THE IMMUNE DISTRIBUTED COMPETITIVE PROBLEM SOLVER USING MAJOR HISTOCOMPATIBILITY COMPLEX AND IMMUNE NETWORK

Naruaki Toma[*1], Satoshi Endo[*2], Koji Yamada[*2] and Hayao Miyagi[*2]

**1 Graduate Student, Doctral Couse in Complex Intelligent Systems Engineering, Graduate School of Science and Engineering, University of the Ryukyus, 1 Senbaru, Nishihara, Okinawa, 903-0213, Japan.*
**2 Department of Information Engineering, Faculty of Engineering, University of the Ryukyus, 1 Senbaru, Nishihara, Okinawa, 903-0213, Japan.*
tnal@eva.ie.u-ryukyu.ac.jp, {endo,koji,miyagi}@ie.u-ryukyu.ac.jp

Abstract The purpose of this paper is to propose an extended immune optimization algorithm using division as well as integration processing based on immune cell-cooperation and to investigate its validity by computer simulations. In the biological immune system, the immune cell-cooperation is a framework including MHC and immune network, the function of which is to eliminate unknown vast antigens. Our algorithm solves the division-of-labor problems for each agent's work domain inside the multi-agent system (MAS) through interactions between two agents, and those of between agents and environment through the work of immune functions. There are three functions in our algorithm: the division as well as integration processing and the co-evolutionary-like approach. The division as well as integration processing optimizes the work domain, and the co-evolutionary approach realizes equal divisions. In order to investigate the validity of the proposed method, this algorithm is applied to the "Nth agent's Travelling Salesmen Problem (called the n-TSP)" as a typical problem of multi-agent system. The property that is believed to function as solution driver for MAS shall be clarified using several simulations.

Keywords: Optimization, MHC and immune network, immune cell-cooperation, multi-agent system, division-of-labor problems.

1. INTRODUCTION

Adaptive problem solving techniques, such as neural networks and genetic algorithms, are based on information processing in biological organisms and are applied in many kinds of optimization problems, see Thomas (1997), David et al (1997), Dasgupta et al (1997), Goldberg (1989), and Holland (1992). A biological immune system is one of the adaptive systems and the studies of the said system are making progress these days, see Dasgupta (1999), Farmer et al (1986), Forrest et al (1990), Ishida et al (1998), and Toma et al (2000-a). In the biological immune system, the following two mechanisms are considered as important functions in eliminating invaded antigens (Janeway et al, 1997). First, major histocompatibility complex (MHC) is used to distinguish a “self” from the other “nonself” when the nonself invades the self. Second, the immune network comprises immune cells and their connections, and then specific antibodies are produced through modification of their immune cells. Furthermore, a framework, with double functions, that sets out to eliminate the unknown vast antigens in parallel, is called `immune cell-cooperation'. From the engineering system's point of view, the immune cell-cooperation is considered a parallel-distributed system with role differentiation.

In this paper, we propose an extended immune optimization algorithm, which is based on the immune cell-cooperation, and to solve the division-of-labor problems for each agent's work domain in multi-agent system (MAS). In previous work (Toma et al, 2000-b), the immune distributed competitive problem solver had two problems; we were not able to compute even work domain and the divergence. The previous algorithm has been improved so that the immune cell-cooperation, which is considered to be a framework in a broad sense of eliminating antigens, may be applied rather than applying the local functions directly to the algorithm.

The extended algorithm solves the division-of-labor problems through interactions between the two agents, and between agents and environment by immune functions. There are three functions in our algorithm: the division as well as integration processing and the co-evolutionary-like approach. The division as well as integration processing optimizes the work domain, and the co-evolutionary approach realizes equal divisions. Through implementation of such functions, we could construct an adaptive algorithm that solves division-of-labor problems. Thereupon, we applied the proposed algorithm to the nth agent's travelling salesman problem (called the n-TSP), which is considered a typical case problem in MAS.

Some computer simulations are designed to clarify the basic performances as well as the characteristics and features of the proposed immune algorithm.

2. ISSUES OF THE TARGET PROBLEMS

2.1. Division-of-Labor Problems

Multi-agent system, which is a study in the field of distributed artificial intelligence, is an information processing technique in which autonomous agents solve problems by interactions among the agents (Russell et al, 1995). The key subject of division-of-labor problems is to focus on the issues of distribution of work for agents. The solvers of the problems are defined as follows: the domain that each agent covers is defined as work domain (WD), and the aggregate total of the work domains is defined as problem domain (PD). There are two issues to the division-of-labor problems. They are: (a) each WDi that is assigned by agent i must be divided evenly and, (b) the system performs effective division-of-labor by optimizing each WDi.

