An Immune Optimization using Biological Immune Co-evolutionary Phenomenon and Cell-cooperation for Division-and-labor Problem

Naruaki Toma*1, Satoshi Endo*2, Koji Yamada*2, Hayao Miyagi*2
**1 Graduate Student, Doctoral Course in Complex Intelligent, Systems Engineering, Graduate School of Science and Engineering, University of the Ryukyus, JAPAN.*
**2 Department of Information Engineering, Faculty of Engineering, University of the Ryukyus, JAPAN.*

Abstract. The purposes of this paper are to propose and evaluate an immune optimization algorithm using biological immune co-evolutionary phenomenon and cell-cooperation. The co-evolutionary models searches the solution through the interactions between two kinds of agents, one of the agent is called immune agent which optimize the cost of its own work. The other is called antigen agent which realize the equal work assignment. This algorithm solves the division-of-labor problems in multi-agent system (MAS) through the three kinds of interactions: *division-and-integration processing* is used for optimization of the work-cost of immune agents and, *immune cell-cooperation* is used to perform equal work assignment as a result of evolving the antigen agents. To investigate the validity, this algorithm is applied to "N-th agent's Travelling Salesmen Problem" as a typical problem of MAS. The good property on solving for MAS will be clarified by some simulations.

1 Introduction

Adaptive problem solving techniques, such as neural networks and genetic algorithms, are based on information processing in biological organisms and are applied on many kinds of optimization problems. A biological immune system is one of the adaptive systems and the studies are making advances [1, 2, 3, 4]. The biological immune system is widely recognized as one of the adaptive biological system whose functions are to identify and to eliminate foreign materials.

In this paper, we propose and evaluate an immune optimization algorithm inspired by biological immune cell-cooperation, and this algorithm solves the division-of-labor problems in multi-agent system (MAS). For the following reason, it is very useful to solve these problems for application. As the reason, we need to correspond to problem space size that becomes more complicated in recent years. MAS is a study in the field of distributed artificial intelligence and attracts attention as a framework for solving effectively using cooperations or competitions through interactions. Therefore, it is very meaningful to examine the application to MAS.

The proposed algorithm solves the problem through interactions between agents (called 'immune agents'), and between agents and environment (called 'antigen agents'). This method for implementation of the interactions uses *division-and-integration processing* inspired by *immune cell-cooperation*. There are three functions in our algorithm: division-and-integration processing and *escape processing*. The division-and-integration processing optimizes the work domain through the interactions between immune agents, and the escape processing performs equal divisions through the interactions between immune agents and antigen agents. Then, in order to investigate the validity of the proposed method, this algorithm is applied to "N-th agent's Travelling Salesmen Problem (called n-TSP)" as a typical problem of multi-agent system. The good property on solving for MAS will be clarified by some simulations.

2 Analogy from biological immune system

In the point of view of engineering system, it is considered the immune cell-cooperation to be a parallel distributed system with role differentiation. The roles in this system are (1) fragmentation and presentation of antigens, (2) activation of producing specific antibodies, (3) elimination of the antigens by specific antibodies, and (4) control of the functions. We have as object to model, implement and apply to MAS. We construct an optimization algorithm based on a concept that (1) fragments a problem, (2) solves the fragmented sub-problems by the specific sub-solutions, and then (3) solves whole the problem through combination of these sub-solutions. We call these procedures *division-and-integration processing*.

In addition, we introduce a co-evolutionary concept that the immune system evolves against the virus which evolves so as to escape the elimination of the system (see figure 1). We call the procedure which attempts to survive antigen itself by counter-checking, *escape processing*. In order to apply this concept as a searching method in MAS, we construct a procedure liken the escaping virus to changing environments.

