A Study on Immune Optimizations for Multimodal Functions and Division-of-labor Problems

Year 2003

Doctoral Course in Interdisciplinary Intelligent Systems Engineering,
Graduate School of Science and Engineering,
University of the Ryukyus

Naruaki Toma

Contents

Chapter 1

Introduction

An Optimization, is one of the field of operations research, is known widely as important study, which is built on any engineering models [Goldberg 1989, Holland 1992, Kitano 1993, Sakawa 1995]. The target problems at an early stage were mainly objective functions such as linear one or continuous one. As optimization methods for the problems, linear programming as typified by simplex method that made use of the continuity and hill climbing based on concept of differentiation were found to be useful. On the other hand, applying their methods to a problem, which is a discontinuous function or the solution space consists of discrete sets, is difficult because they cannot use the continuity to solve it. Suchlike problems whose problem space is a set of discrete are called as combinatorial problems. The solving methods for combinatorial problems classify roughly into an exact algorithm and an approximation (or heuristic) algorithm from the optimality of solution's point of view. An exact algorithm always provides the exact solution, often called global optimum of the optimization problem. Unfortunately, some exact algorithms may have such a high computational cost that their use can be unreasonable on problems of typical size. Hence, approximation algorithms are required. Approximation algorithms are not guaranteed to find the exact solution, but can provide good approximation of the exact solution, which may be valid for practical applications. Approximation algorithms are often called heuristic algorithms, because they use problem-solving techniques based on experience. In this paper, optimization is carried out by a heuristic algorithm.

As approximation algorithms for combinatorial optimization problems, they are making advances in information systems that are based on information processing in biological organisms. In particular, adaptive problem solving techniques, such as neural networks and genetic algorithms, are applied on many kinds of optimization problems and are reported on their availability and efficiency through many kinds of studies [Holland 1992, Russell 1995, Thomas 1997, Kitano 1993].

Also, biological immune system is taken up as the third bioorganic system that lines up with neural system and genetic system [Dasgupta 1999, Ishiguro 1997, Ishida 1998,

Sasaki 1999]. Biological immune system consists of high information processing mechanisms such as diverse antibodies production mechanisms, self-regulating mechanism and primary and secondary immune responses based on antigen's specificity and immunological memory. As studies considered the biological immune system as optimization method, there are many reports in several fields and are argued about their potentialities as engineering models: for example, an immune simulator using cellular automata[Celada 1998], an immune network-based behavior arbitration mechanism for autonomous mobile robots[Ishiguro 1997], a multimodal function optimization[Mori 1993, Mori 1997], and so on.

As mentioned above, optimization problems are investigated from a long time ago. For the meanwhile, the complexity of problems gain more and more in connection with economic development, in addition, the demanded solution's characteristics are changed obsequiously. Especially, in multipurpose optimization problems, to reflect a designer (user) intended to the objective function is difficult work. And in some cases, a problem itself changes owing to changing constraint conditions. In other words, when a user got an exact solution, there are some possibilities that the exact solution is differ from the user intended (e.g., imagine a situation that a machine was break down or received an emergent task in scheduling problem). That is to say, just to find an optimal solution of the problems is insufficient for a user, so otherwise methods to solve the demand are strongly required. One of the answers, which a method supports to obtain robust solutions that can be available against such changes, is "to obtain multi optimal solutions at a problem solving". Herewith it is possible to provide some rooms to choose carrying solutions from multi alternatives.

Genetic algorithms (GA) with sharing[Goldberg 1987, Goldberg 1992] and niche method for GA[Shima 1995] have been proposed as methods which obtain multi optimal solutions before now, however both methods have constraints, which the number of individuals in the population (i.e., the population size) should be greater than the number of optimal solutions, and to make a required function or to adjust parameters are tough works. Immune algorithm (IA)[Mori 1993, Mori 1997] based on the biological immune system can obtain multi optimal solutions without the constraint.

The IA consists of memory cell, which obtains candidate solutions and suppressor cell that moves searching scope. The basic behavior is global search by GA. And according to their cells, the IA can suppress searching obtained solutions already, and then it can obtain multi optimal solutions in multimodal functions. However, there are two issues; (1) appropriate parameters adjustment is critical point, but any method for the adjustment doesn't support, and (2) the use of obtained good solutions carries out only for the second solving time. In short, there will be any rooms for use of obtained solutions.

This paper describes an Adaptive Memorizing Immune Algorithm (AMIA) with two memory mechanisms to improve above-mentioned issues [Toma 2000-a]. In point of first issue, the parameters adjustment gets easily though establishment obtaining memory and restraining re-search individually. About second issue, the search ability is enhanced through use of obtaining memory and secondary immune response in first solving time also. The in-

troduced memory mechanisms consist of following two information processing mechanisms.

primary memory mechanism:
It memorizes common characteristics in the candidate group as a template in the memory cell. And then, it carries out narrowing down of search space through a candidate group is reproduced based on the template.

secondary memory mechanism:
It memorizes a search point that dominates whole population in the suppressor cell. And then, it carries out restraint to search same point again using the memory.

AMIA with the mechanisms can expect to have following characteristics.

- the parameters adjustment is easy because obtaining good solutions and restraining search same point again are separated as the two memory mechanisms independently.
- the local search ability is accelerated through narrowing down search scope.

That is to say, AMIA will carry out efficient search by using appropriately global search and local search.

In order to evaluate the proposed method, AMIA is applied to two kinds of multimodal deceptive problems: Traveling Salesman Problem (TSP) and Bipolar Deceptive Function[Goldberg 1992], and the availability and the efficiency are verified. The experiment results show that AMIA is able to obtain multi optimal solutions with lower generations, and the efficiency for searching sub-optimal solutions is superiority. According these results, this paper concludes AMIA is superior method for multimodal optimization problems.

Furthermore, this paper attempts an application to multi-agent system (MAS) to verify the further potentialities. MAS, which is a study in the field of distributed artificial intelligence. MAS is an information processing technique in which autonomous agents solve problems by interactions among the agents [Russell 1995], and can be expected that has robustness, adaptability and stability [Ishida 1996]. In order to achieve such characteristics, elucidating the following things is required.

1. how to allocate work-domains to multi-agent?
2. how organization is tough for environment changes and/or any troubles?
3. how to communicate between agents efficiency?

In this paper, main objective is to construct two kinds of immune based optimization algorithms for the division-of-labor problems, which attach importance to the first issue. 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. As first algorithm for division-of-labor problems, this paper describes an immune distributed competitive problem solver with MHC and Immune Network. This algorithm solves the division-of-labor problems for each agent's work domain. The MHC is used for eliminations of the states of competition among agents. The Immune Network is used to produce adaptive behaviors for agents. Through implementation of such models, it is constructed that an adaptive algorithm that solves division-of-labor problems.

However, in the previous work [Toma 2000-b], the immune distributed competitive problem solver has two problems; the solver is not able to compute even work domain and the divergence. And so, as second algorithm, this paper describes 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 MAS[Toma 2001, Toma 2002-a, Toma 2002-b]. 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 escape processing. The division as well as integration processing optimizes the work domain, and the escape processing realizes equal divisions. Through implementation of such functions, it is constructed that an adaptive algorithm that solves division-of-labor problems. Thereupon, the proposed algorithm is applied to the n-th agent's traveling salesman problem (called the n-TSP)[Nakamura 1994], 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.

The goal of this thesis is to clarify the potentialities of the engineering models inspired from a biological immune system.

Chapters from 2 to 5 describe a multimodal functions optimization based on immune models. Chapter 2 takes an overview of primary and secondary immune responses and restraint system in a biological immune system, and considers the useful system as an optimization method. Chapter 3 defines the issues of the multimodal functions and clarifies the advantages and disadvantages of the conventional methods. As an immune approach to resolve the disadvantages, chapter 4 describes an Adaptive Memorizing Immune Algorithm with two memory mechanisms. AMIA gets superior efficiency by using appropriately the global search and local search based on the memory mechanisms. Chapter 5 shows the computational simulations to validate the principle behaviors and the performances of AMIA.

