

# A New Approach to Solve the TSP using an Improved Genetic Algorithm

Qian Gao<sup>1,2</sup> · Young Im Cho<sup>1+</sup> · Su mei Xi<sup>1,2</sup>

<sup>1</sup> College of Information Technology, University of Suwon, San 2-2, Bongdam-eup, Hwaseong-si, 445-743, Korea

<sup>2</sup> School of Information, Shandong Polytechnic University, Jinan, 250353, China

gq@spu.edu.cn

ycho@suwon.ac.kr

## Abstract

Genetic algorithms are one of the most important methods used to solve the Traveling Salesman Problem. Therefore, many researchers have tried to improve the Genetic Algorithm by using different methods and operations in order to find the optimal solution within reasonable time. This paper intends to find a new approach that adopts an improved genetic algorithm to solve the Traveling Salesman Problem, and compare with the well known heuristic method, namely, Kohonen Self-Organizing Map by using different data sets of symmetric TSP from TSPLIB. In order to improve the search process for the optimal solution, the proposed approach consists of three strategies: two separate tour segments sets, the improved crossover operator, and the improved mutation operator. The two separate tour segments sets are construction heuristic which produces tour of the first generation with low cost. The improved crossover operator finds the candidate fine tour segments in parents and preserves them for descendants. The mutation operator is an operator which can optimize a chromosome with mutation successfully by altering the mutation probability dynamically. The two improved operators can be used to avoid the premature convergence. Simulation experiments are executed to investigate the quality of the solution and convergence speed by using a representative set of test problems taken from TSPLIB. The results of a comparison between the new approach using the improved genetic algorithm and the Kohonen Self-Organizing Map show that the new approach yields better results for problems up to 200 cities.

**Keywords** : Traveling Salesman Problem, Small tour segments set, Large tour segments set, Improved crossover operator, Improved mutation operator.

## 1. Introduction

TSP-Traveling Salesman Problem is a classic case of combinatorial optimization problem, which is studied widely by many researchers with application in various fields such as VLSI design<sup>[1]</sup>, hole-punching<sup>[2]</sup>, robot control and so on. The idea of TSP is: Given a number of cities and the distances of traveling from any city to any other city, find a minimum-length closed tour that visits each city once and only once and then returns to the starting city. It might look simple, but it is proved that the TSP is a NP-hard problem<sup>[3]</sup>. However, heuristic algorithms such as the Genetic Algorithm (GA) and Self-Organizing Map (SOM), could obtain near-optimal solutions within reasonable time.

Genetic Algorithm (GA) was proposed by Holland as an algorithm for probabilistic search, learning, and optimization, and is based in part on the mechanism of biological evolution and Darwin's theory of evolution<sup>[4]</sup>. Genetic algorithm is a powerful search tool, particularly when applied for combinatorial optimization problems. Genetic algorithm operates on "populations" of potential solutions, usually referred to as "chromosomes"<sup>[5][6]</sup>.

Each chromosome represents a set of parameters for a given problem. The chromosomes evaluate to represent the best solutions for a recombination process, which produces new chromosomes<sup>[7]</sup>. The new, improved chromosomes take the place of those with poorer solutions. In this way, each new generation becomes closer to the optimal solution. This continues for many generations until the termination condition is met. Mutations and different combining strategies ensure that a large range of search space is discovered<sup>[8]</sup>.

However, two major problems arise with the implementation of an efficient GA: on one side, the premature convergence to local optima and on the other the requirements for the GA search of long times in order to reach an optimal or a good suboptimal solution.

In 1975 Teuvo Kohonen introduced a new type of neural network that used competitive and unsupervised learning. This approach is based on WTA (Winner Takes All) and WTM (Winner Takes Most) algorithms. The most basic competitive learning algorithm is WTA. When input vector (a pattern) is presented, a distance to each neuron's synaptic weights is calculated. The neuron whose weights are most correlated to current input vector is the winner. The result of this competition is the activation of only one output neuron at a given moment<sup>[9]</sup>. The purpose of Kohonen SOM is to capture the topology and probability distribution of input data.

