

Clonal Selection Algorithm for Solving Permutation Optimisation Problems: A Case Study of Travelling Salesman Problem

Wei Pang, Kangping Wang, Yan Wang*, Ge Ou, Hanbing Li, Lan Huang*

College of Computer Science and Technology, Jilin University
Changchun, Jilin, 130012, China

*Corresponding authors: wy6868@hotmail.com; huanglan@jlu.edu.cn

Abstract—As an attempt to solve Permutation Optimisation Problems (POP) by using CLONALG (Clonal Selection Algorithm), a well-established artificial immune system, Traveling Salesman Problem (TSP) is studied as an example in this paper. Operators of CLONALG, especially the hypermutation operators are analyzed and modified to make CLONALG adapt to POP. Furthermore, the local search technique is employed to speed up the mature of the repertoire system, and receptor-editing operator is also employed to avoid the premature of the antibody population. Finally, several benchmark problems in TSPLIB are tested to evaluate the best and average performance of the proposed algorithm. Experimental results show the proposed competitive algorithm performs better than both the standard CLONALG and a genetic algorithm.

Keywords- *Clonal Selection Algorithm; Artificial Immune System; Travelling Salesman Problem; Optimisation; Local Search*

I. INTRODUCTION

As a well-established artificial immune system, CLONALG [1,2] has been widely used to solve a variety of problems. In particular, up to now CLONALG and its variants have been intensively studied for solving Traveling Salesman Problem (TSP) [2,3,4,5]. These algorithms shed light on the application of CLONALG to general permutation optimisation problems. In a permutation problem, all values of the problem have to be assigned to the same number of variables, so that on one hand a certain constraints have to be satisfied, and on the other hand the cost function of the problem achieves the optimal value. TSP is one of the best known and most studied permutation optimisation problems for its simple problem description yet hard-to-solve nature.

Similar to other evolutionary algorithms, one of the advantages of CLONALG is that its underlying search mechanism is problem-independent to some extent: once the operators of CLONALG, including the antibody encoding, hyper-mutation, and affinity evaluation operators, are defined for a particular problem, the search in the problem space will be completely determined by the inherent mechanism of CLONALG, and the algorithm designer does not need to further consider the domain specific details of the problem.

However, it is acknowledged that such general-purpose evolutionary algorithms as CLONALG cannot normally outperform the algorithms specifically designed for TSP (and other permutation optimisation problems), for example, the Lin-Kernighan algorithm [6]. Although special-purpose algorithms are very efficient, the lack of generality limits their use: for different problems researchers may design different algorithms to solve them, which might involve very time-consuming and tedious work. More importantly, the performance of these special-purpose algorithms often very much depends on the unique features of the search space and interactions of variables, which might only exist in the particular problems of interest. This means once the definition of the original problem changes or a new problem is proposed, the special-purpose algorithms like the Lin-Kernighan algorithm will not be suitable because of the changes of the features of the search space as well as variable interactions.

In this paper researchers aim to address the above issue and make use of both the generality of CLONALG and the efficiency of special-purpose algorithm to design a hybrid Artificial Immune System algorithm (AIS) for TSP, termed CLONALG-TSP-2opt. On one hand, researchers hope that CLONALG-TSP-2opt has the potential to be applied to other POP problems because it is based on the CLONALG framework, a general-purpose AIS algorithm. On the other hand, researchers want to make the performance of CLONALG-TSP-2opt outperform the standard CLONALG by incorporating a local search technique, which is problem-specific but easy to implement and generalise under the general POP context.

The rest of this paper is organised as follows: in the remaining part of this section, the original CLONALG is briefly described and related work on TSP is overviewed. In Section 2 the design strategy of the algorithm is presented, and this is followed by the local search technique detailed in Section 3. In Section 4 the algorithm is described in detail. In Section 5 the experimental results are reported. Finally Section 6 concludes the paper and explores future work.

A. Clonal selection algorithm

Inspired by the clonal selection theory [7], de Castro *et al.* [1,2] proposed the clonal selection algorithm

(CLONALG). Regarding the clonal selection principle, CLONALG is composed of four processes: selection, clonal expansion, hyper-mutation (affinity maturity) and re-selection. The principal idea of CLONALG is that accumulative but blind mutation can direct the search to better solutions. As an important branch in artificial immune system, after the CLONALG was proposed, it has been applied to many fields, such as multi-model function optimisation and combinatorial optimisation problems.

