Evolutionary Optimization Algorithm using MHC and Immune Network

NARUAKI TOMA, SATOSHI ENDO, KOJI YAMADA, HAYAO MIYAGI

Department of Information Engineering
Faculty of Engineering, University of the Ryukyus
1 senbaru, Nishihara,Okinawa 903-0213, Japan
E-mail: tnal@eva.ie.u-ryukyu.ac.jp, {endo,koji,miyagi}@ie.u-ryukyu.ac.jp

Abstract

The objective of this paper is to propose an evolutionary optimization algorithm using MHC and Immune Network and to verify its validity by means of computer simulations. Our algorithm solves the division-of-labor issues and problems for each agent's work domain in multi-agent system (MAS) by two immune functions. First, the Major Histocompatibility Complex (MHC) distinguishes a "self" from the other "nonself", used in the process of eliminating states of competition. Second, the Immune Network that produces specific antibodies by modification of immune cells is used to produce adaptive behaviors for agents. Then, 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 case problem of multi-agent system. The effectiveness of solving MAS will be clarified through some sets of simulations.

1 Introduction

Such neural networks and genetic algorithms for adaptive problem solving techniques are based on information processing in biological organisms and are applied on many kinds of optimization problems [1]. On the other hand, the biological immune system is widely recognized as one of the adaptive biological systems, the functions of which are to identify and to eliminate foreign materials. In the biological immune system, the following two mechanisms are considered important in performing the immune functions. First, Major Histocompatibility Complex (MHC) is used to distinguish a "self" from the other "nonself" when nonself invades self. Second, the Immune Network is composed of immune cells and their connections, and then specific antibodies are produced by modification of their immune cells.

In this paper, we propose an evolutionary optimization algorithm using two immune functions, the first is MHC and the second is Immune Network. Our algorithm solves the division-of-labor problems for each agent's work domain in multi-agent system (MAS). The MHC is used for elimination of the states of competition among agents. The Immune Network is used to produce adaptive behaviors for agents. Through implementation of such models, we could construct an adaptive optimization algorithm that solves division-of-labor problems. Thereupon, we apply the proposed algorithm to n-th agent's travelling salesman problem (called n-TSP) which is considered as a typical 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. Finally, we discuss remaining problems and future works.

2 MAS and division-of-labor problems

Multi-agent system which is a study in the field of distributed artificial intelligence is an information processing technique that autonomic agents solves problems through interactions among their agents. 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 aims of distributed problem solving]

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. 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 (see figure 1).

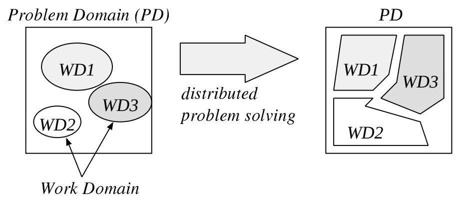


Figure 1: Aims of distributed problem solving.

3 Modeling of biological immune system

In order to construct above distributed problem solver, we developed the models of immune functions in the first instance. Our objective models are shown in figure 2 and the details of those words in *italics* are described below.

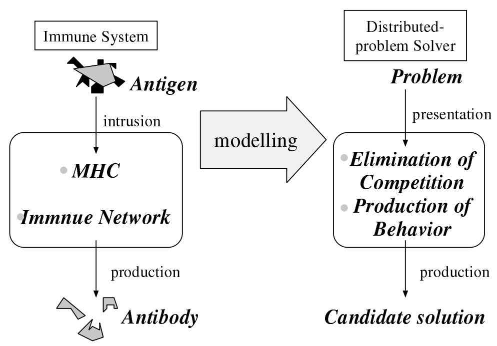


Figure 2: Modeling of biological immune system for distributed problem solver.

3.1 Elimination of competition among agents by *MHC*

In the biological immune system, the most important behavior is to preserve self. One of the self-preservation methods is the use of *MHC* [2]. This *MHC* is defined as unique set of information defining the individual himself. *MHC* is used to identify antigens by the difference of *MHC*, is expressed as MHC with peptide that is a information defining the antigen. If the infected *MHC* on the cells is different from the original (uninfected) *MHC*, the individual can recognize antigens by matching with T cell receptor (called TcR). Subsequently, if nonself, such as antigens, invades the self-system, the immune cell called Macrophage processes and displays invasion of nonself by difference of *MHC*. While the cells with the difference of *MHC* exist, the self-system continues to eliminate nonself system, in the above figure 3.