A competitive training algorithm is used to train the neural network. In this training mode, not only the winning neuron is

---

Manuscript received Nov. 1, 2011; revised Nov. 29, 2011; accepted Dec. 4;

<sup>+</sup>Corresponding author

allowed to learn, but some neurons within a predefined radius from the winning neuron are also allowed to learn with a decreasing learning rate as the cardinality distance from the winning neuron increases. During the training procedure, synaptic weights of neurons are gradually changed in order to preserve the topological information of the input data when introduced to the neural networks [10].

Some experiments show that the conventional genetic algorithms can solve TSP to some extent, but may not produce optimal solutions, and even if they do, they will not be within a reasonable time, and also some answers may get worse and worse with the enlargement of problem.

This paper intends to find a new approach that uses Improved Genetic Algorithm to solve the TSP, and compare with well known heuristic method: Kohonen Self-Organizing Map(SOM).

The proposed Improved Genetic Algorithm (IGA) improves the two operators of the conventional genetic algorithms, and since most of the tour segments are too long to appear in short tours, the proposed new approach divides all the tour segments into two separate sets: the small tour segments set and the large tour segments set.

The new approach is tested by means of different datasets of symmetric TSP from TSPLIB [11], and the experiment demonstrates that the new approach that uses Improved Genetic Algorithm can provide good results within reasonable time.

## 2. Materials and Methods

### 2.1. Small Tour segments Set (STS) and Large Tour segments Set (LTS)

Conventional genetic algorithms start from a population of chromosomes at random selection as the first generation of candidate solutions and evolve toward better solutions by producing a new generation of chromosomes using crossover and mutation operators based on the population of the previous generation. But if the chromosomes of the first generation are generated in low tour cost instead of random selection at the beginning, it will be more conducive to preserve worthy subtours for offspring.

TSP has some geometric properties that are problem specific knowledge. In order to start from a population of chromosome with worthy subtours for offspring, the properties can be utilized to improve the search process of the GA. As is mentioned before, most of the tour segments are too long to appear in short tours, and this information can guide the GA to focus more on small tour segments than large ones. Thus, all the tour segments are divided into two separate sets: the Small Tour segments Set (STS) and the Large Tour segments Set (LTS). The tour segments in the STS would have higher priorities than those in the LTS.

The main idea of this is that the small tour segments are local optimal tour segments, so that the small tour segments might be worthwhile to be selected to create a lower cost tour than LTS.

Therefore, the number of the small tour segments in a tour is maximized to increase the probability of producing a tour of lower cost.

#### 2.1.1. How to divide tour segments into STS and LTS

According to the analysis, almost half of the tour segments in an optimal tour of the benchmark instances in TSPLIB are local optimum. Therefore, the local optimal tour segment can be utilized to identify whether tour segments belong to STS or LTS.

The tour segments  $C_{ij}$  ( $i, j \in N$ ,  $N$  is the set of all cities) which are local optimal tour segments with the lower costs of all the tour segments emanating from city  $i$  are called Small tour segments, and belong to the STS. Others are called Large tour segments belonging to the LTS.

#### 2.1.2. How to use STS and LTS to control the global optimum searching of a GA

An example is taken here to explain how to use the STS and LTS to select tour segments with low cost to form a complete tour as the first generation. Consider STS are as Fig.1 (here, the numbers in the circle indicate the costs):

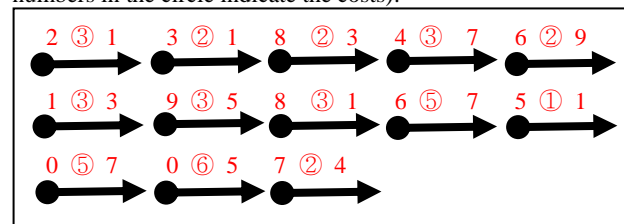


Fig. 1. Small Tour Segments Set

Suppose  $C_{83}$ ,  $C_{51}$ ,  $C_{95}$ ,  $C_{69}$ ,  $C_{07}$ ,  $C_{74}$  have been inserted into the tour as Fig.2:

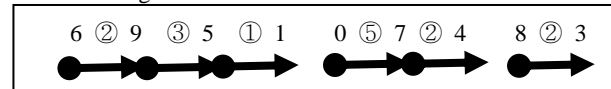


Fig. 2. The selection process 1

Since each city can only be visited once and only once, so  $C_{21}$ ,  $C_{31}$ ,  $C_{47}$ ,  $C_{13}$ ,  $C_{81}$ ,  $C_{67}$ ,  $C_{05}$  will not be inserted to the tour, so as that these subtours belonging to STS will be ignored in the following iterations. Nevertheless, the above tour is not a complete tour. In order to complete the tour, we must choose the tour segments from the LTS.

A key issue is how to decide which tour segments in LTS are worthwhile to be inserted into the tour. Here one can use an objective function to evaluate whether or not a large tour segment is a good tour worthwhile to be inserted into the tour. The objective function can evaluate the quality of tour segments. The objective function  $f(C_{ij})$  is defined as Equation (1):

$$f(C_{ij}) = \cos t(C_{ij}) / \min_{x \in N} \cos t(C_{ix}) \quad i, j \in N \quad (1)$$

Where,  $C_{ij}$  (from city  $i$  to city  $j$ ) belongs to LTS,  $C_{ix}$  (from city  $i$  to city  $x$ ) belongs to STS,  $N$  is the set of all cities,  $\cos t(C_{ij})$  is the weight between city  $i$  and city  $j$  (such as the numbers in the circle in Fig.1). Since the  $C_{ij}$  has less variation between it and  $C_{ix}$ , the lower the value of  $f(C_{ij})$ , the higher the

possibility of an LTS  $C_{ij}$  being a good tour segment and lower probability to be replaced in the future.

In addition, when the subtour from city  $k$  to city  $l$  has the same value of  $f(\cdot)$  with that from city  $m$  to city  $n$ , which one should be chosen? Suppose the  $C_{kl}$  and  $C_{mn}$  have the same value of  $f(\cdot)$ , and also have the same costs, then the subtour can be selected randomly. But if the  $C_{kl}$  and  $C_{mn}$  have the same value of  $f(\cdot)$ , and the  $\text{cost}(C_{kl})$  is smaller than the  $\text{cost}(C_{mn})$ , then  $C_{kl}$  should have be a more competitive candidate of good tour segment than  $C_{kl}$ . Put differently, when two subtours that belong to the LTS with different costs have the same value of  $f(\cdot)$ , the subtour which has lower cost would be selected to form a complete tour.

In this circumstance, another objective function can be used to evaluate whether or not a large tour segment is a good tour worthwhile to be inserted into the tour. The objective function  $f'(C_{ij})$  is defined as Equation (2):

$$f'(C_{ij}) = \text{cost}(C_{ij}) / (\min_{x \in N} \text{cost}(C_{ix}))^2 \quad i, j \in N \quad (2)$$

The higher the value of  $f'(C_{ij})$ , the higher the possibility of an LTS  $C_{ij}$  being a good tour segment and lower probability to be replaced in the future.

Consider LTS are as Fig.3 (here, the numbers in the circle indicate the costs, and the numbers under the arrows indicate the  $f(C_{ij})$ ):

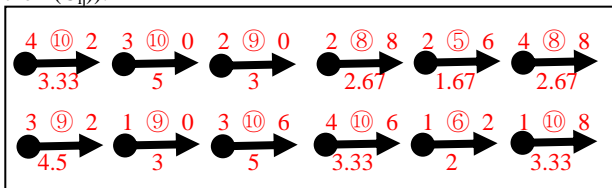


Fig.3. Large Tour Segments Set

The next thing to do is to select the lowest value of  $f(C_{ij})$ . If inserting the subtour from city  $i$  to city  $j$  will not result in a cyclic subtour, then the  $C_{ij}$  will be inserted into the tour. The selection process is as follows:

a. Select the lowest value of  $f(C_{26})$  from the LTS, and because it will not result in a cyclic subtour, so  $C_{26}$  can be chosen to be inserted into the tour, as is shown by Fig.4.

