Block Based Imperial Competitive Algorithm with Greedy Search for Traveling Salesman Problem

Meng-Hui Chen, Chiao-Wei Yu, Pei-Chann Chang

***Abstract*—**Imperial competitive algorithm (ICA) simulates a multi-agent algorithm. Each agent is like a kingdom has its country, and the strongest country in each agent is called imperialist, others are colony. Countries are competitive with imperialist which in the same kingdom by evolving. So this country will move in the search space to find better solutions with higher fitness to be a new imperialist. The main idea in this paper is using the peculiarity of ICA to explore the search space to solve the kinds of combinational problems. Otherwise, we also study to use the greed search to increase the local search ability. To verify the proposed algorithm in this paper, the experimental results of traveling salesman problem (TSP) is according to the traveling salesman problem library (TSPLIB). The results show that the proposed algorithm has higher performance than the other known methods.

***Keywords*—**Traveling Salesman Problem, Artificial Chromosomes, Greedy Search, Imperial Competitive Algorithm.

I. Introduction

THERE are many meta-heuristic algorithms like ant colony optimization (ACO) [1] for solving combinational optimization problems and genetic algorithms (GAs) [2] is proposed many years ago. These algorithms simulate the natural processes. GAs is according to the Darwin's natural selection theorem, which is based on parents generate new offspring by crossover and mutation rule, and select the better offspring to evolve again to get the high performance solutions. Combinational optimization problems are for decreasing the cost by numerous feasible solutions. Discrete optimization is a branch of optimization applied in computer science and mathematics. Discrete optimization contains two main domains. One is integer programming, another one is combinatorial optimization. Combinatorial optimization is a topic in theoretical computer science and applied mathematics that consists of finding the least-cost solution of a mathematical problem in which each solution is associated with a numerical cost. In recent decades, numerous algorithms were proposed to solve the combinatorial optimization problems (COPs).

To find the solution of an optimization problem is a common challenge in which an algorithm may be trapped in the local optima of the objective function when the complexity is high. Tsai et al. [3] and Chun et al. [4] have proposed the algorithms for global optimization problems. The importance of these methods is in many different areas such as modern engineering design and systems operation. Holland [5] and Goldberg [6] proposed Genetic Algorithm (GA) as a tool based on biological mechanisms and natural selection theory. GA has gained much attention regarding its potential as an optimization technique for combinational optimization problems and has been successfully applied in many different areas.

Meng-Huei Chen is with the Department of Information Management, Yuan Ze University, Taoyuan 32026, Taiwan, R.O.C. (Corresponding Author; e-mail: bbb422@hotmail.com).

Chiao-Wei Yu and Pei-Chann Chang are with the Department of Information Management, Yuan Ze University, Taoyuan 32026, Taiwan, R.O.C.

The main idea in this paper is using ICA with group recombination mechanism to increase the search space and enhance the ability of local search by using greedy search. The mechanism of group re-generated is designed to focus on keeping better solutions and the other solutions will recombine randomly to explore the search space. Greedy search is an optimal partial method of a solution. There are more descriptions about our proposed approach in the Section III.

II. Literature Review

A. Imperial Competitive Algorithm

Imperialist competitive algorithm is like a social counter part of genetic algorithms (GAs). Imperialist competitive algorithm is a mathematical model and the computer simulation of human social evolution, while genetic algorithms are based on the biological evolution of species. The concept of ICA is as Fig. 1.

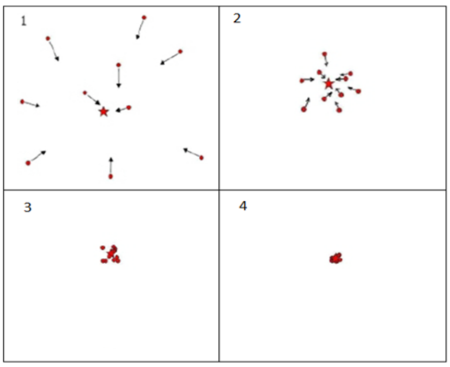

Fig. 1 The concept of ICA

The imperialist competitive algorithm is an evolutionary algorithm based on the imperialistic competition [7]. In recent years, Shirin Nozarian and Majid Vafaei Jahan [8] proposed an improved memetic algorithm for solving meta-heuristic problems. The method uses imperialist competitive algorithm as local search like other evolutionary algorithms. The imperialist competitive algorithm starts with initial populations called countries. In BBICA, country means an artificial chromosomes generation approach.