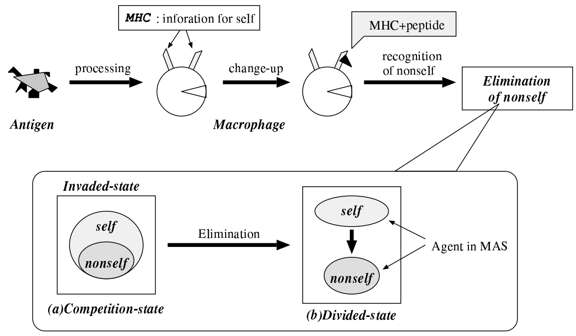


Figure 3: Elimination of competition states is based on elimination of nonself using MHC.

The elimination of nonself means removal of nonself system from the self-system, in figure 3's below. 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.

3.2 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 cells that has the capability to recognize the antigens.
- The activated B cell produces antibodies.
- Finally, the antibodies thus produced begin to eliminate of the invading antigens.

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

- In the production model, first, the agent acquires of information about environment and internal-state in the agent.
- Second, if the states of competition exist in the internal-state, the agent eliminates this states of competition.
- Third, the agent searches for 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, then the agent continues to process their procedures (see figure 4). Our algorithm solves the division-of-labor problems by the elimination of competition by *MHC* and production of behaviors by *Immune Network*.

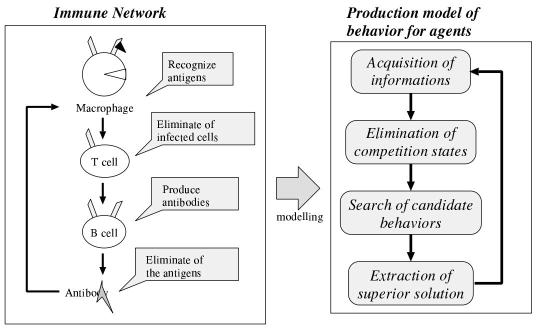


Figure 4: Production of behavior is based on production of an antibody using Immune Network.

4 Proposed method

4.1 Immune optimization algorithm

In our method, transmission of the MHC through the Immune Network and re-construction of the components produce specific antibodies. The MHC is an information carrier that can maintain information both of self and nonself by changing itself into MHC with peptide. The situation in which MHC acquires information of nonself is referred to as the states of competition. The details of our method are described as follows.

[**Step1. Definition of antigens and MHC**] The problem area where agents exist is defined as antigen and the internal state (e.g. solution, behavior, etc.) of each agent is defined as MHC.

[**Step2. Preparation of initial MHC and immune cells**] All the initial MHC and immune cells (Macrophage, T cell and B cell) are prepared at random.

[**Step3. Acquisition of information**] Macrophage acquires both of external and internal sets of information. The pieces of external information are described as MHC with internal information and the MHC is transmitted to T cell.

[**Step 4. Elimination of the States of Competition**] If the states of competition exists in the transmitted MHC, T cell eliminates these states by means of competition processing (the details are described in the next section).

[**Step5. Production of candidate solutions**] On the basis of MHC that comprise both the external and internal information that DOSE NOT include the states of competition, B cell N produces 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.

The four steps, from the 3rd to the 6th steps, are considered as one time step. This algorithm is adapted by repeating itself from step 3 to step 6 until the terminal condition is satisfied.

4.2 Competition processing

The competition processing is executed at the 4th Step in the above algorithm. It is a mechanism that 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. According to the terms, a state of competition is expressed in equation 1.

$$MHC(A) \cap MHC(B) \neq \emptyset \qquad (1)$$

[**Step1. Comparison of fitness in self-agent**] In the competition processing, a comparison is made between $fitness(A)$ and $fitness(A')$. If the $fitness(A')$ without the states of competition is high, the self-agent modifies the fitness from $fitness(A)$ to $fitness(A')$ and the MHC

Table 1: Definitions of terms. Consider case of competition states between agent A and B exist (formula (1)). Here 'after' states means that the time after was eliminated the competition. (ie. the states of A' is different from the states of A.)

	self agent: A	nonself agent: B
work space, now	$MHC(A)$	$MHC(B)$
work space, after	$MHC(A')$	$MHC(B')$
fitness, now	$fitness(A)$	$fitness(B)$
fitness, after	$fitness(A')$	$fitness(B')$

from $MHC(A)$ to $MHC(A')$ 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 the $f(A)$ is large, the states of competition are eliminated from agent B. Otherwise, the states of competition are eliminated from agent A.

4.3 Concept of behaviors and aspects

The proposed method consists of (1) elimination of competition using MHC and (2) production of behaviors using Immune Network. On basis of the construction, this method has antithetic adaptations. According to these interactions, our method efficiently works to solve the division-of-labor problems.

1. Each agent adapts in direction of raising the fitness on the inside itself by the production using Immune Network.
2. Each agent adapts in the direction of eliminating inefficient region in respective work domain through elimination of competition.

5 Experiments

Two computer simulations were run to investigate the performances of our method. In the first simulation, we applied our method to n-th agent's TSP and then, we reviewed the results. In the second simulation, the Genetic Algorithm (GA) was applied to the same problem and then, we compared the performances of the two.