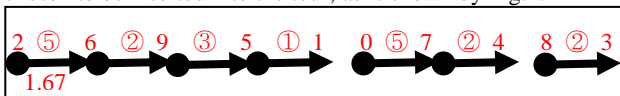


Fig.4. The selection process 2

b. Select the lowest value of  $f(C_{12})$  from the LTS, but if we insert  $C_{12}$  into the tour, a cyclic will be formed in a subtour, so that the  $C_{12}$  will not be chosen, as is shown by Fig.5.

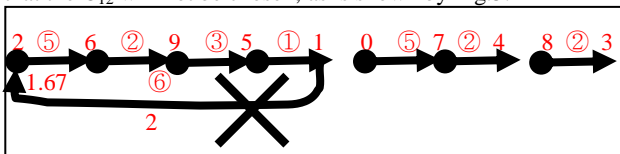


Fig.5. The selection process 3

c. For the same reason, the  $C_{28}$  will not be chosen, and the lowest value of  $f(C_{48})$  will be chosen from LTS and be inserted into the tour, as is shown by Fig.6.

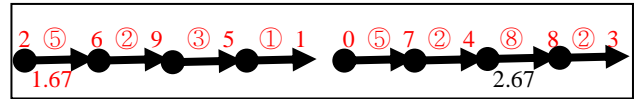


Fig.6. The selection process 4

d. For the same reason, the  $C_{20}$  will not be chosen, and the lowest value of  $f(C_{10})$  will be chosen from LTS and be inserted into the tour, as is shown by Fig.7.

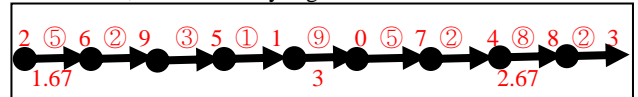


Fig.7. The selection process 5

e. Likewise, the  $C_{42}$ ,  $C_{46}$ ,  $C_{18}$  will not be chosen, and the lowest value of  $f(C_{32})$  will be chosen from LTS and be inserted into the tour, as is shown by Fig.8, then the tour completed.

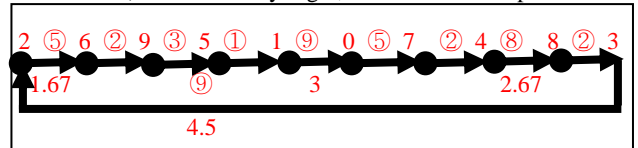


Fig.8. The selection process 6

The whole process to form a complete tour as the first generation can be described as is shown by Fig.9.

## 2.2. Improved crossover operator

Crossover is the most important operation of a GA because in this operation, characteristics are exchanged between the individuals of the population.

During the course of the conventional genetic algorithms, if crossover operator is mainly carried on, the population is easily plunged into premature convergence because of the deficiency of diversity. But this problem can be got over by the improved crossover operator and the improved mutation operator.

Since the  $C_{ij}$  in the STS is more likely to become a good tour segment than the  $C_{ik}$  in the LTS for a local optimum of all the tour segments emanating from city  $i$ , to obtain a low cost tour, the priority of preserving an STS for the offspring is thus higher than that of preserving an LTS.

The main idea of the improved crossover operator is to turn to different means to search for better tours. Before the crossover, divide the tour into two parts: one part is the tour segments belonging to STS called section1; the other part is the tour segments belonging to LTS called section2. Then flip section1 and section2 respectively and swap section1 and section2.

