

# A Method for TSP Immune-Based

Yaping Jiang and Yalin Wang

**Abstract**—A model of TSP problem based on Artificial Immune System is established. In the model, antibody cells and antigen cells was defined, and describe the dynamic process of memory cells to the problem. We refer to the conception of variation GA, and present GFE algorithm of evolution of superior genes, and make use of the principle of immune stability in the clone's procedure to evolve the set of antibody in order to get the global approximate solution rapidly and efficiently. The result of the experiment shows that the algorithm is not only feasible but also a comparative good method, which has a fast convergence speed and global optimization to solve the NP problems.

**Index Terms**—Artificial immune, clone selection, TSP, concentration.

## I. INTRODUCTION

TSP (Traveling Salesman Problem) is a problem about NP. The one in early time to solve this problem is Dijkstra (1959). This algorithm has high accuracy in the shortest path[1]. However, when the value of N is larger, the result is not that efficiency. Genetic algorithm was proposed based on group evolution by J.H. Holland in the 60's[2]. Though we can get a local optimal solution, if convergence is fast, the overall probability of occurrence is not the optimal solution. So the optimal solution for the global optimal solution is doubtful. With the development of modern bionics, M. Dorigo, and others proposed ant colony algorithm by observing carefully and discovered "ants moving out, going to rain," (such as: search for the best return path to food) [3]. By comparing the amount of pheromone secreted by ants, we found that it on the relatively short path, the concentration is high, and according to this we can find the shortest path[4]. But the shortcoming is that computation is too much and the effect is not that good. In 1958, the Danish scholar Jerne proposed a mathematical model on the immune system as the first one [5]. From then on, the immune computation becomes a new area of artificial intelligence research. Artificial immune algorithm has combined the neural network, the immune systems and the principle of evolutionary computation, and is significant for the research of article life, agents, robots information model. So, by simulating the principle of artificial immune systems we have set up a model to solve the problem of TSP which is based on principles of artificial immune and genetic evolution. The result of the experiment shows that compared with genetic algorithm and ant colony algorithm this model is better at convergence speed and entire search ability.

Manuscript received July 20, 2012; revised September 2, 2012.

The authors are with School of Computer and Communication Engineering, Zhengzhou University of light industry, Zhengzhou, China (e-mail: yapingjiang@163.com, Yalinwang1996@126.com).

## II. ALGORITHM

### A. The Mathematical Description of TSP Problem

TSP problem is a shortest path problem, showed in graph  $G$ ,  $G=(V, E)$  and  $V=(V_1, V_2, \dots, V_n)$ , using  $v_i$  represent city  $i$ ;  $E$  is the set of edges in the graph, and  $d(v_i, v_j)$  is the distance from city  $v_i$  to city  $v_j$ . For the TSP problem of several cities, there is a number of paths starting from one city traverse to undirected graph  $G$ . It is a problem to find the shortest path, and this is point in this paper.

Arbitrarily selecting one city as a starting point, traversing graph  $G$ , and return to the starting point, we can get a character sequence in the length of  $n+1$ . This sequence is a solution to the problem, represented with  $P$ ,  $P=v_{\text{start}}, \dots, v_i, \dots, v_j, \dots, v_{\text{start}}$ .  $i, j, \text{start} \in \mathbb{N}$ ,  $1 \leq i \leq n$ ,  $1 \leq j \leq n$ ,  $i \neq j$ .  $\mathbb{N}$  is a set of natural numbers. The value of path length is defined as  $T_d$ , then:

$$T_d = \sum_{i=1}^{n-1} d(v_i, v_{i+1}) + d(v_1, v_n) \quad (1)$$

Generally, (only consider the undirected graph),  $d(v_i, v_j) = d(v_j, v_i) = d_{ij}$ .

We can get the minimum value of the formula (1), and this path is the shortest path (the optimal solution).

### B. Algorithm Description

The correspondence of immune system and the method is showed in Table I:

TABLE I: THE MAPPING OF IMMUNE SYSTEM AND IMMUNE-BASED TSP MODEL

Immune system	The TSP model based on immune
Antigen	Optimal solution
Antibody	Feasible solution
Cell Cloning	Copy antibody
Antibody and antigen binding	Matching algorithm based on distance
B cells, T cells	the path of String representation
Signal	More than match the activation threshold
Increase in antibody concentration	Approximate feasible solution to increase

In the random candidate solutions, according to the calculating antigen and antibody affinity, we can select the greater affinity of antibodies and then evolved into memory cells. In cloning process, memory cells keep a dominant gene, mutate and generate a large number of new antibodies, then re-evaluate new antibody. The new antibodies update the set of memory cells. The renewal mechanism of memory cell: the new cells with higher affinity memory cells eliminate the small affinity memory cells in the set.