B. Greedy Search

Greedy search is a common method to solve the traveling salesman problem, which is focus on casing the solution to be best by recombined. It means the solution may not be an optimal solution, and it is just a local optimal solution. The strength of greedy search is easy to design, but also be easy to escape into local optimal.

As a general rule, greedy search has three components as follows:

1. A candidate dataset of a solution
2. Using a evolving function to the solution to get feasible solutions
3. To choose the best feasible solution when we have discovered a complete solution

In 1993, Bart Selman et al. [9] studied on how to prove that the greedy search is useful to increase the ability of local search. The application for traveling salesman problem, the experimental results of Geng et al. [10] are shown that greedy search could enhance the speed of evolving. Wang et al. [11] proposed a new greedy search method which could help to generate initial solutions with higher fitness to raise the quality of evolving.

III. Methodology

In this paper, we study how to increase the search space to find feasible solutions. On the other hand, we also try to enhance the ability of local search and keep the useful evolving information by the block mechanism to combine artificial chromosomes. As mentioned above, we use ICA with group combinational mechanism to explore the search space. ICA could provide multi-agent evolving system to find more feasible solutions, in other words, ICA could increase the diversity of solutions. After exploring the search space, we focus on the ability of local search. So there are two mechanisms to do local search. One is to generate the blocks to combine the artificial chromosomes, the other is to use greedy search to make the solution evolving. The algorithm proposed is called block based imperial competitive algorithm with greedy search (BICAGS). BICAGS is an effective meta-heuristic algorithm which contains two phases. The first phase is to consider the mining blocks approach which is adopted to discover the effective blocks. Meantime, these blocks will recombine by the way of the competition and keep the blocks with high quality. The second phase is to develop the approach of the artificial chromosomes composition. In this paper, we designed four different kinds of approaches about artificial chromosomes composition. Each artificial chromosome composition means a country which is according to the strength of the country is given a different number of generations. In the BICAGS, we have four approaches to generate artificial chromosomes as the kingdoms in this paper. Then the reassembled solutions will be considered to select the solutions with good performance. In BICAGS we will use mutation of solutions and then the reassembled solutions will be considered to select the solutions with good performance to update the probability matrix. The process will be repeated until the defined iterations are satisfied.

The process of BICAGS is shown as the Fig. 2. There are five steps as followed to describe the workflow of BICAGS. In first step, we generate the initial population by randomly to explore the search space. We find the elite solutions by calculating the fitness of each solution and make four groups as four kingdoms by fitness. Then to store the information from elite solutions builds the dominance matrix. At the second step, to generate the blocks by the useful information from the dominance matrix and combine the artificial chromosomes by blocks. Then to calculate the power of each kingdom by fitness is to show the competition of each kingdom and reset the source of each kingdom. The final step is to use the greed search to case each solution evolving and check the population need to start the regroup mechanism or not.

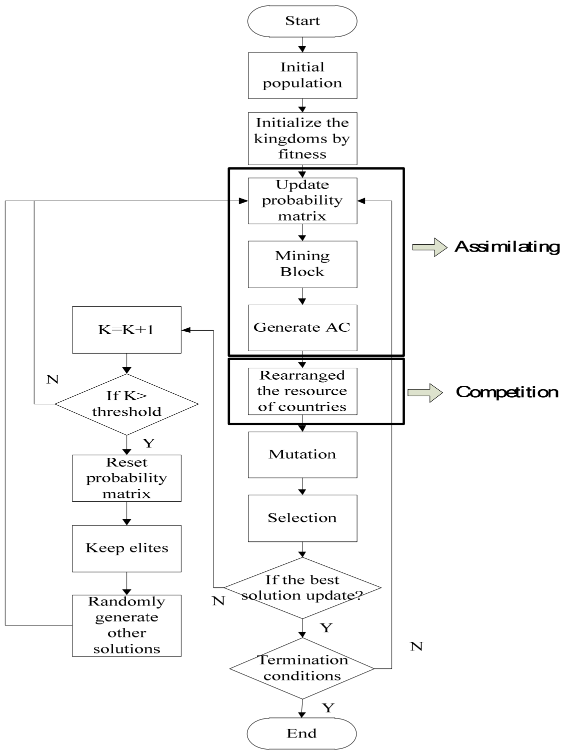

Fig. 2 BICAGS flowchart

Then there are four sections to describe the details of important mechanism as followed.