Like genetic algorithms (GAs), CLONALG is known as an evolutionary algorithm and incarnates the Darwin's evolution theory. However, CLONALG employs clone and hyper-mutation operator instead of selection and crossover operation as in GAs. The optimisation version [1] of CLONALG is briefly described as follows:

Step1: A specific encoding strategy is applied to "encode" an antibody to represent a potential solution to the problem under investigation.

Step2: The antibody repertoire is composed of M such antibodies, and M is a given number determined by the complexity of the problem.

Step3: CSA executes in an iterative manner. For each generation:

i) First, all antibodies are selected to enter the next generation (*selection*).

ii) Each antibody is cloned for the same number of copies (*clonal expansion*).

iii) All these cloned copies undergo the hyper-mutation scheme, in which they will be mutated with a high probability.

iv) All the mutated antibodies are evaluated for their affinity values with the antigen (*Affinity Evaluation*). The evaluation is achieved by setting up a scoring system based on different problem context, which is the same as the fitness evaluation in GAs.

v) M antibodies (the same number as that of antibodies in the initial antibody repertoire) with the highest affinity values will be re-selected to form a new generation of the antibody repertoire (*re-selection*).

Step4: The algorithm will terminate if the termination conditions are satisfied, such as the optimal solutions have been found or the algorithm has run for a given number of generations.

From the above description, researchers can see that CSA can be considered as a population-based evolutionary algorithm. Each individual (antibody) is essentially a local optimum searcher, and the hyper-mutation can be deemed as a process of blind neighbourhood exploration in the search space in relation to the original antibody. In the re-selection process, the proliferated antibodies are selected according to their affinity values, which is a greedy strategy. After a sufficient number of generations, the antibodies are expected to find the optimal solution(s), or the approximate optimal solution(s).

B. CLONALG for solving TSP

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the

entire proceedings, and not as an independent document. Please do not revise any of the current designations.

As one of the classical and representative permutation optimisation problems, also as a notorious NP-hard problem, TSP has been intensively studied in the last several decades. There have been many effective algorithms specially designed to solve the TSP problems, for instance, the aforementioned Lin-Kernighan algorithm [6].

In addition, TSP has become a benchmark for evaluating the performance of many soft computing algorithms such as Genetic Algorithm (GA) [8,9], Ant Colony Algorithm [10,11] and Particle Swarm Optimisation [12,13]. Although the above soft computing algorithms are not as effective as the specially designed algorithms when dealing with TSP, the research on the design and analysis of these algorithms for solving TSP is still very important. This is because through the design and analysis processes researchers can figure out principles and intrinsic mechanisms that govern these algorithms in the search process and thus find ways to improve the efficiency of these algorithms for solving other permutation optimisation problems in general.

Consequently after the proposal of CLONALG, de Castro and Von Zuben performed some preliminary research on using CLONALG to solve TSP [2] based on the above considerations. In their work, the CLONALG was tested with a randomly generated instance of 30 cities, and the algorithm obtained comparatively satisfying results. Since de Castro and Von Zuben's first attempt of applying CLONALG to TSP, there have been many variants of CLONALG developed to investigate how CLONALG combined with other mechanisms can better deal with TSP, and among them the following algorithms are noteworthy: (1) inspired by the receptor editing process [14], Gao et al. [4] introduced the receptor-editing operator to CLONALG to obtain better solutions; Zhu et al. [15] employed a CLONALG using receptor editing and an additional self-crossover operator to improve the performance of CLONALG for TSP; recently Liu et al. [5] studied how to apply CLONALG to large-scale TSP problems; in addition, Chen et al. [3] proposed a probabilistic modelling clonal selection algorithm for TSP.

Different from the above-mentioned existing algorithms, in this paper researchers focus on investigating how to incorporate problem-dependent local search techniques into CLONALG. In particular, researchers are interested in how to design the local search algorithm so that the balance between efficiency and the quality of solutions can be achieved. In addition, researchers aim to investigate the interactions between the local search technique and other immune operators, especially the hyper-mutation operator when performing the search. The research on the above mentioned topics would result in the proposal of CLONALG-TSP-2opt, a novel hybrid CLONALG approach to TSP. To verify the effectiveness of this novel algorithm, researchers would like to make comparisons between the proposed algorithm and other two algorithms, namely GA and the original CLONALG. This is achieved by testing these three algorithms with the benchmark TSPs taken from the well-established TSPLIB [16].

