

# Immune-Genetic Algorithm for Traveling Salesman Problem

Jingui Lu<sup>1</sup> and Min Xie<sup>2</sup>  
*Nanjing University of Technology,  
 P. R. China*

## 1. Introduction

The Traveling Salesman Problem (TSP), first formulated as a mathematical problem in 1930, has been receiving continuous and growing attention in artificial intelligence, computational mathematics and optimization in recent years. TSP can be described as follows: Given a set of cities, and known distances between each pair of cities, the salesman has to find a shortest possible tour that visits each city exactly once and that minimises the total distance travelled.

The mathematical model of TSP is described below:

Given a set of cities  $C = \{C_1, C_2, C_3 \dots C_n\}$ , the distance of each pair of cities is  $d(C_i, C_j)$ . The problem is to find a route  $(C_1, C_2, C_3 \dots C_n)$  that visits each city exactly once and makes

$$\sum_{i=1}^n d(C_i, C_{i+1}) + d(C_n, C_1) \text{ to have a minimum value.}$$

TSP is the problem of the permutation of  $n$  cities. For  $n$  cities, there should be  $n!$  different permutations. For the symmetric TSP, each route has two different ways to represent. Therefore, the size of its search space is:  $S = n!/2n = (n-1)!/2$ . As TSP is an 'NP-hard' problem, researchers all over the world try to solve the problem with various algorithms. Genetic Algorithm (GA), with the advantages of robustness, flexibility and versatility, has been widely studied to solve large-scale combinatorial and optimization problems. However, Genetic Algorithm has some significant drawbacks, for instance, the pre-mature convergence of computations, the poor use of system information during computational evolutions, expensive computation from evolutionary procedures and the poor capability of "local" search (Potvin, 1996) (Jin et al., 1996) (Wei & Lee, 2004) (Lu et al., 1996).

Although TSP itself seems very simple, as the number of visited cities increases, the computation of the problem can be extremely time-consuming (in the order of exponential growth) or even results in no optimal solution in the worst case. Developing effective algorithms for the TSP has long been a topic of interest in both academic research and engineering applications, ranging from transportation optimization to the sequencing of jobs on a single machine.

The methods commonly employed to solve the TSP include simulated annealing algorithm, artificial neural networks, tabu search algorithm, genetic algorithm (GA), and so on. Each method has different advantages and disadvantages. For example, GA combines many positive features (such as robustness, flexibility, and versatility), leading to its widespread

applications in engineering optimization. However, GA also has some significant drawbacks, for instance, the pre-mature convergence of computations, the poor use of system information during computational evolutions, expensive computation from evolutionary procedures, and the poor capability of local search.

The immune system, which is made up of special organs, tissues, cells and proteins, is the body's defence against infectious organisms and other invaders (Liu, 2009). The immune system detects and attacks antigens that invade the body through different types of lymphocytes. Artificial immune systems are adaptive systems inspired by the functions, principals and models of the vertebrate immune system. When artificial immune systems are attacked, the immune mechanisms are started to guarantee the basic functions of the whole intelligent information system. Researches on artificial immune systems aim to set up engineering models, algorithms and advanced intelligent information system through intensive study on the information processing mechanisms of biological immune systems.

In the 1970s, Jerne first propounded the hypothesis of the immune network system and founded the basic theories of the artificial immune systems, Jerne's idiotypic network model. Perelson studied on a number of theoretical immune network models proposed to describe the maintenance of immune memory, which accelerated the development of artificial immune systems in computer science. In 1986, Farmer built a dynamic model of the immune system and brought in the concept of learning. Farmer's work contributed to turning artificial immune systems to practical application. One important aspect of the research on artificial immune systems is to develop effective learning and optimization algorithms. Immune algorithms are one of heuristic search algorithms inspired by immune principals. In 1990, Bersini put immune algorithms into practice for the first time. By the end of the 20th century, Forrest et al. started to apply immune algorithms to computer security field. At the same time, Hunt et al. began to use immune algorithms in machine learning.

Immune algorithms (IAs), mainly simulate the idea of antigen processing, including antibody production, auto-body tolerance, clonal expansion, immune memory and so on. The key is the system's protection, shielding and learning control of the attacked part by invaders. There are two ways considered to design an immune algorithm: one is to abstract the structure and function of the biological immune system to computational systems, simulating immunology using computational and mathematical models; the other is to consider whether the output of the artificial immune systems is similar with that of the biological immune system when the two systems have similar invaders. The latter doesn't focus on the direct simulation of the process, but the data analysis of the immune algorithm. As the immune system is closely related to the evolutionary mechanism, the evolutionary computation is often used to solve the optimization problem in immune algorithms. The research on artificial immune systems lays a foundation for further study on engineering optimization problems. On the one hand, it aims to build a computer model of biological immune system, which contributes to the study of the immune system operation. On the other hand, it supplies an effective way to solve many practical problems.

Immune Algorithms inspired by biological immune mechanism, can make full use of the best individuals and the information of the system, and keeps the diversity of the population. In the optimization process, Immune Algorithms take the useful ideas of existing optimization algorithms, combine random search with deterministic changes, reduce the impact of the random factors to the algorithm itself and can better eliminate the premature convergence and oscillation.

One kind of immune algorithms is immunity based neural method, such as the neuro-immune network presented in (Pasti & De Castro, 2006), which is a meta-heuristics for solving TSP based on a neural network trained using ideas from the immune system. In addition, an immune-inspired self-organizing neural network proposed by Thiago is showed to be competitive in relation to the other neural methods with regards to the quality (cost) of the solutions found (Thiago & Leandro, 2009).

