0 or 1. In the stable state of the network, the position of the city is determined by the neuron which has the greatest output signal from among neurons assigned to this city. If a neuron of the greatest output signal has an output signal greater than 0.6 for each city, the constant *D* in Eq. 7 should be increased in the next trial. Otherwise, such a situation is the signal that the network is working on the border line between valid and invalid solutions. In that case, the constant *D* should be decreased in the next trial, otherwise the solution for that trial may be invalid. If the constant *D* is increased by a small value, network solutions are valid for almost all trials.

All problems were solved using the modified Hopfield network with auto-tuning of the constant *D*, which was simulated in software using a program written in C++. In this program solutions presented in Section 3 and new solutions described in this section were used. In simulations the same values of network parameters as in Section 3 were used. For each trial, the initial input signals of neurons were independently randomly chosen in the interval between $-0.1U_0$ and $0.1U_0$. The constants in Eqs. 7-8 were set $A = 5$, $B = 5$, and $C = 0.5$. The constant *D* in Eq. 7 was increased or decreased by ±0.1 after each trial. The initial value of the constant *D* was equal to 2. For each problem the optimal solution was found using the exhaustive search. For each problem the percentage of valid and optimal solutions and the average ratio of obtained tour lengths to the optimal tour length of the problem for all 100 trials were calculated. Table 4 presents the results achieved for all problems. Table 4 contains: the minimum, maximum, and arithmetic average percentage of valid solutions for all problems; the minimum, maximum, and average percentage of optimal solutions for all problems; the minimum and maximum ratio of the average tour length to the optimal tour length for all 100 trials of the problem, which were achieved for all problems. Ratios of the average tour length to the optimal tour length for all problems were arithmetically averaged and this result is also included in Table 4.

Results obtained in this section confirm the very good quality of the modified Hopfield network solutions for 10-city problems. The tour length for all problem trials is on average only about 0.8% greater than the optimal tour length. About 99% of all problem trials on average provided valid solutions. For 10-city problems the modified Hopfield network found an optimal solution with about 63% success rate on average. It should be noted that simulations were carried out with auto-tuning of the constant *D*. None of the network parameters was tuned for a specific 10-city problem. The constant *D* was increased or decreased according to the values of the greatest neuron output signal from among neurons assigned to each city. The threshold value of these greatest output signals, according to which the constant *D* is increased or decreased, can be equal to 0.5 but the average percentage of valid tours for all problem trials is then a bit smaller and equal to 95.34%. The average percentage of optimal tours is then equal to 64.20% and the ratio of the average tour length to the optimal tour length for all 100 trials of the problem, which is additionally averaged for all problems, is equal to 1.0059.

Conclusions

The modified Hopfield network was presented. The aim of this work was to examine the efficiency of this network for the TSP. The presented network is destined for hardware implementation, which could significantly decrease computation time in comparison with programs executed on sequential computers. Generally for 10-city problems the average tour length of obtained solutions is near the optimal tour length. Other works did not report such results using the classical Hopfield network. For greater city number problems, the solution quality is significantly better in comparison with the quality of results achieved using the classical Hopfield network in other works. The method of auto-tuning of the constant *D* in the cost function is presented. It should be noted that, thanks to that, none of the network parameters is tuned for a specific problem. It is a novel method because in other works such network parameters were tuned manually and so it is very useful method in real-time systems. This method ensures very good quality results for randomly generated instances of the 10-city TSP. The paper shows there is great potential in the Hopfield network. The author is working towards elaboration of the method for larger TSP problems because the main difficulty with such TSP problems is an extremely high number of network weights.