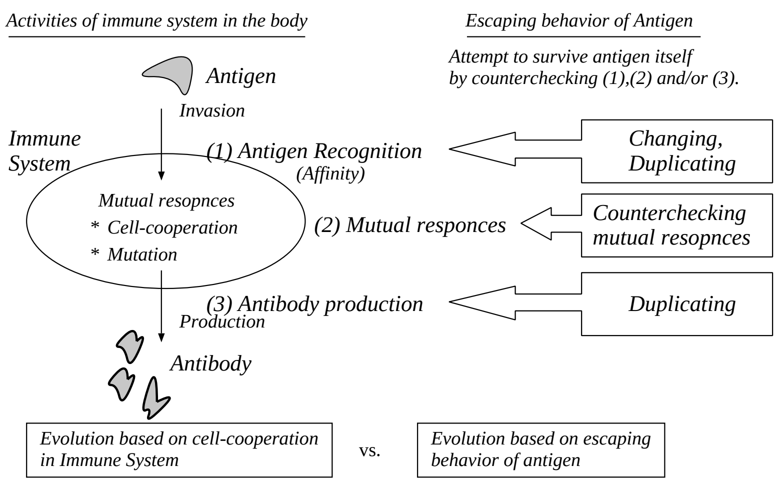

Figure 1: Concept of immune co-evolutionary phenomenon

3 Issues of division-of-labor problems

The key subject of division-of-labor problems is to focus on the issues of distribution of works for agents. The solvers of the problems are defined as follows: the domain that each agent covers is defined as work domains (WD), and the aggregate total of the work domains is defined as problem domain (PD). There are two aims in the division-of-labor problems. They are: (a) each WD_i that is assigned by $agent_i$ must be divided evenly and, (b) the system performs effective division-of-labor by optimizing each WD_i. As a typical case problem of the division-of-labor, we deal with n-TSP to investigate an adaptation ability of our model. The objective is finding the minimal tour by division among salesman. Note that one condition that all salesmen use the same city as the starting point was set in our experiments.

4 Proposed method

The algorithm solves the problems through two searching ways, (1) *division-and-integration processing* by salesman agents and (2) *escape processing* by city agents in the environment. The procedures of the algorithm against an n-TSP are described as below (see figure 4).

[Step1. Definition of problems and immune functions.]
Cities and Salesmen as the problem must be defined. Salesman's unique ID and division-and-integration processing for tours of each salesman must be defined. At this point, each city has the ID for expression of salesman visiting the city.

[Step2. Calculation of objective function.]
The cost of salesman is calculated by following function.

$$Cost(S_i) = \sum distance(tour_{S_i}) \tag{1}$$

- S_i: Salesman i.
- $tour_{S_i}$: the tour of Salesman i.

[Step3. ***division processing*** **with Simulated Annealing.]**
One salesman tries to divide own tour into two subtours for searching lower costs according to these steps (see also figure 2): decide of any one $subtour_j$, make new $tour_i$ except the $subtour_j$, calculate costs of both tours and then, if the cost is improved, the division will be performed (as a consequence, a new agent is generated). Note that, in order to increase the number of execution times of division processing, 'Simulated Annealing' which is a kind of methos using Monte Carlo is implemented.

[Step4. ***integration processing*****.]**
Two salesmen try to integrate the tours for searching lower costs by following steps (see also figure 3): decide of any two tours, make new tour by binding, calculate cost of the new tour and then, if the cost is improved, the integration performed (as a consequence, a original salesman is deleted).

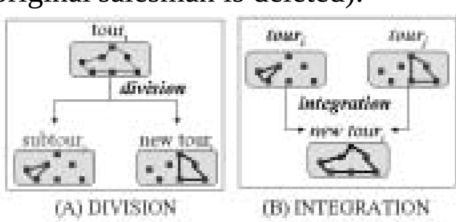

Figure 2: Examples of Division and Integration.