A. A Block Mining Procedure

A block mining procedure is applying the probability matrix to extract the blocks from the set of high fit chromosomes. It is a process of block learning which is applied to discover the hidden knowledge within the dependent variables. The block consists of a series of genes linked to each other continuously. To mine the blocks from the set of high fit chromosomes two methods can be applied: static block size, which are created blocks with equal sizes. The other one is dynamic block size where are created blocks with random sizes. In this research, we will focus on static block size.

A static block with size K can be generated according to the following procedures which have two different rules used in BBICA. One is used in first ten percentages of iterations:

For example, the instance has 10 cities. First, city 3 is selected randomly as start city and the next K, i.e., 3 here. Cities with the distance in descending orders will be 9, 5, and 8. Finally, block {3, 9} will be selected since it has the biggest probability, i.e., 0.52. A branching strategy is applied in generating the possible blocks. The final set of blocks, i.e., puzzles, mined from the distance are stored in the archive as shown in Fig. 3.

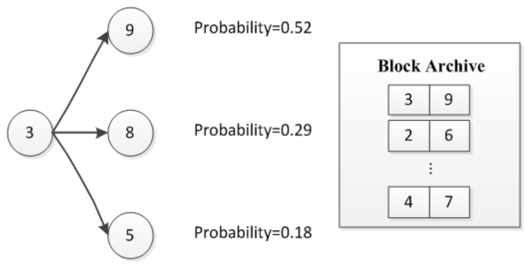

Fig. 3 The set of Blocks mined from the distance

Another is used in ninety percentages of iterations:

For example, the instance has 10 cities. First, city 3 is selected randomly as start city. Cities with probability bigger than a threshold will be 8 and 9. Finally, block {3, 9} will be selected. Since city 9 has been chosen by roulette selection as shown in Fig. 4.

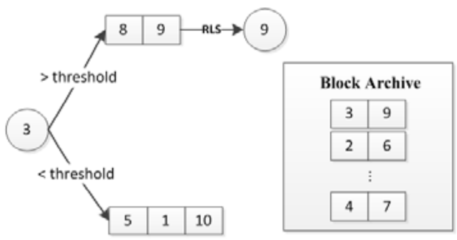

Fig. 4 The block mining mechanism I

Otherwise, cities with probability bigger than a threshold are null. Finally, block {3, 8} will be selected since it has the shortest distance, i.e., 2 as shown in Fig. 5.

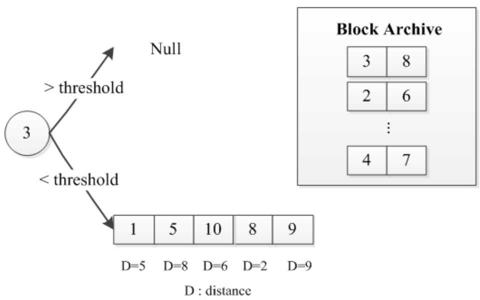

Fig. 5 The block mining mechanism II

By the two rules, there will be many blocks generated. To keeping the high quality of blocks, we design a mechanism to weed out the inefficiency of blocks.

In order to ensure qualities of the blocks with better result, we have a block select strategy. The strategy is to calculate the probability of each block according the probability matrix, and then sort the calculated values. In this research there will be only remaining twenty percentages of the block selected by values of rank.

B. Generate Artificial Chromosomes Procedure

In previous section, we mention about an artificial chromosomes generation approach means a country. An artificial chromosome can be generated according to the following procedures which have four different approaches used in BBICA. The propose approach procedure of the first country is as shown in Fig. 6:

1) A city is picked randomly as head.
2) To select a city form the residual cities using roulette wheel selection as the continuous city. If the selected city is contained in a block then the block will be selected and put it in this chromosome instead of the selected city. The last city of the block as a new head.
3) Repeated step2 until the residual city number is zero.
4) The procedures will be repeated again and again until a pre-defined number of the country's population is met.