References

1. Brandt RD, Wang Y, Laub AJ, Mitra SK (1988) Alternative networks for solving the traveling salesman problem and the list-matching problem. In: Proceeding of the International Joint Conference on Neural Networks, vol. 2. pp 333-340
2. Dorigo M, Gambardella LM (1997) Ant colonies for the travelling salesman problem. BioSystems 43:73-81
3. Durbin R, Willshaw D (1987) An analogue approach to the travelling salesman problem using an elastic net method. Nature 326:689-691
4. Freeman JA, Skapura DM (1991) Neural networks: algorithms, applications, and programming techniques. Addison-Wesley, Reading
5. Hertz J, Krogh A, Palmer RG (1991) Introduction to the theory of neural computation. Addison-Wesley, Redwood City
6. Hopfield JJ, Tank DW (1985) “Neural” computation of decisions in optimization problems. Biological Cybernetics 52:141-152
7. Kamgar-Parsi B, Kamgar-Parsi B (1990) On problem solving with Hopfield neural networks. Biological Cybernetics 62:415-423
8. Kirkpatrick S, Gelatt CD Jr, Vecchi MP (1983) Optimization by simulated annealing. Science 220:671-680
9. Li SZ (1996) Improving convergence and solution quality of Hopfield-type neural networks with augmented Lagrange multipliers. IEEE Transactions on Neural Networks 7:1507-1516
10. Mańdziuk J (2000) Hopfield-type neural networks. Theory and applications. AOW EXIT, Warsaw (in Polish)
11. Mańdziuk J (2000) Optimization with the Hopfield network based on correlated noises: Experimental approach. Neurocomputing 30:301-321
12. Mańdziuk J (1996) Solving the travelling salesman problem with the Hopfield-type neural network. Demonstratio Mathematica 29:219-231
13. Michalewicz Z (1996) Genetic algorithms + data structures = evolution programs. Springer-Verlag, Berlin, New York
14. Michalewicz Z, Fogel DB (2004) How to solve it: modern heuristics. Springer, Berlin, New York
15. Nagórny Z, Kos A (2005) Optimization by using a modified Hopfield network. Electronics and Telecommunications Quarterly 51:255-275 (in Polish)
16. Pham DT, Karaboga D (2000) Intelligent optimisation techniques. Springer, London, New York
17. Qu H, Yi Z, Tang H (2006) Improving local minima of columnar competitive model for TSPs. IEEE Transactions on Circuits and Systems-I 53:1353-1362
18. Reinelt G (1995) TSPLIB 95, http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp/. University of Heidelberg, Department of Computer Science, Heidelberg
19. Tadeusiewicz R (1993) Neural networks. AOW, Warsaw (in Polish)
20. Van den Bout DE, Miller TK III (1989) Improving the performance of the Hopfield-Tank neural network through normalization and annealing. Biological Cybernetics 62:129-139

21. Wilson GV, Pawley GS (1988) On the stability of the travelling salesman problem algorithm of Hopfield and Tank. Biological Cybernetics 58:63-70
22. Zurada JM (1992) Introduction to artificial neural systems. West, St. Paul