Combining GA with immune algorithms is another kind of method in TSP solving, such as an immune genetic algorithm based on elitist strategy proposed in (Liang & Yang, 2008). A genetic algorithm based on immunity and growth for the TSP is also showed to be feasible and effective, in which a reversal exchange crossover and mutation operator is used to preserve the good sub tours and to make individuals various, an immune operator is used to restrain individuals' degeneracy, and a novel growth operator is used to obtain the optimal solution with more chances (Zeng & Gu, 2007).

Besides, Clonal Selection Algorithm is widely used to solve TSP. For example, a Hypermutation Antibody Clone Selection Algorithm (HACSA) shows the advantage of enhancing the local search performance of the antibody in solving TSP (Du et al., 2009). A novel Clonal Selection Algorithm(CSA), which extends the traditional CSA approach by incorporating the receptor editing method, is proved to be effective in enhancing the searching efficiency and improving the searching quality within reasonable number of generations (Gao et al., 2007).

Moreover, a number of improved artificial immune algorithms are studied and show the capability for TSP solutions. For example, an immune algorithm with self-adaptive reduction used for solving TSP improves the probability that it finds the global optimal solution by refining the reduction edges which gradually increase in the number and enhance in the forecasting accuracy (Qi et al., 2008).

To partially overcome the above-mentioned shortcomings of GA, an immune-genetic algorithm (IGA) is introduced in this book chapter, and then an improved strategy of IGA for TSP is also discussed. Section 3 is related to a selection strategy incorporated into the conventional genetic algorithm to improve the performance of genetic algorithm for TSP. The selection strategy includes three computational procedures: evaluating the diversity of genes, calculating the percentage of genes, and computing the selection probability of genes. The computer implementation for the improved immune-genetic algorithm is given in section 4, and finally the computer numerical experiments will be given in this book chapter.

## **2. The immune-genetic algorithm**

### **2.1 Immune algorithms**

Immune Algorithms can be divided into Network-based immune algorithm and Population-based algorithm.

Immune Network theory was first proposed by Jerne in 1974. Currently the most widely used is Jerne's network based thinking: immune cells in the immune system link each other through the mutual recognition. When an immune cell recognizes an antigen or another immune cell, it is activated. On the other hand, the immune cell is inhibited when it is recognized by other immune cells.

There are two kinds of Immune Network models: the continuous model and the discrete model. The continuous Immune Network model is based on ordinary differential equations. The typical models include the model proposed by Farmer et al. in 1986 and the model

proposed by Varela and Coutinho in 1991. These models have been successfully applied to continuous optimization problems, automatic navigation system and automatic control field. However, the equations of continuous immune network model can not always be solved and usually it needs numerical integration to study the behavior of the system. To make up the drawbacks of continuous immune network model, the discrete immune network model is produced, which is based on a set of differential equations or an adaptive iteration. Population-based Immune Algorithm mainly includes Negative Selection Algorithm and Positive Selection Algorithm. Negative Selection Algorithm, proposed by Forrest et al. from the University of Mexico, is a kind of selection Algorithms used to test data change. The algorithm embodies the ideas of the ideological principles of negative selection (Ge & Mao, 2002). The immune system works out mainly by successfully detecting abnormal changes of the system. Negative selection refers to the identification and deletion of self-reacting cells, that is, T cells that may select for and attack self tissues (Forrest et al., 1994). The immune system removes the immune cells that response to autologous cells to realize self-tolerance through Negative selection algorithm. There are mainly two procedures contained in Negative selection algorithm: tolerance and detection. The task in tolerance procedure is to produce mature detector. In the detection phase, the detector detects the protected system. Negative selection Algorithm does not directly use self-information, but generates testing subset by self-assembly through Negative selection. The algorithm is robust, parallel, distributed detected and easy to implement. However, as its computational complexity increases exponentially, Negative selection algorithm is not conducive to handling complex problems. Positive selection algorithm is very similar to Negative selection algorithm. But it works contrary to the Negative selection algorithm. Negative selection algorithm removes the self-reacting immune cells, while Positive selection algorithm keeps them. Besides, Clonal Selection Algorithm is also a widely used immune algorithm. It is inspired by the clonal selection theory of acquired immunity that explains how B and T lymphocytes improve their response to antigens over time. The algorithm solves problems through the mechanisms of cell cloning, high-frequency variation, clonal selection and dying. It is high parallel and can be used in machine learning, pattern recognition and optimization domains. Standard Clonal selection algorithm achieves population compression and ensures the quality of antibody population in the optimal solution through local search. But Positive selection algorithm requires the system to be static. To make up that defect, Kim and Bentley proposed Dynamic Clonal algorithms in 2002, mainly for Network Intrusion Detection, to meet the real-time network security requirements.

## 2.2 The immune-genetic algorithm

The Immune-Genetic Algorithm (IGA) is an improved genetic algorithm based on biological immune mechanisms. In the course of immune response, biological immune system preserves part of the antibodies as memory cells. When the same antigen invades again, memory cells are activated and a large number of antibodies are generated so that the secondary immune response is more quickly than the initial response. In the meanwhile, there are mutual promotion and inhibition between antibodies. Therefore, the diversity and immune balance of the antibodies are maintained. That is the self-regulatory function of the immune system. The Immune-Genetic Algorithm simulates the process of adaptive regulation of biological antibody concentration, in which the optimal solution of the objective function corresponds to the invading antigens and the fitness  $f(X_i)$  of solution  $X_i$  corresponds to the antibodies produced by the immune system. According to the

concentration of antibodies, the algorithm adaptively regulates the distribution of the search direction of solutions and greatly enhances the ability to overcome the local convergence.