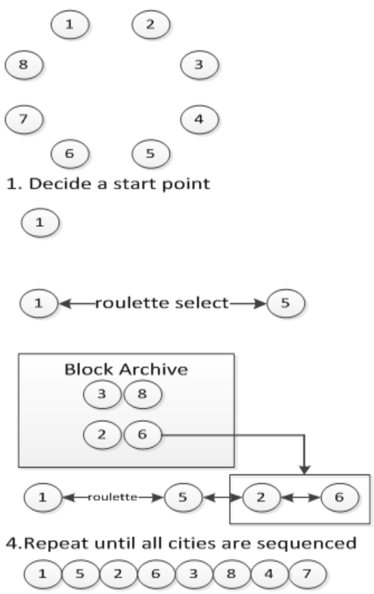

Fig. 6 The approach procedure of the first country

The proposed approach procedure of the second country is as shown in Fig. 7:

1) A city is picked randomly as start point.
2) To select a city which is selected from the residual cities by the shortest distance connected with head. If the selected city is contained in a block, the block will be selected and put it in this chromosome instead of the selected city. The last city of block as new head.

3) Repeated step2 until the residual city number is zero.
4) The procedures will be repeated again and again until a pre-defined number of the country's population is met.

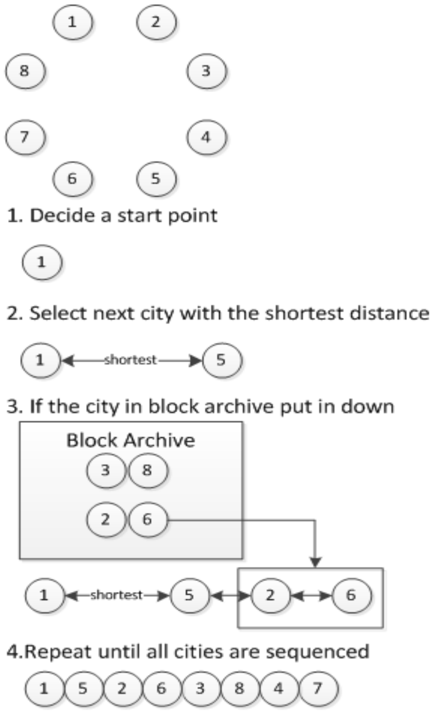

Fig. 7 The approach procedure of the second country

The proposed approach procedure of the third country as Fig. 8:

1) A city is picked randomly as a head and the block archive, save some blocks which are randomly selected from block archive.
2) To select a city which is selected from the residual cities by roulette selection connected with head. If the selected city is contained in a block from block archive, then the block will be selected and put it in this chromosome. The last city of the block as a new head.
3) Repeated step2 until the residual city number is zero.
4) The procedures will be repeated again and again until a pre-defined number of the country's population is met.

The proposed approach procedure of the fourth country is as Fig. 9:

1) A city is picked randomly as a head and the block archive. Save some blocks which are randomly selected from block archive.
2) To select a city which is selected from the residual cities by the shortest distance connected with a head. If the selected city is contained in a block from block archive, the block will be selected and put it in this chromosome. The last city of the block as a new head.
3) Repeated step2 until the residual city number is zero.
4) The procedures will be repeated again and again until a pre-defined number of the country's population is met.