Chapters from 6 to 10 describe two division-of-labor problem optimizations based on immune models. Chapter 6 defines the issues of the division-of-labor problems and takes an overview of MHC and immune network, and considers the engineering models for the problems. Chapter 7 describes the details of an immune distributed competitive problem solver, which solves the problems through competitions between the immune agents that behaves to expand the work domain and to reduce one. The basic performances are investigated in chapter 8, and it is clarified the disadvantages of this competitive approach. Chapter 9 proposes and evaluates an immune optimization algorithm using a biological immune co-evolutionary phenomenon and cell-cooperation. This immune co-evolutionary algorithm solves division-of-labor problems in MAS through the three kinds of interactions: *division-and-integration processing* is used for optimization of the work-cost of immune agents and, *escape processing* is used to perform equal work assignment as a result of evolving the antigen agents. And the antigen agent computes even division of work domain using escape processing based on a phenomenon that the antigen evolves to escape from the elimination of immune system. The availability and the validity are investigated in the chapter 10.

Finally, from engineering's point of view, the potentialities of the engineering models inspired from a biological immune system are summarized in chapter 11. I hope you notice that the models exploited a biological immune system will have many important considerations.

Chapter 2

Biological immune system

2.1 Introduction

In the field of artificial intelligence (AI), studies about information processing systems that imitated human recognition and analogism, and approximations of mathematical logic to human analogism are investigated. As results, such knowledge-based systems, which based on symbol processing and so on have been produced as engineering applications [Russell 1995, Hirota 1996]. In late years, information systems, which take a hint from information processing in biological organisms, are gained force for their investigations (Figure 2.1). Genetic algorithms (GAs) and evolutionary computations (ECs) are algorithms inspired from principles of a biological evolution, and now they are made studies as new stochastic search methods, learning methods and optimization methods. And artificial neural networks (NNs) are engineering models inspired from neural system, and are investigated in the field of pattern recognition [Holland 1992, Russell 1995, Thomas 1997, Kitano 1993].

On the other hand, a biological immune system, which is a kind of intelligent systems in the human body, begins to make a mark as a paradigm of new bioorganic system [Dasgupta 1999, Ishida 1998]. In this chapter, an overview of the biological immune system is described, and some models for engineering applications are defined.

2.2 An overview of biological immune system

A biological immune system illustrated on figure 2.2. The system is a kind of defense mechanism to eliminate antigens using subsystems. Some essential subsystems are antigen recognition system, memory mechanism, antibody production system and antibody restraint system[Janeway 1997]. From an entineering's point of view, this system can be regarded as an adaptive optimization system on the various kinds of dynamic environments.

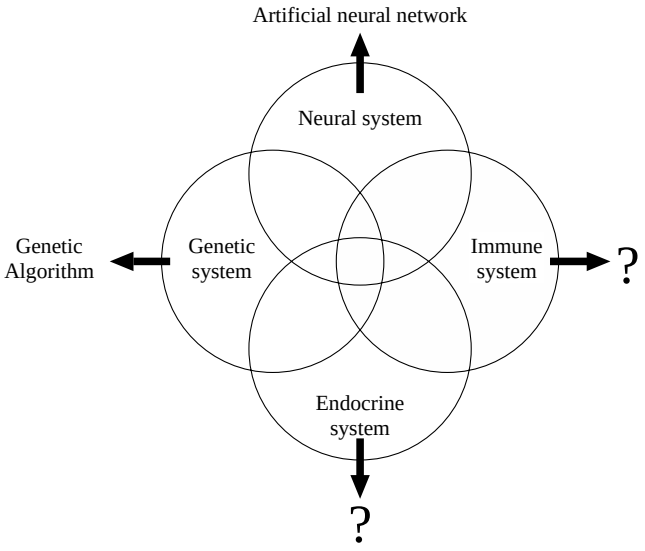


Figure 2.1: Information systems inspired from information processing in biological organisms.

A series of the following processing against an antigen is called immune response.

Step1. antigen recognition
When lymph system recognizes an invaded antigen, then begins immune response.

Step2. antibody reproduction
Antibody forming cell produces appropriate antibodies, which can eliminate the antigen through reproductions (selection, crossover, mutation, duplication and so on).

Step3. antigen elimination by antibodies
By decomposition or neutralization of the antigen, it carries out biophylaxis.

Step4. memory mechanism of antibodies used for elimination
For the antigen, which eliminated once, it carries out antigen elimination rapidly by using the memory that is memorized in memory cell.

Step5. suppression of production for antibodies
After it produces many antibodies, the diversity of a set of antibodies in the human will be lack. It controls the diversity to improve the situation.

The biological immune system consists of some functional subsystems, in particular there are three subsystems for an adjustment of antibody production.

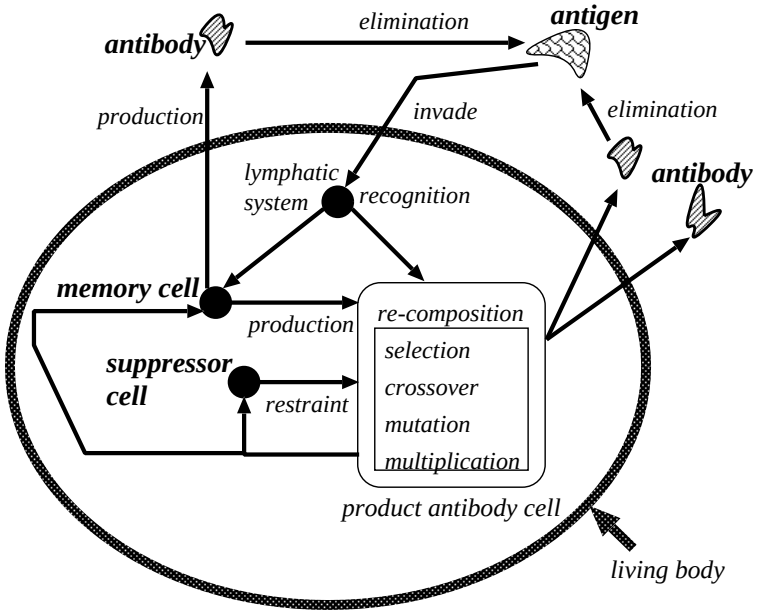


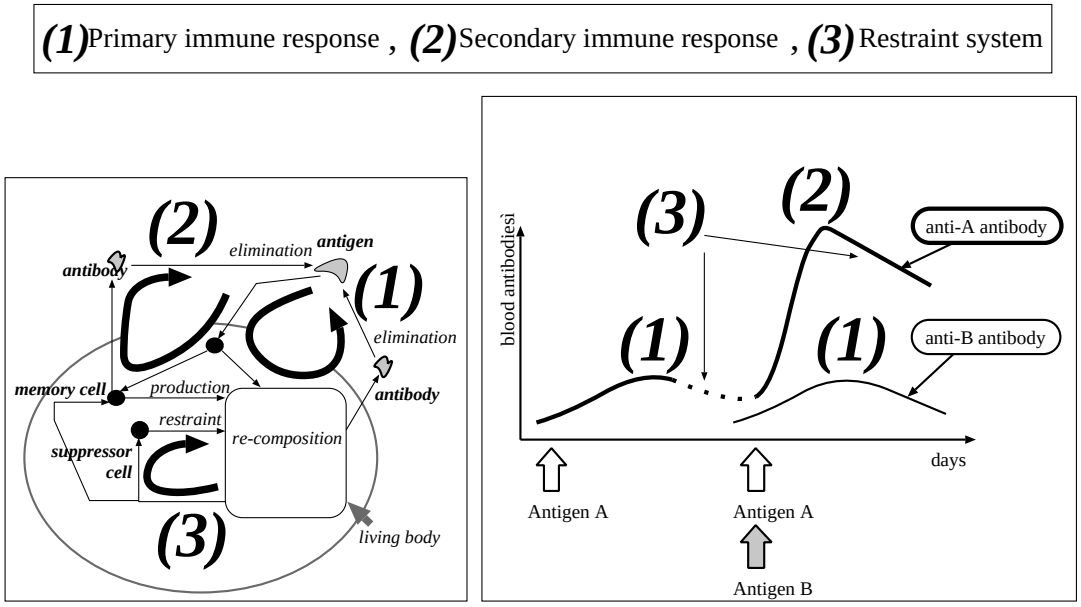
Figure 2.2: Biological immune system.