The antibody is arranged as path length in the set of memory cells. After the given generations of evolution, the optimal extraction of the memory cells is the optimal

solution.

We should treat the optimal solution problem as the antigen abstractly. There are  $N$  unrepeated nodes arranged as an antibody randomly, and each antibody represents a candidate solution.

**Definition 1** In graph  $G$ , start from any of the Vertex, visit all the other vertices, and each vertex is visited only once (except for starting and ending vertex), so the vertices of the shortest path in the loop is defined as antigen (ag).

**Definition 2** In graph  $G$ , start from any of the Vertex, visit all the other vertices and each vertex is visited only once (except for starting and ending vertex), so the sequence of vertices in the loop is defined as the antibody (ab). We define the set of antibodies as:

$$D = \{ \langle s, dist \rangle \mid s \in P, dist \in N \} \quad (2)$$

( $S$  is the sequence of  $n$  nodes in a loop (loop path),  $dist$  is the path length in the loop.)

Generate  $n$  non-repeat city nodes which consist of a string in the length of  $n+1$  (add a initial node forming a loop), and this string consist a initial antibody (ab),  $ab \in D$ . Set the initial antibody in the size of  $k$ .

Affinity is a measure to match antibody and antigen. This paper uses the distance-based matching algorithm. Define the affinity between antibody (ab) and antigen (ag) as:

$$f_{dist}(ab, ag) = 1 / (ab.dist - ag.dist) \quad (3)$$

For formula (3), the greater the affinity of antibodies and antigens, they will match better, which indicates that it get closer to the optimal solution. However, we do not know the value of  $ag.dist$ . So, according to the specific scale of the problem and the initial value of city, we give a parameter  $T$ :

$$T = \left( \sum_{i=1}^n \sum_{j=1}^n d(i, j) \right) / (2n) \quad (4)$$

$$f_{dist}(ab, ag) = 1 / (ab.dist - T) \quad (5)$$

Therefore, the affinity of the formula is modified as:

$$R = \{ x \mid f_{dist}(x, ag) \geq \theta, x \in D \} \quad (6)$$

Set the size of memory cells as  $q$ , and the initial set of memory cells empty. Calculate the initial affinity between antibodies and antigens using formula (5) calculation. For those surpass the threshold value of  $\theta$  are activated and will evaluate into memory cell.  $R$  defined as the set of memory cells, such as formula (6).

In body immune systems, when antigens invade, the immune system will produce a large number of antibodies to match with antigen, and the concentration of antibodies which is better with antigen affinity will increase, helping to eliminate the antigen; when antigen died, the generation of such antibodies will be inhabited, and at the same time the concentration will decrease. By this way the immune system will always keep balance [6].

At the antibody collection, the advantage antibody which is close to antibody antigen is activated, and becomes a memory cell. Memory cells will generate a lot of cloning to eliminate antigen, clone selection process in Fig. 1, and then through the clone selection process [7] a new set of

generated cells will combine with the antibody, antibody concentrations rose, indicating that the number of approximate solutions is increasing. However, if these antibodies are too concentrated, the concentration is too high, it is difficult to maintain the diversity of antibodies, and we will lose the potential of good evolution of antibodies and get into a local optimum [8].

This article limits the number of memory cells in cloning, as is shown in the formula (7),

$$f_{num} = \sum_{i=1}^q \left\lceil \frac{\beta \cdot q}{i} \right\rceil \quad (7)$$

Procedure Clone\_Select()

Begin

Set antigen  $ag$ ; /\* s.t.  $\min(Td)$  \*/

Initialization antibody collection;

While (Evolution algebra  $< m$ ) /\*  $m$  is Evolution algebra \*/

Begin

While (Antibody non-empty set)

Begin

If ( $f_{dist}(ab, ag) < \theta$ ) Then

Begin

delete  $ab$ ; /\* Remove low affinity antibodies \*/

End;

Else

Begin

If (The number of memory cells  $< q$ ) Then

Memory cells will be added to the set of  $ab$ ;

Else

Begin

Select from the memory cell affinity lowest concentration of cells;

If (Select the cell affinity  $> f_{dist}(ab, ag)$ ) Then

delete  $ab$ ;

Else

Remove from the set of selected memory cells

added to  $R$  in the  $ab$ ;

clone the number  $f_{num}(ab(i))$  of  $ab$ ;

After joining antibody collection variation;

End

End

End

End

Which,  $f_{num}$  is the total memory cell number of clones, the first  $i$  items on behalf of the number of clones  $i$  cells,  $\beta$  is the default parameter factor,  $q$  is the number of memory cells. By the formula (7) we know that, the greater the affinity of the cell, the more the clone number will be; the other hand, the less. For example:  $\beta=1$ ,  $q=100$ , because the order of memory cells is decided by affinity, and therefore the largest number of affinity memory cells in cloning is 100, followed, clone number 50, and so on.