In general, the Immune-Genetic Algorithm includes:

1. Antigen definition: Abstract the problem to the form of antigens which the immune system deals with and the antigen recognition to the solution of problem.
2. Initial antibody population generation: The antibody population is defined as the solution of the problem. The affinity between antibody and antigen corresponds to the evaluation of solution, the higher the affinity, the better the solution.
3. Calculation of affinity: Calculate the affinity between antigen and antibody.
4. Various immune operations: The immune operations include selection, clone variation, auto-body tolerance, antibody supplementation and so on. The affinity and diversity are usually considered to be the guidance of these immune operations. Among them, select Options usually refer to the antibody population selected from the population into the next operation or into the next generation of the immune antibody population. Clone variation is usually the main way of artificial immune algorithm to generate new antibodies. Auto-body tolerance is the process of judging the rationality of the presence of the antibodies. Antibody supplementation is the accessorial means of population recruitment.
5. Evaluation of new antibody population: If the termination conditions are not satisfied, the affinity is re-calculated and the algorithm restarts from the beginning. If the termination conditions are satisfied, the current antibody population is the optimal solution.
6. Evolution of the antibody using standard Genetic Algorithm: crossover and mutation.

This model makes the immune system learn to identify the antibodies that are helpful to the antigen recognition. Moreover, the introduction of fitness further improves the immunogenicity, ensuring the diversity of antibody population in Genetic Algorithm.

Immune-Genetic Algorithm introduces the "immune operator", genes inoculation and selection, and simulates the specific auto-adaptation and artificial immune of the artificial immune system, possessing good properties of fast global convergence. The specific workflow of Immune-Genetic Algorithm is described in figure 1.

- Step 1.** Randomly generate  $\mu$  individuals of parent population. The search space of those quasi-optimal values  $x^*$  is composed of mesh points in  $R^n$ . Each part of these points is an integral multiple of  $\Delta$ . Each individual in the population is presented as  $(x, \sigma)$ , where  $x = (x_1, x_2, \dots, x_n) \in X \subset R^n$ , is a solution to the problem.  $x^* \in X$  is the expected solution.  $f(x^*) = \max f(x)(x \in X) = f^*$ , where  $f^*$  is the max fitness of  $X$ .
- Step 2.** Generate the intermediate population by crossing, with the size  $2\mu$ . The specific process is that for each individual  $(x, \sigma)$  of parent population, select another individual  $(x', \sigma')$  to crossover with  $(x, \sigma)$  in a crossover-point to generate  $y$  and  $y'$ .
- Step 3.** Mutate on the individual  $(x, \sigma)$  and generate a new one  $(x', \sigma')$
- Step 4.** Inoculate genes. Inoculating the individual  $(x, \sigma)$  means to modify the value of  $x$  and  $\sigma$  in the range of variation or the restrictions in some parts of the optimal individuals. The inoculation process satisfies: if  $f(x) = f^*$ ,  $(x, \sigma)$  turns to itself with probability  $l$ .