1. **primary immune response** (Figure2.3 (1) :
   This response is one of the antigen elimination procedures against unknown antigens. Immune cells such as antibodies which used for antigen eliminations in this response have been differentiated as memory cells (the concept is called as **immunological memory**).

2. **secondary immune response** (Figure2.3 (2)) :
   This response is one of the antigen elimination procedures against the antigens which are eliminated previously at least once. It rapidly carries out antigen elimination with the assistance o using the memory cells obtained by the primary immune response.

3. **restraint system** (Figure2.3 (3)) :
   By keeping the diversity of antibodies through restraint of producing similar antibodies, it get prepared for invading unknown antigens.

The secondary immune response can carry out rapidly antigen elimination because the antibodies which are capable to respond with the invading antigen in early stage than the primary immune response (Figure2.3 (b)). A biological immune system uses the **immunological memory** and the **specificity** to achieve the above-mentioned subsystems.



(a) immune system (b) differences between the two immune responses

Figure 2.3: Primary and secondary immune responses and restraint system

2.3 Immune system as an engineering model

As they expressed with the previous section, the two kinds of immune responses are achieved by using the immunological memory and the specificity. In general, it is considered that an understandable approach is to make a model for them. However, the implementation of the specificity requires constructing following subsystems:

- recognizes a problem should be solved,
- abstracts the characteristics the problem has, and
- distinguishes them.

These issues will provide useful guidelines how to design a problem. However, the model implemented the specificity depends on a preference of a designer, the problem it self and so on. In other words, the model will be lack generalization. Consequentially, this paper deals with the construction and the implementation of a model for the secondary immune response and the restrain system, which the model is introduced the immunological memory.

Figure 2.4 shows a relationship between the biological immune system and a search system as a kind of engineering applications. The relationship is outlined below.

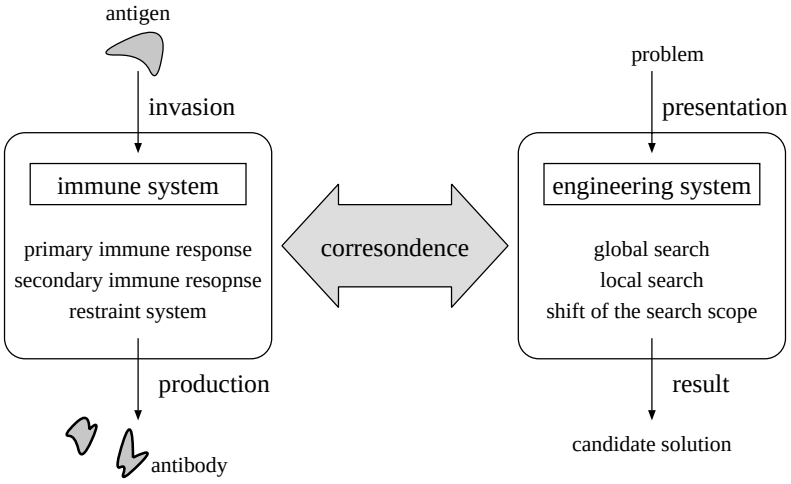


Figure 2.4: Biological immune system and engineering system.

2.3.1 Antigen and antibody

First of all, when we want to apply the biological immune system as an engineering model, we have to decide two points: (1) what is the antigen for the immune system and (2) what does the antibody to eliminate the antigen mean. The objective in the biological immune system is to carry out a biophylaxis through elimination of the invaded antigens. On the other hand, the objective in the search system is to find out optimal solutions against the problems. Here, by analogy the objective in the biological immune system from the engineering system's point of view, we can obtain following definitions.

- the invaded antigen = the problem
- the produced antibody to eliminate the antigen = the optimal solution against the problem

Below, it is described how to define the framework of the biological immune system, which produces appropriate antibodies to eliminate the invaded antigen using the subsystems such as the primary immune response, as a search method to find optimal solutions using the subsystems.

2.3.2 Primary and secondary immune responses

In the biological immune system, the secondary immune response, which can carry out effective problem solving as antigen elimination, exploits memories obtained at primary immune response on the after second elimination only. But, if the use of the secondary immune response on the first problem solving is implemented, the search method will be get superior performance. In this paper, the models of the three subsystems (the primary and secondary immune responses and the restraint system) in the biological immune system are constructed and applied as an approach for a search system.

At first, we start to consider the roles of the immune responses to apply as a search algorithm. The memory obtained at the primary immune response is considered as a kind of learning results, which is carried out for antigen elimination. The biological immune system is achieving effective antigen elimination by using the learning results when the eliminated antigen previously invades the system again. By analogy for the immune responses from search method's point of view, it is considered that (1) the primary immune response has to search on a situation without foresighted information against unknown antigens, on the other hand, (2) the secondary immune response produces the appropriate antibodies rapidly by exploitation of the learning results as the foresighted information. In this paper, the two immune responses are defined as followings to apply as search method.

- primary immune response = global search on a situation without the foresighted information
- secondary immune response = local search exploited the foresighted information

According to the definitions, we can achieve to construct a search method introduced a local search with the foresighted information, like as an exploitation of the secondary immune response at the first problem solving.

2.3.3 Restraint system

This section describes a consideration of the role of the restraint system in order to exploit as a search algorithm. In the biological immune system, the antibodies, which are restrained by the restraint system, dominate a set of antibodies. The state, which has dominated antibody, decreases the diversity, and denotes that the adaptability against unknown antigens is lower. The biological immune system adjusts the diversity through the restraint of antibody production in order to improve the adaptability. This restraint system is considered as an adjustment mechanism, which adjusts the diversity of the search population like GA, by analogy from a search method's point of view. In particular, the restraint system is defined as a following mechanism, which restrains to search the solutions that have high-dominated rate in the population.

- restraint system = shift of the search scope for the dominated population

2.3.4 Search algorithm using an immune system

In order to construct a search algorithm exploited the biological immune system, the primary and secondary immune responses and the restraint system are defined as search models. As mentioned Figure2.5, the methods are summarized as follows:

1. carries out a global search on a situation without any foresighted information,
2. carries out a local search exploited the obtained memories by the global search, and
3. adjustments the diversity of a set of solutions through a shift of the dominated search scope.

The Immune Algorithm (IA) described in chapter3 is a method, which introduced the primary immune response and the restraint system in Figure2.5. The secondary immune response isn't adopted to the IA. In contrast, the Adaptive Memorizing Immune Algorithm (AMIA) proposed in this paper adopts the three methods in Figure2.5 included the secondary immune response. Consequently, AMIA gets superior efficiencies by using appropriately the global search and the local search. The details are described in chapter4.

