8

A High Performance Immune Clonal Algorithm for Solving Large Scale TSP

Fang Liu, Yutao Qi[1], Jingjing Ma[2], Maoguo Gong[3], Ronghua Shang[4], Yangyang Li[5] and Licheng Jiao[6]
Xidian University, China

1. Introduction

Traveling Salesman Problem (TSP) is one of the most challenging combinatorial optimization problems. As the city number of TSP grows, the feasible solution space size increases factorially. For the small to mid-size TSP, the Lin-Kernighan (D. S. Johnson, 1990) (LK) and Lin-Kernighan Heuristic (C. Walshaw, 2001) (LKH) algorithms are very effective. However, these two algorithms are local search methods which find the best TSP tour in the k-change neighborhoods of the given initial TSP tour. Thus, they can only find a local optimal tour for TSP with complex solution space. Accordingly, the LK and LKH algorithms become very sensitive to the initial solution and often fail to find the global optimal tour within a reasonable time for solving large scale TSP. To remedy this problem, we make use of the global search ability of the immune clonal algorithm. Especially, we combine the two types of approaches (i.e. LK and immune clonal algorithm) to achieve high performance of the immune clonal algorithm, which can be run on loose-coupled computing environment for solving the large scale TSP.

The immune clonal algorithm inspired by biological immune system is a type of evolutionary random search algorithms. More and more research achievements indicate that immune clonal algorithm can maintain good population diversity and strong global search capability. Under the searching framework of the immune clonal algorithm, heuristic search strategies can be conveniently employed to enhance its local search capability. Such combinations take into account both global and local search strategies, and thus can realize a good tradeoff between effectiveness and efficiency. Moreover, the parallelizability of the biological immune system ensures the immune clonal algorithm can be run on loose-coupled computing environment which is advantageous to solve massive optimization problems such as the large scale TSP.

Simulation and analysis results show that the edges in the intersection set of several local optimal tours obtained by LK approach appear in the global optimal tour with high probability and the probability increases rapidly as the amount of local optimal tours increases. Using this phenomenon, an intersection set based vaccination strategy is designed in this chapter to accelerate the convergence speed of the immune clonal algorithm for TSP. In the immune clonal algorithm, vaccine is a set of genes which are estimations of the genes expected to appear in the global optimal antibody. The proposed approach in this chapter takes the intersection gene set of several memory antibodies as vaccine and injects the set

into antibody populations which are distributed on different computing nodes. This information-delivery approach between antibody populations, which take vaccine as carrier, not only accelerated the procedure of the evolution but also promoted the co-evolution between antibody populations.

The main content of this chapter is arranged as follows. Section 2 provides a brief description of the related background including the development of the immune inspired optimization algorithm and its general flow chart. Section 3 describes the main loop of the proposed high performance immune clonal algorithm for TSP. Section 4 gives a detailed description of the intersection set based vaccination strategy for TSP. Section 5 investigates the experimental study of the proposed approach. Finally, concluding remarks are made in Section 6.

2. Immune optimization

Immunization is a physiological function of biological immune systems, which identify and remove the invading "non-self" antigen, mutated and damaged cells to maintain the bodies' physiological balance and stability. Human immune system is a complex system consists of organs, cells and molecules with immune functions that can protect the body against pathogens, harmful foreign bodies and other disease factors. The same as neurological and endocrine systems, immune system has its own operation mechanism and can mutual cooperate and restraint with other systems, common to maintain the bodies' physiological balance in the life processes. Since 1940s, with the development of medical research on the biological immune systems, people's awareness and understanding of the immune system has been continuously improved, a complete biological immune science system had gradually formed (Jiao Licheng et al., 2006).

2.1 Some inspiring biological mechanism of immune system

Inspired by the biological immune systems, the model and algorithm of artificial immune systems are proposed. The key inspiring biological mechanism of immune system includes: immune recognition, immune memory, immune diversity, immune tolerance, parallelism and other biological immune mechanism.

1. Immune recognition

Modern immunology believes that immune function is a response to stimulation from antigens, which is shown as the immune systems' ability of identifying themselves and excluding non-self materials. Identification is an important prerequisite in the process of immune system functions. For the phenomenon of immune recognition, clonal selection theory believes that because of the differentiation of embryonic cell, the body has formed many lymphocytic series, each lymphocyte cell's surface has a specific set of antigen receptors. When antigens enter the body, they select the corresponding lymphocytes and specifically bind to the antigen receptors of their surfaces, led to the lymphocyte activation, propagation, differentiation, and thus lead to specific immune response. In addition, the antibody itself has antigen determinant, which can be recognized by other internal antibodies and lead to a reaction with them. So, antibodies have the dual nature of recognizing antigens and being recognized by other antibodies (F.M. Burnet,1978).

2. Immune memory

Immune memory is another important feature of the immune system. Experimental results show that it can produce not only B memory cells but also TH memory cells during the

immune response process. Immune memory can be explained as the phenomena of the increasing of the number of lymphocytes which have responses to specific antigens. When immune system first encounters an antigen, lymphocytes have to take some time to adjust themselves to identify antigens and save the memory information of the antigen after recognizing. When body meets the same antigen again, the effect of the associative memory, the Incubation Period of the appearance of antibodies reduced clearly and the content of antibody increased substantially, Erju duration Chang. Such phenomenon is called immunological memory (A. Tarakanov & D. Dasgupta, 2000).

Simulation on the immune memory is an important feature of artificial immune algorithms that distinction from other classic evolutionary algorithms. Farmer first presented an artificial immune model with memory which regards immune memory mechanism as an associative memory (J. D. Farmer et al, 1986). Smith compared the immune memory model and the sparse distributed memory model (SDM) and indicated that initial response corresponds to the procedure of information storage in SDM, the second response and the cross-immune response correspond to the procedure of reading memory (D. J. Smith et al,1998). Immune memory mechanisms can greatly accelerate the searching process of the optimization, speed up the learning process and improve the quality of learning. The introduction of immune memory mechanisms is an effective means to improve the efficiency of artificial immune system algorithm.

3. Immune diversity

In biological immune system, the number of antibody type is much larger than that of known antigen. There are two types of theories to explain the mechanism of immune diversity, the germlinetheory theory and the somatic mutation hypothesis. According to these theories, immune diversity may lies in the diversity of the connection of gene segments and it may be influenced by the complex pairing mechanism of the heavy chain and light chain. The immune diversity mechanism can be used for the searching procedure of optimization, it does not try global optimization, but deal with different antigens evolutionary, so as to enhancing the global search ability and keep the algorithms from falling into local optimum.

4. Immune tolerance

Immune tolerance is another important type of immune response and also one element of immunoregulation. Its performance is contrary to the positive immune response, and also different from a variety of non-specific immune suppressions which have no antigen specific, and can response or low response to various antigens. Immune tolerance is a phenomenon of body fail to response to a certain antigen which is caused by the lost function or death of specific antigen-induced lymphocyte. The general characteristics of immune tolerance are mainly in the following aspects: 1) For T or B cells were excluded or inhibited, immune tolerance is specific. 2) It's easier to introduce immature lymphocyte tolerance than mature cells. 3) The tolerance induction and maintenance of tolerance need the persistence of toleragen.

5. Parallelism

Biological immune system is a complex parallel system. Lymphoid organs, lymphoid tissue within other organs, lymphocytes cells and antigen presenting cells distribute to all parts of the body. Lymphocytes travel around the body by the blood, from one lymphoid organ or lymph tissue to another, so as to scattered the lymphoid organs and lymphoid tissue together all around the body into a functional whole. Various components of the immune system work in parallel and coordinated jointly, achieve all the features of the immune

system. Simulation of biological immune system is very important to taking full advantage of loosely coupled computing resources and improving the efficiency of immune system.

2.2 Artificial immune system model

Compared with other intelligent computing systems, a complete set of mathematical theory has not yet developed in the research areas of artificial immune system. Since the immune system itself is rather complicated, there are relatively a few research findings on artificial immune system model. In 1973, Jerne proposed the idiotypic network model (N. K. Jerne, 1973) and initiated the study of artificial immune system model in 1980, Herzenberg, etc. presented a loose-coupled network architecture which is more suitable for distributed problems (L. A. Herzenberg & S. J. Black, 1980). In 1986, Hoffmann put forward symmetric network model (G. W. Hoffmann,1986) based on the immune neuron model according to the similarity between immune system and nervous system. In 1989, Perelson presented a probability model of unique type network based on previous studies. (A. S. Perelson, 1989). In 1990, Farmer proposed dynamic system model (J. D. Farmer, 1990) based on connectionism, after compared and analyzed the similarities, differences and characteristics among the immune system, neural network and genetic system.. In 1995, Ishiguro etc. presented coupled immune network model. In 1997, Tang proposed multi-valued immune network model based on the mechanism of interaction between B cells and T cells (Z. Tang et al, 1997). In 1997, borrowing ideas from the mechanism that the system balance can be maintained by the interaction between B cells, Mitsumoto proposed immune response network model, which is used for scheduling and controlling the distributed autonomous robots group (N. Mitsumoto & T. Fukuta, 1997). In 2000, Zak proposed an immune system stochastic model, according to the principle of response under stress.