5.1 N-th agent's TSP

We applied to n-th agent's Travelling Salesman Problem to verify adaptability of our model. Travelling salesman problem (TSP) is one of the most typical combinatorial optimization problems. The objective is to find a roundtrip of minimal total length visiting each node exactly once. The number of salesman in N-th agent's travelling salesman problem (n-TSP) was extended from singular to plural number. The objective is finding the minimal tour by division among salesman. Note that two conditions were set. First, the number of salesman is fixed while the system runs to solve. Second, all salesmen use the same city as the starting point.

5.2 Experiment 1. Definition of Simulation

In order to confirm the basic performances of our method, we apply the method to following n-TSP. The problem is defined in table 2. The parameters of proposed method are set in figure 5-(A). The $fitness_i$, the fitness of agent i, is expressed in equation 2.

$$fitness_i = Cities(MHC(i))/Cost(MHC(i)) + 1/Cost(MHC(i)) \quad (2)$$

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

Table 2: Parameters of n-TSP.

the number of cities	25
start city	center of circle
position of cities	dual circle
radius	out=0.5, in=0.4
the number of salesmen	3
terminal condition	200 time step

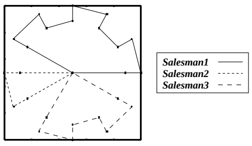
5.3 Experiment 1. Results and Discussions

Figure 5-(B) shows an acquired solution. The solution of each agent is considered as optimum solution in its own work-space. In case of simple nTSP with *a single circle*, it is clear that (a) the number of cities (without the starting point) visited by each agent are 8 and the even division of work-space is computed and (b) the solution of each agent is considered as the optimum solution [6]. However, even division of work-space isn't computed, in case of table 2. The fitness function caused the results because interactions for work-space are strictly based on the value of fitness. Therefore, designing the fitness function should be improved directly or indirectly.

Next, in order to verify the behaviors of our method, the transition of the fitness and the number of visited

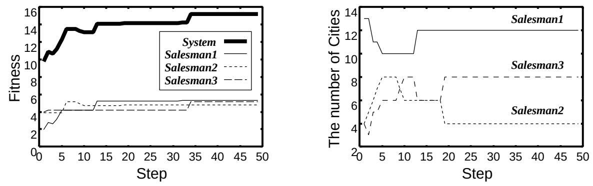
Macrophage	1
T cell	1
B cell	25
Activated B cell	5
add city rate	0.5
swap mutation rate	0.5

A) Parameters of proposed method



(B)Acquired a solution for division-of-labor problems.

Figure 5: Parameters and result.



(A) Transition of fitness for system and each agent.

(B) Transition of the number of visited cities for each agent.

Figure 6: Results: transition of fitness and the number of visited cities, in case of our method.

cities are shown in figure 6. In figure 6-(A), System is the total fitness of all agents. Figure 6-(A) shows that the each fitness transits monotonous increase or flatness as a whole although one or more decreases, and the increase of fitness occurs at the steps where the movements of their work-spaces are large (figure 6-(B)). In our method, the decrease of fitness will occur when elimination processing is executed by opponent agents. However, the decrease almost never. It is considered that the work domain of each agent is moved while maintaining the fitness because the fitness shows a stable increase, although their work domains are fluctuating widely. This is a good property in solving the division-of-labor problems.

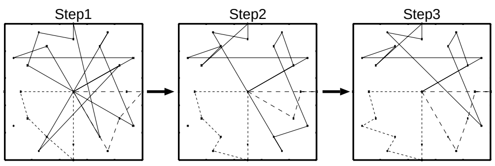


Figure 7: Changing process of solutions for each agent.

Figure 7 shows the changing of solutions for each agent in initial steps. Multiple modifications (addition of new cities or changing visiting order, elimination of states of competition) are shown all at once while maintaining the fitness. This property is good for modifying solutions, but it is weak point for dividing evenly.

5.4 Experiment 2. Definition of Simulation

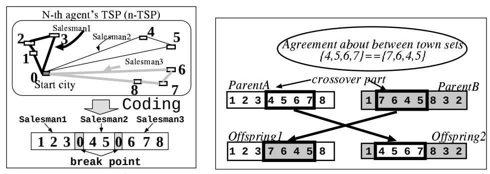
We compared the results of both our method and those of GA. The parameters of GA are set in figure 8-(A). The $fitness_i$, the fitness of agent i, is expressed in equation 3.