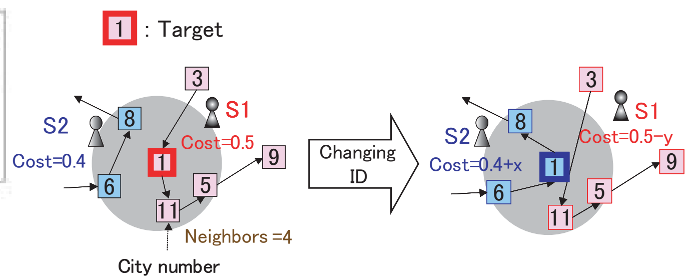

Figure 3: Escape processing.

[Step5. Mutation.]
Swap any two cities. If the cost, after swap, is improved, the swap will be performed.

[Step6. Calculation of objective function.]
The city agent's cost (same as a cost of visited salesman) is calculated by the function 1.

[Step7. Escape processing.]
This approach changes ID of a city depending on neighbors cost to process following steps (see also figure 3). First, check costs of salesmen that visited neighbor cities of target city. Second, if the other salesman's cost lower than its cost, the city changes its ID into the other. As a consequence, in this example figure 3, the ID of target city is changed to $S2$, and then, the target city is visited by $S2$ like the right side of figure 3.

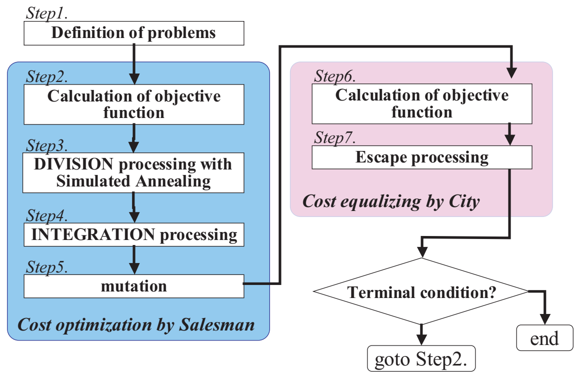

Figure 4: Flowchart of proposed method.

5 Experiment

5.1 *Problem definition and design of the proposed method:*

In order to investigate the basic performance of our method, we apply this algorithm to the following n-TSP. Definition of problem is set in table 5.1. The parameters of the proposed algorithm are shown in table 5.1.

Table 1: Definition of problem.

the number of cities	25
the number of salesmen	3
the arrangement of cities	dual circle
start city	center of circle
radius (Out,In)	0.4, 0.384

Table 2: Parameters of the algorithm.

the number of neighbors	4
the number of initial agents	under 24
terminal steps	2000

5.2 *Results and discussions:*

In order to verify the behaviors of our method, the transition of the cost, sum of whole agent's cost, and the number of visited cities (dividing-number) are shown in figure 5. The cost improves dramatically up to step 50. And an appearance of dividing changes from a combination of a few number of visited cities with great number of agents into a combination of approximately eight cities with 3-5 agents. The appearance is caused under the primary influence of the equalizing divisions in step 7 of the extended algorithm. Then, a rise of average cost of all agents occurs also. In subsequent steps, the optimization is performed repeatedly while maintaining the approximately even divisions. Finally, the transition of fitness and dividing-number converges on the optimum solution that is a combination of eight cities with three agents.

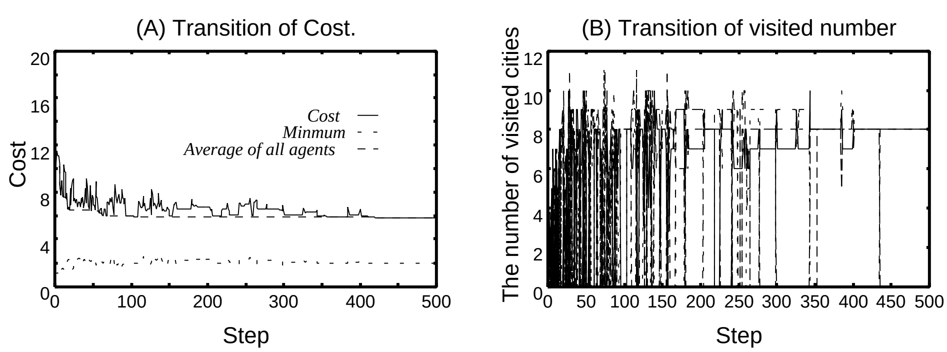

Figure 5: Transition of cost and the number of visited cities.