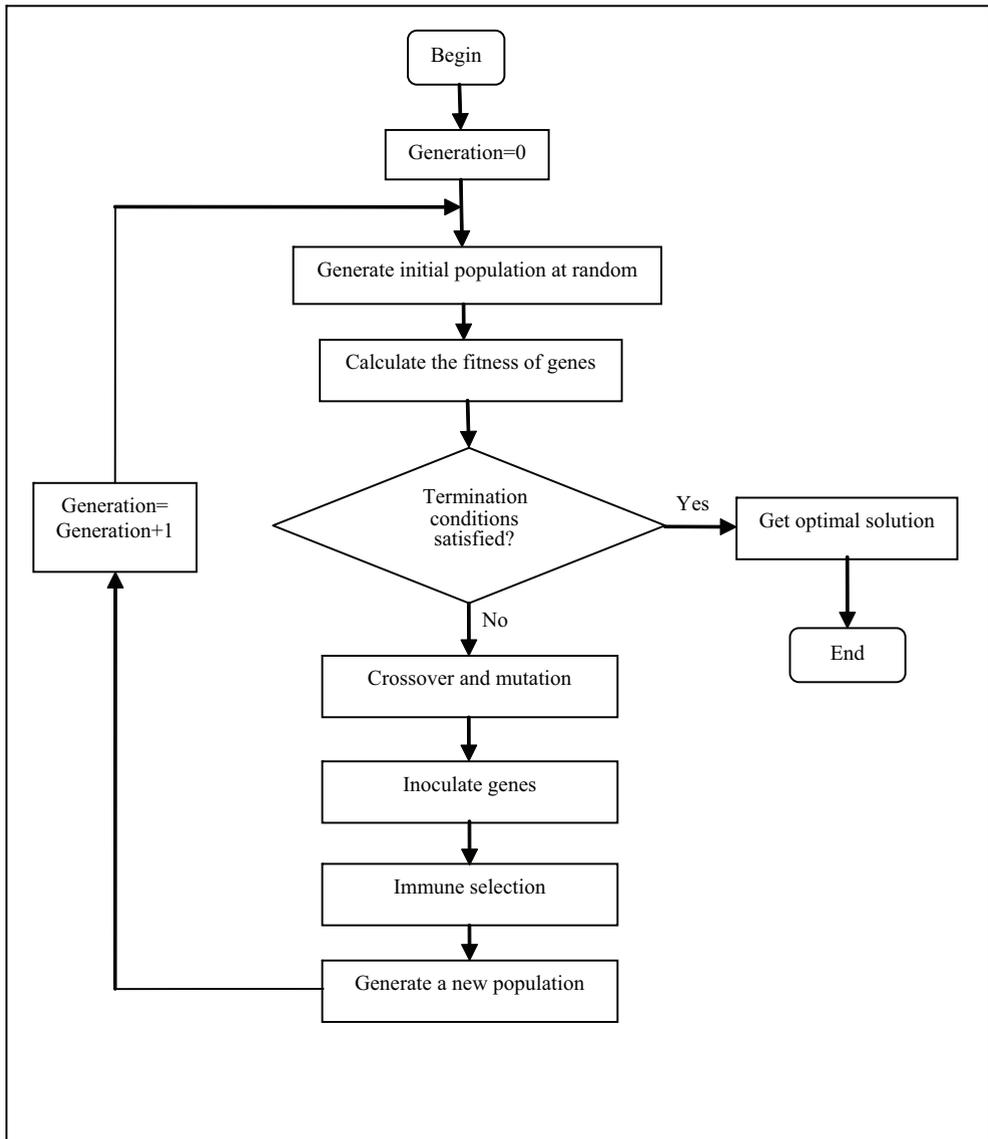


Fig. 1. Immune-genetic algorithm workflow

**Step 5.** Immune selection. It consists of two procedures: Immunity testing and selection. The first procedure is to test the inoculated individuals. If its fitness is smaller than its parent's fitness, there has been a serious degradation on the inoculated individual and its parent individual is used instead for the next competition. Immune Selection is to select  $\mu$  individuals from  $2\mu$  individuals according to their fitness to compose a new parent population.

**Step 6.** If the termination conditions are not satisfied, generate a new generation and go back to step 2.

IGA has two advantages: 1. inoculating genes and adding a priori knowledge can effectively accelerate the convergence speed and improve the quality of the solution; 2. Concentration based immune selection method can prevent premature phenomenon and make sure the optimization process toward the global optimum direction. The disadvantage is that the selection of genes and inoculation approach should be analyzed according to specific situations.

### 3. The Improved immune-genetic algorithm for TSP

This section is related to improvements on the standard immune-genetic algorithm for TSP. An improved immune-genetic algorithm of the author's research work (Lu et al., 2007) for TSP is introduced in this section. The algorithm effectively integrates immune algorithm into GA (Jiao & Wang, 2000) using an improved strategy of IGA and applies a new selection strategy in the procedure of inoculating genes. The computer implementation for the improved algorithm is also discussed in this section.

#### 3.1 The Improved immune-genetic algorithm

The improved immune-genetic algorithm uses a sequential representation to present the visited cities listed in the order (Lu et al., 2007). For example, the journey (5-7-8-9-4-3-2-6-1) can be expressed as (578943261). The path based coding method requires that the genetic code in the chromosome of an individual (a route) is not repeated. That is, any city should be visited once and only once.

The Roulette Wheel selection is employed where parents are selected according to their fitness (Lu et al., 2007). The individuals are generated using the Greedy crossover algorithm, which selects the first city of one parent, compares the others left in both parents, and chooses the closer one to extend the traveling way. If one city has been chosen, another city will be selected. And if both cities have been chosen, a not-yet-selected city will be randomly selected.

A swapping method is used for the TSP in IGA instead of the conventional mutation method. The method selects a binary code in random, which represents two cities from two individuals. The binary code is then swapped if the distance (length) of the traveling way for a new individual is shorter than that of the old one. (Lu et al., 2007)

In the procedure of developing and inoculating genes, the quality of genes has decisive influence on the convergence speed of the immune algorithm. Therefore, we make full use of prior knowledge and first develop a good gene pool that includes different genes representing the shorter traveling way of the TSP. After that, genes are randomly selected from the gene pool and finally inoculated into individuals.

In solving TSP, if the coordinates of 10 cities are in a circle for example, the route along the circle is the optimal solution and the optimal gene. The prior knowledge is applied to develop a gene pool. The gene pool is a two-dimensional 10 x 2 matrix. Having been calculated and optimized the gene pool can be the best, showed in table 1.

from	0	1	2	3	4	5	6	7	8
to	1	2	3	4	5	6	7	8	9

Table 1. The Gene Pool

When inoculating genes, a new selection strategy is applied to keep the excellent genes included in the population of individuals within a reasonable percentage. Those excellent genes are further used to generate other individuals. The selection strategy is developed based on the evaluation of the diversity of genes that are involved in the population of individuals. There are three computational procedures included in the strategy (Lu et al., 2007): 1) evaluating the diversity of genes included in the population of individuals, 2) calculating the percentage of genes included in the population of individuals, 3) computing the selection probability of genes. The details of the procedures are explained as follows.

1. Evaluating the diversity of genes

The diversity of genes is first evaluated by comparing the information entropy of every two genes. For example, giving two bit strings, each one has two alternative letters in its alphabet, thus the information entropy for the N genes is given by:

$$H(N) = \frac{1}{M} \sum_{i=1}^n H_i(N) \tag{1}$$

where  $H_j = \sum_{i=1}^2 -P_{ij} \log P_{ij}$ ;  $H_j(N)$  is the information entropy of the jth binary bit of two genes,  $P_{ij}$  is the probability of the jth binary bit of two genes being equal to  $k_i$ .  $P_{ij}$  is equal to 0 if the binary bits of two genes are the same; otherwise  $P_{ij}$  is equal to 0.5. M is the number of genes.