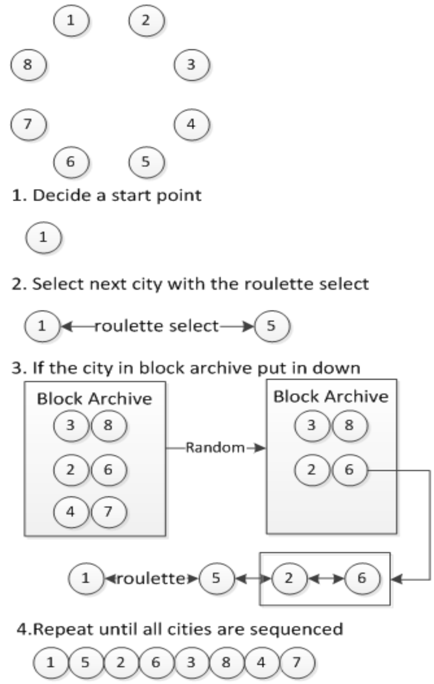

Fig. 8 The approach procedure of the third country

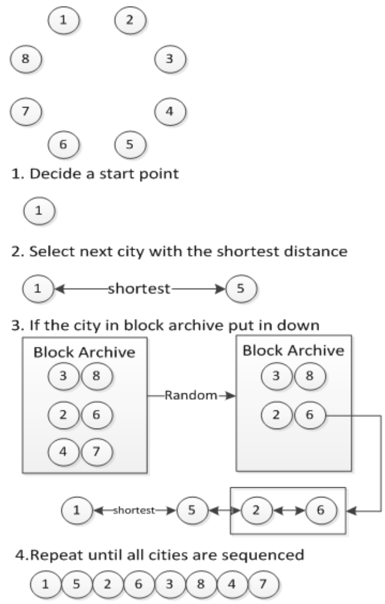

Fig. 9 The approach procedure of the fourth country

In Fig. 10, continually strong country and obtain more population is core concept of imperialist competitive algorithm. This is a reason why BBICA designed an approach which is continually revise number of the each country's population. The description of the approach is as following: First, we will calculate average fitness of each country. Then revise the number of each country's population according to the rank which is sorting by the average fitness.

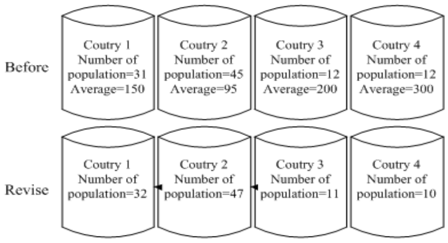

Fig. 10 Revise number of countries population

Finally, according to the rank of the averages fitness we will use different mutation rules. For this reason, we hope that weak countries use powerful mutation rule which is causes the weak countries to be better. On the other hand, strong countries use the mutation rule with more diversity to keeping high diversity of each country population.

C. Regroups Mechanism

To increase the diversity of solutions, there is a mechanism in the paper. As shown in Fig. 11, when the population evolves for lots of generations and did not find a better solution than at present. The population might be into local optimal. So we keep the elites of the population, than the others are removed and re-generate by randomly. After this mechanism, the population will be included the elite solutions and new solutions with higher diversity. The mechanism provides that elite solutions still have the chance to evolve to be better, and new solutions give the evolution a new direction.

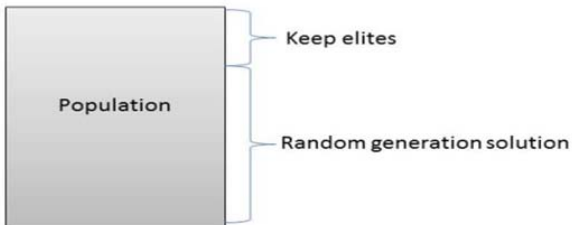

Fig. 11 The concept of Regroup

D. Greedy Search

In this paper, we use the method for two reasons. One is to enhance local search, the other is this method might keep the better link from the two chosen solutions. As Fig. 12 shown, the two chosen solutions will choose one city randomly at first, and there are four candidate cities neighbored with the first city. Then the second city will be chosen by the shortest distance between the first city and the four candidate cities. Repeat the step of the mentioned above until all cities are sequenced.

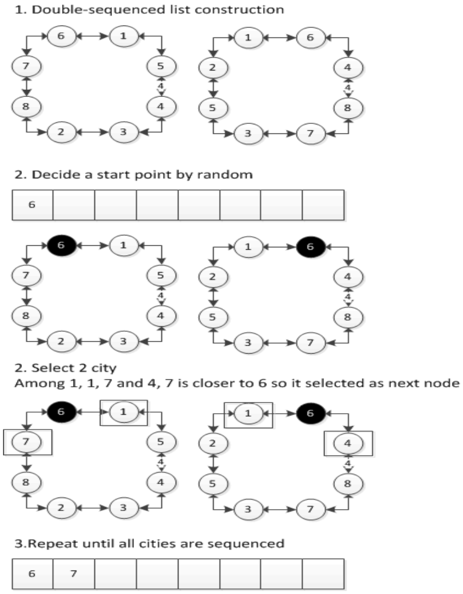

Fig. 12 The concept of the greedy search mechanism

IV. Experimental Results

In this section we present the experimental results of the BICAGS and compare the performance of BICAGS with other algorithms. Each algorithm is executed for 30 times on each instance, and the computing hardware consists of Intel Core (3.40GHz) and with DDR2 800 (2GB Memory). The programming language is Microsoft Visual C# 2010 Express. In Table I, it shows the parameters which we used in our research.

