THE IMMUNE DISTRIBUTED COMPETITIVE PROBLEM SOLVER WITH MHC AND IMMUNE NETWORK

NARUAKI TOMA, SATOSHI ENDO, KOJI YAMADA, HAYAO MIYAGI
University of the Ryukyus
Department of Information Engineering, Faculty of Engineering
Okinawa 903-0213, Japan

ABSTRACT
The objective of this paper is to propose an immune distributed competitive problem solver with MHC and Immune Network and to verify its validity. Our algorithm solves the division-of-labor problems for each agent's work domain by two immune functions. First, the MHC distinguishes "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. In order to investigate the validity of the proposed method, this algorithm is applied to N-th agent's Travelling Salesmen Problem (n-TSP) which is considered as a typical case problem in combinatorial optimization problems. The effectiveness of solving n-TSP will be clarified through some sets of 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 (Back, 1997). A biological immune system is one of the adaptive systems and the stdudies are making advances (Bersini 1991, Forrest 1990, Ishida 1996). In the biological immune system, the following two mechanisms are considered important in eliminating invaded antigens. 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 immune distributed competitive problem solver with MHC and Immune Network. Our algorithm solves the division-of-labor problems for each agent's work domain. 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 algorithm that solves division-of-labor problems. Thereupon, we apply the proposed algorithm to n-th agent's travelling salesman problem (n-TSP) which is considered as a typical case problem in combinatorial optimization problems. Some computer simulations are designed to clarify the basic performances as well as the characteristics and features of the proposed immune algorithm. Finally, we discus remaining problems and future works.

2 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. 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).

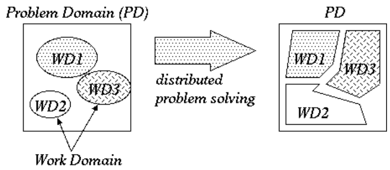


Figure1. Aims of distributed problem solving.

3 PROPOSED METHOD

3.1 Immune distributed competitive problem solver

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 (a piece of information of nonself). 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.

3.2 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.

[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 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 *fitness(A)* and *fitness(B)*. If the *fitness(A)* is large, the states of competition are eliminated from agent *B*. Otherwise, the states are eliminated from agent *A*.

Table 1. Definitions of terms. Consider case of competition states between agent A and B exist. Here 'after' states means that the time after was eliminated the competition.

	self agent: A	nonself agent: B
work domain, now	*MHC(A)*	*MHC(B)*
work domain, after	*MHC(A')*	*MHC(B)*
fitness, now	*fitness(A)*	*fitness(B)*
fitness, after	*fitness(A')*	*fitness(B')*

3.3 Concept of behaviors and aspects

Our method consists of elimination of competition using MHC and production of behaviors using Immune Network. On basis of the construction, this method has antithetic adaptations, (1) Each agent adapts in direction of raising the fitness on the inside itself by the production using Immune Network and (2) Each agent adapts in the direction of eliminating inefficient region in respective work domain through elimination of competition. According to these interactions, our method efficiently works to solve the division-of-labor problems.

4 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, we reviewed the results. In the second simulation, Genetic Algorithm (GA) was applied to the same problem and then, we compared the performances of the two.

4.1 N-th agent's TSP

We applied to *n*-TSP to verify adaptability of our model. 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*-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.

4.2 Experiment 1. Definition of Simulation

We compared the results of both our method and those of GA. The problem is defined in table 2. The parameters of proposed method are set in figure 2-(A). The $fitness_i$, the fitness of agent *i*, is expressed in 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 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

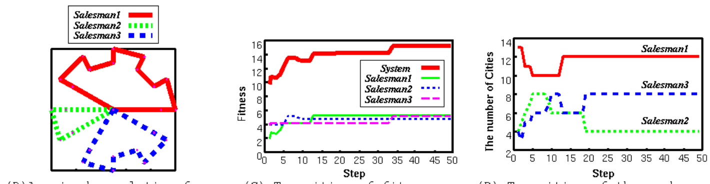
4.3 Experiment 1. Results and discussions

Figure 2-(B) shows an acquired solution. The solution is considered as optimum solution in its own work domain. In case of simple nTSP with *a single circle*, it is clear that (a) the number of cities visited by each agent are 8 and the even division of work domain is computed and (b) the solution of each agent is considered as the optimum solution (Toma, 2000). However, even division of work domain isn't computed, in this time. The fitness function caused the results because the interactions 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 cities are shown in figure 2-(C),(D). In figure 2-(C), System is the total fitness of all agents. Figure 2-(C) 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 domains are large (figure 2-(D)). 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.