II. ALGORITHM DESIGN

In this section researchers present a solution of adapting CLONALG to solve TSP by designing the antibody coding strategy as well as the selection, clonal expansion, and hyper-mutation operators. In addition, researchers also propose a receptor editing approach suitable for POP.

A. Antibody encoding strategy

Similar to GA for solving TSP, the intuitive integer coding strategy is adopted. For a TSP with N cities, an antibody is represented by a vector $\mathbf{A}_b = \{s_1, s_2, \dots, s_N\}$, where each element of the vector $s_i (i \in [1, N])$ is an integer in the range of 1 to N and denotes a city in TSP.

B. Selection operator

The selection operator follows the original de Castro's CLONALG for TSP. Due to the complexity of TSP problem space [17], the corresponding shape space [18] of the immune system is extraordinary complicated. To acquire better search ability, the exploration of the shape space should be strengthened. Consequently, all the antibodies in the population will participate in the processes of clonal expansion and hyper-mutation, that is, all the antibodies will be selected to explore the shape space.

C. Clonal expansion

An excellent style manual for science writers is [7].

The clone size is an important factor that largely influences the algorithm speed as well as the quality of solutions. With the increase of the clone size, the speed of the algorithm will slow down, while the solution quality may be improved. Consequently an appropriate clone size should be chosen to make a trade off between the convergence speed and the solution quality.

Based on the above consideration, the dynamic self-adaptive clone size is proposed: in each generation, each antibody is cloned with the same size. However, with the increase of the iterations, the clone size may increase by a given step until it attains the limitation of the maximum clone size.

D. Hyper-mutation

Considering the context of POP, two hyper-mutation operators are investigated: the point mutation and inverse mutation. Although the point mutation is commonly used in the original CLONALG, in this research the inverse mutation is employed for solving TSP because it is more intensive than the former and more suitable for exploration in complicated shape space such as the one in TSP. Later experiments will show that the inverse mutation performs better than the point mutation.

E. Receptor editing

Since all the antibodies are involved in the searching process, after the hyper-mutation the receptor-editing operator is employed to increase the probability of jumping out from the local optima and thus to avoid the premature of the antibody population. Researchers design the receptor-editing operator as follows: first choose a random continuous subsequence of length d

($d \in [1, N]$), and N is the number of the cities in TSP) from the antibody, and then the subsequence of d elements is rearranged in a range into one of $d!$ possible arrangements selected at random.

III. LOCAL SEARCH

With the increase of the clone size, the algorithm will be more computationally expensive. It is necessary to speed up the convergence of the algorithm by introducing additional operations. Local search is one good option, and as mentioned before, the local search can make better use of the domain specific knowledge. For solving the TSP, the classical 2-opt approach [19] can effectively eliminate the cross edges and obtain a better solution, but it is very complicated and time-consuming. Apparently this kind of computationally expensive operation is not suitable for population-based iterative search, as during the search process the local search function will be executed many times, and an expensive local search function will result in a significant increase in the algorithm running time. Based on this consideration, researchers propose a compromise approach termed quasi-2opt, which partially improves the quality of solutions and thus is not as expensive as the original 2-opt approach. The pseudo code of the quasi-2opt is given as follows:

```

For an antibody S of length N,  $S = \{s_1, s_2, s_3, \dots, s_N\}$ 
BEGIN: For i=1 to N-1, i++
    For j=i+1 to N, j++
    {
        Swap and , get the new sequence SS;
        Calculate the cost of S;
        If (Cost (SS) < Cost (S)) {
            S=SS;
            Goto BEGIN, Restart the process;
        }
    }

```

In the above, S is an antibody as described in Section 2.1, and SS is the newly obtained antibody after a swap operation. $\text{Cost}(S)$ and $\text{Cost}(SS)$ are the cost functions (the total length of the tour) of S and SS , respectively. It should be pointed out that when researchers calculate the cost of SS generated by the swap operator, there is no need to recalculate all the edges, because the swap operator only influences the edges that are related to the two swapped cities. Furthermore, from the above pseudo code researchers can draw the conclusion that after the quasi-2opt process, swaps of any two cities in the final resulting antibody S cannot further improve the solution.