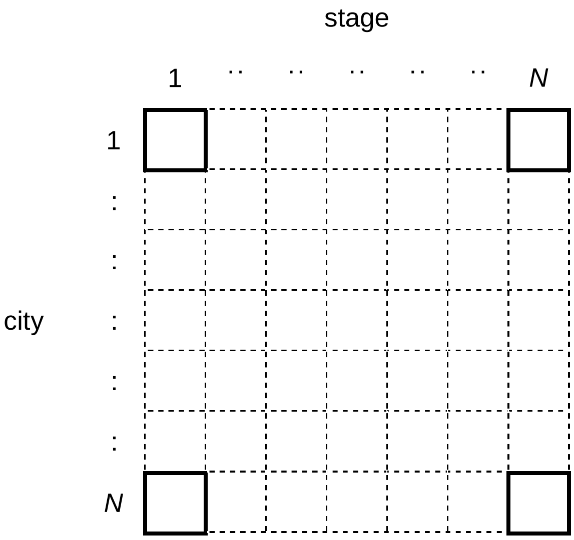


Fig. 1 The division of the network

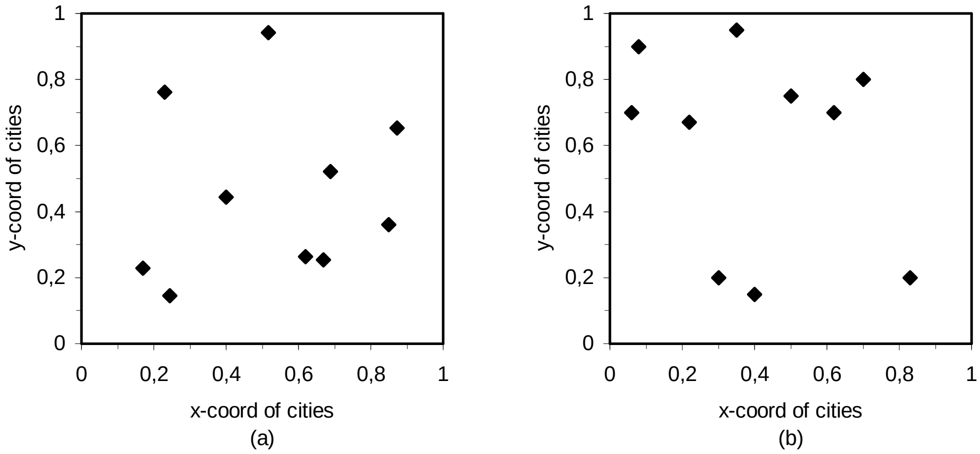
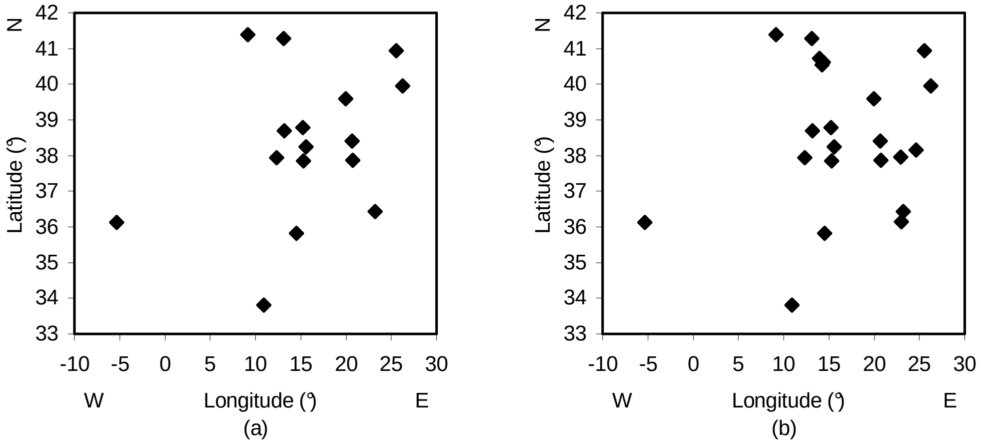


Fig. 2 The position of the cities in the unit square for: (**a**) 1st benchmark; (**b**) 2nd benchmark



(a)

(b)

Fig. 3 The position of the points on the earth for: (**a**) 3rd benchmark; (**b**) 4th benchmark

Table 1 The coordinates of the cities for 1st and 2nd benchmarks

Bench.	X, Y	City									
		1	2	3	4	5	6	7	8	9	10
1	X	0.4000	0.2439	0.1707	0.2293	0.5171	0.8732	0.6878	0.8488	0.6683	0.6195
	Y	0.4439	0.1463	0.2293	0.7610	0.9414	0.6536	0.5219	0.3609	0.2536	0.2634
2	X	0.06	0.08	0.22	0.30	0.35	0.40	0.50	0.62	0.70	0.83
	Y	0.70	0.90	0.67	0.20	0.95	0.15	0.75	0.70	0.80	0.20

Table 2 The results for all benchmarks

Bench.	Average tour length	Best tour length (Achieved)	Optimal tour length	Average length/ Optimal length	Constant *D*
1	2.6907	2.6907	2.6907	1	2.2
2	2.7818	2.7818	2.7818	1	2.4
3	2.5108	2.3811	2.3632	1.0625	0.9
4	2.6718	2.4522	2.4128	1.1073	0.9

Table 3 The results for the constant *C* = 5 and the threshold equal to 0.9

Bench.	Average tour length	Best tour length (Achieved)	Optimal tour length	Average length/ Optimal length	Constant *D*
1	3.4504	2.7693	2.6907	1.2823	2.2
2	3.7607	2.7818	2.7818	1.3520	1.8
3	3.3985	2.7978	2.3632	1.4381	2.0
4	3.9334	3.1585	2.4128	1.6302	1.8

Table 4 The results for all 10-city problems

Valid tours (%)			Optimal tours (%)			Average length/ Optimal length		
Min.	Max.	Average (100-problem average)	Min.	Max.	Average (100-problem average)	Min.	Max.	Average (100-problem average)
82	100	99.10	0	100	63.03	1	1.0537	1.0076