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. (C) Transition of fitness for system and each agent. (D) Transition of the number of visited cities for each agent.

Figure2. Parameters and results, in case of our method.

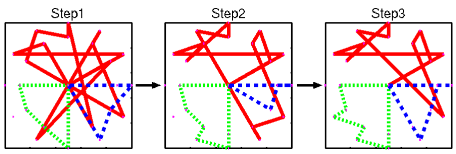


Figure 3.Changing process of solutions for each agent, in initial steps.

Figure 3 shows the changing of solutions for each agent. 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.

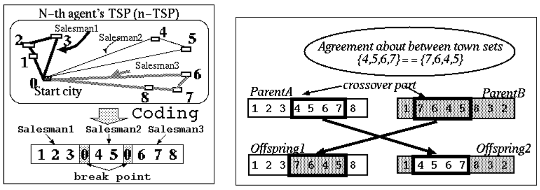
4.4 Experiment 2. Definition of simulation, in case of GA

The parameters of GA are set in figure 4-(A). The $fitness_i$, the fitness of agent i, is expressed in equation 2. The candidate solutions must be encoded to apply GA to n-TSP. We adopted the pass representation method originally proposed by Yamamura (Yamamura, 1992). Coding and genetic operator are shown in figure 4.

$$fitness_i = 1/(Cost \times Cost_w + Penalty \times Penalty_W) \qquad (2)$$

- $Cost$: 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

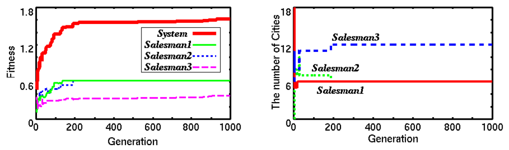


(A) Parameter of GA. (B) An example of coding. (C) An example of subtour crossover.

Figure 4. Parameters, coding and subtour crossover.

4.5 Experiment 2. Results and discussions

Figure 5-(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 $25\times3\times100=7,500$ in the case of our method. In the case of GA, the number is $100\times1,000=100,000$. On average results, our method's cost is less than one-tenth of GA.



(A) Transition of fitness for system and each agent. (B) Transition of the number of visited cities for each agent.

Figure 5. Results, transition of maximum fitness and the number of visited cities, in case 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 domain in parallel. On the other hand, the GA has only but to search the whole of a problem domain at a time. Actually, figure 5-(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 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 domain, 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.

5 CONCLUSIONS

We proposed an immune distributed competitive problem solver with two immune functions. 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 following properties are confirmed through some sets of simulations. Our method has two advantages (1) modifying the work domain 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. As future works, 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 domains.

ACKNOWLEDGEMENTS

We would like to extend our appreciation to Prof. Y. Itoh (Lymphocyte Biology Section, Laboratory of Immunology). The same appreciation is extended to the Members of Laboratory of Harmonious Systems Engineering, Hokkaido University.

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

REFERENCES

T. Back (editor), 1997, “Proceedings of The Seventh International Conference on Genetic Algorithms, Morgan Kaufmann,” Michigan State University, East Lansing, MI.

H. Bersini and F. J. Varela, 1991, “The Immune Recruitment Mechanism : A Selective Evolutionary Strategy,” Proc. of ICGA 91.

D.E. Goldberg, 1989, “Genetic algorithm, search optimization and machine learning,” Addison Wesley.S.

Forrest and A. S. Perelson, 1990, “Genetic Algorithm and the Immune System,” Proc. of PPSN 90, pp.320-325.

Y. Ishida, and N. Adachi, 1996, “Active Noise Control by an Immune Algorithm,” Proc. ICEC 96, pp.150-153.

N. Toma, E. Satoshi, K. Yamada, H. Miyagi, 2000, “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.

M. Yamamura, T. Ono, S. Kobayashi, 1992, “Character-Preserving genetic algorithms for Traveling salesman problem,” Jurnal of JSAI, Vol.7 No.6, pp.1049-1059.