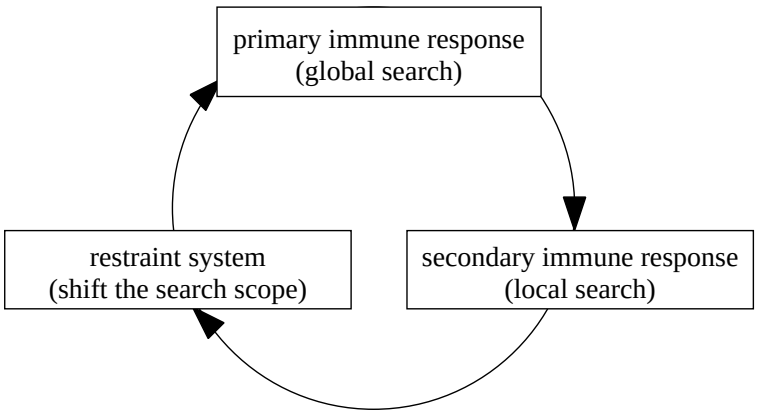


Figure 2.5: Biological immune system and an approach for search method

2.4 Conclusion

This chapter described an overview of a biological immune system, which begins to make a mark as a new paradigm based on the bioorganic system, and defined the subsystems in

the biological immune system as an engineering system. Especially, we can construct an optimization system based on the following principal of behaviors through an exploitation of the secondary immune response and the restraint system as search mechanisms.

- a global search by antibodies production with high diversity.
- a local search exploited the memories as the foresighted information.
- an adjustment of the diversity of the set of solutions through a shift ot the dominated search scope.

Next chapter describes the multimodal functions, which are used for the investigation of the efficiency of proposed method.

Chapter 3

Multimodal functions optimization

3.1 Introduction

Optimization, is one of the field of operations research, has been known widely as an important study that is built on any engineering models because any optimization method is required when we want to solve the faced problems [Goldberg 1989, Holland 1992, Kitano 1993, Sakawa 1995].

A general definition is formulated as a issue that "find a solution x, which maximizes or minimizes an objective function $f(x)$, and the solution x fills a constraint condition F in the solution space X" (equation(3.1), (3.2) and (3.3)). The solutions fill a constraint condition F are called as executable solutions.

$$opt_x \ f(x) \tag{3.1}$$

$$subject\ to\ x \in F \tag{3.2}$$

$$F \subseteq X \tag{3.3}$$

Optimization problems have been investigated from a long time ago. For the meanwhile, the complexity of problems gains more and more in connection with economic development, in addition, the demanded solution's characteristics are changed obsequiously.

A target problem at an early stage was mainly an objective function such as a linear one or a continuous one. As optimization methods for the problems, linear programming as typified by simplex method, which made use of the continuity and hill climbing based on concept of differentiation, were found to be useful. On the other hand, applying their methods to problems, which is a discontinuous function or the solution space consists of discrete sets, is difficult work because they couldn't use the continuity to solve it. Suchlike optimization problems whose problem space is a set of discrete are called as combinatorial optimization problems. The solving methods for combinatorial problems classify roughly

into exact algorithm and approximation (or heuristic) algorithm from the optimality of solution's point of view. In this paper, optimization is carried out by a heuristic algorithm.

As optimization methods to find the sub-optimal solution in the multimodal functions, there are the followings algorithms [Kitano 1993, Hirota 1996].

- heuristics methods such as Greedy Algorithms and a local search.
- methods exploited biological information processing systems such as Neural Networks and Genetic Algorithms.

In addition, hybrid algorithms, which are combined these methods, and specialized algorithms against the target problems are proposed in large numbers, and then, their methods obtain an exact solution frequently. However, there are many situations, which change some constraint conditions and/or objective function itself in the real world. Therefore, a user requires not only exact optimal solution, but also plural number of alternative solutions [Sakawa 1995].

This chapter describes the characteristics of the multimodal functions, and takes a general view of some conventional methods. In particular, the issues from optimality's point of view are considered about the Immune Algorithm, whish is a new optimization method based on a biological immune system.

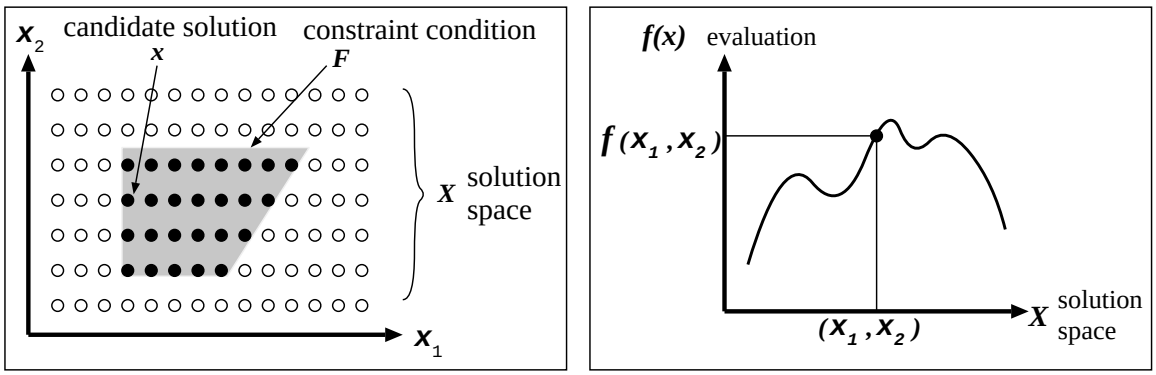
3.2 Multimodal functions optimization

A typical case problem of combinatorial optimization problems consists of the discrete solution space and the fitness landscape with multimodality(Figure 3.1). A solving method for the combinatorial optimization problems will use an enumerative approach because it cannot exploit the continuity of the solution space to solve it. Such many problems are NP-complete or NP-hard one are reported. And also, to enumerate all possible solutions is unrealistic approach even the total number of executive solutions is finite. Therefore, how to limit the scope of enumerating solution for a shortening of search time is important.

The solving methods for combinatorial problems classify roughly into the following ones.

- exact algorithm: always provides the exact solution.
- approximation (or heuristic) algorithm: are not guaranteed to find the exact solution, but can provide good approximation rapidly of the exact solution, which may be valid for practical applications.

Some methods used in the exact algorithm narrow the enumerating scope in the all-possible solution space, but they require huge computational costs because they search the limited space stem to stem. The approximately algorithms narrow the search space exploited the



(1) solution space with a set of discrete (2) multimodal function

Figure 3.1: Combinatorial optimization problems

foresighted information about the problem without exactly processing like exact algorithm. On the occasion of narrowing the search space, it is important to mount in a well-balanced for the following two cross demands:

- it is desired to search space as larger as (i.e., global search) in order to obtain good quality solution, on the other hand,
- it is desired to narrow the search space as much as possible (i.e., local search) in order to solve the problem rapidly.

In the cross two demands, the methods, which attach a high value to former demand, are random search (or Monte Carlo method), and the latter ones are hill climbing methods or greedy algorithms.

These solving methods are single-point search methods, which move a solution in the solution space based on some rules. Genetic Algorithm (GA), which has so many varied studies, is a multi-point search methods, so GA can search plural number of solutions in the solution space in parallel. GA is one of the approximately algorithms because the local search ability is low even has strong global search ability as known.

Next, this section takes a general view of the change of demands in real worlds, which the demands are required for obtained results in scheduling problems. The scheduling problem is optimizing the task schedules on the machine(s) when the tasks are processed by one or multi machines. In the scheduling problems, it is considered some inconvenient situations: the change of any constraint conditions by increase and/or decrease of the number of machines, the change of the objective functions by additions and omissions of tasks, and so on. In addition, in multipurpose optimization problems, to reflect a designer (user)'s intent

to the objective function is difficult work. And in some cases, the problems changes owing to changing constraint conditions. In other words, when an user got an exact solution, there are possibilities that solution is differ from the user intended. That is to say, just to find an optimal solution of the problems is insufficient for a user, so otherwise methods to solve the demand are strongly required. One of the answers, which a method supports to obtain robust solutions that can be available against such changes, is "to obtain multi optimal solutions at a problem solving". Herewith it is possible to provide some rooms to choose carrying solutions from multi alternatives.

The next section describes convenient approaches to obtain multi optimal solutions, and considers about their characteristics and the remaining issues.