At present, two influential artificial immune network models are the Resource Limited Artificial Immune System (RLAIS) proposed by Jonathan Timmis etc.(J. Timmis & M. Neal, 2001) and the aiNet proposed by De Castro etc. (L. N. De Castro & F. J. Von Zuben,2000). Timmis put forward RLAIS on the basis of Cook and Hunt's research. He also presented the concept of Artificial Recognition Ball (ARB). Timmis considered that the role of ARB and B cell function is similar, artificial immune system is composed by a fixed number of ARB. Further more, by analogy with the natural immune system,, he thought that the stimulation from which ARB suffered includes the main stimulation, the stimulation and restrain from adjacent antibody. In addition, the capability of cloning can be determined by the stimulation given to ARB. De Castro's aiNet algorithm which simulates the stimulation process of the stimulation from the immune network to antigens, mainly includes the antibody-antigen recognition, immune cloning proliferation, affinity maturation and network suppression Immune network is considered as an enabling undirected graph, and is not fully connected. However, the current prevalence of adaptive immune network model is rather poor, contains more parameters, and over-reliance on changes in the network nodes to maintain network dynamics, the lack of the understanding of immune network of nonlinear information processing capacity are also weak points.At the same time, the design of the algorithm generally starts from focusing on data compression, therefore, the scope of the application of the algorithm is limited. It should be noted that the relevant mechanism in the immune network has been widely used in computer networks, particularly network security study, but these applications are mostly ideological. There are still no specific algorithms.

In addition to the network models described above, there are also two different non-network models presented respectively by Alexander Tarakanov (A. Tarakanov & D. Dasgupta, 2000) and Nohara (B. T. Nohara & H. Takahashi, 2000) in 2000.Alexander Tarakanov etc. tried to establish the form of protein models based on artificial immune system for the establishment of a formal model. After that, they indicated that the improved model can be used for the evaluation of the complex calculations of Kaliningrad's Ecological Atlas. In their works,much attention was paid to the interaction between the immune function of cells, and the network was not much involved. Based the feature of antibody units, Nohara etc. presented a non-network model of artificial immune system.

2.3 Artificial immune system algorithm

As the understanding of the mechanism of the immune system is not yet very deep, there is not much research on the artificial immune system algorithm. Common artificial immune algorithms include the following four types: artificial immune network algorithm, negative selection algorithm, immune evolutionary algorithm and immune clonal selection algorithm.

1. Artificial immune network

The simulation researches of the immune network mainly focus on the application of computer network security, while the study on immune algorithm is rarely seen at present. Now two typical artificial immune network algorithms are the Resource Limited Artificial Immune System Algorithm proposed by Timmis etc. (J. Timmis & M. Neal, 2001) and the aiNet algorithm proposed by De Castro etc. (L. N. De Castro & F. J.Von Zuben, 2000). However, the current prevalence of adaptive immune network model is rather poor, contains more parameters, and the over-reliance on changes in the network nodes to maintain network dynamics, the lack of the understanding of immune network of nonlinear information processing capacity are also weak points.At the same time, the design of the algorithm generally starts from focusing on data compression, therefore, the scope of the application of the algorithm is limited.

2. Negative selection algorithm

Computer security problems and immune system problems encountered with striking similarities, they both have to be constantly changing environment to maintain system stability. Distribution, flexibility, adaptively and robust solution of the immune system are exactly what the field of computer security expects. According to self / non-self distinction principle of the immune system, Forrest etc. proposed a negative selection algorithm which can detect changes in computer system (S. Forrest et al, 1994). The algorithms simulate the "negative selection" principal of T cell maturation process: randomly generated detectors, remove detectors which detect themselves and preserve those detect non-self. Negative selection algorithm has laid a theoretical foundation for the application of the immunity in computer network security areas.

3. Immune evolutionary algorithm

As a kind of random search optimization method, evolutionary algorithm has been widely used. However, it still needs improving in practice. For example, evolutionary algorithm can not guarantee grantee getting the globally optimal solution, it may lose the best individual in the population and it also has the problems of premature convergence. More effective optimization algorithms will be got if evolution and immunity are combined. Under the framework of evolutionary algorithm, researchers have introduced many features of the

immune system and developed a number of immune optimization algorithms. Such as immune optimization algorithm with vaccination (Jiao Licheng & Wang Lei, 2000), immune optimization algorithm with self-regulation mechanism (Zhang Jun et al., 1999), immune optimization algorithm based on immune response (J. S. Chun et al., 1998), and immune optimization algorithm with immune memory (S. Endoh et al., 1998). These improved algorithms can quickly find the optimal solution meeting the requirements of certain accuracy and are useful to solve engineering problems (Jiao Licheng &Du Haifeng, 2003).

4. Immune clonal selection algorithm

Clonal selection algorithm is an important type of immune optimization algorithm, and it has been widely used in the artificial immune system. In 2000, De Castro etc. concentrated the clonal selection mechanism of immune system with the help of previous studies, and proposed a clonal selection based immune algorithm, which was successfully used to solve pattern recognition, numerical optimization and combinatorial optimization problems (L. N. De Castro & F. J. Von Zuben, 2000). In 2002, Kim etc. proposed a dynamic clonal selection algorithm and it was used to solve the anomaly detection problem in the continuous changing environment (J. Kim & P. J. Bentley, 2002) In 2005, Jiao Licheng, Du Haifeng etc. proposed Immune polyclonal Strategy based on the work of predecessors, and what was more they proposed Immune clonal selection algorithm for solving the problem about high dimensional function optimization (Du Haifeng et al, 2005), which achieved good results. Jiao Licheng and others have also presented some other high-level algorithm, Such as the Immune Memory Clonal Programming Algorithm (Du Haifeng et al, 2004), Adaptive chaos clonal evolutionary programming algorithm (Du Haifeng, 2005) and so on.

2.4 Artificial immune optimization algorithm

Engineering Optimization technology is a technology for solving various engineering optimal problems. As an important branch of science, engineering optimization technology has been attracting widespread attention, and been applied in many engineering fields, such as system control, artificial intelligence, pattern recognition, production scheduling, VLSI technology, fault diagnosis, computer engineering and so on. Engineering process optimization plays an important role in improving the efficiency and effectiveness and saving resources. Theoretical study of optimization algorithms also plays an important role in improving performance of algorithm, broadening the application field of algorithm, improving algorithms system. Therefore, study of the optimization theory and algorithm is important both theoretically and practically.

As science and technology continues to progress and the computer technology has been widely used, the scale and complexity of engineering optimization problems are increasing. Because of some inherent limitations and shortcomings, traditional optimization methods fail to meet such requirements to solve complex optimization problems. Researchers have to find new ideas to solve problems. Since the 1980s, a number of novel optimization algorithms have been proposed, such as artificial neural networks, simulated annealing, tabu search, evolutionary algorithms, ant colony optimization, particle swarm optimization, artificial immune algorithms, EDA algorithms and hybrid optimization strategy. These algorithms develop through simulating or revealing certain natural phenomena or a process, and the ideas and content relate to mathematics, physics, biological evolution, artificial intelligence, neuroscience and statistics, so it provides new methods to solve the complex problems. These new algorithms can often get rid of the limitations of traditional

optimization algorithm, using heuristic optimization strategies to explore the optimal solution, and has been used in a large number of practical applications and achieved encouraging results. With the development of interdisciplinary research, new intelligent optimization algorithms are emerging and bring new solutions to the optimization problems.

Artificial immune system is an adaptive system for solving the complex problems by simulating the function and principle of biological immune system. Immune algorithm retains many intelligent features of the biological immune system, so it has great diversity maintaining mechanisms, global search capability and robustness, and enables parallel search. Artificial immune algorithm is getting more and more attentions from researchers, and it is widely used for numerical optimization (Gong Maoguo et al, 2007), combinatorial optimization, multicast routing (Liu Fang et al, 2003), job shop scheduling (Z.X. Ong et al, 2005) multi-objective optimization (Shang Ronghua et al, 2007) and other engineering optimization problems.

2.5 The concept of immunology used in immune optimization algorithm

Immune optimization algorithm simulates immune mechanisms of biological immune system to deal with engineering optimization problems. Before Immune optimization algorithm is constructed, we need to map the various elements in engineering optimization problems to related concepts in immunology. As the biological immune system is very complex, it is impossible and unnecessary to completely apply biology definition in the artificial immune system. In order to better describe the artificial immune system algorithm, the following will briefly explain a few common used immune academic terms and their meaning in immune optimization algorithms.

1. Antigen

In the artificial immune system, it generally refers to the problem and its constraints, which is similar with fitness function in evolutionary algorithm. Specifically, it is a function of the objective function, and is the initiating factor and the important metrics of artificial immune algorithms.

2. Antibody

In the artificial immune system, it generally refers to candidate solutions of the problem, which is similar with individual in evolutionary algorithm. Collection of antibodies is called antibody group. In practice, the antibody generally appears in the form of coding.

3. Antibody-antigen affinity

It shows that the Antibody's binding capacity to Antigen, and reflects the binding site of a single antibody and the binding force of the unit antigen (or epitopes). In artificial immune system, it is used to show how the antibody at different locations (code) affects the antigen (or objective function).