5.3 *Comparison with GA:*

We compare our method's results with GA using a subtour crossover. The candidate solutions must be encoded to apply GA to n-TSP. We adopted the pass representation method originally proposed by Yamamura[8]. Furthermore, subtour crossover corresponding to the pass representation method is adopted. In subtour crossover, two subtours which have the same elements are looked up, and those subtours are replaced. It becomes possible that the destruction of the important subtour is prevented by adopting this method.

5.4 Results:

The searching performances are compared through simulation results of 20 trials. Table 3 shows, (1) the number of obtained best solution, (2) minimum, average, maximum of cost, (3) the number of searched solutions (e.g. a sum of individuals in the case of GA), and (4) run time. In all terms, our method results with smaller run time are better than GA. In particular, this algorithm can be capable of searching superior solution quickly on average.

Table 3: Simulation results of 20 trials.

		GA	IA
Best	Times	3	15
Cost	Min(best)	5.845231	5.845231
	Ave	6.403267	5.858212
	Max(worst)	6.949701	5.858212
The number of searched solutions	Min(best)	74,100	1,616
	Ave	167,100	8,488
	Max(worst)	227,100	16,580
Run time		50sec	10sec

6 Conclusion

In this paper, we proposed and evaluated an immune optimization algorithms in order to verify the engineering application possibility of artificial immune system Our method solves the problems through combination of division, integration and co-evolutionary approach. These functions are based on local interactions between agents, and between agents and environment. In MAS, clarifying the objective function considered all agents and components in the environment is too hard problem, and then it is important optimizing of the whole problem by using the local interactions. Since our algorithm can optimize division-of-labor problems, it can expect what is functioned effectively as an optimization algorithm in MAS. Other results which was applied to some asymmetric arrangements will be shown in the presentation.

The first author acknowledges the Grant-in-Aid for Scientific Research (Grant-in-Aid for JSPS Fellows).

References

[1] H.Bersini and F.J.Varela (1991). The Immune Recruitment Mechanism : A Selective Evolutionary Strategy. Proc. of ICGA 91.

[2] J.D. Farmer, N.H. Packard, A.A. Perelson (1986). The Immune system adaptation, and machine learning. Physica 22D, pp.187-204,1986.

[3] S. Forrest, A.A. Perelson (1990) Genetic algorithm and the Immune system. Proc. of 1st Workshop on Parallel Problem Solving from Nature, Dortmund, Federal Republic of Germany, 1-3, October, 1990.

[4] Y.Ishida and N.Adachi (1996). Active Noise Control by an Immune Algorithm : Adaptation in Immune System as an Evolution. Pro. ICEC 96, pp.150-153.

[5] Charles A. Janeway, Jr., Paul Travers ; with assistance of Simon Hunt, Mark Walport (1997) Immunobiology : The Immune System in Health And Disease. Garland Pub.

[6] N. Toma, S. Endo, K. Yamada (2000). The Proposal and Evaluation of an Adaptive Memorizing Immune Algorithm with Two Memory Mechanisms. Journal of JSAI, Vol.15, No.6, pp.1097-1106.

[7] N. Toma, S. Endo, K. Yamada, H. Miyagi (2000). The Immune Distributed Competitive Problem Solver with MHC and Immune Network. Proceeding of ANNIE 2000, Volume 10, pp.317-322.

[8] M. Yamamura, T. Ono, S. Kobayashi (1992) Character-Preserving genetic algorithms for Travering salesman problem. Jurnal of JSAI, Vol.7 No.6, pp.1049-1059, 1992.