The affinity of genes shows the similarity between the two genes. The affinity of gene v and gene w is:

$$ay_{v,w} = \frac{1}{1 + H(2)} \tag{2}$$

2. Calculating the percentage of genes

The percentage of gene v is  $C_v$ , given by:

$$c_v = \frac{1}{N} \sum_{w=1}^N a c_{v,w} \tag{3}$$

Where  $ay_{v,w}$  is the affinity of gene v and gene w :

$$ac_{v,w} = \begin{cases} 1, & ay_{v,w} \geq Tac1 \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

, Tac1 is a predefined threshold

If  $C_v$  is bigger than a predefined threshold, the gene will be inhibited (removed from the population), otherwise it remains. This step is to remove the extra candidates.

3. Computing the selection probability of genes

The selection probability of gene v is:

$$e_v = \frac{ax_v \prod_{s=1}^N (1 - as_{v,s})}{c_v \sum_{i=1}^n ax_i} \quad (5)$$

$$(6) \quad \text{where } \begin{cases} ax_{v,s}, ay_{v,s} \geq Tac2 \\ as_{v,s} = \end{cases} \quad , Tac2 \text{ is a predefined threshold}$$

$$0, \quad \text{otherwise}$$

$$ax_v = \text{fitness}(v) \quad (7)$$

$$\sum_{i=1}^n ax_i = \text{sum fitness} \quad (8)$$

This formula controls the concentration and diversity of genes. The genes with high affinity to the antigen will be selected to regenerate. The genes with high percentage are inhibited. The improvement of the improved genetic algorithm is mainly reflected in that: 1) Improve the fitness of individual genes and the quality of the individual by inoculating genes. This way the convergence rate is significantly sped up. 2) Concentration based immune selection method not only encourages the solution with high fitness, but also inhibits the solution with high percentage, ensuring the convergence of the algorithm and the diversity of the solution population. It's also suitable for multimodal function optimization.

### 3.2 Computer implementation for improved immune-genetic algorithm

The workflow of the improved immune-genetic algorithm is showed in figure 2.

The Computational Flow of the improved immune-genetic algorithm is showed in the following:

Begin

Initiation: develop a gene pool using prior knowledge; select genes from the gene pool randomly;

Repeat

Calculate the fitness of each gene;

Calculate the probability that genes are selected;

Generate individuals using the Greedy crossover;

Gene mutates;

Inoculate genes using inoculation algorithm;

Select genes using the selection strategy based on the evaluation of the

diversity of genes: Calculate the information entropy, the percentage and the selection probability of genes;  
 Replace the removed genes with the new developed genes to produce a new generation of genes;

Until ( the genes satisfy the termination conditions)

End

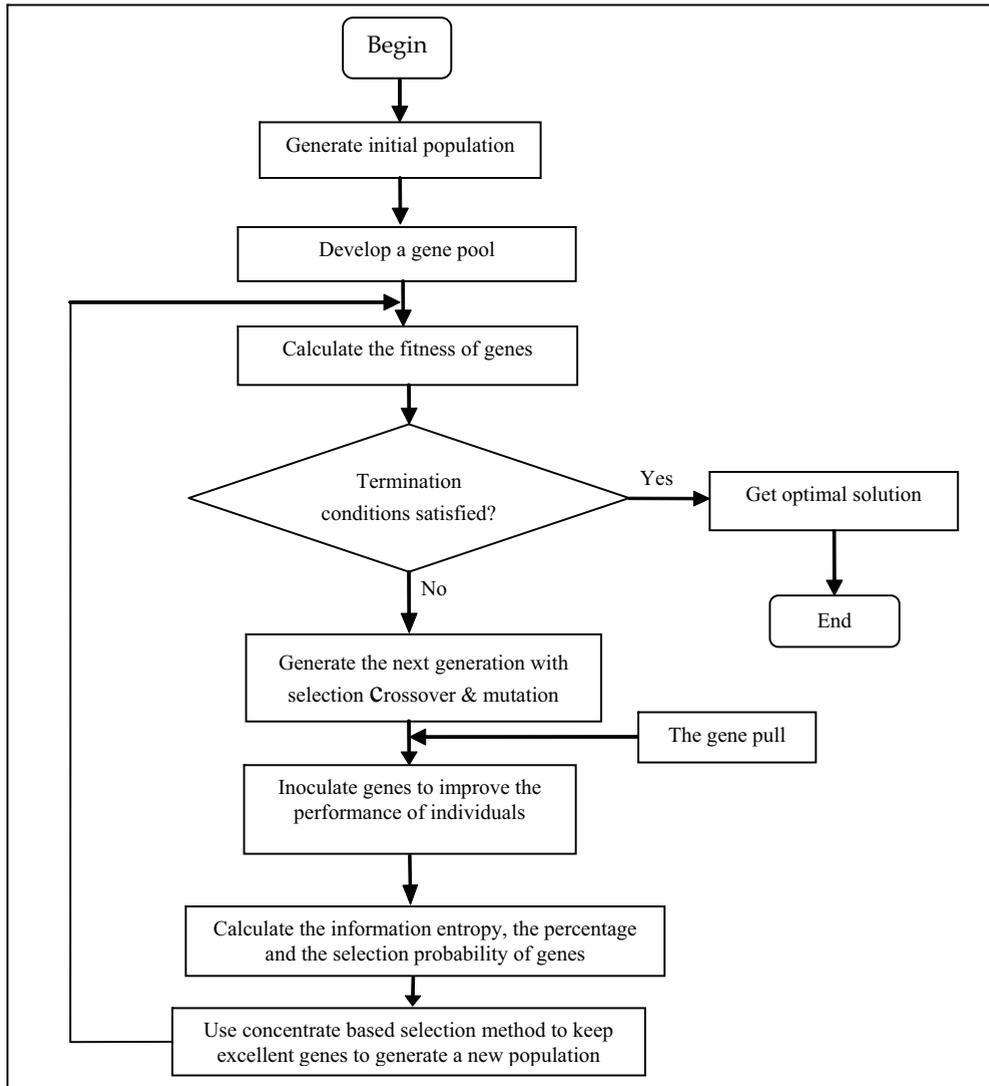


Fig. 2. The Improved Immune-Genetic Algorithm

The information of a gene individual includes gene chromosomes, chromosome length, individual fitness and the individual's corresponding variable. In the program, the structure of a gene individual is defined as follows.

```
typedef struct individual{
    int chrom[n];      /* chromosomes */
    float fitness;     /* individual fitness */
    int totaldistance;
    int lchrom;        /* chromosome length */
    double variable;   /* individual's corresponding variable */
};
```