4. Vaccine

Vaccine is defined as the estimate of the best individual gene, resulting from evolutionary environment or prior knowledge of the unknown problem.

5. Memory unit

In the artificial immune system, memory unit is an antibody group composed by specific antibody, which is used to maintain species diversity and the optimal solution in the process of solving problem.

6. Clone

Clone is the proliferation processes of biological Immune systems. In the artificial immune system cloning operator, based on clonal selection theory, clone is a composite operator which is a combination of selection, expansion, mutation and crossover operators.

3. Parallel immune memory clonal selection algorithm for large scale TSP

Traveling salesman problem (TSP) is a classical combinatorial optimization problem, with a strong engineering background and extensive application. TSP problem can be formally described as: given N cities $\mathbf{C}=\{C_1,C_2,\cdots,C_N\}$, and the distance between any two cities $d\left(C_i,C_j\right)$, find a closed path $\mathbf{C}_\pi=\left\{C_{\pi(1)},C_{\pi(2)},\cdots,C_{\pi(N)}\right\}$, through all cities in C only once, making minimal total distance $\sum_{i=1}^{N-1}d\left(C_{\pi(i)},C_{\pi(i+1)}\right)+d\left(C_{\pi(N)},C_{\pi(1)}\right)$ (D.S.Johnson and L.A.McGeoch, 1997). The solution space of TSP problem increases rapidly as the size of the problem increases, as a result, traditional methods (such as the exhaustive method, dynamic programming, branch and bound, etc.) have been powerless. It has proven that TSP problem is NP-hard combinatorial optimization problem, and it is difficult to find an effective algorithm to obtain the optimal solution in polynomial time. Therefore for Large-scale problems, people are more inclined to seek an algorithm that can find acceptable approximate solution in a limited time. Approximation algorithm for solving TSP is divided into two categories: tour construction algorithm and loop improved algorithm. Tour construction algorithms start from an illegal solution and gradually change the path until to get up a legitimate path. Such algorithms include: nearest neighbor algorithm, greedy algorithm, Clarke-Wright algorithm, Christofides algorithm (D.S.Johnson & L.A.McGeoch, 1997) and so on. After given an initial legitimate solution, circle improved algorithm uses a certain strategy to find solutions of better quality. Such algorithms include: local search strategy (r-Opt, LK, LKH, cycle LK (D. S. Johnson & L. A. McGeoch, 2002), etc.), tabu search (D.S.Johnson & L.A.McGeoch, 1997), simulated annealing (D.S.Johnson & L.A.McGeoch, 1997), genetic algorithm (T. Guo & Z. Michalewicz, 1998), ant colony algorithm (X.M. Song et al., 1998), particle swarm optimization (X.X.He et al., 2006), multi-level algorithms (C. Walshaw, 2001), immune algorithms (Wang Lei et al., 2000) and so on.

For large search spaces of massive TSP, the computing power of single computer is far from being able to satisfy the search algorithm on the request of the time. At the same time, with the development of network technology, there exists a large number of loosely coupled idle computation resource. It is practicable that cluster these computing resources to handle large and complex problems. Therefore, the research on parallel algorithms running in a loosely coupled environment has a very important significance. Parallel algorithms for solving large-scale TSP have attracted more and more attention. There has been some research results about parallel ACO (Lv Qiang et al., 2007), however currently just in its infancy. This chapter attempts to design a parallel immune algorithm to solve this complex problem.

At present, the achievements of parallel artificial immune system research are mostly parallel immune algorithms that have been existed. Artificial immune system model on parallel research is still rare. However, the parallel algorithm is not simply the only existing serial algorithm using multiple processors in parallel to achieve. In the parallel genetic algorithm results, many mature parallel modes are in emergence, such as: Master (Master-Slave) model, fine-grained (Fine-grained) model, coarse-grained (Coarse-grained) model,

mixed (Hybrid) model (Erick Cantú-Paz, 2000) and so on. Especially, the coarse-grained parallel model which is widely used can not only speed up the speed of algorithm for large-scale complex problems, and a variety of groups in the search can make the algorithm more stable, avoid local optima.

In this chapter, based on the successful experience of the Parallel genetic algorithm, artificial immune system, TMSM and PIMCSA are designed to solve the large-scale TSP problem according with the features of artificial immune system. TMSM is a coarse-grained two parallel artificial immune model, which stimulate the distributed immune memory and immune response mechanisms based on TMSM of PIMCSA. the migration of vaccines instead of individual migration in PIMCSA, not only reduces the cost of communication greatly, but also accelerate the convergence. Either the simulations of the symmetric or asymmetric TSP problem show that, PIMCSA compared with the most effective one of the local search algorithm cycle LK(D. S. Johnson, 1990) algorithm and the pure random search algorithm Guo Tao evolutionary algorithm (T. Guo & Z. Michalewicz, 1998) whose performance is best recognized , is much better. Meanwhile PIMCSA has good scalability.

3.1 Towerlike master-slave model

Coarse-grained parallel model of genetic algorithm is on the basis of the model of multi-populations evolution (Fig. 1), that is, each sub-population evolves independently, and sub-populations do the individual migration to a certain interval. On the research of coarse-grained parallel genetic algorithm, sub-species topology (X.M. Song et al., 2006), chromosome migration strategies (X.X. He et al., 2006), sub- population division strategy(C. Walshaw, 2001) and so on are key points of the algorithm design.

When designing a parallel artificial immune system, we should not only consider the division and organizational structure of sub-population antibody along with the way to information interactions of sub-populations of antibodies. The proposed tower master-slave model (TMSM) is a coarse-grained two parallel artificial immune system model, which is not only parallel but also embodies the distributed characteristics of antibodies populations and immune memory characteristics.

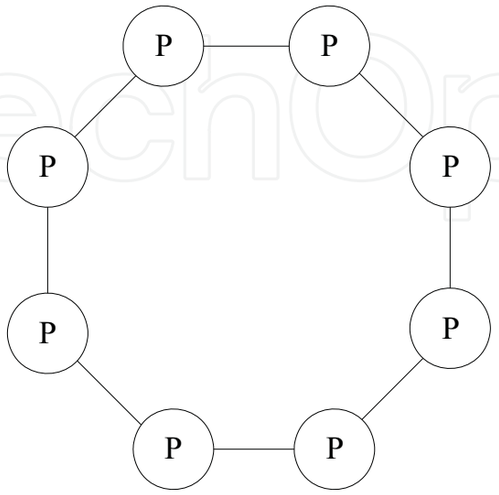


Fig. 1. Coarse-grained parallel genetic algorithm model

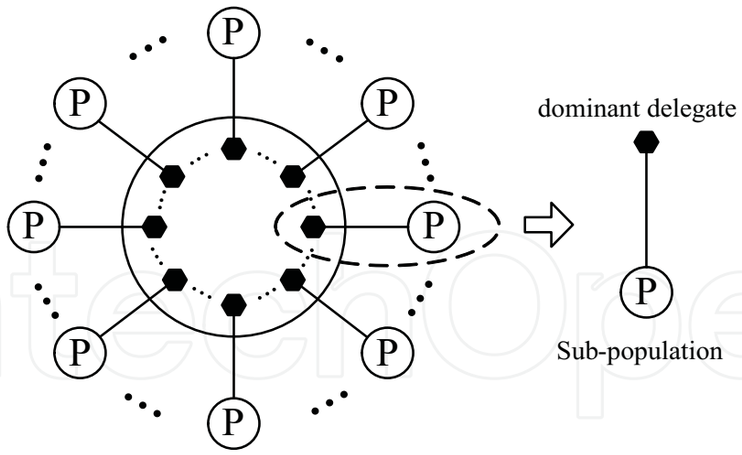


Fig. 2. Tower-like master-slave model

Definition 1: The towerlike master-slave model (TMSM), as shown in Fig. 2, is a two layer coarse-grained parallel model. The model consists of two types of populations organized as in Fig. 2, including a memory population M and several antibody populations P.

The top layer of the TMSM is the memory population $\mathbf{M}=\left(\hat{\mathbf{A}}_1,\hat{\mathbf{A}}_2,\cdots,\hat{\mathbf{A}}_m\right)$ formed by m memory antibodies. Each memory antibody of M corresponds to an antibody sub-population. The under layer of TMST are m sub-populations of antibodies with size n.

Definition 2: Each memory antibody in memory population M, $\hat{\mathbf{A}}_i$ ($i=1,2,\cdots,m$), is mapped to a sub-population $\mathbf{P}_i$. We call $\hat{\mathbf{A}}_i$ is the dominant delegate of population $\mathbf{P}_i$.

TMSM inherits the advantage of the evolution of a variety of groups of a coarse-grained parallel model, while with modifies according to the characteristics of the immune system. In TMSM, the antibody population is divided into a memory and several sub-populations. Moreover, a one to one map between antibody in memory and sub-population is established. Such a design not only makes the immune system get the memory function, but also produce the distributed immune memory with the below driven algorithm used to a distributed population of antibodies. Meanwhile, the corresponding immune mechanism, the self adaptive extraction and inoculation mechanism o f the immune vaccine and can be expanded with this model.