3.3 Conventional methods for obtaining multi optimal solutions

3.3.1 Genetic Algorithm, GA

Many methods for obtaining multi optimal solutions are based on GA, which is multi-point search method in parallel. So, at first, an overview of GA is described. GA is a method to obtain high-evaluated solution through an evolution of a set of individuals to adapt an environment, which the evolution is carried out based on an evaluation from the environment [Kitano 1993]. Figure3.2 shows the algorithm of GA. GA can find good quality solution when the following things are appropriate designed by trial and error in general.

- cording,
- fitness function,
- genetic operator (selection, crossover, mutation),
- some parameters adjustment.

Many of their design guidelines are the issues depended on the environments (problems), so the hybrid algorithms, which is introduced heuristics or local search method have been investigated in order to run GA appropriately.

3.3.2 Sharing

When an optimization was carried out by GA, we obtain frequently no more one optimal solution even though there are multi optimal solutions. This is caused by the initial convergence, which is characteristics of GA, after that, we couldn't obtain other optimal

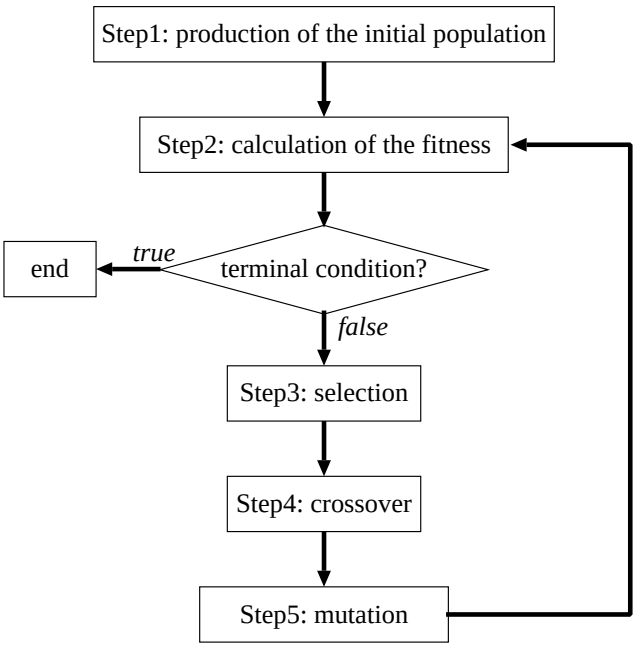


Figure 3.2: Framework of GA

solutions. For this issue, Goldberg et al. succeeded to obtain multi optimal solutions using sharing, which the individuals were allocated on multi spaces head off the initial convergence [Goldberg 1987, Goldberg 1992].

Sharing is an operator to achieve a relative uniformization in the distribution of the function value through larger weighting to the function value on which the individuals are centered, and smaller weighting to the function value on which the individuals are isolated. In mathematically, they think a distance $d(s_i, s_j)$ between corded strings s_i, s_j, termed a function $sh(\cdot)$ satisfied the following conditions as 'sharing function'.

$$0 \leq sh(d) \leq 1, \ \forall d \in [0, \infty) \tag{3.4}$$

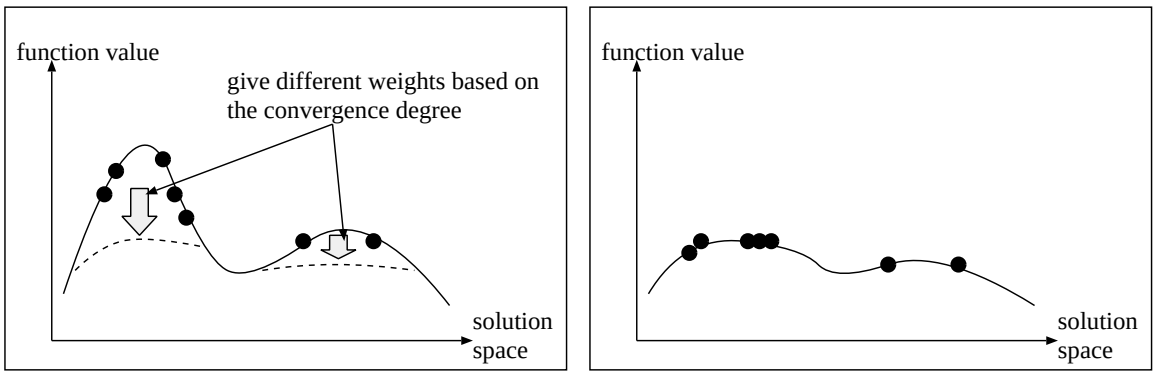
$$sh(0) = 1 \tag{3.5}$$

$$\lim_{d \to \infty} sh(d) = 0 \tag{3.6}$$

And then, they proposed that an exponential function as follows was useful to prevent the initial convergence:

$$sh(d) = \begin{cases} 1 - \left(\frac{d}{\sigma_{share}}\right)^{\alpha} & d < \sigma_{share} \\ 0 & otherwise, \end{cases} \tag{3.7}$$

where σ_{share} and α are constants.



(1) actual solution space (2) weighted solution space

Figure 3.3: Sharing.

Assuming that the fitness function for the individual s_i is $f(s_i)$ and the population size is N, the modifications of the fitness are carried out as below.

$$f(s_i) := \frac{f(s_i)}{m_i} \tag{3.8}$$

$$m_i = \sum_{j=1}^{N} sh(d(s_i, s_j)) \tag{3.9}$$

In this way, the prevention of the initial convergence of GA was achieved by an uniformization in the distribution of the function value. On the other hand, there are some remaining issues:

- the population size must be larger than the number of optimal solutions, and
- the adjustment of the parameters σ_{share} and α are difficult.

3.3.3 Niche method for GA

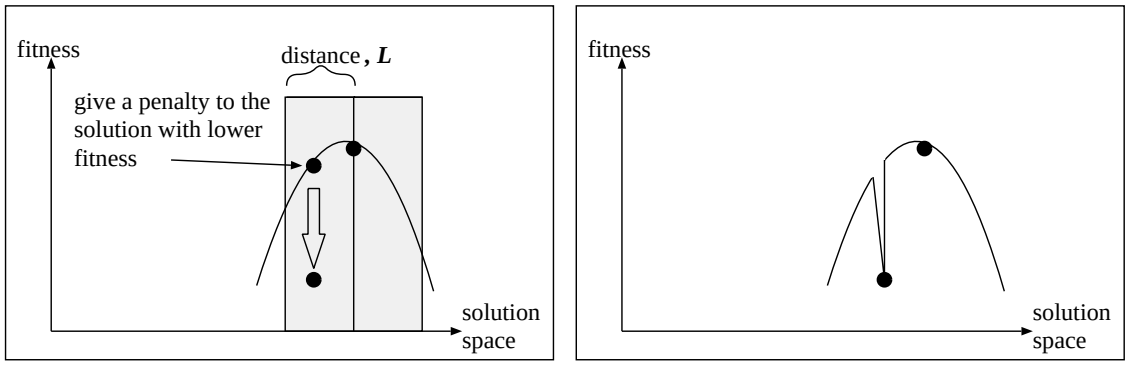
As a similar method to the above-mentioned sharing, there is a niche method for GA, which the method succeeded to obtain multi optimal solutions through introduction of a penalty for the distance between the individuals. [Shima 1995].

Let's assuming, for an explanation, that the number of the parent individuals is N, each parent makes 1 offspring, and now, there are $2N$ individuals. In order to achieve niche, the following procedure is introduced as a penalty between the $2N$ individuals.

- compare the distance between the $2N$ individuals in increments of 2,

- if the 2 individuals get close to each other smaller than a constant value L, compare each fitness, and then, the fitness of worse one is taken a penalty even though the fitness of better one leave well alone.