The fitness function employed is showed as equation (9).

$$\text{Fit}(x) = \frac{1}{\sum_{i=1}^n d(C_i, C_{i+1}) + d(C_n, C_1)} \quad (9)$$

The probability that genes are selected is calculated using the roulette wheel selection with standard genetic algorithm

A swapping method is used for gene mutation. In the swapping method, a binary code that represents two cities from two individuals is randomly selected, and is then swapped if the distance (length) of the traveling way for a new individual is shorter than that of the old individual (Lu et al., 2007). The figure 3 shows an example of gene mutation: two locations are randomly selected to mutate. The probability of mutation is between 0.5 and 0.1;

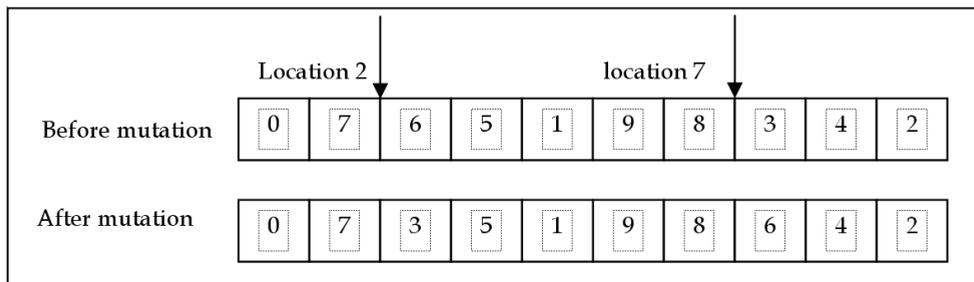


Fig. 3. Gene mutation

The computational flow of the gene mutation is as follows.

```
Begin
    Select locations to mutate at random;
    If (mutation probability < predefined value) swap the codes on the locations;
End;
```

The inoculation process is:

```
Begin
    Select a gene randomly from the gene pool;
```

Find a reasonable inoculating location;  
 Find and modify the conflict location of the gene to be inoculated;  
 Inoculate to the location of the gene;

End;

The process of the selection strategy based on the evaluation of the diversity of genes is:

Begin

Calculate the affinity of gene  $\nu$  and  $\omega$  according to the equation (2), where the diversity of two genes  $H(2)$  is calculated using the equation (1);

Calculate the percentage  $C_\nu$  of each gene in the population according to the equations (3) and (4),

If ( $C_\nu \geq \text{Tac1}$ ) remove the gene from the population;

Else the gene remains;

Calculate the selection probability  $e_\nu$  of genes according to the equations (5) and (6):

If ( $e_\nu \geq \text{Tac2}$ ) the gene is selected to regenerate;

Else the gene is inhibited;

End;

#### 4. Numerical experiments

Two case studies on 21-city and 56-city traveling salesman problems are given in this section to compare the solutions generated by IGA and the conventional GA.

The comparisons of the number of evolutionary iterations in IGA and conventional GA of two cases are showed in Figures 4 and 5 respectively. In both figures, the upper curve shows the evolutionary process of IGA and the lower one shows the evolutionary process of GA. Figures 6 and 7 show the optimal path for the TSP in the two case studies.

The results prove that the number of evolutionary iterations is significantly reduced by using IGA. As seen from Figure 4, IGA takes only five iterations to reach the optimal solution while GA takes about 30 evolutionary iterations. The selection strategy and the procedure of inoculating genes used by the improved immune-genetic algorithm proposed are effective to improve the performance of the individuals.

Therefore, the improvement on the performance of the individuals of IGA is helpful in accelerating the iterative process of GA. Although the convergence of the algorithm proposed need to be investigated, the computer numerical experiments on two case studies demonstrate preliminarily that the improved strategy of IGA is helpful for improving the evolutionary iterations of genetic algorithms for traveling salesman problem (Lu et al., 2007).

The main data of IGA:

1. The probability of crossover is 0.8~0.9.
2. The probability of mutation is 0.05~0.2.
3. The size of the population is 100.
4. The probability of inoculating is 0.85~1.
5. The evolutionary generation is 200.

Seen from Figure 4, the optimal solution has been reached in the first 2 generations. Inoculating genes significantly improves the convergence speed of the algorithm.