As the memory population served to initiate and terminate the process of calculation, as well as schedule the task of information exchange between sub-populations of antibodies, the concept of primary and secondary comes out. Memory population is the "primary" and the sub-species antibody is "secondary." This is essentially different from the primary-secondary parallel model of the parallel genetic algorithm (Erick Cantú-Paz, 2000).

3.2 PIMCSA for solving TSP

To solve the TSP problem, Parallel Immune Memory Clonal Selection Algorithm (PIMCSA) adopted the encoding method of path representation. The antibody affinity A, is defined as:

$$\text{Affinity}\left(\mathbf{A}\right)=\left(\text{Length}\left(\mathbf{A}\right)-HKB\right)/HKB \tag{1}$$

$\text{Length}\left(\mathbf{A}\right)$ indicates the path length after antibody A decoded, HKB indicates Held-Karp Bound of TSP problem which is the estimation of optimal path length of TSP.

In accordance with the description of the general framework of PIMCSA, PIMCSA includes two parts: the memory population immune algorithm (Memory Immune Algorithm, MIA) and sub-populations of antibody immune algorithm (Population Immune Algorithm, PIA). MIA and PIA process were designed to solve large-scale TSP problems.

3.2.1 Immune algorithm of memory population

Memory population immune algorithm (MIA) is running in memory antibody population driven algorithm of TMSM. MIA is the initiator of the parallel algorithm and termination of immune persons, not only in the memory to complete the memory antibody population evolution that is self-learning and memory mature operation, while responsible for extraction and distribution of vaccines to the antibody sub-populations. The pseudo code of memory immune algorithm population is described as follows:

Memory population immune algorithm (MIA):

Set algorithm termination conditions, and let evolution generations r＝0；
Initialize population of memory antibodies randomly $\mathbf{M}(r)$ and calculate of affinity, then set Collection of vaccines $\mathbf{V}(r)$ to be Empty set;
While (algorithm termination conditions are not satisfied)
{
Try to receive every antibody $\bar{\mathbf{A}}_i$ which is sended from $\mathbf{P}_i$ to $\mathbf{M}(r)$.If $\bar{\mathbf{A}}_i$ is received and it's affinity is larger than $\hat{\mathbf{A}}_i$,
Then $\hat{\mathbf{A}}_i$ is replaced by $\bar{\mathbf{A}}_i$;
Run mature implementation of memory: $\mathbf{M}(r+1)=\text{Maturation}(\mathbf{M}(r))$;
Run Dynamic vaccine extraction operation: $\mathbf{V}(r+1)=\text{Extraction}(\mathbf{M}(r+1))$;
Run vaccine distribution operation: $\text{Dispatch}(\mathbf{P}_1,\mathbf{P}_2,\cdots,\mathbf{P}_m)$;
r＝r+1;
}
Count, output the result, and send termination signal of the algorithm to each sub-population.

Mature implementation of memory: Cyclic LK algorithm process is adopted in the mature implementation of memory. For each memory antibody population, $\hat{\mathbf{A}}'_i$ is got after local search 4-Opt is done firstly, then after optimizing $\hat{\mathbf{A}}'_i$ through LK algorithm, we can get local optimal solution $\hat{\mathbf{A}}''_i$. If the affinity of $\hat{\mathbf{A}}''_i$ is greater than $\hat{\mathbf{A}}_i$, the alternative $\hat{\mathbf{A}}_i$ with $\hat{\mathbf{A}}''_i$, otherwise $\hat{\mathbf{A}}_i$ is retained. Mature memory populations in post-operation is $\mathbf{M}(r+1)$.

The strategy of the extraction of dynamic memory antibody vaccines will be divided into two parts, Vaccine is extracted from a part and is inoculated to another part of the memory of antibody sub-populations corresponding antibodies. Therefore, the moving into sub-populations of antibody vaccine is the evolutionary experience concluded from the representative of their own advantages other than some good memories antibodies. Extraction and distribution of this vaccine strategy is conducive to the exchange between the antibody sub-populations experience and accelerates the evolution of species.

Meanwhile, when the vaccine is extracted each time, it will be re-divided into two groups of memory. This will prevent too many antibodies being assigned to the sub-populations, so as to prevent the algorithm prematurity which is caused by the loss of diversity. The operation of dynamic extraction and distribution of vaccines will be given in detail in the next section.

3.2.2 Immune algorithm of antibody population

The immune algorithm of antibody sub-populations (PIA) is a driven algorithm running on every sub-population of antibodies in TMSM. PIA receives two kinds of information from the immune algorithm of memory populations in the process of iteration: Algorithm termination information and vaccines information.

Immune algorithm of antibody sub-populations maintains the iterative evolution of antibody sub-populations. The operation process of clonal selection can be adopted in the iterative evolution. The immune genetic operation to Sub-population of the antibodies is composed by the vaccination operation and Inver-over operator. It will be given in detail in the next section. The pseudo-code of the immune algorithm which runs in the sub-populations of antibody $\mathbf{P}_i(i=1,2,\cdots,m)$ is given as follows:

Immune algorithm of antibody sub-populations(PIA):

Let iteration times t＝0, set termination signal Halt＝False;
Initialize the population of memory antibodies $\mathbf{P}_i(t)=\left(\mathbf{A}^i{}_1(t),\mathbf{A}^i{}_2(t),\cdots,\mathbf{A}^i{}_n(t)\right)$ randomly, and calculate of affinity;
Assume $\bar{\mathbf{A}}_i(t)$ to be optimal antibody in $\mathbf{P}_i(t)$ ，Set current optimal antibody affinity: *CurrentBest*=Affinity$\left(\bar{\mathbf{A}}_i(t)\right)$;
Send $\bar{\mathbf{A}}_i(t)$ to the population of memory M;
Set current vaccine $v(t)$ to be Empty.
While (Halt is not True)
{
Try to receive the termination signal algorithm from the population of memory M, If received, Set Halt＝True. Otherwise, jump out of the loop;
Try to receive the vaccine from the population of memory M, If received, replace v with the new vaccine;
Run clonal operation: $\mathbf{P}_i'(t)=\mathrm{CL}\left(\mathbf{P}_i(t)\right)$;
Run immunity operation: $\mathbf{P}_i''(t)=\mathrm{IG}\left(\mathbf{P}_i'(t)\right)$;
Run clonal selection operation: $\mathbf{P}_i(t+1)=\mathrm{SL}\left(\mathbf{P}_i''(t)\right)$;
Find the optimal antibody from $\mathbf{P}_i(t+1)$,If Affinity$\left(\bar{\mathbf{A}}_i(t+1)\right)>$ *CurrentBest* , then set *CurrentBest*=Affinity$\left(\bar{\mathbf{A}}_i(t+1)\right)$ and send $\bar{\mathbf{A}}_i(t+1)$ to M;
t = t+1;
}

Cloning operation: cloning operation CL is defined as follows,

$$\mathrm{CL}\left(\mathbf{P}_i(t)\right)=\left(\mathrm{CL}\left(\mathbf{A}^i{}_1(t)\right),\mathrm{CL}\left(\mathbf{A}^i{}_2(t)\right),...,\mathrm{CL}\left(\mathbf{A}^i{}_n(t)\right)\right) \tag{2}$$

In the formula, $\mathrm{CL}\left(\mathbf{A}^i{}_j(t)\right)=\vec{\mathbf{I}}_j\times\mathbf{A}^i{}_j(t)$ $(j=1,2,\cdots,n)$, $\vec{\mathbf{I}}_j$ is Unit row vector with q_j dimensions), which is called the q_j colon of antibody $\mathbf{A}^i{}_j(t)$. Clone size q_j and the antibody affinity $\mathbf{A}^i{}_j(t)$ are related. q_j is greater while affinity is greater.

Let $\mathbf{Y}_j'(t)=\mathrm{CL}\left(\mathbf{A}^i{}_j(t)\right)=\left\{y'_{j1}(t),y'_{j2}(t),...,y'_{jq_j}(t)\right\}$, then the sub-populations of antibodies after the CL operation can be written as:

$$\mathbf{P}_i'(t)=\left\{\mathbf{Y}_1'(t),\mathbf{Y}_2'(t),...,\mathbf{Y}_n'(t)\right\} \tag{3}$$

Immune gene action: immune genetic operation is defined as follows，

$$\mathrm{IG}\left(\mathbf{P}_i'(t)\right)=\left(\mathrm{IG}(\mathbf{Y}_1'(t)),\mathrm{IG}(\mathbf{Y}_2'(t)),\cdots,\mathrm{IG}(\mathbf{Y}_n'(t))\right) \tag{4}$$

Assume $\mathbf{Y}_j''(t)=\mathbf{IG}\left(\mathbf{Y}_j'(t)\right)=\left\{y''_{j1}(t),y''_{j2}(t),...,y''_{jq_j}(t)\right\}$, then the sub-populations of antibodies after the IG operation can be written as:

$$\mathbf{P}''(t)=\left\{\mathbf{Y}_1''(t),\mathbf{Y}_2''(t),...,\mathbf{Y}_n''(t)\right\} \tag{5}$$

Immune genetic manipulation IG acts on the antibody with the operator which is chosen with equal probability between vaccination operator Vaccination and Inver-over operator .Inver-over operator which is famous for Guo Tao algorithm designed for TSP problems an effective operation of the genetic evolution. Vaccination operator is designed according to the dynamic vaccine extracted from the memory population M. Detailed operational procedures will be written in the next section.