Memory cells in the evolutionary process is updating dynamic. Each memory cell is selected from the current as the best antibody.

Each time we select a collection of several  $c$  from the current best antibody, if the memory cell collection are not

full, then set them directly into the memory cells; otherwise, firstly, determine whether the memory cells are already existing in the memory cell collection, if they exist, eliminate the memory cells; or compare the relative new cells produced by the affinity with the current memory cells, if the former is smaller, the new memory cells will be eliminated; otherwise, eliminate the several  $c$  which are smaller at affinity, and then incorporated into the newly created memory cell. In such process, memory cells can update dynamically [9].

For the TSP problem, maybe there exist repeated sequences after variation in the traditional way. This paper presents a new type of mutation, GSE (Gene Segments Exchange) algorithm, which not only solve the recurring problem, but also maintain good genes and inherit to the next generation [10]. This algorithm achieves the exchange of the two alleles of the antibody; when the length of gene fragment is 1, the algorithm degenerates to a single point mutation--gene mutation.

At the same time, the algorithm can also be extended to Crossover mutation of multiple gene fragments. The algorithm is as follows:

Given two sequences  $gs1, gs2$ , gene sequence's exchange beginning and ending point is  $(i, j)$ , and  $i, j \in [1, n]$

$$N=34, k=300, m=0.0002, c=5, \beta=2, q=100$$

$$gs1: (v_1, v_2, \dots, v_i, \dots, v_j \dots v_n)$$

$$gs2: (v_1, v_2, \dots, v_i', \dots, v_j' \dots v_n')$$

First, create the mapping relationship between the allele  $(v_m, v_m')$  from  $v_i$  to  $v_j$  and  $v_i'$  to  $v_j'$ , that is,  $v_m = \text{map}(v_m')$ ,  $v_m' = \text{map}(v_m)$ ,  $i \leq m \leq j, m \in n$ .

Then, after the transformation of formula (8), we can get two new series.

$$\begin{cases} v_m = v_m, v_m' = v_m' & m \notin [i, j] \wedge v_m \notin [v_i, v_j] \wedge v_m' \notin [v_i', v_j'] \\ v_m = \text{map}(v_m'), v_m' = \text{map}(v_m) & \text{otherwise} \end{cases} \quad (8)$$

If there is passed relations between mapping relations, select a new code to exchange. The new sequences, on one hand, preserve the advantages of gene fragments (for example  $v_i, \dots, v_j$ ), on the other hand, ensure that there is no repeated new gene sequence.

For example:

$$gs1: (1,2,3,4,5,6,7,8,9)$$

$$gs2: (4,2,1,3,7,9,6,5,8)$$

Then, Transform according to the mapping. Apparently, 4,5,6,7 exchange with 3,7,9,6 directly; the  $x$  exchanges based on mapping relationship table, for example, in  $gs1$ , the third fig. is  $x$  (while the original one is 3), because the mapping(4,3), the present  $x$  becomes as 4. After such operation, the final variation sequence is:

$$gs1' (1,2,4,3,7,9,6,8,5)$$

$$gs2' (3,2,1,4,5,6,7,9,8)$$

### III. THE RESULTS OF EXPERIMENTAL

In Linux algorithm and using C language, the operating

environment is: Celeron 2.4GHz, 512M memory. Set:  $N=34, k=300, m=0.0002, c=5, \beta=2, q=100$ . The distance between cities is air straight-line distance.



Fig. 1. Immune algorithms for optimal path

After 10 times with 200 generations of repeated experiments, we get the results: among 34 cities the shortest path is (Fig. 2): Beijing-Hohhot-Yinchuan-Lanzhou-Xining-Urumqi-Lhasa-Chengdu-Chongqing-Guiyang-Kunming-Nanning-Haikou - Macau - Hong Kong - Guangzhou - Fuzhou - Taipei - Hangzhou - Shanghai - Nanjing - Hefei - Nanchang - Changsha-Wuhan-Xian-Zhengzhou-Taiyuan-Shijiazhuang-Jinan-Tianjin-Shenyang-Changchun-Harbin-Beijing. The total distance is 15786 km. This path is shorter than the optimized neural network algorithm by 118 km.