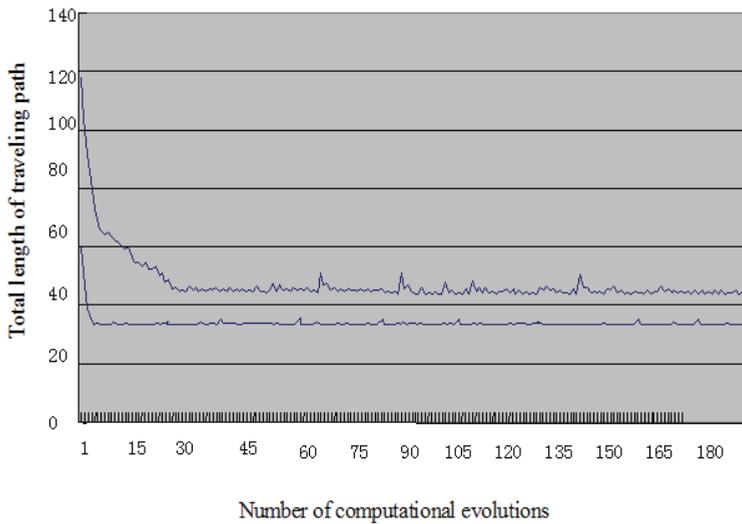


Fig. 4. Comparison of the number of evolutionary iterations in IGA and the conventional GA: 21-city case

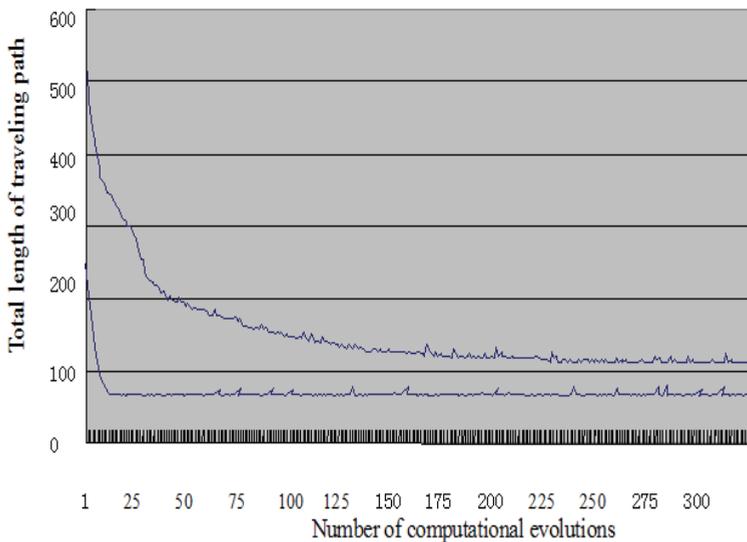


Fig. 5. Comparison of the number of evolutionary iterations in IGA and the conventional GA: 56-city case

The main data of IGA:

1. The probability of crossover is 0.8~0.9.
2. The probability of mutation is 0.05~0.2.
3. The size of the population is 100.
4. The probability of inoculating is 0.85~1.

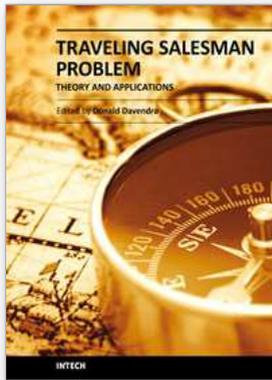


and 56-city TSPs show that IGA significantly reduces the number of evolutionary iterations for reaching an optimal solution.

## 6. References

- Aarts, E. H. L.; Korst, J. H. M. & Laarhoven, P. J. M. (1988). A quantitative analysis of the simulated annealing algorithm: a case study for the traveling salesman problem, *J. Stats. Phys*, vol. 50, pp. 189-206
- Aarts, E. H. L. & Stehouwer, H. P. (1993). Neural networks and the traveling salesman problem, *Proc. Intl. Conf. on Artificial Neural Networks*, Berlin Heidelberg: Springer Verlag
- Applegate, D.; Bixby, R. E.; Chvatal, V. & Cook, W. (1998). On the solution of traveling salesman problems, *Documenta Mathematica* (Extra Volume, ICM III), pp. 645-658
- Barnhart, C.; Boland, N. L.; Clarke, L. W.; Johnson, E. L.; Nemhauser, G. L. & Sheno, R. G. (1998). Flight string models for aircraft fleet and routing, *Transportation Science*, vol. 32, pp. 208-220
- Choi, B. K. & Yang, B. S. (2001). Optimal design of rotor-bearing systems using immune-genetic algorithm, *Transactions of the ASME Journal of Vibration and Acoustics*, vol. 123, pp. 398-401
- Du, Wei; Du, Haifeng & Li, Maolin (2009). Hyper-mutation antibody clone algorithms for TSP, *Journal of Xidian University*, vol.36, pp.527-534, ISSN: 10012400
- Fiechter, C. N. (1994). A parallel tabu search algorithm for large traveling salesman problems, *Discrete Applied Mathematics*, vol. 51, pp. 243-267
- Forrest, S.; Perelson, A.S.; Allen, L. & Cherukuri, R. (1994). Self-nonsel discrimination in a computer, *Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy*, Los Alamitos, CA. pp. 202-212
- Gao, Shangce ; Dai, Hongwei ; Yang, Gang & Tang, Zheng (2007). A novel Clonal Selection Algorithm and its application to Traveling Salesman Problem, *Proceedings of IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, pp. 2318-2325, ISSN : 09168508, Maruzen Co., Ltd., Tokyo, Japan
- Ge, Hong & Mao, Zongyuan (2002). Improvement for Immune Algorithm, *Computer Engineering and Applications*, pp. 47-49, ISSN: 1002-8331-(2002)14-0047-03, Beijing
- Gilmore, P. C. & Gomory, R. E. (1964). Sequencing a one state-variable machine: a solvable case of the traveling salesman problem, *Operations Research*, vol. 12, pp. 655-679
- Jiao, L. C. & Wang, L. (2000). A novel genetic algorithm based on immunity, *IEEE Transactions on Systems, Man, and Cybernetics*, Part A - Systems and Humans, vol. 30, pp. 552-561
- Jin, H. D. ; Leung, K. S. ; Wong, M. L. & Xu, Z. B. (2003). An efficient self-organizing map designed by genetic algorithms for the traveling salesman problem, *IEEE Trans. on Systems, Man, and Cybernetics Part B: Cybernetics*, vol. 33, pp. 877-888
- Laarhoven, P. & Aarts, E. H. L. (1987). Simulated Annealing: Theory and Applications, MA: Kluwer Academic Publishers, Norwell
- Lawler, E. L.; Lenstra, J. K.; Kan, A. H. G. R. & Shmoys, D. B. eds. (1985), The Traveling Salesman Problem, *Chichester: John Wiley*
- Li, Tao (2004). Computer immunology , Publishing House of Electronics Industry, ISBN: 7-120-00107-8, Beijing
- Liang, Yan & Yang, Kongyu (2008). Immunity genetic algorithm based on elitist strategy and its application to the TSP problem, *Proceedings of 2nd 2008 International Symposium on Intelligent Information Technology Application Workshop, IITA 2008*