The candidate solutions must be encoded to apply GA to n-TSP. We adopted the pass representation method originally proposed by Yamamura [7]. In sub-tour crossover, the two sub-tours that have the same elements are looked up and those sub-tours were replaced. It proved possible that adoption of this method prevents the destruction of the important subtour. Coding and genetic operator are shown in figure 8.

$$fitness_i = 1/(Cost * Cost_W + Penalty * Penalty_W) \tag{3}$$

- $Cost$: length of the tour.
- $Cost_W$: weight of Cost.
- $Penalty$: penalty for the equality.
- $Penalty_W$: weight of penalty.

population number	100
crossover rate	1.0
mutation rate	0.01
terminal generations	1000



(A) Parameter of GA.

(B) An example of coding.

(C) An example of subtour crossover.

Figure 8: Parameters, coding and subtour crossover.

5.5 Experiment 2. Results and Discussions

Figure 9-(A) shows that the maximum fitness of GA transits monotonous increase as it was in our method, and the efficiency of evolution is good until generation 200. However, the costs that the solution converge in the associate solution are very high. The numbers of searched points in two methods are widely different. In fact, the number is $25 \times 3 \times 100 = 7,500$ in the case of our method. In the case of GA, the number of $100 \times 1,000 = 100,000$. On average results, our method's cost is less than one-tenth of GA.

The reasons that the cost of GA being so large are as follows. In our method, each agent can search independently and owns a part of a problem space in parallel. On the other hand, the GA has only but to search the whole of a problem space at a time. Actually, figure 9-(B) shows that the movement of the number of visited cities is only partly confirmed. In normal cases using GA, if sub-tour crossover is performed, the number of visited cities for each salesman should change. But the changes are minimal. It

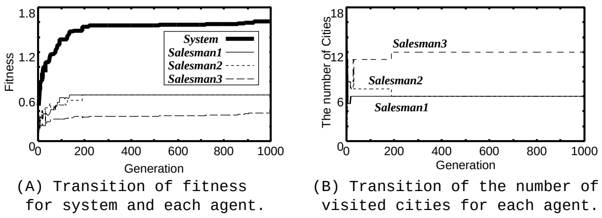


Figure 9: Results, transition of maximum fitness and the number of visited cities, in case of GA.

means that a genetic operator that performs evolution effectively will be mutation. The crossover, sub-tour crossover, cannot actively clear the obstructive part of problem space, although it is a very useful operator to maintain good schemata. As our conclusion, a certain function with the property of removing an obstructive part of problem must be mounted in order to solve effectively the division-of-labor problems in MAS.

5.6 Summary of the experiments

The following properties are confirmed through some sets of simulations. Our method has two advantages (1) modifying the work-space is changed while maintaining the fitness, and (2) the cost are very small compared with GA, on average. By contrast, the downside is, if the affect of the maintaining is too strong, dividing evenly is difficult. According to these results, improvement for the equality in the task of division-of-labor problems are required.

6 Conclusion

We proposed models of immune functions and an evolutionary optimization algorithm using their models. Our method consists of (1) elimination of competition using MHC and (2) the production of behaviors using Immune Network. This method was applied to n-TSP and its validity was examined. In the experiments, the property that the work domain of each agent is moved while maintaining the fitness was confirmed. Also, we can state that the investigation has proved that our method performs better than the GA. As future work, we're going to extend our method (e.g. extending the number of activated cells from one to multiple number, introducing two situations for cell interaction) and make comparisons with other methods. Furthermore, we'll consider a combinatorial optimization algorithm allowed integration or combination of the divided work-spaces.

Acknowledgments

We would like to extend our appreciation to: Prof. Y. Itoh (Lymphocyte Biology Section, Laboratory of Immunology, National Institute of Allergy and Infectious Diseases, National Institutes of Health, Bethesda MD 20892). The same appreciation is extended to the 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).

References

[1] Thomas Back (editor), "Proceedings of The Seventh International Conference on Genetic Algorithms, Morgan Kaufmann," *Michigan State University, East Lansing, MI.*, 1997.

[2] Charles A. Janeway, Jr / Paul Travers / Simon Hunt / Mark Walport, "Immunobiology : The Immune System in Health And Disease," *Garland Pub*, 1997.

[3] Y. Ishida, H. Hirayama, H. Fujita, A. Ishiguro, K. Mori, "Immunity-Based Systems and Its Applications," *CORONA*, 1998.

[4] D.E. Goldberg, "Genetic algorithm, search optimization and machine learning," *Addison Wesley*, 1989.

[5] S. Forrest, A.A. Perelson, "Genetic algorithm and the Immune system," *Proc. of 1st Workshop on PPSN*, pp.320-325, 1990.

[6] N. Toma, E. Satoshi, K. Yamada, H. Miyagi, "The Immune Distributed Competitive Problem Solver Using MHC and Immune Network," *Proc. of The 2nd Joint International Workshop - ORSJ Hokkaido Chapter and ASOR Queensland Branch -*, pp82-89, 2000.

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