In this way, the individual with low fitness will get worse fitness because the individual is taken a penalty, so the possibility of a selection pressure for the individual will be very high (Figure3.4). As results, each individual be dispersed in the solution space, which the distances keep larger than L.



(1) actual solution space (2) solution space with penalty

Figure 3.4: Niche method for GA

The new point is to introduce the penalty for the distance between the individuals in order to achieve niche. In the computational experiments, this method got better results than sharing, which the target problem is Shubert function (equation(3.10), (3.11)).

$$\min f(x_1, x_2) = \left\{ \sum_{i=1}^{5} \cos[(i+1)x_1 + i] \right\} \times \left\{ \sum_{i=1}^{5} \cos[(i+1)x_2 + i] \right\}, \tag{3.10}$$

$$subject\ to \quad -10 \leq x_i \leq 10 \quad (i = 1, 2). \tag{3.11}$$

In this wise, the niche method gets superior method to obtain multi optimal solutions, but it has some issues similar to sharing:

- the population size must be larger than the number of optimal solutions, and
- the adjustment of the constant value L must be decided by trial and error because the constant depends on the problems.

3.3.4 Immune optimization

First of all, the definition of an immune optimization is described before the IA, which was proposed Mori et al[Mori 1993, Mori 1997].

The fitness landscape in a general combinatorial problem makes multimodal functions (Figure3.5). The objective in a general optimization is to find an exact solution, which has maximum fitness in the fitness landscape. However, when we use the exact solution for problem solving, some issues arise. As mentioned to keep repeating arguments, to reflect precisely a designer (user)'s intent to the objective function is difficult work. And there are any troublesome situations in the real world. One of the answers, which a method supports to obtain robust solutions that can be available against such changes, is "to obtain multi optimal solutions at a problem solving". Herewith it is possible to provide some rooms to choose carrying solutions from multi alternatives.

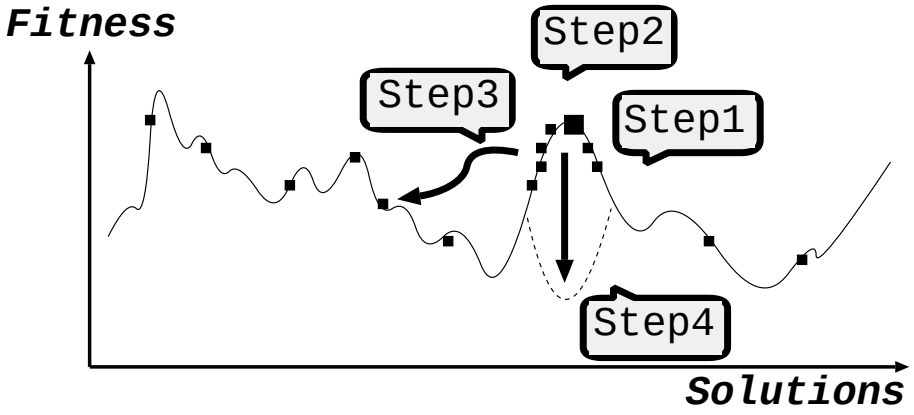


Figure 3.5: Principle of the behaviors of the immune optimizations in the multimodal functions.

The immune optimization obtains multi optimal solutions by additions of the following four processings while GA based search is running (Figure3.5).

Step1. Calculation of the degree of the convergence in the search scope. When the degree of the convergence exceeds a threshold value, the following steps are executed.

Step2. Acquisition of the solution, which exceeds the threshold, as a memory.

Step3. Shift the search scope to other scope.

Step4. Restraint to search the same solution again.

In order to achieve these processings, Immune Algorithm use fitness, similarity, concentration and expectation value.

[**fitness** $fitness_v$] evaluation of a solution v.

[**similarity** $ay_{v,w}$] the rate of common genes between 2 solutions v, w.

[**concentration** c_v] the rate of existence of a solution v in the population.

$$c_v = (\sum_{w=0}^{PopSize} ac_{v,w})/PopSize \tag{3.12}$$

$$ac_{v,w} = \begin{cases} 1 & ay_{v,w} \geq TAC1 \\ 0 & otherwise \end{cases} \tag{3.13}$$

- $PopSize$: population size in GA.
- $TAC1$: threshold.

[**expectation** e_v] the rate that one solution v remains to next generation.

$$e_v = fitness_v \times \prod_{s=1}^{S}(1 - as_{v,s}) \tag{3.14}$$

$$as_{v,s} = \begin{cases} ay_{v,s} & ay_{v,s} \geq TAC2 \\ 0 & otherwise \end{cases} \tag{3.15}$$

- S: the number of suppressor cells
- $ay_{v,s}$: similarity between solution v and suppressor cell.
- $TAC2$: threshold.

The formulations of the fitness and the similarity are depends on the problem. Immune algorithm calculates the degree of the convergence in the processing Step 1 as a concentration, and if there is a solution, which exceeds a threshold, executes the following processings Step 2. The solution with high concentration is dominating in the population; in other words, the solution will have many partial solutions with high fitness (i.e., schemata). IA achieves to get superior efficiency through acquisition of such solutions as memories, and restraint of searching similar solution again, The restraint is achieved by decreasing expected value when the similarity between solution and suppressor cells exceeds a threshold $TAC2$.

3.3.5 Immune Algorithm

Sharing or niche method for GA had restrictions, which the population size must be larger than the number of optimal solutions and the adjustment of their function or parameters

is difficult work. Immune Algorithm (IA) proposed by Mori et al. is a search algorithm imitated antibody production system and autoregulation system, and can find multi optimal solutions independent the restriction [Mori 1993, Mori 1997].

Characteristics of IA

Though the early IA was designed to find one optimal solution, it was extended to find multi optimal solutions by additions of the following steps:

- differential mechanism (Step4) for the obtained solutions into memory cells and suppressor cells, and
- restraint mechanism (Step5) for antibody production by suppressor cells.

The extended IA carries out reproduction of solutions based on a strong global search of GA, acquisition of superior solutions by memory cell, and restraint of the obtained solution (Figure3.6). Figure 3.7 shows the algorithm of IA.

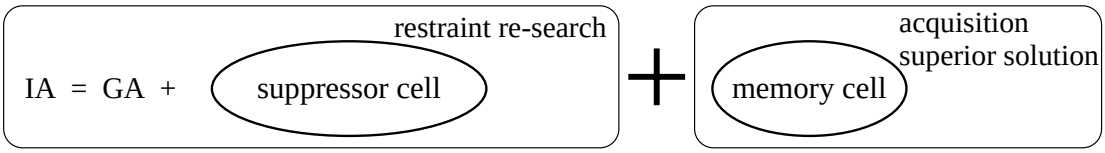


Figure 3.6: Framework of IA.

The remaining issues in IA

The immune optimization, which is basis on IA, is a search approach exploited the biological immune system. The immune optimization consists of the following four steps.

1. Calculation of the degree of the convergence in the search scope. When the degree of the convergence exceeds a threshold value, the following steps are executed.
2. Acquisition of the solution, which exceeds the threshold, as a memory.
3. Shift the search scope to other scope.
4. Restraint to search the same solution again.