Inver-over operator is the local search which runs in the encoded space around with antibody, using the heuristic information within the sub-populations of antibodies. Vaccination operator will introduce the knowledge learned from the memory population M to the antibodies, in the use of heuristic information from other sub-populations of antibodies.

Clonal selection operation: clonal selection operation are defined as follows,

$$\mathrm{SL}\left(\mathbf{P}_i''(t)\right)=\left(\mathrm{SL}\left(\mathbf{Y}_1''(t)\right),\mathrm{SL}\left(\mathbf{Y}_2''(t)\right),\cdots,\mathrm{SL}\left(\mathbf{Y}_n''(t)\right)\right) \tag{6}$$

If $\mathbf{A}^i{}_j(t+1)=\mathrm{SL}\left(\mathbf{Y}_j''(t)\right)$ $(j=1,2,\cdots,n)$,then the population in SL post-operation is ,

$$\mathbf{P}_i(t+1)=\left(\mathbf{A}^i{}_1(t+1),\mathbf{A}^i{}_2(t+1),\cdots,\mathbf{A}^i{}_n(t+1)\right) \tag{7}$$

The process of the operation $\mathrm{SL}\left(\mathbf{Y}_j''(t)\right)$ acted on $\mathbf{Y}_j''(t)$ is as follows: the antibody with maximum optimal affinity chosen from $\mathbf{Y}_j''(t)$ can be written as,

$$y_j''(t)=\left\{\mathbf{y}''_{jk}(t)\,|\,\max\text{affinity}(\mathbf{y}''_{jk}(t)),k=1,\cdots,q_j\right\} \tag{8}$$

If $\text{Affinity}\left(y''_j(t)\right) > \text{Affinity}\left(\mathbf{A}^i{}_j(t)\right)$, then let $\text{SL}\left(\Upsilon''_j(t)\right) = y''_j(t)$. Otherwise, $\text{SL}\left(\Upsilon''_j(t)\right) = \mathbf{A}^i{}_j(t)$.

4. Dynamic vaccination

In artificial immune system, the vaccine is an estimate of the best individual gene on the basis of evolution environment or the apriori knowledge of unknown problem. Vaccine is not an individual, which can not be decoded to a solution of a problem as antibodies can be done. It just has the characteristics on some places of the genes. Therefore vaccine can be regarded as a single gene or a set of gene sequences fragment. The right choice for the vaccine will have a positive role in promoting population evolution, and thus have a very vital significance to the operating efficiency of algorithm. But, the quality of selection of vaccine and generated antibodies will only affect the function of the vaccination of immune operator, but will not involve to the convergence of algorithm.

4.1 Selection and distribution of vaccine

For TSP problem and PIMCSA algorithm, we design a dynamic vaccine extraction (Dynamic Vaccination, DV) strategy and a vaccine allocation strategy as described below.

Dynamic vaccination strategy will first divide current memory antibody population into two antibody sets: the set of vaccines extraction $\mathbf{M}_1(r+1)$ and the vaccination set $\mathbf{M}_2(r+1)$. Let k is the largest positive integer less than or equal to m/2 (m is the size of memory population), then randomly select k memory antibodies to compose $\mathbf{M}_1(r+1)$, and the remaining antibodies compose $\mathbf{M}_2(r+1)$.Then, do intersection operation for all the memory antibodies, and get the set $\mathbf{E}(r+1)$ of the public sides on the k paths. And next, we merge the sides with public cities into public sub-paths, and store the received public sub-paths and the rest of public sides as a multi-gene vaccine group and single-gene vaccine respectively into vaccine set $\mathbf{V}(r+1)$. In $\mathbf{V}(r+1)$, a single gene vaccine with length 1 represents an edge of the path and its storage form is city sequences of the end of a edge. Vaccine group represents a section of sub-paths such that a sequence of the edges of the head-to-serial, and its storage form is the sorted arrangement of a number of cities, and its length is the number of edges that sub-paths include.

After producing the vaccines, we will distribute them to the antibody sub-populations. We design the following vaccine distribution operation. First of all, randomly choose a vaccine v_i from the vaccine set $\mathbf{V}(r+1)$, which may be a single gene vaccine or a multi-genes vaccine group. Then, randomly choose a memory antibody $\hat{\mathbf{A}}_j$ from the vaccination set $\mathbf{M}_2(r+1)$ and send the vaccine v_i to the antibody sub-population $\mathbf{P}_j$ that $\hat{\mathbf{A}}_j$ corresponds to.

4.2 Vaccination

As the vaccination operations will bring the loss of population diversity when accelerating the convergence of algorithm. In Section 1.2 where we will operate it as part of immune genes operation rather than independent step, will help alleviate the loss of the diversity of antibody population. For the TSP problem, we take the implementation of the Inver-over operator and vaccination operator in equal probability, which jointly constitute the immune genes operators. The immune genes operation process of the antibody sub-population $\mathbf{P}'_j(t)$ (j = 1, 2,..., m) is described as follows:

The operation process of $\mathrm{IG}\left(\mathbf{P}'_j(t)\right)$:

```
for each antibody A'_i(t) in P'_j(t)
{
if ( Rand(0,1) ≤ 0.5 )
   Viccination(A'_i(t));
else
Inver_over(A'_i(t));
}
```

Here, $\mathrm{Rand}(0,1)$ is a random number between 0 and 1. The vaccination $\mathrm{Viccination}(\mathbf{A})$ on antibody A is described as follows:

The operation process of $\mathrm{Viccination}(\mathbf{A})$:

```
Viccination(A)
{
If (the current vaccine v of antibody sub-population is not empty)
{
c is the first city in v , c' is the next city of c in v;
While ( c is not the last city in v )
{
    turn the city between the next city of c and c' in A'
    c = c';
    c' =The next city of c' in v
}
}
}
```

Here, the process of vaccination operation using Inver-over operator plant the edges of a single-gene vaccine into the operated antibody or plant the edges of multi-genes vaccine group into the operated individuals.

5. Simulating results and analysis

The simulating software of the proposed approach PIMCSA was developed by C&MPI and ran on the HPC (Cluster) parallel computing platform. We got several typical symmetric and asymmetric TSP instances from TSPLIB and tested them.

As the memory mature operation of PIMCSA used steps of ILK, and immune genetic manipulation of antibody sub-population introduced the Inver-over operator, we compare the performance of PIMCSA with iterated Lin-Kernighan (ILK) algorithm (D. S. Johnson, 1990) and GuoTao (GT) algorithm (T. Guo & Z. Michalewicz, 1998). ILK algorithm is one of the most effective algorithms based on local search, and GT algorithm is the best pure evolutionary stochastic searching algorithm. In PIMCSA, antibody sub-population number m is set as 8, antibody sub-population size n is set as 30. Population size of ILK and GT algorithm are both set as 30. Termination condition of these three algorithms is that current

optimal path length of memory population is less than the best known path length or equal to it, or the current optimal path remains unchanged in 50 iterations. All the experimental data are statistical results of 20 independent runs.

5.1 Performances on symmetric TSP instances

Table 1 shows the performances of compared algorithms on symmetric TSP problems with different type and size. In the table, "Percent over Opt" refers to the percentage of difference between path length and the optimal path length, and "Running Time (seconds) "refers to the average time of computing.

Instance	Cities	Opt	Percent over Opt (mean)			Running Time (mean)		
			ILK	GT	PIMCSA	ILK	GT	PIMCSA
ATT532	532	27686	0.086	0.114	**0**	127.5	83.5	**17.4**
GR666	666	294358	0.061	0.176	**0.003**	292.1	102.4	**59.6**
DSJ1000	1000	18659688	0.133	0.152	**0.008**	418.6	372.4	**52.2**
PR2392	2392	378032	0.142	0.357	**0.006**	102.5	87.3	**16.6**
RL5915	5915	565530	0.163	0.879	**0.047**	293.7	**226.9**	239.1
PLA7397	7397	23260728	0.059	0.356	**0.007**	8843.2	8221.1	**1762.4**
RL11849	11849	923288	0.191	0.824	**0.105**	6311.3	5352.8	**2581.3**
USA13509	13509	19982859	0.163	1.209	**0.067**	10352.3	8931.5	**3520.7**
PLA33810	33810	66050499	0.184	1.813	**0.152**	76315.8	53356.6	**18137.4**
PLA85900	85900	142382641	0.246	1.115	**0.214**	214307.5	113755.9	**29883.2**
Average			0.1428	0.6995	**0.0609**	31736.45	19049.04	**5626.99**

Table 1. Performance comparisons on Symmetric TSP Instances

As it can be seen from Table 1, for symmetric TSP problems, both the tour quality and computing time of the proposed PIMCSA are superior to other two compared algorithms. With the increase of problem scale, the advantage of PIMCSA is getting more obvious. ILK can obtain TSP solution tour with higher quality, but the time cost of ILK is large, so it needs too long computing time for solving large scale TSP instances. GT expenses little computing time at each generation, however, it is easy to fall into local optimum. The running time of GT is shorter, but the quality of solution tour is not very good. PIMCSA combines the strengths of these two types of methods, it use the random search with small time cost for global search, and then it use the heuristic search with large time cost for local search. Experimental results indicate that PIMCSA achieves a good trade off between solution quality and computing time.