For example, consider parent tours:

Parent1 (9 8 7 6 5 4 3 2 1) Parent2 (6 8 7 1 4 5 9 2 3)

Suppose that the section1 of parent1 is (9 8 7 6), the section2 of parent1 is (5 4 3 2 1), the section1 of parent2 is (6 8 7 1), the section2 of parent2 is (4 5 9 2 3). Then the section1 of parent1 will be flipped and become (6 7 8 9). Next, these cities are removed from parent2, and parent2 will be (1 4 5 2 3), and  $C1$  and  $C2$  will be:  $C1$  (6 7 8 9 1 4 5 2 3),  $C2$  (1 4 5 2 3 6 7 8 9). The same procedure is done for parent2, so that the flipped section1 of parent2 is (1 7 8 6), and that of parent1 is (9 5 4 3 2), and the produced offspring  $C3$  and  $C4$  will be:  $C3$  (1 7 8 6 9 5 4 3 2),  $C4$  (9 5 4 3 2 1 7 8 6).

We can see that different methods can change the structure of the tour by exchanging the two sections of two parents; hence different characteristics can be inherited from their parents.

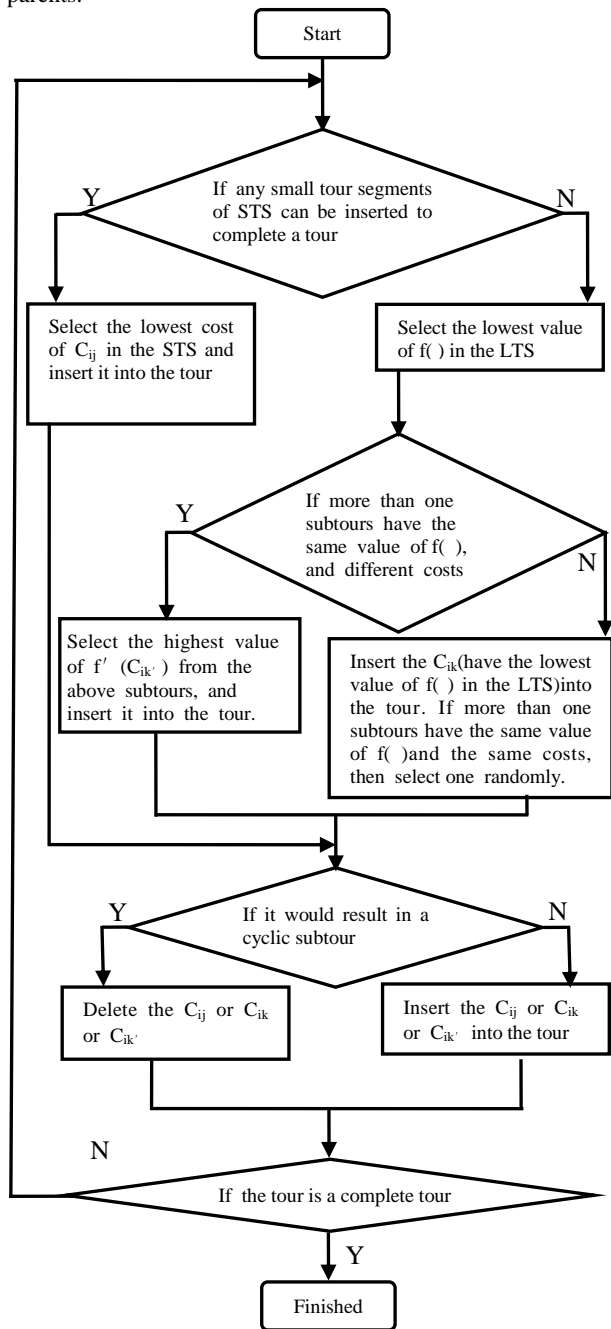


Fig.9. The flowchart to form a complete tour

2.3. Improved mutation operator

Mutation options specify how the genetic algorithm makes small random changes in the individuals in the population to create mutation children. Mutation provides genetic diversity and enables the genetic algorithm to search a broader space.