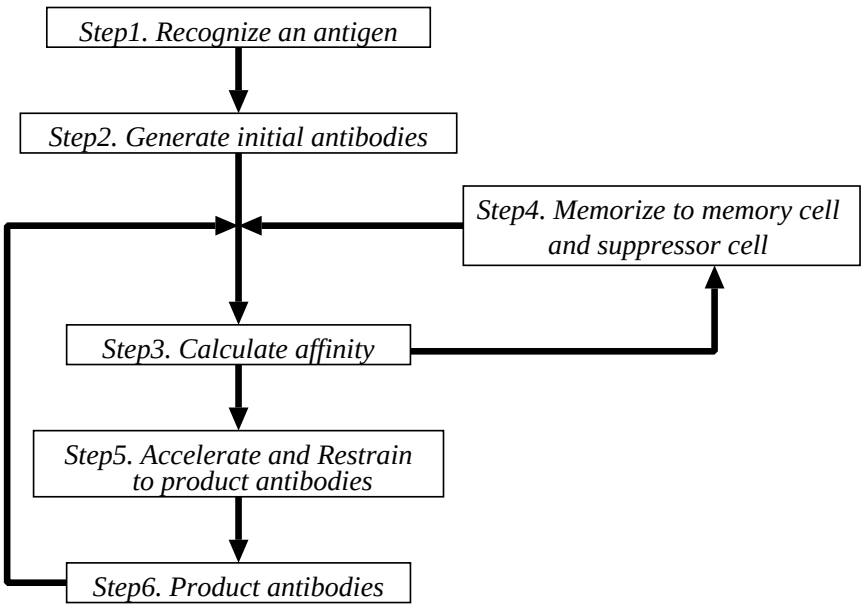


Figure 3.7: Immune Algorithm.

According to these configurations, the basic behavior of IA is carrying out (1) acquisition of candidate solutions and (2) restraint to search the obtained solutions again when the degree of the convergence exceeds a threshold value. This behavior will be achieved when the threshold in the 1. is designed appropriately, however, the adjustment of the parameter is difficult because:

- IA couldn't be obtained optimal solutions when the threshold is low, which IA obtains the solution as memory, and begins to restrain immediately.
- The calculation cost to obtain optimal solutions will become so huge when the threshold is high.

In this paper, Adaptive Memorizing Immune Algorithm in order to resolve this issue and improve the search ability has been decribed in the next chapter.

3.4 Acquisition of multi optimal solutions in the multimodal functions

An optimal solution obtained by an optimization becomes frequently to be lack the availability, the adaptability and the optimality owing to a change of some constraint conditions

and so on. As methods to resolve the issue, we took a general view of sharing, niche method for GA and IA such as a method to obtain multi optimal solutions. Especially, IA is considered as superior method than sharing and niche method because IA can find multi optimal solutions independent the restriction. However, the adjustment of the parameters is essential required, but this work is difficult.

In order to resolve the issue of the adjustment of the parameters, this paper proposes Adaptive Memorizing Immune Algorithm (AMIA), which achieves to be easy the adjustment and to enhance the local search ability. In AMIA, the parameters adjustment gets easily though an establishment obtaining memory and restraining search individually by separation as two kinds of memory mechanisms. In the meantime, the biological immune system achieves the secondary immune response, which carries out antigen elimination rapidly and efficiency using the memory cells obtained by the primary immune response. IA also exploits the memory cells and after the second time. In AMIA, by the use of the secondary immune response is exploited on the first problem solving, it is an aim for improving the search ability. The details of AMIA are described next chapter.

3.5 Conclution

This chapter described that to obtain multi optimal solutions, which will can provide some rooms to choose carrying solutions from multi alternatives, are strongly required because one exact solution is not available or useful sometimes in the real world. And as the conventional methods to obtain multi optimal solutions, we took an overview about sharing, niche method and IA.

In the next chapter, the details of AMIA as superior methods than the conventional methods are described.

Chapter 4

Adaptive Memorizing Immune Algorithm

4.1 Introduction

Before now,

- the definitions of a biological immune system as a engineering system in chapter 2, and
- the characteristics and the issues of the conventional methods to obtain multi optimal solutions in the multimodal functions,

were described. This chapter aims at the improvement of the efficiency by resolving the issues of IA through construction of a search approch based on the immune models mentions in chapter 2.

In this chapter, Adaptive Memorizing Immune Algorithm (AMIA) with two memory mechanisms is proposed as a method to obtain multi optimal solutions in the multimodal functions. At first, the issues of IA[Mori 1993, Mori 1997] are clarified, and AMIA, which improves the issues, are explained. Note that, simply "immune algorithm" denoted implies both IA and AMIA.

4.2 Characteristics and issues of IA

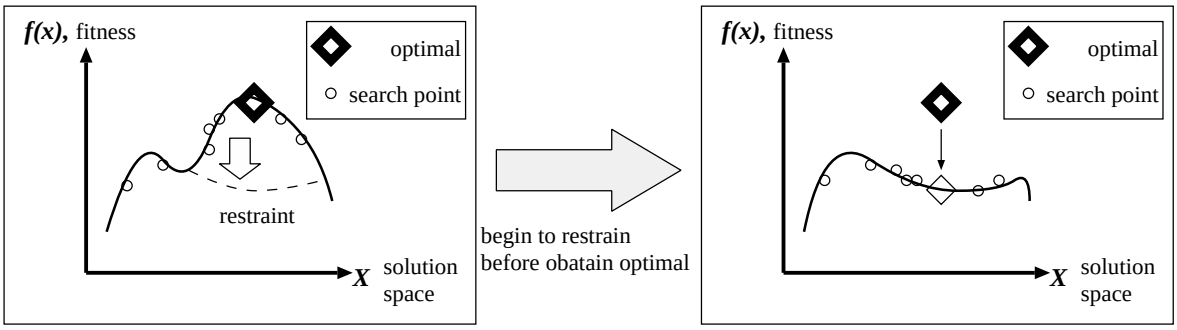
IA proposed by Mori et al. consists of reproduction of solutions based on GA, acquisition of superior solutions by memory cell, and restraint of the memorized solution (Figure3.6).

According to these constructions, IA achieves the behaviors: the acquisition of superior solutions and the shift of search scope, and so, IA can obtain multi optimal solutions.

However, (1) the issues of the adjustment of the parameters aren't resolving even though IA requires to adjust appropriately for the obtaining multi optimal solutions, and (2) there are any rooms for consideration about the use of memorized superior solution, which is exploited on and after second problem solving in IA.

(1) the difficulty of the parameter adjustment: IA carries out the acquisition of memory and the restraint to search the memorized solution again simultaneously when a solution, which have a high concentration exceeded than a threshold, arises. For that purpose, when IA memorizes a low fitness solution before obtains optimal solution in the current scoped peak, IA cannot obtain the optimal solution because a search for the peack is restrained after that (Figure4.1). However, when the threshold is configured as a high value to run memorizing after sufficient search, the acquisition of memory itself becomes to be difficult. Contrary, when the threshold is configured as a low value, IA will get done with inappropriate solutions.

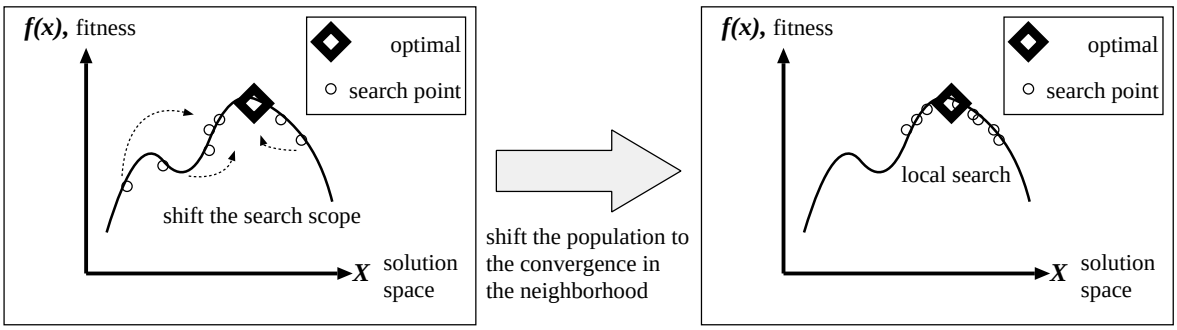
(2) the improvement of the search ability exploited memory cells: It is possible to enhance the local search ability by achieving a local search for the search scope similar to memorized solution (Figure4.2). In addition, the decrease of the generations for the problem solving will be achieved through using the local search than using only a global search.