It should be also noticed that the hyper-mutation and selection operations imply the hill-climbing process, and this process is problem-independent. The local search method is also a hill-climbing strategy, but it is problem-dependent. Take the TSP as an example: the quasi-2opt operation considers the features of TSP, that is, the connections between the node cities and the network structure of the problem. This makes the quasi-2opt only explore feasible solutions of TSP without violating problem constraints.

IV. ALGORITHM DESCRIPTION

Having described all the design details of CLONALG-TSP-2opt, in this section researchers give a formal description of the algorithm:

A. Population initialisation

Set the population size M , the initial clone size $initial_clone_size$, current clone size $clone_size=initial_clone_size$, maximum generation number gen_num , maximum clone size max_clone_size , the increase step of the clone size $increase_step$, and the probability of local search and receptor editing. Let P_i denote the antibody population in the i -th generation. First all the antibodies are generated randomly: $\mathbf{P}_0 = \{A_{b1}, A_{b2}, A_{b3}, \dots, A_{bM}\}$, where \mathbf{P}_0 is the initial population, and for each antibody A_{bi} in \mathbf{P}_0 , $A_{bi} = \{s_1, s_2, \dots, s_N\}$, where $i \in [1, M]$, $s_j \in [1, N]$, $j \in [1, N]$, and N is the number of the cities in TSP.

B. Selection

All the antibodies in P_i will be selected.

C. Clonal expansion

Adjustment of the clone size: after a given number of generations ($increase_generation$) is exceeded, if there is still no improvement of the solution, the clone size will increase by $increase_step$, but the maximum clone size max_clone_size cannot be exceeded.

Each antibody in the population will be cloned $clone_size$ copies. After the clonal expansion the cloned population is represented by $\mathbf{P}_c = \{\mathbf{Q}_1, \mathbf{Q}_2, \mathbf{Q}_3, \dots, \mathbf{Q}_M\}$, where \mathbf{Q}_i ($i \in [1, M]$) is a $clone_size$ -dimensional vector denoted by $\mathbf{Q}_i = (A_{bi}, A_{bi}, \dots, A_{bi})'$.

D. Hyper-mutation

All the antibodies will suffer a hyper-mutation process; the inverse sequence operator described in Section 2.4 is taken in this algorithm. Let the symbol \otimes denote the hyper-mutation operator, and the population after hyper-mutation is represented by $\mathbf{P}_H = \{\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3, \dots, \mathbf{R}_M\}$, where $\mathbf{R}_i = \otimes \mathbf{Q}_i = (\otimes A_{bi}, \otimes A_{bi}, \dots, \otimes A_{bi})'$, $i \in [1, M]$.

E. Local search

The quasi-2opt operation is applied to \mathbf{P}_H with a given probability, if the symbol χ denotes the local search operator, the population after the local search is as follows: $\mathbf{P}_R = \chi \mathbf{P}_H$.

F. Receptor editing

The receptor editing is applied to \mathbf{P}_L with a very small given probability, after this operation, if the symbol θ denotes the receptor-editing operator, the population after receptor editing is as follows: $\mathbf{P}_R = \theta \mathbf{P}_L$.

G. Re-selection

Select the M best antibodies from \mathbf{P}_R , and finally the new generation \mathbf{P}' is obtained.

Step 2~Step 7 are executed repeatedly until the maximum number of generations is reached or the quality of solutions is acceptable.

V. EXPERIMENTAL RESULTS

In this section the best and average performance of the algorithm will be evaluated by benchmark problems. All the algorithms were implemented in C++ with STL (Standard Template Library), and all experiments were performed on a Macbook Pro with an Intel Core i7 3.0GHz CPU and 8GB RAM.

A. Best Performance Comparison

Several Benchmark problems from TSPLIB are selected to evaluate the best performance of the algorithms. All the algorithms were run ten trials and the best results were recorded.