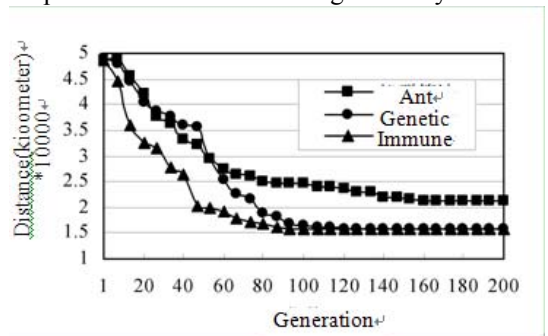


Fig. 2. Comparison immune algorithms with genetic algorithm and ant algorithm

We have selected 32 cities' nationwide, evolution algebra 200, and by comparing the immune algorithm, genetic algorithm and ant colony experimental (Fig. 3), the result showed that the convergence rate of immunization is superior to genetic algorithm and ant colony.

Memory cells update efficiency shows that the algorithm has strong entire search ability and faster convergence. As is shown in Table II :

TABLE II: EVOLUTION OF THE TABLE MEMORY CELLS

Evolution algebra	Update the number of memory cells	Running time (s)	Shortest distance (km)
1	20	0.010	25698
10	2	0.206	20479
100	5	2.385	15786
1000	1	20.492	15786
10000	0	33.140	15786

#### IV. CONCLUSION

Based on clone selection principle and the regulation mechanism of antibody concentration, we can get a solution for TSP. The result of the experiment shows that the algorithm is better than the genetic algorithm and ant colony algorithm in efficiency, and more powerful in entire search ability. The model based on immune group evolutionary is good at distributed parallel processing and self-learning and adaptive characteristics. All of this have given us a new thoughts and measures for solving the problems of optimized combination, pattern recognition and network security.

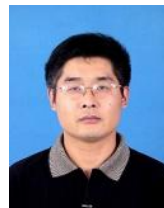
#### ACKNOWLEDGMENT

This work is partially supported by the Doctor Foundation of Zhengzhou University of Light industry (No:2009-010), the natural science foundation of Henan province of China (No:2010A520044), basic and frontier technology research of Henan province( No:122300410255).

#### REFERENCES

- [1] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, "An Introduction to Algorithms," *The MIT Press*, pp. 527-532, 2000.
- [2] J. H. Holland, "Adapation in Natural and Artificial System," *The University of Michigan Press*, 1975, pp. 150-195.
- [3] N. K. Jerne, "Towards a network theory of the immune system," *Annals of Immunology*, 1974, vol. 125, pp. 373-389.
- [4] M. Dorrigo, E. Bonabeau, and G. Theraulaz, "Ant algorithms and stigmergy," *Future Generation Computer Systems*, 2000, vol. 16, pp. 851-871.

- [5] O. Nasaroui, F. González, C. Cardona, and D. Dasgupta, "A Scalable Artificial Immune System Model for Dynamic Unsupervised Learning," *Genetic and Evolutionary Computation Conference (GECCO)*, Chicago, USA, 2003, vol. 7, pp. 219-230.
- [6] T. li, "Computer immunology," Beijing: *Publishing House of electronics industry*, 2004.
- [7] L. N. Castro and J. I. Timmis, "Artificial Immune Systems as a Novel Soft Computing Paradigm," *Soft Computing journal*, 2003, vol. 7, no. 8, pp. 526-544.
- [8] K. Jungwon and J. P. Bentley, "Towards an Artificial Immune System for Network Intrusion Detection: an investigation of dynamic clonal selection," *the congress on evolutionary computation (CEC-2002)* [C], Honolulu, May 12-17, 2010, pp. 1015 - 1020
- [9] K. Junwon and J. P. Bentley, "A Model of Gene Library Evolution in the Dynamic Clonal Selection Algorithm," in *Proceedings of the First International Conference on Artificial Immune Systems (ICARIS) Canterbury*, September 9-11, 2012, pp. 175-182.
- [10] Y. Jiang and T. Lv, "A Network Security Prevention Model Based on Vaccine," *Journal of Northeastern University*, vol. 32, no. 11, pp. 204-207, 2011.



**Yaping Jiang** was born in October, 1970 at Yong Cheng, Henan, China. Associate professor, Ph.D. School of Computer and Communication Engineering, Zhengzhou University of light industry, Zhengzhou, China, (e-mail: yapingjiang@163.com)

**Yalin Wang** was born in October, 1971 at Yong Cheng, Henan, China. Associate professor, Dept. of Electronic and Information Engineering, Yongcheng Vocational College, Yongcheng, China, (e-mail: Yalinwang1996@126.com)