- Workshop*, pp. 3-6, ISBN-13: 9780769535050, Shanghai, China, January 21, 2008 - December 22, 2008, Inst. of Elec. and Elec. Eng. Computer Society, Piscataway
- Liu, C. Y. (2006). Immune genetic algorithm and its application on TSP, *Technical Report of Nanjing University of Technology*, Nanjing
- Liu, Jinkun (2009). *Intelligent control*, Second Edition, Publishing House of Electronics Industry, ISBN: 9787121011184, Beijing
- Lu, J. G. ; Ding, Y. L. ; Wu, B. & Xiao, S. D. (1996). An improved strategy for GAs in structural optimization, *Computers & Structures*, vol. 61, pp. 1185-1191
- Lu, Jinggui ; Fang, Ning ; Shao, Dinghong & Liu, Congyan (2007). An Improved Immune-Genetic Algorithm for the Traveling Salesman Problem, *Proceedings of Third International Conference on Natural Computation (ICNC 2007)*, vol. 4, pp.297-301, ISBN: 0-7695-2875-9, Haikou, Hainan, China, August 24-August 27,2007
- Pasti, Rodrigo & De Castro, Leandro Nunes (2006). A neuro-immune network for solving the traveling salesman problem, *Proceedings of IEEE International Conference on Neural Networks*, pp.3760-3766, ISSN : 10987576, Vancouver, BC, Canada, July 16, 2006 - July 21, 2006, Institute of Electrical and Electronics Engineers Inc., Piscataway
- Plante, R. D.; Lowe, T. J. & Chandrasekaran, R. (1987). The production matrix traveling salesman problem: an application and solution heuristic, *Operations Research*, vol. 35, pp. 772-783
- Potvin, J. V. (1996). Genetic Algorithms for the Traveling Salesman Problem, *Annals of Operations Research*, vol. 63, pp. 339-370
- Qi, Yutao ; Liu, Fang & Jiao, Licheng (2008). Immune algorithm with selfadaptive reduction for large-scale TSP, *Journal of Software*, vol. 19, pp. 1265-1273, ISSN: 10009825
- Thiago A.S. Masutti & Leandro N. de Castro(2009). A self-organizing neural network using ideas from the immune system to solve the traveling salesman problem, *Information Sciences* 179 (2009), W. Pedrycz (Ed.), pp. 1454-1468, ISSN: 0020-0255
- Timmis, J.; Neal, M. & Hunt, J. (2000). An artificial immune system for data analysis, *BioSystems* 55 (1), pp. 143-150
- Wei, J. D. & Lee, D. T. (2004). A new approach to the traveling salesman problem using genetic algorithms with priority encoding, *Proc. 2004 IEEE Congress on Evolutionary Computation, Portland*, pp. 1457-1464
- Wu, Jianhui ; Zhang, Jin & Liu Zhaohua (2009). Solution of TSP problem based on the combination of ant colony algorithm and immune algorithm, *Journal of Hunan University Natural Sciences*, vol. 36, pp. 81-87, ISSN: 16742974
- Zeng, Congwen ; Gu, Tianlong (2007). A novel immunity-growth genetic algorithm for traveling salesman problem, *Proceedings of Third International Conference on Natural Computation, ICNC 2007*, pp. 394-398, ISBN-10: 0769528759, Haikou, Hainan, China, August 24, 2007 - August 27, 2007
- Zheng, J. G. ; Wang, L. & Jiao, L. C. (2001). Immune algorithm for KDD, *Proc. SPIE*, vol. 4553, pp. 215-220



## **Traveling Salesman Problem, Theory and Applications**

Edited by Prof. Donald Davendra

ISBN 978-953-307-426-9

Hard cover, 298 pages

**Publisher** InTech

**Published online** 30, November, 2010

**Published in print edition** November, 2010

This book is a collection of current research in the application of evolutionary algorithms and other optimal algorithms to solving the TSP problem. It brings together researchers with applications in Artificial Immune Systems, Genetic Algorithms, Neural Networks and Differential Evolution Algorithm. Hybrid systems, like Fuzzy Maps, Chaotic Maps and Parallelized TSP are also presented. Most importantly, this book presents both theoretical as well as practical applications of TSP, which will be a vital tool for researchers and graduate entry students in the field of applied Mathematics, Computing Science and Engineering.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Jingui Lu and Min Xie (2010). Immune-Genetic Algorithm for Traveling Salesman Problem, Traveling Salesman Problem, Theory and Applications, Prof. Donald Davendra (Ed.), ISBN: 978-953-307-426-9, InTech, Available from: <http://www.intechopen.com/books/traveling-salesman-problem-theory-and-applications/immune-genetic-algorithm-for-traveling-salesman-problem>

# **INTECH**

open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.