Eilon problems, including Eil51, Eil75, and Eil101, are three problems from TSPLIB and have been used as benchmark problems by many researchers such as Fogel [20]. First the mutation strategy in hyper-mutation is investigated. The point mutation and inverse sequence mutation operators are tested with the Eil51 problem, of which the best-known solution is 427.855. In the experiment the parameters are taken as follows: the population size $M=100$, $initial_clone_size=10$, $gen_num=1000$, $max_clone_size=20$, $increase_step=1$, $increase_generation=100$. In addition, the probability of receptor editing is 0.1%. The experimental results are shown in Fig. 1.

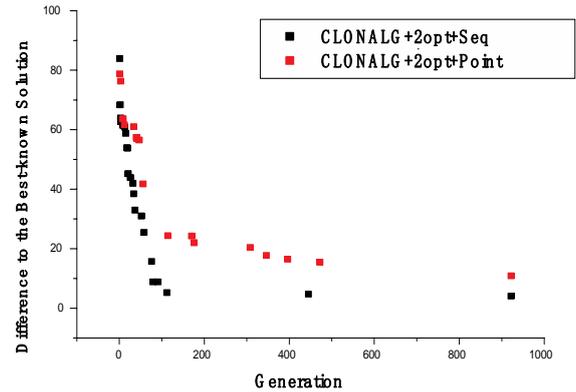


Figure 1. The Effect of Different Hyper-mutation Strategies on the Search.

Second, the inverse sequence is chosen as the mutation strategy and under this circumstance the influence of quasi-2opt is tested, the results are shown in Fig. 2. The probability of local search is 1% and the other parameters are the same as the previous experiment. The results shown in Fig. 2 indicate that before introducing the quasi-2opt operator, the algorithm starts from a random generated solution set and can explore many candidate solutions in the earlier generations, but the improvement among these solutions is not obvious, while after introducing the 2opt, the solution can be improved within a

short period. Finally the CLONALG+2opt can find better solution than CLONALG without 2opt.

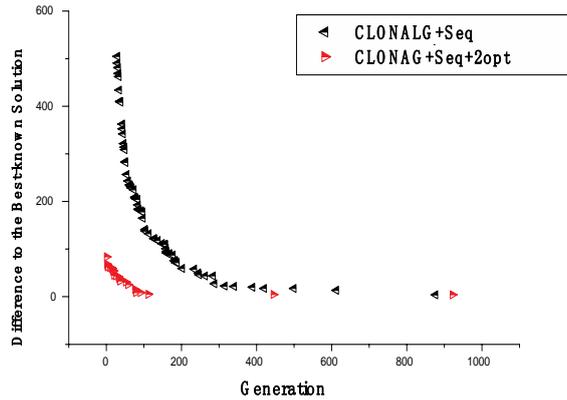


Figure 2. The Effect of Quasi-2opt On the Search.

Third, the algorithm is compared with a general genetic algorithm designed for TSP. In this GA the population size is 100; the most commonly used Partial Match Crossover (PMX) [8] is employed and the cross probability is 0.8; the inverse sequence mutation strategy is selected and the mutation probability is 0.5. The results were shown in Fig. 3, from which researchers can see that the best solution obtained by CLONALG+2opt is also better than that from GA.

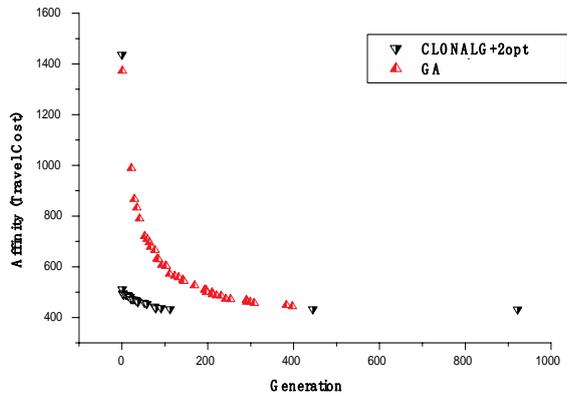


Figure 3. Comparison between GA and CLONALG+2opt.

Lastly the three algorithms: standard CLONALG, GA and CLONALG+2opt are compared together and the results are shown in Fig. 4.