5.2 Performances on asymmetric TSP Instances

For asymmetric TSP with N cities, we use Jonker and Volgenant's method to transform it into symmetric TSP with 2N cities (Noda E et al., 2002).

Note $\mathbf{C} = \left[c_{ij} \right]_{N \times N}$ as the cost matrix of asymmetric TSP problem, we can use equation (9) to transform it into $\mathbf{C}' = \left[c'_{ij} \right]_{2N \times 2N}$ which is the cost matrix of the transformed symmetric TSP instance. L is a sufficiently large real number. In this paper, we set $L = \max\left(c_{ij} \right)$.

$$
\begin{aligned}
&c'_{N+i,j} = c'_{j,N+i} = c_{ij} \quad i,j = 1,2,\cdots N \text{ and } i \neq j \\
&c'_{N+i,i} = c'_{i,N+i} = -L \quad i = 1,2,\cdots N \\
&c'_{ij} = L \qquad\qquad\qquad \text{otherwise}
\end{aligned} \tag{9}
$$

Instance	Cities	Opt	Percent over Opt (mean)			Running Time (mean)		
			ILK	GT	PIMCSA	ILK	GT	PIMCSA
BR17	17	39	0	0	**0**	2.14	3.57	**0.12**
P43	43	5620	0.082	0.372	**0.002**	33.2	**21.7**	25.3
RY48P	48	14422	0.037	0.506	**0.015**	23.6	24.2	**1.34**
FTV70	71	1950	0.132	0.358	**0.039**	42.9	38.1	**2.33**
KRO124P	100	36230	0.031	0.114	**0.007**	72.3	55.6	**1.15**
FT53	106	6905	**0**	0.037	**0**	31.2	63.4	**1.85**
FTV170	171	2755	**0.018**	0.326	0.044	69.3	42.7	**1.03**
RBG358	358	1163	0.139	1.977	**0.006**	137.5	114.6	**36.6**
RBG 403	403	2465	0.082	0.973	**0**	291.7	174.3	**59.4**
RBG 443	443	2720	0.074	0.661	**0**	372.5	351.1	**84.9**
Average			0.0595	0.5324	**0.0113**	107.634	88.927	**21.402**

Table 2. Performance comparisons on asymmetric TSP problems

Table 2 shows asymmetric TSP simulation experiment results. Every term's meaning is as same as Table 1. According to the data of table 2, we can come to the same conclusions as that of table 1.

5.3 Performances on Large Scale Art TSP Instances

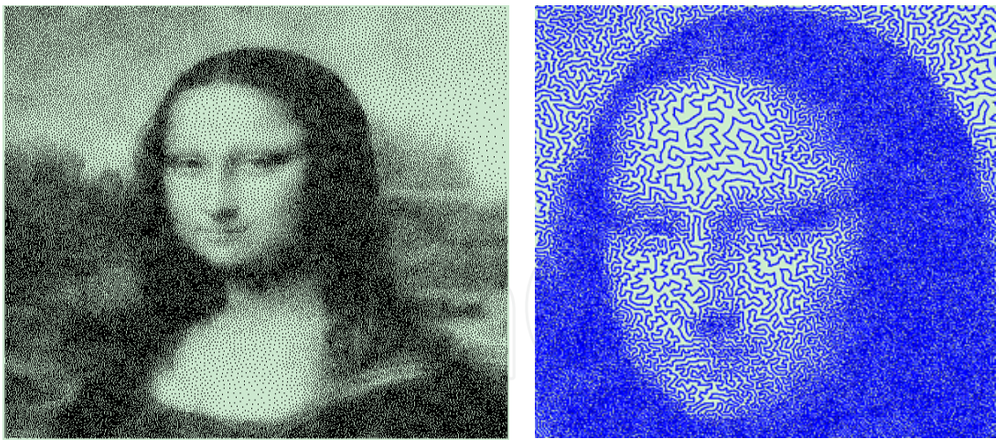
Robert Bosch has created a fascinating series of instances of the traveling salesman problem (TSP) that provide continuous-line drawings of well-known pieces of art. In this part, large scale art TSP instances with sizes from 100,000 to 200,000 were adopted to verify the efficiency of the proposed PIMCSA.

Fig. 3, Fig. 4 and Fig.5 are the city location and TSP tour obtained by the proposed PIMCSA. It can be seen that the obtained tours have no road crossing which indicates the tour is of good quality.

Table 3 is the numerical results of PIMCSA on six large scale art TSP instances. The best known tour lengths (BT) in table 3 are given by Keld Helsgaun and published on the website TSP Homepage (http://www.tsp.gatech.edu/data/art/index.html).

Instance	Cities	Best Known Tour Lengths (BT)	Tour Lengths	Percent over BT
Monalisa100K	100K	5,757,199	5,758,769	0.0273
Vangogh120K	120K	6,543,643	6,5455,94	0.0298
Venus140K	140K	6,810,730	6,812,641	0.0281
Pareja160K	160K	7,620,040	7,622,486	0.0321
Curbet180K	180K	7,888,801	7,891,518	0.0344
Earring200K	200K	8,171,733	8,174,864	0.0383
Average				0.0317

Table 3. Experimental results of PIMCSA on large scale art TSP instances

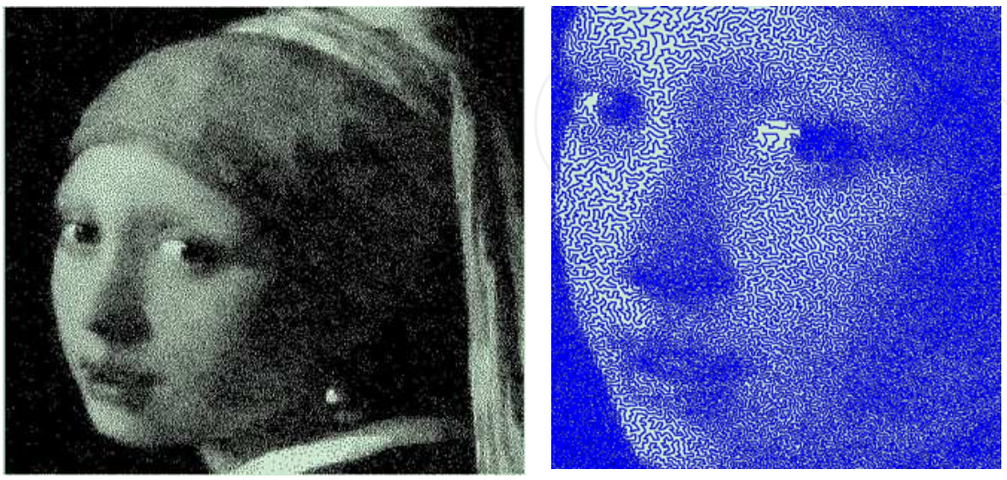


(a) City Location of Monalisa100K Instance (b) Part of the obtained Tour

Fig. 3. Performance of PIMCSA on Monalisa100K Instance

(a) City Location of Venus140K Instance (b) Part of the obtained Tour

Fig. 4. Performance of PIMCSA on Venus140K Instance



(a) City Location of Earring200K Instance (b) Part of the obtained Tour

Fig. 5. Performance of PIMCSA on Earring200K Instance

5.4 Effectiveness of the vaccine extraction strategy

This part of experiments is used to verify the effectiveness of PIMCSA vaccine strategy. Fig. 6 and Fig.7 shows the percentage of the edges' appearance in the known best tour. The higher the percentage the more superior the vaccine is. These data are statistic results of 20 independent runs.