2.2. Nth agent's TSP

As a typical case problem of the division-of-labor, we dealt with the nth agent's Travelling Salesman Problem to investigate the adaptability of our model. Travelling salesman problem (TSP) is one of the most typical combinatorial optimization problems. The objective is to find a minimal roundtrip route visiting each node exactly once. The number of salesmen in the n-TSP was increased from singular to plural number. The objective is to find the minimal tour route by division among salesmen (see Figure 2). Note that two conditions were set. First, the number of salesmen is fixed while the system runs to solve. Second, all salesmen use the same city as the starting point.

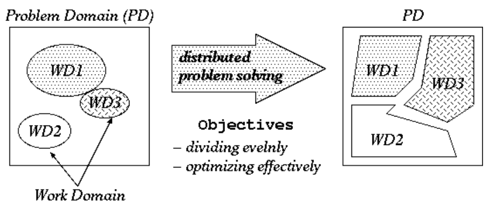

Figure 1. Aims of distributed problem solving.

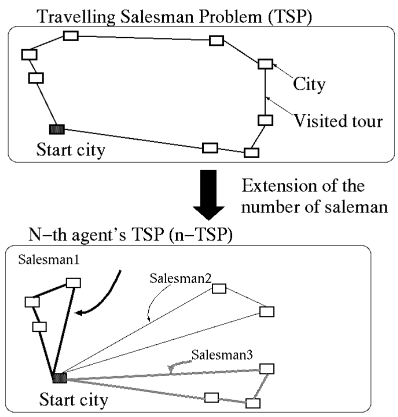

Figure 2. Notion of n-TSP.

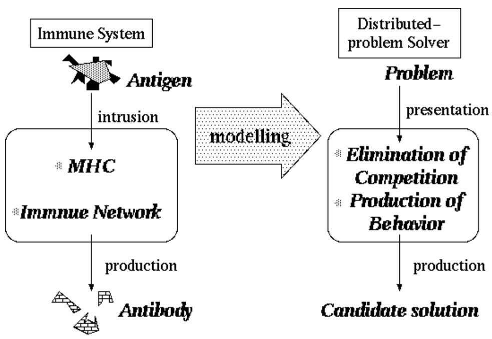

Figure 3. Modeling of biological immune system for distributed problem solver.

3. PREVIOUS WORK

3.1. Modeling of an Biological Immune System

In order to construct an immune optimization for the division-of-labor problems, we developed the models of immune functions in the first instance. Our objective models are shown in Figure 3 and the details are described below in italics.

3.1.1. Antigens and Antibodies

Antigens are defined as problems fed into the systems or environments in which the agents exist. Antibodies are defined as solutions against the problems or ways to adapt to the environments. In accordance with these definitions, it is possible to develop the problem solver, which is adapted to the framework of a biological immune system that performs self-preservation by searching specific antibodies that react against unknown vast antigens.

3.1.2. Elimination of Competition-States among Agents by MHC

In the biological immune system, the most important behavior is the preservation of the self. One of the self-preservation methods is the use of MHC (Janeway, 1997). This MHC is defined as a unique set of information determining the individual himself. MHC that is used to identify antigens by the difference of MHC is expressed as `MHC with peptide', which is a set of information that defines the antigen. If the infected MHC on the cells is different from the original (uninfected) MHC, the individual can recognize antigens by matching them with a T cell receptor (called TcR).

Subsequently, if nonself, such as antigens, invade the self-system, the immune cell called Macrophage processes and displays invasion of nonself by differences in MHC. While the cells with different MHC exist, the self-system continues to eliminate the nonself system as shown in the above Figure 4.

The elimination of nonself means removal of the nonself system from the self-system, expressed in the lower part of Figure 4. By regarding the self and nonself as agents in MAS, the invaded-state considers that some agents compete with each other's work domain. In such a case, if the ability of each agent is equal, the overlapping of the respective work domain places it in an ineffectual state. The MHC is used to eliminate such states of competition.

3.1.3. Production of Behaviors by Immune Network

In the biological immune system, the Immune Network is composed of immune cells and their connections, and the antibodies are produced through modification of their immune cells. The one piece of information transmitted through the network is MHC. The procedures of the immune system producing a specific antibody are described below.