TABLE I
SET OF PARAMETERS

Parameter	Value of parameter
Number of the initial solutions	100
Number of the artificial chromosomes	100
Number of the elite solutions	20
K	50
Iteration	City * 50
Length of block	2

In this paper, the block size means length of mining blocks. The block counts are equals that we will mine that how many number of blocks and the generation is defined the number of the total doing. All test cases are chosen from website TSPLIB and with the best known solutions.

These three approaches, i.e., BBEA [12], RABNET-TSP [13] and SME [14], are selected for comparison with our proposed approach. BBEA is based on block with great performances. RABNET and SME are based on Self-Organized Map (SOM) network with very efficient and effective performances. The comparisons of the experimental results for BBEA, RABNET and SME are presented in Tables II and III.

TABLE II
PERFORMANCE OF BEST SOLUTION IN TSP'S INSTANCE

		BICAGS	BBEA	RABNET	SME
Instance	Opt.	Error Rate	Error Rate	Error Rate	Error Rate
eil51	426	1.39%	0.47%	0.23%	1.64%
eil76	538	2.48%	1.12%	0.56%	2.60%
eil101	629	2.19%	2.07%	1.43%	1.75%
berlin52	7542	0.03%	0.03%	0.00%	2.29%
bier127	118282	0.53%	7.32%	0.58%	1.32%
ch130	6110	1.61%	1.11%	0.57%	1.52%
ch150	6528	0.51%	0.32%	1.13%	1.58%
rd100	7910	0.01%	1.18%	0.91%	1.49%
lin105	14379	0.03%	0.02%	0.00%	0.00%
lin318	42029	0.86%	2.95%	1.92%	2.68%
kroA100	21282	0.02%	0.01%	0.24%	0.60%
kroA150	26524	0.87%	0.94%	0.58%	1.53%
kroA200	29368	0.56%	0.89%	0.79%	2.64%
kroB100	22141	0.84%	0.00%	0.91%	1.84%
kroB150	26130	0.98%	0.23%	0.51%	0.81%
kroB200	29437	1.41%	2.36%	0.68%	0.90%
kroC100	20749	0.01%	0.00%	0.80%	0.83%
kroD100	21294	0.86%	0.00%	0.38%	0.97%
kroE100	22068	0.67%	0.17%	1.48%	1.41%
rat575	6773	2.88%	9.67%	4.05%	4.68%
rat783	8806	2.66%	7.00%	5.00%	5.79%
Avg.		1.30%	1.80%	1.08%	1.85%

TABLE III
PERFORMANCE OF BEST SOLUTIONS AVERAGED IN TSP'S INSTANCE

		BICAGS	BBEA	RABNET	SME
Instance	Opt.	Error Rate	Error Rate	Error Rate	Error Rate
eil51	426	1.76%	0.47%	2.70%	3.43%
eil76	538	3.08%	1.62%	3.40%	4.52%
eil101	629	3.24%	3.32%	3.12%	4.23%
berlin52	7542	0.03%	0.03%	5.18%	6.41%
bier127	118282	0.93%	7.76%	2.20%	2.92%
ch130	6110	2.59%	1.84%	2.82%	3.23%
ch150	6528	0.82%	1.53%	3.22%	3.42%
rd100	7910	0.40%	1.18%	0.91%	1.49%
lin105	14379	0.76%	0.02%	0.15%	0.67%
lin318	42029	1.93%	3.90%	3.97%	4.51%
kroA100	21282	0.31%	0.01%	1.13%	1.57%
kroA150	26524	1.79%	1.47%	3.14%	3.31%
kroA200	29368	0.82%	2.26%	2.80%	3.57%
kroB100	22141	1.16%	0.30%	2.35%	2.17%
kroB150	26130	1.60%	0.82%	1.92%	2.59%
kroB200	29437	1.87%	3.24%	2.37%	2.89%
kroC100	20749	0.35%	0.06%	1.07%	1.93%
kroD100	21294	1.39%	0.44%	1.89%	2.59%
kroE100	22068	1.17%	0.33%	2.93%	2.78%
rat575	6773	3.40%	10.85%	5.06%	5.91%
rat783	8806	3.02%	7.95%	6.11%	6.60%
Avg.		1.54%	2.35%	2.78%	3.37%

To calculate the error rate we used the following formula as shown in Table II,

$$ErrorRate(\%) = \frac{Best - Opt}{Opt} * 100\% \quad (1)$$

The result shows that the average error rate of best solution in BICAGS is 1.30% which is better than BBEA and SME. Even through the performance of best solution in BICAGS isn't as good as in RABNET. In some instances, the BICAGS still has better performance than RABNET.