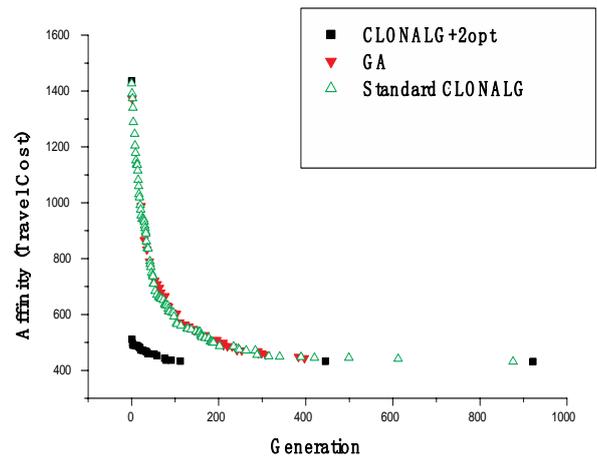


Figure 4. Comparison of the Three Algorithms.

B. Average Performance Comparison

The proposed algorithm is then compared with standard CLONALG and GA in terms of average performance using four benchmark problems from TSBLIB: the three Eilon problems and Berlin52 problem. The Berlin52 problem is the most difficult problem among the four problems. The parameter settings of all algorithms are the same as those described in Section 5.1. All the algorithms were run in ten times and the average values were recorded. The results are shown in Table 1. From this table researchers can see that the average performance of CLONALG+2opt is always better than GA and CLONALG, which demonstrates the effectiveness of CLONALG+2opt. In addition, from all experiments researchers can see that the standard deviation of CLONALG+2opt tends to be smaller than the other two algorithms, which shows its stability.

TABLE I. COMPARISON BETWEEN ALGORITHMS

Problem	Generations	GA		CLONALG		CLONALG+2opt	
		Cost	Deviation	Cost	Deviation	Cost	Deviation
Eil51	1000	443.15	3.574%	441.36	3.156%	433.71	1.368%
Eil75	1000	566.32	3.027%	565.23	2.923%	562.45	2.416%
Eil101	1000	777.31	23.58%	704.32	11.975%	670.49	6.596%
Berlin52	1000	7704.67	2.136%	7752.47	2.758%	7598.44	0.717%

VI. CONCLUSIONS

In this paper, researchers proposed a hybrid AIS system that incorporates the local search technique into CLONALG for solving TSP. The immune operators of CLONALG were also modified to make CLONALG adapt to TSP. Experimental results show that the proposed approach, CLONALG with quasi-2opt and sequence mutation strategy, is a promising algorithm for solving TSP and has the potential to be applied to other permutation optimisation problems.

The work presented in this paper is a preliminary research on solving general permutation optimisation problems by immune-inspired algorithms combined with problem-dependent local search. The design of the quasi-2opt operator is an example showing that by making use of the information about problem features the performance of immune-inspired algorithms could be significantly improved. The fact that there are already existing local search techniques specially designed for other POP problems encourages us to further investigate the potential of the hybrid approach to POP, which combines CLONALG with the local search approach. More work in this field could be done in the near future. One promising future work is to apply CLONALG to the Vehicle Routing Problem with Time Windows (VRPTW) [21], a more complicated and realistic POP than TSP. The local search techniques for VRPTW, such as PFIH (Push Forward Insertion Heuristic) [22] and λ -interchange [23] techniques, and their variants could be integrated into CLONALG, and the performance of the resulting algorithms can be compared with the previous work on using GA to solve the same problem [24].

ACKNOWLEDGMENT

This work was supported by the Natural Science Foundation of China (No. 61472159), Jilin Innovation Team Project (No. 20122805) and Development Project of Jilin Province of China (No. 20130522111JH, 20140101180JC)