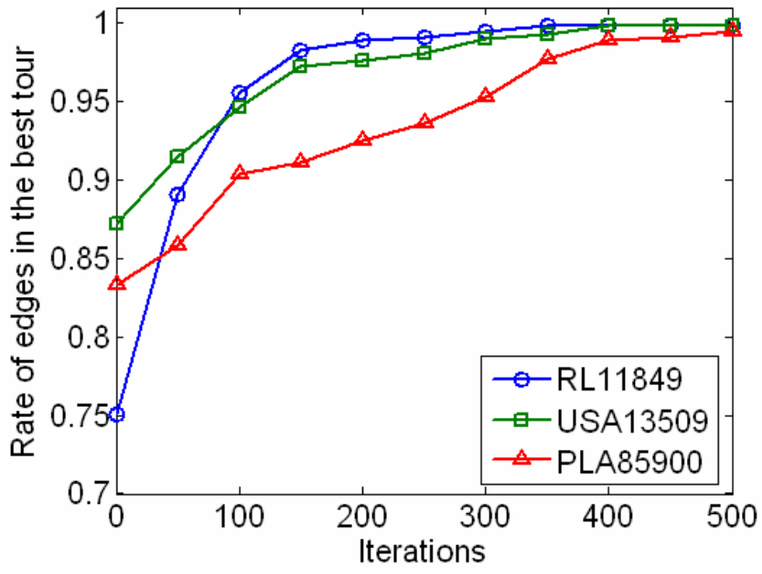


Fig. 6. Vaccine prediction accuracy for symmetric TSP

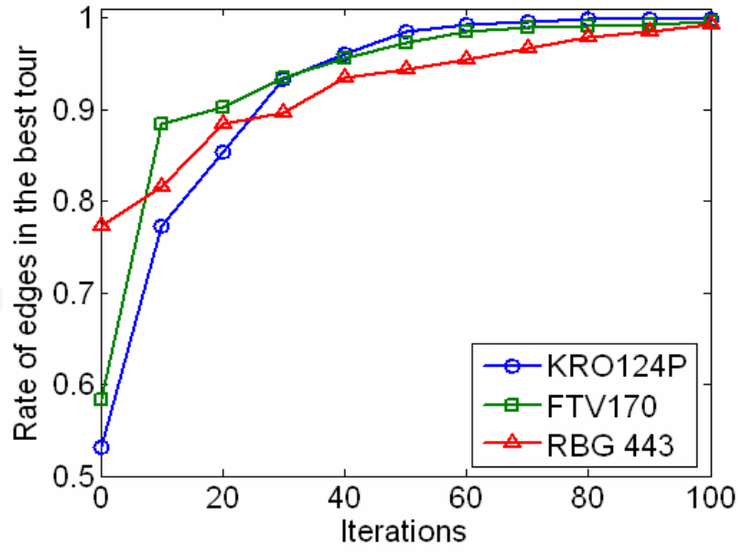


Fig. 7. Vaccine prediction accuracy for asymmetric TSP

Whether the problem is symmetrical or asymmetrical, the prediction accuracy of the vaccine increases rapidly along with the iteration times and the percentage is gradually close to one. It indicates that heuristic information of vaccine is helpful to speed up the algorithms' convergence. Further more, when the scale of the problem is large, the initial prediction

accuracy of vaccine becomes high, however, the increasing rate of prediction accuracy is slow. When problem size is small, the initial vaccine prediction accuracy is low, but it increased rapidly along with iterations.

5.5 Scalability of parallel algorithm

Speedup ratio is an evaluation of the time gain of parallel algorithms. For a given application, speedup ratio of parallel system indicates how many times parallel algorithm is faster than serial algorithm. If T_S is the time that we need from start of algorithm to the last on a serial computer and T_p is the time we need on a parallel computer, the speedup ratio S is defined as:

$$S = \frac{T_s}{T_p} \tag{10}$$

We use the efficiency to measure the rate of a processor's effectively used computing power. If the CPU number is p, the efficiency E is defined as:

$$E = \frac{S}{p} \tag{11}$$

If W is the total computation of problem, $T_0(W,p)$ is additional expenses (which is a function of W and p), then T_p can be expressed as:

$$T_p = \frac{W + T_0(W,p)}{p} \tag{12}$$

Thus, the speed-up ratio S and efficiency E can be expressed as:

$$S = \frac{W}{T_p} = \frac{pW}{W + T_0(W,p)} = \frac{p}{1 + T_0(W,p)/W} = \frac{p}{1+\Omega} \tag{13}$$

$$E = \frac{S}{p} = \frac{1}{1 + T_0(W,p)/W} = \frac{1}{1+\Omega} \tag{14}$$

From equation (13) and (14) we can see that $\Omega = T_0(W,p)/W$ is the key factor affect the efficiency of algorithms and processor. If W is certain, $T_0(W,p)$ is only associated with p, it can be written as $T_0(p)$. This function is determined by the algorithm. $T_0(p)$ increase more slowly, the scalability of algorithm is better, otherwise be worse. From equation (14) we can deduce:

$$\Omega = \frac{T_0(W,p)}{W} = \frac{p}{S} - 1 = \frac{1}{E} - 1 \tag{15}$$

Fig. 8 and Fig. 9 show the relationship between Ω and p. Antibody sub-population size is 30, memory population size is p-1. The data are average results of 20 independent runs.

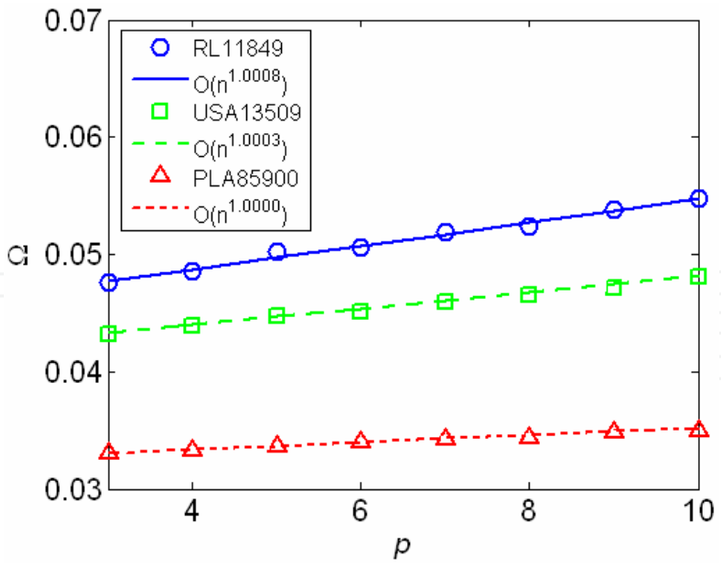


Fig. 8. Scalability of symmetric TSP

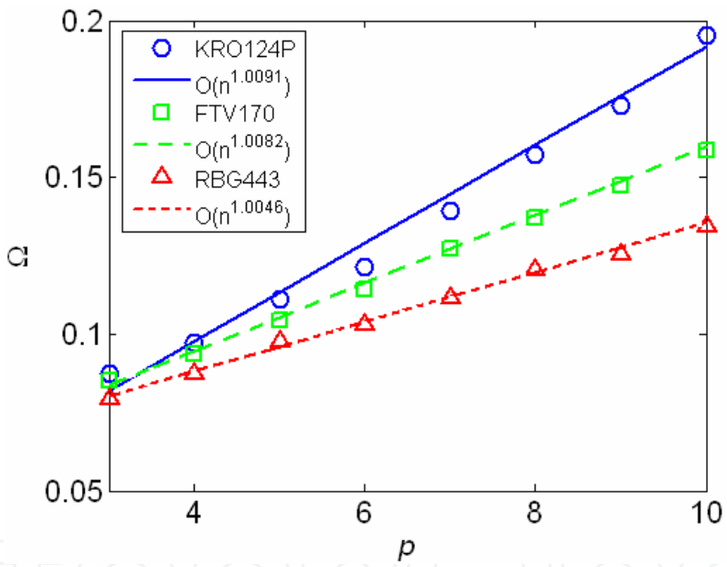


Fig. 9. Scalability of asymmetric TSP

When the scale of problem is certain, as the number of processors (p) increases, Ω shows linear increasing trend, it indicates that PIMCSA has good scalability. When problem becomes larger, Ω becomes smaller and increases more slowly, it indicates that the scalability of PIMCSA is better on large scale TSP problems.

These results above are reasonable. Additional expenses mainly include three parts: the expense of communication (denoted as C_1), vaccine producing of memory population (denoted as C_2) and vaccine delivery costs (denoted as C_3). C_1 will linearly increase when p increase under certain scale of problem. C_2 will linearly increase with the increase of p too. p has no influence on C_3. C_1 and C_2 are the major overhead costs for large-scale TSP problem. Compared to C_1 and C_2, C_3 can be neglected. Therefore, if the scale of problem is certain,

$T_0(W,p)$ will increase linearly with the increasing of p. When the scale of problem becomes larger, the total computation of problem (W) will increase, so Ω will increase more slowly.

6. Conclusion

This chapter first introduces the immune system and immune optimization algorithm, and then proposes the parallel immune memory clonal selection algorithm (PIMCSA) for solving large scale TSP problem. In the proposed PIMCSA, a dynamic vaccine extraction (DV) strategy is designed for solving large-scale TSP problem. Based on the general framework of PIMCSA, a special designed memory population immune algorithm (MIA) and a specific antibody sub-populations immune algorithm (PIA) are also proposed for solving TSP problems. Simulating results on the symmetric and asymmetric TSP instances in TSPLIB indicate that PIMCSA has good performance on both tour quality and running time. We also verify the validity of PIMCSA vaccine extraction strategy. Experimental results show that the rate of accuracy increases rapidly with the process of iteration and gradually close to 1. In addition, this chapter also analyses in theory that speedup ratio of parallel algorithms and the processor efficiency are related to variables Ω (the ratio of the extra overhead of algorithm and the total calculated amount of the problems). Experimental results show that, the parameter Ω of PIMCSA generally tends to enlarge linearly with the increase of the number of processors p, indicating that PIMCSA have good scalability.

It can be seen that the dynamic vaccine strategy designed in this chapter is very effective for the combinatorial optimization problems just as TSP problem. PIMCSA is a parallel artificial immune algorithm suitable for solving large-scale and complex optimization problems. For the parallel artificial immune algorithm, it is an important direction for further research that how to determine the size, quantity of the antibody sub-populations, and the relationship between them and the number of processors, computing power.