First, Macrophage recognizes antigens and transmits MHC (with the information about the antigens) to T cell. Next, the T cell that received MHC eliminates the infected cells and then it activates specific B cell that has the capability to recognize the antigens. The activated B cell produces specific antibodies. Finally, the antibodies thus produced begin the elimination of the invading antigens.

To apply such procedures as a production model of behavior of agents, the following procedures were constructed.

In the production model, first, the agent acquires the information about the environment and internal-state in the agent. Second, if the states of competition exist in the internal-state, the agent eliminates these states of competition. Third, the agent searches for a set of candidate behaviors in order to increase the fitness of internal-state in the environment. Finally, the agent extracts superior solution in the candidate behaviors.

The extracted solution is applied to the environment, and then the agent continues to process their procedures (see Figure 5). Our algorithm solves the division-of-labor problems through the elimination of competition by MHC and production of behaviors by Immune Network.

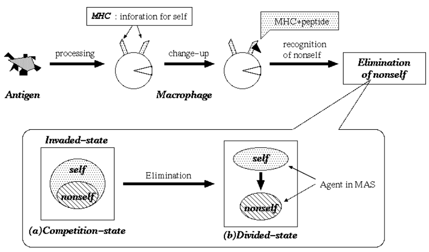

Figure 4. Elimination of Competition-States is based on the Elimination of Nonself using MHC.

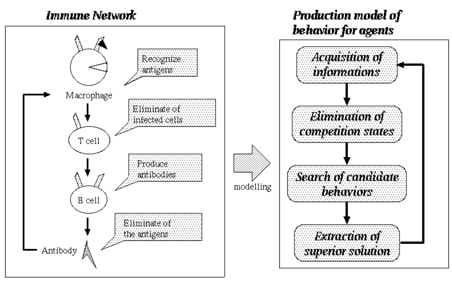

Figure 5. Production of behavior is based on production of an antibody using Immune Network.

3.2. The Immune Distributed Competitive Problem Solver

In our method, specific antibodies are produced by transmission of the MHC through the Immune Network and re-construction of the components.

The MHC is an information carrier capable of maintaining information of both the self and nonself by transforming itself into MHC with peptide (a piece of information of nonself). The situation in which MHC acquires information of nonself is referred to as the states of competition. In order to eliminate the states, we implemented the algorithm for computer simulations as described in Section 3.1. The details of our method are described below, and Figure 6 illustrates the implementation of n-TSP.

[Step1: Definition of antigens and MHC]

The problem area where agents exist is defined as antigen and the internal state (for example, solution, behavior) of each agent is defined as MHC.

[Step2: Preparation of initial MHC and immune cells]

All the initial MHC and immune cells are prepared at random for initialization.

[Step3: Acquisition of information]

Macrophage acquires both the external and internal sets of information for calculation of the fitness and recognition of competition-states. The pieces of external information are described as MHC with internal information and the MHC is transmitted to T cell.

[Step4: Elimination of the States of Competition]

If the states of competition exist in the transmitted MHC, T cell eliminates these states by means of competition processing. (The details are described in the following section).

[Step5: Production of candidate solutions]

On the basis of MHC that comprise both the external and internal information that DOES NOT include the states of competition, B cell produces N number of candidate (similar) solutions that are approximated to the MHC. The design of the similarity and the way of production of similar solutions depend on the problem.

[Step6: Extraction of a superior solution]

Each agent extracts a superior solution from the above similar solutions and executes the solution. If the terminal condition is not satisfactory, go to Step 3.

Four steps, from the third to the sixth steps, are one-time step. This algorithm is adapted by repeating from step 3 to step 6 until it comes to the terminal condition.

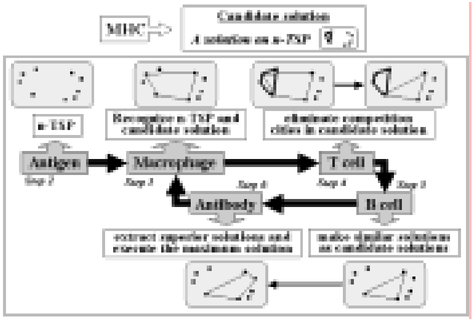

Figure 6. A Behavior example of the Immune Distributed Competitive Problem Solver.

3.3. Competition Processing

The competition processing is executed at the fourth Step in the above algorithm. A mechanism eliminates states of competition among competing agents by comparing respective fitness. The definition of terms that is used here to explain this mechanism is described in Table 1. Figure 7 illustrates a situation, which is void of competition-states based on the following two comparisons.