In addition to using improved crossover operator, mutation operator can also keep the population diversity. However,

mutation itself is a blindfold operation, which is relatively effective at the earlier stage of evolution. At the later stage of evolution, the probability to optimize a chromosome with mutation successfully seems infinitesimal, which requires the improvement of the mutation operation.

The main idea of the improved mutation operator is to alter the mutation probability dynamically through the following dynamic function as Equation (3):

$$MP = MP - 0.1 * MP * N / M \tag{3}$$

Here the MP indicates the probability of the mutation; the N indicates the current generation, and the M indicates the number of the generation when the IGA to TSP is ended. If a random number between 0 and 1 (random (0, 1)) is smaller than MP, then the mutation would be carried on.

3. Results

The simulation experiments are executed to investigate the quality of the solution and convergence speed for various benchmark problems that have different numbers of nodes.

Two types of tests were administered: using city sets taken from TSPLIB and some optimal solutions are already there; using randomly chosen cities.

In the simulation study, for a fair comparison with the other researchers' work, it is reasonable to compare the proposed IGA with available SOM research using the same dataset of TSP instances in the same area. By using the same dataset of TSP instances, the number of iterations in the test is the same, and the distance between two cities is calculated through the Euclidian norm.

In order to be better understood, the results were presented with tables and visual graphics. Table1 shows the TSP instances that have been used.

Tab.1. TSP instances used for computational experiments

Instances	No. of cities	Optimum length
Bier	127	118,282
kroA200	200	29,368
Lin105	105	14,379
Pr107	107	44,303
ed100	100	7,910
Pcb442	442	50,778
Rat195	195	2,323
st70	70	675
Pr152	152	73,682

The relative error, which indicates how close the solution is to the known global solution, is considered in the performance measurement and calculated by Equation (4):

$$\text{Relative Error (\%)} = ((RL - OL) / OL) \times 100 \tag{4}$$

Where RL is the Results from the IGA or SOM under each problem, OL is the Optimum length. Table 2 shows solution already known, results from the IGA and SOM, and relative error for the problems with the same dataset of TSP instances.

Tab.2. Results from the IGA and SOM, and Relative Error

Instances	Optimum length	IGA	SOM	relative error of IGA	relative error of SOM
Bier127	118,282	121,321	132,537	3%	12%
kroA200	29,368	30,947	30,438	5%	4%
Lin105	14,379	14,985	15,281	4%	6%
Pr107	44,303	45,921	47,231	4%	7%
ed100	7,910	7,998	8,103	1%	2%
Pcb442	50,778	53,964	51,237	6%	1%
Rat195	2,323	2,498	2,536	8%	9%
st70	675	683	701	1%	4%
Pr152	73,682	75,159	76,326	2%	4%

Fig.10. shows the results obtained from the application of the IGA to some TSP instance and comparisons with the results presented in SOM.