7. References

L.C. Jiao, H.F. Du, F. et al. (2006). *Immune optimization calculation, learning and recognition*. Science publishing Company, Beijing.

F. M. Burnet. (1978). Clonal selection and after, In: *Theoretical Immunology*, 63-85, Marcel Dekker Inc, New York.

L. N. de Castro, F. J. Von Zuben. (2000). *Artificial Immune System: Part I basic theory and applications*. Technical Report DCA-RT.

J. D. Farmer, S. Forrest, A. S. Perelson. (1986). The Immune System, Adaptation and Machine Learning. *Physical*, Vol. 22(D), 187-204.

D. J. Smith, S. Forrest, A. S. Perelson. (1998). Immunological memory is associative, In: *Artificial Immune System and their Applications*, Dasgupta, (Ed.), 105-112., Springer, Berlin

N. K. Jerne. (1973). The Immune System. *Scientific American*, Vol. 229, No.1, 52-60.

L. A. Herzenberg, S. J. Black. (1980). Regulatory circuits and antibody response. *European Journalist of Immune*, Vol. 10, 1-11.

G. W. Hoffmann. (1986). A Neural Network Model Based on the Analogy with the Immune System. *Journal of Theoretic Biology*, Vol. 122, 33-67.

A. S. Perelson. (1989). Immune Network Theory. *Immunological Review*, Vol. 1, No. 10, 5-36.

J. D. Farmer. (1990). A Rosetta Stone for Connectionism. *Physical*. Vol. 42 (D), 153-187.

Z. Tang, T. Yamaguchi, K. Tashima, et al. (1997). Multiple-Valued immune network model and its simulation, *Proceedings of the 27th International Symposium on multiple-valued logic*, Best Western, Autigonish, Canada, 1997.5.

N. Mitsumoto, T. Fukuta. (1997). Control of Distributed Autonomous Robotic System Based on Biologically Inspired Immunological Architecture, *Proceedings of IEEE International Conference on Robotics and Automation*, Albuquerque Convention Center, Albuquerque, NM, USA, 1997.4.

M. Zak. (2000). Physical Model of Immune Inspired Computing. *Information Sciences*, Vol. 129, 61-79.

J. Timmis, M. Neal. (2001). A resource limited artificial immune system for data analysis. *Knowledge Based Systems*, Vol. 14, No.3-4, 121-130.

L. N. De Castro, F. J.Von Zuben. (2000). An Evolutionary Immune Network for Data Clustering. *Proceedings of Sixth Brazilian Symposium on Neural Networks*, Rio de Janeiro, Brazil.

A. Tarakanov, D. Dasgupta. (2000). A formal model of an artificial immune system. *Bio-Systems*, Vol. 5, No.5, 151-158.

B. T. Nohara, H. Takahashi. (2000). Evolutionary Computation in Engineering Artificially Immune System, *Proceedings of the 26th Annual Conference of the IEEE Industrial Electronics Society*.

S. Forrest, A. Perelson, R. Cherukuri. (1994). Self non-self discrimination in a computer. *Proceedings of 1994 IEEE Computer Society Symposium on Research in Security and Privacy*, Oakland, CA, USA.

L.C. Jiao, L. Wang. (2000). A novel genetic algorithm based on immunity. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, Vol. 30, No. 5, 552-561.

J. S. Chun, H. K. Jung, S. Y. HaHn. (1998). A study on comparison of optimization performance between immune algorithm and other heuristic algorithms. *Magnetics*. Vol. 34, No. 5, 2972-2975.

S. Endoh, N. Toma, K. Yamada. (1998). Immune algorithm for n-TSP. *IEEE International Conference on System, Man, and Cybernetics*.Vol 4, 3844-3849.

L.C. Jiao, H.F. Du. (2003). Development and Prospect of the Artificial Immune System. *Acta Electronica Sinica*, Vol. 31, No. 10, 1540-1548.

L. N. De Castro, F. J. Von Zuben. (2000). The Clonal Selection Algorithm with Engineering Application, *Proceedings of Workshop on Artificial Immune System and Their Applications*, 36-37.

J. Kim, P. J. Bentley. (2002). Towards an Artificial Immune System for Network Intrusion Detection: An Investigation of Dynamic Clonal Selection. *Proceedings of Congress on Evolutionary Computation*, 1015-1020.

H. F. Du, M. G Gong, L. C. Jiao, R. C. Liu (2005). A novel artificial immune system algorithm for high-dimensional function numerical optimization. *Progress in Nature Science*. Vol. 15, No. 5, 463-471.

F. Du, M. G Gong, L. C. Jiao. (2004). A immune memory clonal programming algorithm for high-dimensional function optimization. *Progress in Nature Science*. Vol. 14, No. 8, 925-933.

H. F. Du, M. G Gong, R. C. Liu (2005). Adaptive chaos clonal evolutionary programming algorithm. *Science In China, Ser. E*, Vol. 35, No. 8, 817-829.

M. G Gong, L. C. Jiao, H. F. Du, W. P. Ma (2007). A Novel Evolutionary Strategy Based on Artificial Immune Response for Constrained Optimizations. *Chinese Journal of Computers*, Vol. 30, No. 1, 37-46.

Y.Y. Li, L. C. Jiao (2007).Quantum-Inspired Immune Clonal Algorithm for SAT Problem. *Chinese Journal of Computers*. Vol. 30, No. 2,176-182.

F. Liu, X. J. Feng. (2003). Immune Algorithm for Multicast Routing. *Chinese Journal of Computers*, Vol. 26, No. 6, 676-681.

Y. T. Qi, L. C. Jiao, F. Liu (2008). Multi-Agent Immune Memory Clone based Multicast Routing. *Chinese Journal Electronics*. Vol. 17, No. 2, 289-292.

Z.X. Ong, J.C. Tay, C.K. Kwoh. (2005). Applying the Clonal Selection Principle to Find Flexible Job-Shop Schedules. *Proceedings of the 4th International Conference on Artificial Immune Systems*, Banff, Canada.

J. Chen, M. Mahfouf (2006). A Population Adaptive Based Immune Algorithm for Solving Multi-Objective Optimization Problems. *Proceedings of the 5th International Conference on Artificial Immune Systems*, Portugal, 2006.

R. H. Shang, L. C. Jiao, M. G. Gong, W. P. Ma (2007). An Immune Clonal Algorithm for Dynamic Multi-Objective Optimization. *Journal of Software* . Vol. 18, No. 11, 2700-2711.

D.S.Johnson, L.A.McGeoch. (1997). *The Traveling Salesman: A Case Study in Local Optimization. Local Search in Combinatorial Optimization*.

D. S. Johnson, L. A. McGeoch. (2002). Experimental analysis of heuristics for the STSP. In: *The Traveling Salesman Problem and its Variations*. G. Gutin, A. Punnen, (Ed.), 369-443, Kluwer Academic Publishers, Boston.

T. Guo, Z. Michalewicz. (1998). Inver-over operator for the TSP. In: *Proc of the 5th Parallel Problem Solving form Nature*, 803-812, springer, Berlin.

X.M. Song, B. Li and H.M.Yang. (2006). Improved Ant Colony Algorithm and its Applications in TSP. In: *Proc of Intelligent Systems Design and Applications*. 1145 – 1148.

X.X.He et al.. A New Algorithm for TSP Based on Swarm Intelligence. (2006). In: *Proc of Intelligent Control and Automation*, 3241–3244.

C. Walshaw. (2001). A Multilevel Lin-Kernighan-Helsgaun Algorithm for the Travelling Salesman Problem. *Computing and Mathematical Sciences*, University of Greenwich, Old Royal Naval College, Greenwich, London, SE10 9LS, UK.

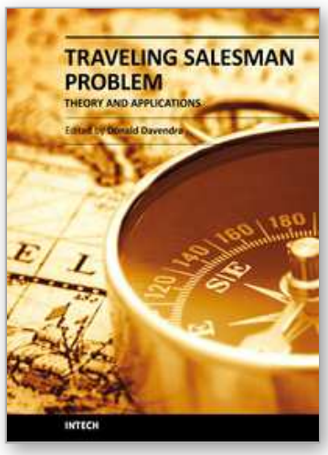
P. Zou, Z. Zhou, G. L. Chen, J. Gu (2003). A Multilevel Reduction Algorithm to TSP. *Journal of Software*, Vol. 14, No. 1, 35-42.

L. Wang, J. Pan, L. C. Jiao (2000). The Immune Programming. *Chinese Journal of Computers*, Vol. 23, No. 8, 806-812.

Erick Cantú-Paz. (2000). Efficient and Accurate Parallel Genetic Algorithms. *Kluwer Academic Publishers*, Holland.

D. S. Johnson. (1990). Local optimization and the traveling salesman problem. *Proceedings of the 17th Colloquium on Automata, Language, and Programming*, *Lecture Notes in Computer Science 443*, 446-461, Springer-Verlag, Berlin.

E. Noda, A. L. V. Coelho, I. L. M. Ricarte, A. Yamakami and A. Freitas. (2002). A Devising adaptive migration policies for cooperative distributed genetic algorithms. *Proceedings of Congress on Systems, Man and Cybernetics*, vol.6.



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.

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

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

IntechOpen