[Step1: Comparison of fitness in self-agent]

In the competition processing, a comparison is made between f (A) and f (A'). If the f (A') without the states of competition is high, the self-agent modifies the fitness from f (A) to f (A') and the MHC from A_{MHC} to A'_{MHC} by turning over the states of competition to agent B. Otherwise go to Step2.

[Step2: Comparison of fitness between self and nonself agents]

A comparison is made between f (A) and f (B). If f (A) is large, the states of competition are eliminated from agent B. Otherwise; the states are eliminated from agent A.

3.4. Concepts and Aspects

The proposed method consists of (1) elimination of competition using MHC and (2) production of behaviors using Immune Network. Based on the construction, this method has antithetic adaptations (also, see Figure 8). According to these interactions, our method is capable of solving the division-of-labor problems efficiently.

1. Each agent adapts in the direction of raising the level of fitness on the inside of itself through production utilizing Immune Network (see left-hand side of Figure 8).
2. Each agent adapts in the direction of elimination of inefficient region in each work domain through elimination of competition (see right-hand side of Figure 8).

Table 1. Definitions of terms: Here `after' denotes the situation that the states of competition are eliminated by the following two comparisons.

	Self agent: A	Nonself agent: B
Work domain, now	A_{MHC}	B_{MHC}
Work domain, after	A'_{MHC}	B'_{MHC}
Fitness, now	F(A)	f(B)
Fitness, after	F(A')	f(B')

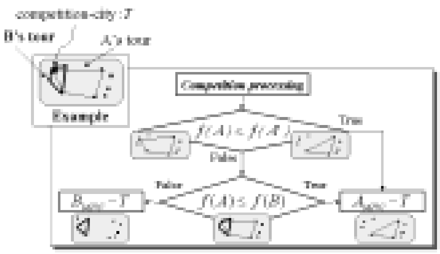

Figure 7. An example of competition processing.

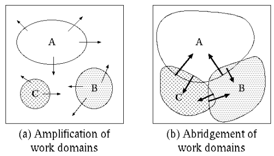

Figure 8. Basic concept of the proposed method.

4. EXPERIMENT FOR IMMUNE DISTRIBUTED COMPETITIVE PROBLEM SOLVER

4.1. Design of Simulation

In order to investigate the basic performance of our method, we applied this algorithm to the following n-TSP. The definition of problem and parameters of proposed method are set forth in Tables 2 and 3. The fitness i, the fitness of agent i, is defined as equation 1.

$$fitness_i = Cities(MHC(i)/Cost(MHC(i))+1/Cost(MHC(i)) \quad (1)$$

- *MHC(i)*: the tour of agent i.
- *Cities(MHC(i))*: the number of cities in *MHC(i)*
- *Cost(MHC(i))*: the length of *MHC(i)*

4.2. Results and Problems

Figure 9-(A) shows the acquired solution at step 50. The solution is considered as optimum solution in its own work domain. In case of simple n-

TSP with a single circle, it is clear that (a) the number of cities visited by each agent is 8 and the even division of work domain is computed and (b) the solution of each agent is considered as the optimum solution. However, in case of the dual circle even division of work domain is not computed.

In addition, this method has a critical problem of divergence. Figure 9-(B) shows the transition of the fitness and the number of the cities. As shown in the figure, the fitness decreases whenever the number of cities changes, although it is improving up to step 50. The reason for these results is that the interactions are strictly based on the value of fitness. We adjusted the weights and proportion of the formula to improve it through trial and error, but we were not able to obtain the optimum that has even divisions and minimum cost.

Table 2. 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=0.4, In=0.384
Terminal steps	2000

Table 3. Parameters of n-TSP.

The number of B cells	25
The number of activated B cells	25
Add city rate	0.5
Swap mutation rate	0.5

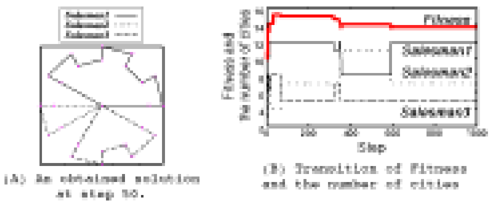

Figure 9. Results.