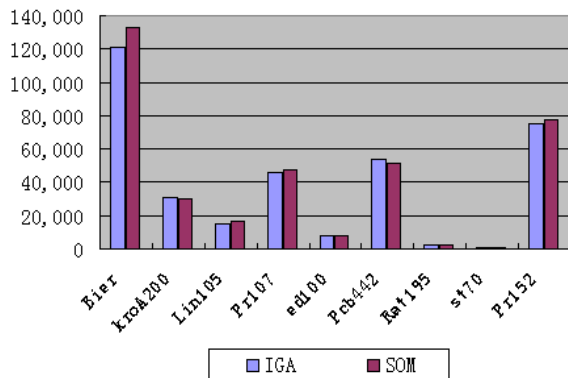


Fig.10. Results of the IGA and SOM

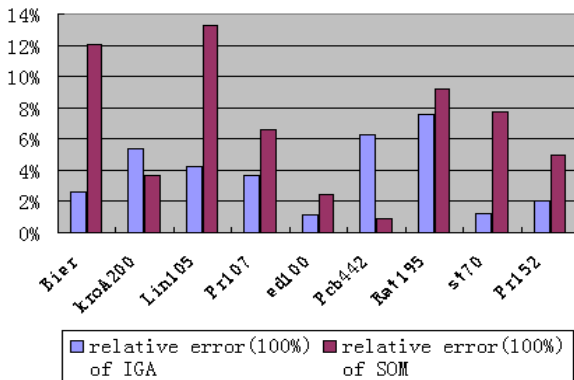


Fig.11. The relative errors of IGA and SOM

Fig.11. shows the relative errors according to each TSP instance optimum length (specified in Table 1) for IGA and SOM algorithm analyzed.

In general, the relative errors of the SOM are higher than the IGA for instances with up to 200 cities. However, the SOM could find a solution with low relative error for problems of larger amount of cities (442 and more).

#### 4. Discussion

Genetic algorithm is a powerful search tool, particularly when applied for combinatorial optimization problems and it works on “populations” of potential solutions, which are usually referred to as “chromosomes”<sup>[5][6]</sup>.

In order to solve TSP more efficiently, the proposed new approach consists of three strategies: two separate tour segments sets, the improved crossover operator, the improved mutation operator.

The two separate tour segments sets are construction heuristic which produces tours with the lowest costs. The improved crossover operator finds the candidate fine tour segments in parents and preserves them for descendants. The mutation operator is an operator that can successfully optimize a chromosome with mutation by altering the mutation probability dynamically. The two improved operators can be used to avoid the premature convergence.

There are a variety of types of Self-Organizing Maps; however, they all share a common characteristic, the ability to assess the input patterns presented to the networks, organize themselves to learn, on their own, based on similarities among the collective set of inputs, and categorize them into groups of similar patterns <sup>[12]</sup>.

To solve TSP problem, the Kohonen Self-Organizing Map<sup>[13]</sup> which I used to compare with the IGA adopted the well known 2opt algorithm to obtain the local optimum.

In short, the SOM does not need a target output. That is to say, SOM is an unsupervised network, but the Genetic Algorithms is a way of solving problems by mimicking the same processes Mother Nature uses, and it needs a target output. However, the SOM Algorithms and the Genetic Algorithms are all good at optimization of the problem.

#### 5. Conclusion

Genetic Algorithm seeks the solution of a problem in the form of strings of numbers, by applying operators such as recombination and mutation. It innovates in self-adaptive inversion operators to diversify the individuals efficiently and avoid premature convergence to some extent. Then it can find out good answers rapidly. But during the course of the conventional genetic algorithms, if crossover operator is mainly carried on, the population is easily plunged into premature convergence because of the deficiency of diversity, thus some improved methods must be taken.

In this paper, a new approach using Improved Genetic Algorithm is presented for solving the TSP. To improve the search process for the optimal solution, the proposed new approach uses Small Tour segments Set (STS) and Large Tour segments Set (LTS) to create a lower cost tour. Furthermore, the proposed Improved Genetic Algorithm uses improved crossover operator and improved mutation operator to avoid the premature convergence.

Simulation experiments are executed to investigate the quality of the solution and convergence speed using a

representative set of test problems taken from TSPLIB. Considering the optimum length and the relative error in terms of quality of solution, the obtained results of a comparison between the IGA and the SOM show that IGA has rapidly converged to a solution with up to 200 cities. However, the SOM could find a solution with low relative error for the problem of larger amount of cities (442 and more).

Besides, more priority levels for classifying the tour segments could be considered in future studies, and in order to speed up ratio and enlarge the scale of city, maybe parallel virtual machine (PVM) should be used in the future.