According to Table III, we use the following formula to calculate error rate.

$$ErrorRate(\%) = \frac{Best - Opt}{Opt} * 100\% \quad (2)$$

The result shows that the average error rate of mean solution in BICAGS is 1.54% better than BBEA, RABNET and SME. The result means that the steadiness of searching solution is the best in these four approaches.

V. CONCLUSION

In order to expand the search space, we study to using ICA to design the base architecture. ICA is like that several independent kingdoms evolving at the same time. So we design four different mechanisms for each kingdom to case this four kingdoms have different development. It means that there are four different directions to search feasible solutions in search space. In addition, we design the block mechanism to share the higher performance information to each country. This will case each country doesn't only search by the direction of its kingdom, but also received the useful information of stronger countries at present. Finally, we also use the regroups mechanism to enhance the diversity of population and increase the deep exploration by the greedy search. The regroups mechanism is not only to enhance the diversity of population, but also conserves the elite solutions having choice to keep evolving. The method of the greedy search will increase the deep exploration and hold back the good link of the two selected solutions. As mentioned above, the BICAGS provide increasing search space and enhance the deep exploration. According to the experimental results, it shows that BICAGS could find the better solutions than the other methods. The continue research in future will be focus on how to design the numbers of kingdoms will case the better results and try to test the robust of BICAGS.

REFERENCES

[1] M. Mavrovouniotis and S. Yang, "Ant colony optimization with direct communication for the traveling salesman problem," in Proc. of the 2010 UK Workshop on Computational Intelligence (UKCI2010),2010.

[2] A. Shakeel, S. Yang. "A hybrid genetic algorithm and inver over approach for the travelling salesman problem," Evolutionary Computation (CEC), 2010 IEEE Congress on. IEEE, pp. 1-8,2010.

[3] J. T. Tsai, W. H. Ho, T. K. Liu, and J. H. Chou, "Improved immune algorithm for global numerical optimization and job-shop scheduling problems," Applied Mathematics and Computation, Vol. 194, pp. 406-424, Dec. 2007.

[4] J. S. Chun, H. K. Jung and S. Y. Hahn, "A Study on Comparison of Optimization Performances between Immune Algorithm and other Heuristic Algorithms," IEEE Transactions on Magnetics, vol. 34, No. 5, Sept. 1998.

[5] J. H. Holland, "Genetic Algorithms and the Optimal Allocation of Trials," SIAM J. Comput, vol. 2, pp. 88-105, 1973.

[6] D. E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning (Book style)," Boston, MA: Addison-Wesley, 1989.

[7] Atashpaz-Gargari, E. Lucas, C. 2007. Imperialist Competitive Algorithm: An algorithm for optimization inspired by imperialistic competition. IEEE Congress on Evolutionary Computation, pp.4661–4667.

[8] Shirin Nozarian and Majid Vafaei Jahan , "A Novel Memetic Algorithm with Imperialist Competition as Local Search," IACSIT Hong Kong Conferences vol. 30 (2012).

[9] B. Selman, & H. A. Kautz, "An empirical study of greedy local search for satisfiability testing," AAAI, vol.93, pp.46-51, 1993.

[10] X. Geng, Z. Chen, W. Yang, D. Shi, K. Zhao, "Solving the traveling salesman problem based on an adaptive simulated annealing algorithm with greedy search," Applied Soft Computing, vol.11, pp. 3680-3689,2011.

[11] Z. Wang, H. Duan, X. Zhang, "An Improved Greedy Genetic Algorithm for Solving Travelling Salesman Problem," Natural Computation, ICNC'09 Fifth International Conference on, vol.5, pp.374-378, 2009.

[12] Huang, Wei-Hsiu Chang, Pei-Chann, Wang, Lien-Chun, "A Fast Block-based Evolutional Algorithm for Combinatorial Problems," World Academy of Science, Engineering and Technology 67 (2012).

[13] R. Pasti, L. N. de. Castro, "A Neuro-immune network for solving the traveling salesman problem," Proceedings of International Joint Conference on Neural Networks, vol. 6, pp. 3760-3766, 2006.

[14] S. Somhom, A. Modares, T. Enkawa, "A self-organizing model for the travelling salesman problem," Journal of the Operational Research Society, vol. 48, pp. 919-928, 1997.