4.3. Summary of the Experiment

According to the above results and Toma et al (2000-b), our method has two advantages (1) modification of the work-domain is changed while maintaining the fitness, and (2) the average costs are very small compared with GA. By contrast, the downside is, if the effect of maintaining is too strong, dividing evenly becomes difficult. According to these results, improvement for the equality in the task of division-of-labor problems is required. In this instance, our algorithm has been improved so that the immune cell-cooperation which is a framework including the MHC and the immune network may be applied rather than having it applied directly to the local functions.

5. EXTENDED ALGORITHM

In this section, we propose an extended immune optimization algorithm, which is based on the immune cell-cooperation, to solve the division-of-labor problems. In previous work for MAS (Toma et al, 2000-b), the immune distributed competitive problem solver had two problems; the even work domain and divergence could not be computed as mentioned in Section 4.3. The previous algorithm has been improved so that the immune cell-cooperation, which is a framework in a broad sense of elimination of antigens, may be applied rather than having it applied directly to the local functions.

The extended algorithm solves the division-of-labor problems through the interactions between the agents, and between agents and environment by immune functions. There are three functions in our algorithm: division as well as integration processing and co-evolutionary-like approach. The division as well as integration processing optimizes the work domain, and the co-evolutionary approach realizes equal divisions. Through implementation of such functions, it was possible to construct an adaptive algorithm that solves the division-of-labor problems. Thereupon, the proposed algorithm was applied to n-TSP. Some computer simulations are designed to clarify the basic performances as well as the characteristics and features of the extended algorithm.

5.1. Modeling of Immune Cell-Cooperation

5.1.1. Biological Immune Cell-Cooperation

In the biological immune system, the two mechanisms (MHC and immune network) are considered important in eliminating invaded antigens (see Janeway et al, 1997). Thereupon, a framework, which together with the two functions eliminates the unknown vast antigens in parallel, is called the `immune cell-cooperation'. From the engineering system's point of view, the immune cell-cooperation is considered 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 their functions (see Figure 10). Note that, the cell-cooperation has a set of rules not only as `function’ (fragmentation, activation) but also as `work domain’ (as defined in Section 2.1). The object of the model is to implement and to apply them to the multi-agent system.

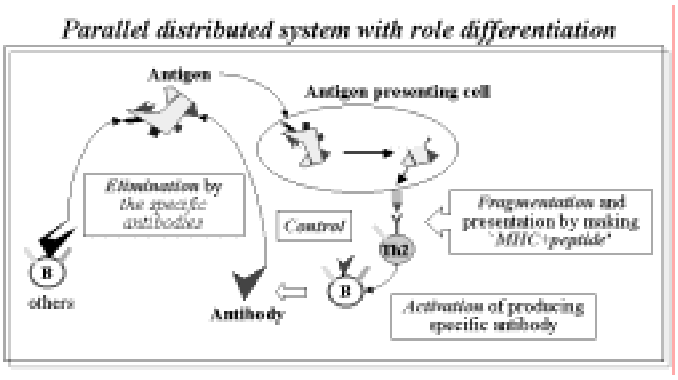

Figure 10. Concept of immune cell-cooperation.

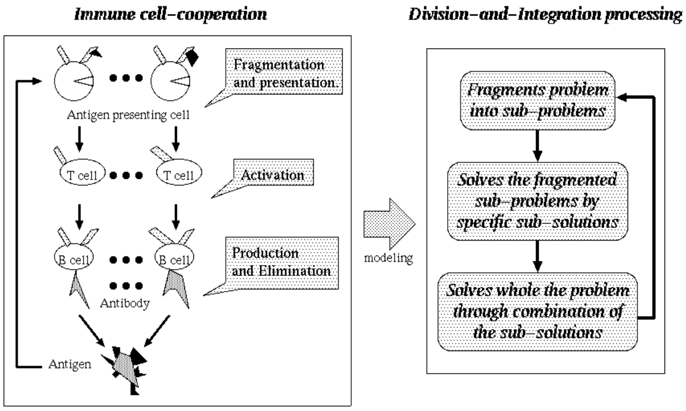

Figure 11. Analogy from immune cell-cooperation.

The fragmentation, the activation and the elimination functions have been especially chosen in this issue because their functions are minimum components to act as problem-solvers.

5.1.2. Analogy

As described in the beginning of Section 5.1.1, the biological immune cell-cooperation has four roles: fragmentation, activation, elimination and control of these functions. In order to apply the system to the division-of-labor problems, the three functions were chosen in this issue, namely the fragmentation, activation and elimination, because their functions are the minimum components necessary to perform the function as the problem-solvers. An optimization algorithm was constructed based on the concept that (1) fragments a problem, (2) solves the fragmented sub-problems by the specific sub-solutions, and (3) solves the whole problem through combination of these sub-solutions.