## References

- [1] K. Kurbel, B. Schneider, and K. Singh, "Solving optimization problems by parallel recombinative simulated annealing on a parallel computer—an application to standard cell placement in VLSI design," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol.28, no.3, pp.454–461, 1998.
- [2] G. Reinelt, "TSPLIB—A Traveling Salesman Problem Library," *ORSA Journal on Computing*, vol. 3, no. 4, pp. 376–384, 1991.
- [3] P. Larranaga, C. M. H. Kuijpers, R. H. Murga, I. Innza and S. Dizdarevic, "Genetic Algorithms for the Travelling Salesman Problem: A Review of Representations and Operators," *SI: A rtif. Intell.*, vol. 13, no. 2, pp.129-170, 1999.
- [4] J. Holland, "Adaptation in Natural Artificial Systems," *the University of Michigan Press (Second edition; MIT Press)*, 1992.
- [5] M. Yoshikawa, T. Fujino, and H. Terai, "A Novel Genetic Algorithm Routing Technique in 3-Dimentional Space," *Proceedings of the 10th World Multiconference on Systemics, Cybernetics and Informatics*, vol.1, pp.70-75, 2006.
- [6] S. Nakaya, Koide and T. Wakabayashi, S, "An adaptive genetic algorithm for VLSI floor planning based on sequence-pair," *Proc. IEEE ISCAS*, vol.3, pp.65-68, 2000.
- [7] V. Ravi, B.S.N. Murty, and J. Reddy, "Nonequilibrium simulated-annealing algorithm applied to reliability optimization of complex systems," *IEEE Transactions on Reliability*, vol. 46, no.2, pp.233-239, 1997.
- [8] K. A. Smith, "Neural networks for combinatorial optimization: A review of more than a decade of research," *INFORMS J. on Computing*, vol. 11, no.1, pp. 15–34, 1999.
- [9] Kohonen, T, "The self-organizing map," *Proceedings of the IEEE*, vol.78, no.9, pp.74-90, 1990.
- [10] T. Kohonen, S. Kaski, K. Lagus, J. Salojärvi, V. Paatero, and A. Saarela, "Organization of a massive document collection," *IEEE Trans. Neural Networks*, vol. 11, pp. 574–585, May 2000.
- [11] TSBLIB, <http://www.iwr.uni-eidelberg.de/iwr/comopt/soft/TSPLIB95/TSPLIB.html>. [Online] [Cited: 12 22, 2008.
- [12] A. S. Nissinen and H. Hyotyniemi, "Evolutionary training of behavior based self-organizing map," in *Proc. 1998 IEEE Int. Joint Conf. Neural Networks*, vol. 1, pp. 660–665, 1998.
- [13] Yanping Bai, Wendong Zhang, and Zhen Jin, "An new self-organizing maps strategy for solving the traveling salesman problem," *Chaos, Solitons and Fractals*, vol.28, pp.1082-1089, 2006.
- [14] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biol. Cybern.*, vol. 43, no. 2, pp. 59–69, 1982.
- [15] S. Lin and B.W. Kernighan, "An Effective Heuristic Algorithm for the Traveling Salesman Problem," *Operations Research*, vol. 21, no.2, pp. 498-516, 1973.



**Qian Gao**

2001. Bachelor, Shandong University of Science and Technology  
2008. Master, Shandong University  
Current Ph.D. Course student, University of Suwon  
Research Area: Neural Networks, Genetic Algorithm

E-mail : gq@spu.edu.cn



**Young Im Cho**

1988. Bachelor, Korea University  
1990. M.Sc. Korea University  
1994. Ph.D. Korea University  
Current. Professor at Univ. of Suwon  
Dept. of computer science  
Research Area: Artificial Intelligence, Agent System, Ubiquitous System etc.  
E-mail : ycho@suwon.ac.kr



**Su mei Xi**

2001 Bachelor, Shandong University of Science and Technology  
2009 Master, Shandong University  
Current Ph.D. Course student, University of Suwon  
Research Area: Artificial intelligence  
E-mail : xsm@spu.edu.cn