REFERENCES

- [1] L. N. De Castro and F. J. Von Zuben, "Learning and optimization using the clonal selection principle," *Evol. Comput. IEEE Trans. On*, vol. 6, no. 3, pp. 239–251, 2002.
- [2] L. N. De Castro and F. J. Von Zuben, "The clonal selection algorithm with engineering applications," in *Proceedings of GECCO*, 2000, vol. 2000, pp. 36–39.
- [3] H. Chen, S. Gao, S. Li, and Z. Tang, "A probabilistic modeling clonal selection algorithm and its application to traveling salesman problems," *IJCSNS*, vol. 10, no. 6, pp. 57–62, 2010.
- [4] S. Gao, H. Dai, G. Yang, and Z. Tang, "A novel clonal selection algorithm and its application to traveling salesman problem," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci. E Ser. A*, vol. 90, no. 10, pp. 2318–2325, 2007.
- [5] F. Liu, J. Ma, L. Jiao, M. Gong, R. Shang, Y. Li, and Y. Qi, *High Performance Immune Clonal Algorithm for Solving Large Scale TSP*. INTECH Open Access Publisher, 2010.
- [6] S. Lin and B. W. Kernighan, "An effective heuristic algorithm for the traveling-salesman problem," *Oper. Res.*, vol. 21, no. 2, pp. 498–516, 1973.
- [7] S. F. M. Burnet and others, *The clonal selection theory of acquired immunity*. University Press Cambridge, 1959.
- [8] Goldberg, David E., J. J. Grefenstette and N. L. Cramer, "Proceedings of an International Conference on Genetic Algorithms and Their Applications," Held Carnegie-Mellon Univ. Pittsburgh Pa, 1985.
- [9] Grefenstette J J, Gopal R, Rosmaita B J, Gucht D V. *Proceedings of the 1st international conference on genetic algorithms*, Pittsburgh, PA, USA, , pp.160–168, 1985.
- [10] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *Evol. Comput. IEEE Trans. On*, vol. 1, no. 1, pp. 53–66, 1997.
- [11] L. Huang, C. Zhou, and K. Wang, "Hybrid ant colony algorithm for traveling salesman problem," *Prog. Nat. Sci.*, vol. 13, no. 4, pp. 295–299, 2003.
- [12] W. Pang, K. Wang, C. Zhou, and L. Dong, "Fuzzy discrete particle swarm optimization for solving traveling salesman problem," in *Computer and Information Technology, 2004. CIT'04. The Fourth International Conference on*, 2004, pp. 796–800.
- [13] K.-P. Wang, L. Huang, C.-G. Zhou, and W. Pang, "Particle swarm optimization for traveling salesman problem," in *Machine Learning and Cybernetics, 2003 International Conference on*, vol. 3, pp. 1583–1585, 2003.
- [14] A. J. George and D. Gray, "Receptor editing during affinity maturation," *Immunol. Today*, vol. 20, no. 4, pp. 196–196, 1999.
- [15] Y. Zhu, S. Gao, H. Dai, F. Li, and Z. Tang, "Improved clonal algorithm and its application to traveling salesman problem," *Int J Comput Sci Netw. Secur.*, vol. 7, no. 8, pp. 109–113, 2007.
- [16] Reinelt G. TSPLIB, [2008-08-06], <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>.
- [17] S. Kirkpatrick and G. Toulouse, "Configuration space analysis of travelling salesman problems," *J. Phys.*, vol. 46, no. 8, pp. 1277–1292, 1985.
- [18] A. S. Perelson and G. F. Oster, "Theoretical studies of clonal selection: minimal antibody repertoire size and reliability of self-non-self discrimination," *J. Theor. Biol.*, vol. 81, no. 4, pp. 645–670, 1979.
- [19] G. A. Croes, "A method for solving traveling-salesman problems," *Oper. Res.*, vol. 6, no. 6, pp. 791–812, 1958.
- [20] D. B. Fogel, "Applying evolutionary programming to selected traveling salesman problems," *Cybern. Syst.*, vol. 24, no. 1, pp. 27–36, 1993.
- [21] G. Laporte, "The vehicle routing problem: An overview of exact and approximate algorithms," *Eur. J. Oper. Res.*, vol. 59, no. 3, pp. 345–358, 1992.
- [22] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Oper. Res.*, vol. 35, no. 2, pp. 254–265, 1987.
- [23] I. H. Osman and N. Christofides, "Capacitated clustering problems by hybrid simulated annealing and tabu search," *Int. Trans. Oper. Res.*, vol. 1, no. 3, pp. 317–336, 1994.
- [24] L. Huang, W. Pang, K.-P. Wang, C.-G. Zhou, Y. Xiao, "Improved Genetic Algorithm for Vehicle Routing Problem with Time Windows," *Advances in Systems Science and Applications*, vol.4, pp.118-124, 2004.