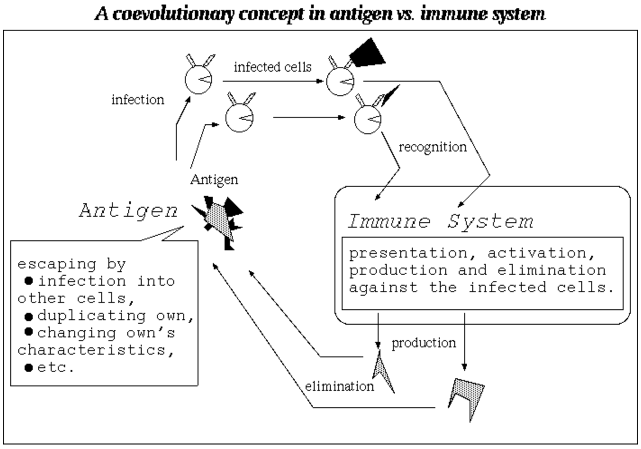

Figure 12. Illustration of a co-evolutionary like concept in the case of antigen vs. immune system.

Figure 13. Procedures of modified algorithm.

We call these procedures division-and-integration processing (Figure 11).

In addition, we introduced a co-evolutionary concept that the immune system evolves against the virus, which also evolves to escape from the elimination of the system. In order to apply this concept as a searching method in MAS, we constructed a procedure in the likeness of the escaping virus from the changing environments (Figure 12).

5.2. Extended Algorithm

As shown in Figure 13, the algorithm solves the problems through the two searching methods, (1) division-and-integration processing by agents and (2) co-evolutionary like procedure by components in the environment. The procedures of the extended algorithm against n-TSP are as described below. In the procedures of proposed method, Step 1 is processed only once by the System for initialization.

Each agent processes are repeated from Step 2 to Step 6 for the optimization of the problem.

[Step1: Definition of problems and immune functions.]

Cities and Salesmen as the problems must be defined. Salesman's unique ID and division as well as integration processing for tours of each salesman must be defined. At this point, each city has the ID as the expression of salesman visiting the city.

[Step2: Calculation of objective function.]

The cost of salesman is calculated by following function.

$$Cost(S_i) = \sum distance(tour_{S_i}) \qquad (2)$$

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

[Step3: Division processing with Simulated Annealing.]

One salesman tries to divide his own tour into two sub tours in search of lower costs according to these steps (see also Figure 14-(A)). First, we need to decide on one sub $tour_j$ for new $salesman_j$. Second, we need to map a new $tour_i$ for $salesman_i$ except for the sub $tour_j$ and third, we need to calculate the cost of both new sub tours.

When the cost after the division is improved, then the division shall be carried out. Therefore, in the case of DIVISION performed, the original salesman generates a new agent for attaining results that are more efficient.

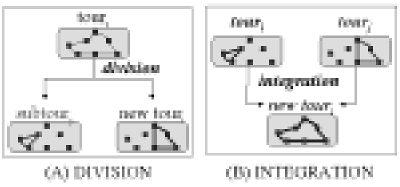

Figure 14. Examples of Division and Integration.

[Step4: Integration processing.]

The two salesmen try to integrate the tours in search of lower costs by following the steps (see also Figure 14-(B)). First, we need to decide on any two tours and integration point ptr has to be made. Second, routing of new tour crossing the ptr has to be scheduled. Third, the costs of the new tour have to be computed. If the cost, after integration, is improved, the integration will be carried out. Therefore, in the case of INTEGRATION performed, the original salesman kills the other salesman for making an efficient result.

[Step5: Mutation.]

Swap any two cities. In the event the costs after swapping are improved, the swap will be carried out.

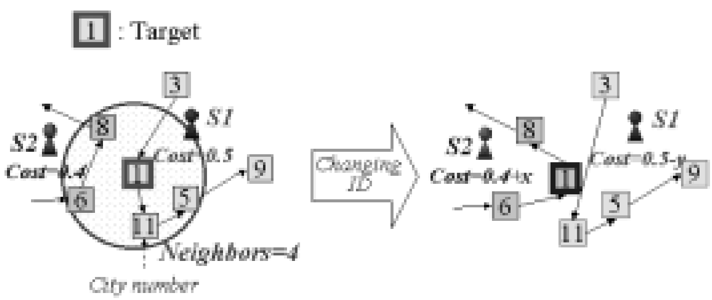

Figure 15. A co-evolutionary like approach.

[Step6: Co-evolutionary like Procedure.]

This approach changes the ID of a given city depending on the costs to and from the neighboring city and the costs to process the following steps (see also Figure 15). First, check the costs of salesmen that visited neighboring cities of the target city. The neighbors are defined as N-cities close to the target city. In this example, the neighbors are visited by S1 and S2. Second, if the other salesman's cost is lower, then the city changes its ID into that of the other. Consequently, in this example, the ID of the target city is changed to S2, and then, the target city is visited by S2 as shown in the right side of Figure 15.

In this manner, each agent searches for lower costs, and each component in the environment searches for even divisions. Note that, in order to increase the number of execution times of division processing, `Simulated Annealing' is implemented at Step 3.

6. EXPERIMENT FOR EXTENDED ALGORITHM

6.1. Design of Simulations

Two computer simulations are performed to investigate the basic performance of the extended algorithm. In the first simulation, we applied the algorithm to the problem in Table 2 and then we considered the results. In the second simulation, we applied the Simple Genetic Algorithm (SGA) to the same problem and the two performances were compared. The parameters of the extended algorithm are defined in Table 4.

Table 4. Parameters of the extended algorithm.

The number of neighbours	4
The number of initial agents	Under 24
Terminal steps	2000

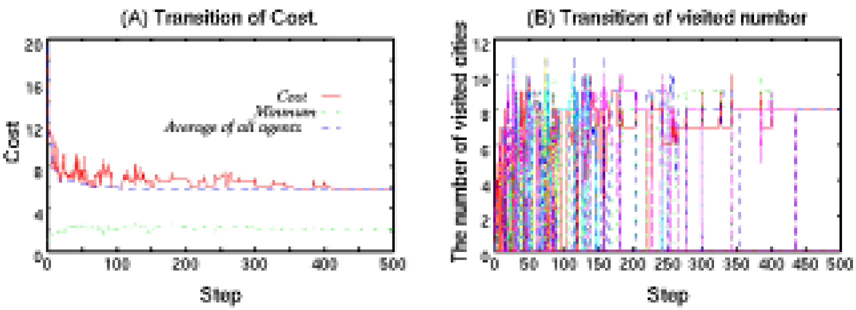

Figure 17. Transition of cost and the number of visited cities.

6.2. Experiment 1: System Behavior

Figure 16 shows an acquired solution. Because of the extension of our method, the best solution led to equal divisions and optimized the costs. The transition of the number of visited cities for each agent is shown in Table 5. The number of executions of the division processing improved owing to the Simulated Annealing. As a result, the integration processing is performed frequently and effectively.

Table 5. Distribution of the number of searching solutions.

	Trials	Times of execution
DIVISION	6,242	211(6)
INTEGRATION	546	225(16)
Mutation	6,037	50
CITY operation	47,876	47,876
Total	60,701	48,372

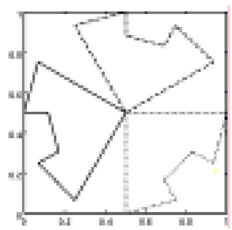

Figure 16. Results.

In order to verify the behaviors of our method, the transition of the cost, the sum of the whole agent's costs, and the number of visited cities (dividing-number) are shown in Figure 17. The cost improves dramatically up to Step 50. A new combinational situation that alters the division-of-labor from multiple agents visiting multiple cities to three to five agents visiting approximately eight cities occurred. This occurrence is caused under the primary influence of the equalizing divisions in Step 6 of the extended algorithm. Here, a rise in the average costs for all agents is observed. In the subsequent steps, the optimization is performed repeatedly while maintaining the approximately even divisions. Finally, the transition of fitness and dividing-number converges in the optimum solution, which is a combination of eight cities with three agents in Figure 16.

6.3. Experiment 2: Comparison with Simple GA

6.3.1. Design of Simple GA

The results were compared with both methods and SGA. The parameters of SGA are set in Table 6. The $fitness_i$, which is the fitness of agent i is expressed in equation 3.

The candidate solutions must be encoded for it to be applied to GA in n-TSP. We adopted the pass representation method originally proposed by Yamamura et al (1992).

Table 6. Parameters of SGA.

Population	100
Crossover rate	1.0
Mutation rate	0.01
Terminal generations	5000

$$fitness_i = 1/(Cost + Penalty) \qquad (3)$$

- *Cost*: Length of the tour
- *Penalty*: Penalty for the equality

6.3.2. Results

The searching performances are compared through simulated results of 20 trials. Table 7 shows, (1) the number of the best solutions obtained, (2) the minimum, average, and the maximum cost, (3) the number of searched solutions (for example, a sum of individuals in the case of SGA), and (4) run time. In all terms, our method had better results with smaller run time than SGA. In particular, this algorithm is capable of searching for superior solution quickly on the average.

Table 7. Simulation results of 20 trials.

		SGA	Extended Algorithm
Best Cost	Times	3	15
	Min(best)	5.845231	5.845231
	Ave	6.403267	5.858212
	Max(worst)	6.949701	5.858212
Tne number of searched solutions	Min(best)	74,100	1,616
	Ave	167,000	8,488
	Max(worst)	227,100	16,580
Run tme		50sec	10sec

7. CONCLUSION

An extended immune optimization using division as well as integration processing based on the immune cell-cooperation is proposed, and the target problem has to be the optimization of the division-of-labor problems.

This method solves the problems through combination of division, integration and co-evolutionary approach. These functions are based on local interactions between the agents, and between agents and environment. In MAS, clarification of the objective function considering all agents and components in the environment is too hard a problem but it is important in the optimizing of the whole problem by using the local interactions. Since this algorithm is capable of optimizing the division-of-labor problems, it could be expected to function effectively as the optimization algorithm in MAS. Future work would (a) analyze behaviors, adaptabilities and performances of our method, (b) define applicable problems and required conditions for application, (c) compare with similar and conventional methods, and (d) investigate its application to a more specific problem as a suitable subject for this algorithm. In addition, further work will (e) implement the other functions in the immune cell-cooperation as described in the beginning.

8. ACKNOWLEDGE-MENTS

The authors would like to acknowledge Prof. Y. Itoh (Lymphocyte Biology Section, Laboratory of Immunology, National Institute of Allergy and Infectious Diseases, National Institutes of Health, Bethesda MD 20892) and members of Laboratory of Harmonious Systems Engineering, Research Group of Complex Systems Engineering, Division of Systems and Information Engineering, Graduate School of Engineering, Hokkaido University.

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

9. REFERENCES

Back, Thomas. Proceedings of The Seventh International Conference on Genetic Algorithms. 1997; Michigan State University, East Lansing.

Dipankar Dasgupta and Zbigniew Michalewicz, Evolutionary Algorithms in Engineering Applications, Springer-Verlag, 1997.

Dipankar Dasgupta. Artificial Immune Systems and Their Applications. Springer, 1999.

David Corne, Marco Dorigo, and Fred Glover. New ideas in optimisation. 1997; McGraw-Hill Publishing Company.

J.D. Farmer, N.H. Packard, and A.A. Perelson. The Immune system adaptation, and machine learning. Physica 22D, 1986; 187-204.

S. Forrest, and A.A. Perelson. Genetic algorithm and the Immune system. Proc. of 1st Workshop on Parallel Problem Solving from Nature 1990; 320-325.

D.E. Goldberg. Genetic algorithm, search optimization and machine learning. 1989; Addison Wesley.

Holland, J.H.. Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology. 1992; MIT Press.

Ishida, Y., Hirayama, H., Fujita, H., Ishiguro, A. and Mori, K. Immunity-Based Systems and Its Applications. 1998; CORONA.

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

S. Russell and P. Norvig. Artificial Intelligence: A Modern Approach. 1995; Prentice-Hall.

Toma, N., Endo, S. and Yamada, K. The Proposal and Evaluation of an Adaptive Memorizing Immune Algorithm with Two Memory Mechanisms. Journal of Japanese Society for Artificial Intelligence. 2000; Vol.15, No.6, pp.1097-1106.

Toma, N., and Endo, S., Yamada, K. and Miyagi, H. The Immune Distributed Competitive Problem Solver with MHC and Immune Network. Intelligent Engineering Systems through Artificial Neural Networks 2000; vol.10 (editor C.H. Dagli et al.), ASME PRESS SERIES (ISBN 0-7918-0161-6), pp.317-322

Yamamura, M., Ono, T., and Kobayashi, S. Character-Preserving genetic algorithms for Travering salesman problem. Journal of JSAI 1992; Vol.7 No.6, pp.1049-1059.