(a) actual solution space (b) restrained solution space

Figure 4.1: The difficulty of the parameter adjustment in IA (lower threshold case)

In order to improve the above-mentioned issues, we introduce the two kinds of memory mechanisms to AMIA. In point of first issue, the parameters adjustment gets easily though establishment obtaining memory and restraining re-search individually. About second issue, the search ability is enhanced through use of obtaining memory and secondary



(a) the population before local search (b) the population after local search

Figure 4.2: Enhancement of the search ability exploited memory cells

immune response at the first solving time also. The introduced memory mechanisms consist of following two information processing mechanisms.

4.3 Adaptive Memorizing Immune Algorithm (AMIA)

An extended algorithm AMIA, which two kinds of memory mechanisms are introduced to the Step4, is shown in Figure4.3.

[Step1. Recognize an antigen]
An antigen is recognized as input information. That is decided the evaluation value $fitness_v$ of the solution v.

[Step2. Generate initial antibodies]
A set of antibodies is produced from the memory cells, which are good quality solutions at the previous problem solving. When there are no memory cells, the initial antibodies are made at random, which the genes of antibodies are selected randomly.

[Step3. Calculate fitness and affinity]
A fitness value $fitness_v$, which is a binding rate between the antibody v and the angiten, and an affinity $ay_{v,w}$, which is a recognition rate between the antibodies v, w, are calculated.

[Step4. Memorize (Differentiate) effective antibody to memory cell and suppressor cell]
The concentrations for all antibodies are calculated. When there is a antibody v

whose concentration c_v exceeds a threshold TC, it carries out memorizing the antibody into two kinds of memory mechanisms. In case of normal (global) search, the antibody is differentiated to primary memory mechanism. In other case (i.e., local search), the antibody is differentiated to secondary memory mechanism.

- **[Step4$_{Mem}$. primary memory mechanism]**
  The common characteristics of the candidate group are memorized as a template in the memory cell. An candidate group is reproduced based on the template, and a local search is accelerated.
- **[Step4$_{Sup}$. secondary memory mechanism]**
  The candidate solution, which has maximum concentration in the candidate group, is memorized in suppressor cell. The candidate solutions memorized in suppressor cell are restrained to search ones again. In addition, the search scope is shifted through a random reproduction of the current antibodies.

[Step5. Accelerate and restrain to product antibody]
A expectation e_v for the antibody v is calculated. According to the expectations, it is possible to modify the fitness landscape based on the similarity value of each solution.

[Step6. Product antibody]
A reproduction of an antibody is carried out through selection, crossover and mutation based on the expectations. AMIA continues to repeat from Step3 to Step6 until terminal generations.

Immune algorithms calculate the convergence rate of the search scope based on the affinities and the concentrations. By using the convergence rate to the expectations, immune algorithms are carrying out an autoregulation of the antibody production. The fitness and the affinity are designed inherently against each problem. The concentration and the expectation is calculated based on equation (3.12), (3.13), (3.14), (3.15) in section 3.3.4.

4.4 Memory mechanisms introduced to AMIA

This section describes the details of the two kinds of memory mechanisms:

- primary memory mechanism to achieve a local search exploited memory cell, and
- secondary memory mechanism to shift the search space exploited suppressor cell.

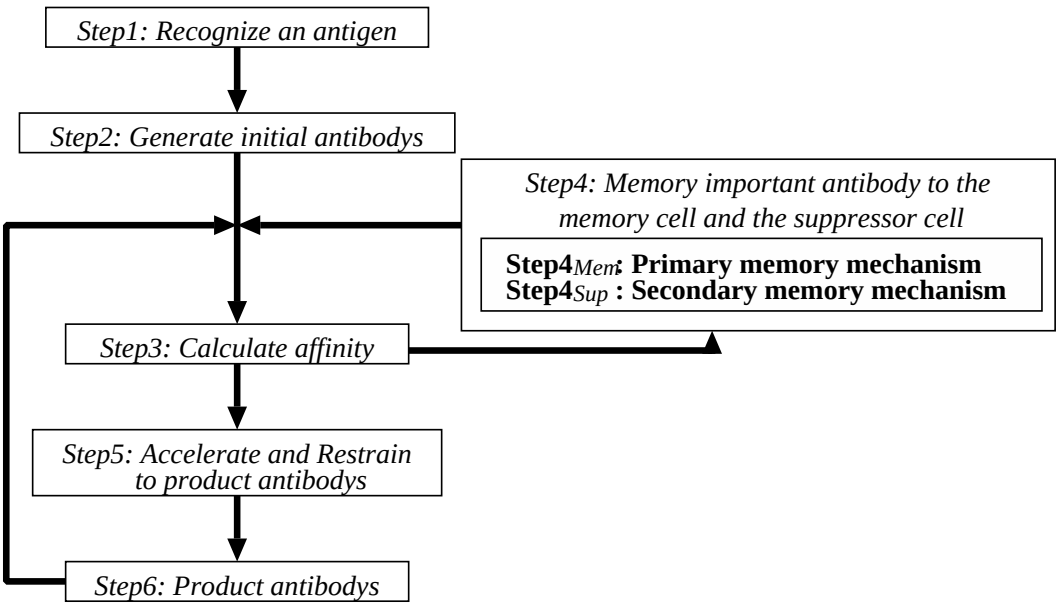


Figure 4.3: Adaptive Memorizing Immune Algorithm with two kinds of memory mechanisms.

4.4.1 The characteristics and the concept of the behavior of the memory mechanisms

The principal behaviors of the memory mechanisms consist of the three cycles: (1) a global search based on GA, (2) a local search based on the primary memory mechanisms, and (3) a search restraint based on the secondary memory mechanism (Figure4.6). The relationship between the immune system and the memory mechanisms, and the characteristics are described in the following sections.

The relationship between the immune systems and the memory mechanisms

In section 2.3.4, the two immune responses and the restraint system were defined as search methods. The correspond between the immune subsystems and the memory mechanisms are shown Figure4.4.

The principle behavior of AMIA

The search processing procedure takes a transition through a global search, a local search and a shift of the search scope in Figure2.5. However, in the introduced memory mecha-

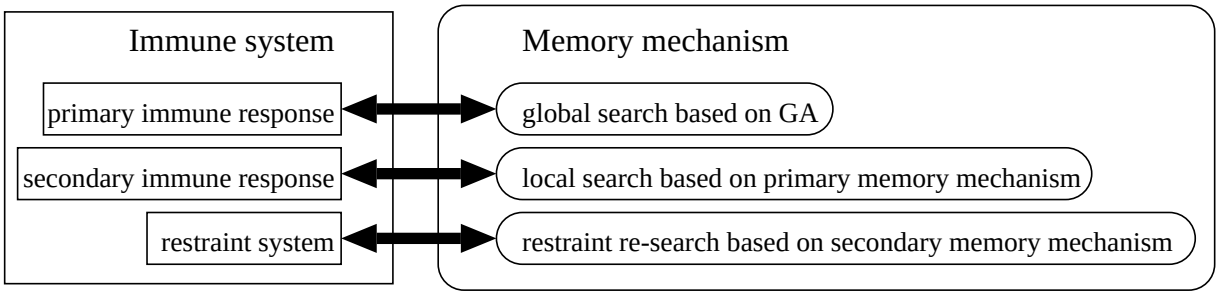


Figure 4.4: The immune system and the memory mechanisms.

nisms, the actual search is achieved based on GA, which is exploited the memory cells or the suppressor cells. Namely, the implemented memory mechanisms carry out (1st) GA (global search), (2nd) an antibody reproduction for a local search by the primary memory mechanism, (3rd) GA (local search), and (4th) an antibody reproduction to shift for the search scope by the secondary memory mechanism. Here, the transfer from GA to the memory mechanisms (i.e., from 1st to 2nd, and from 3rd to 4th) is carried out only when the convergence rate exceeds a pre-configured threshold.

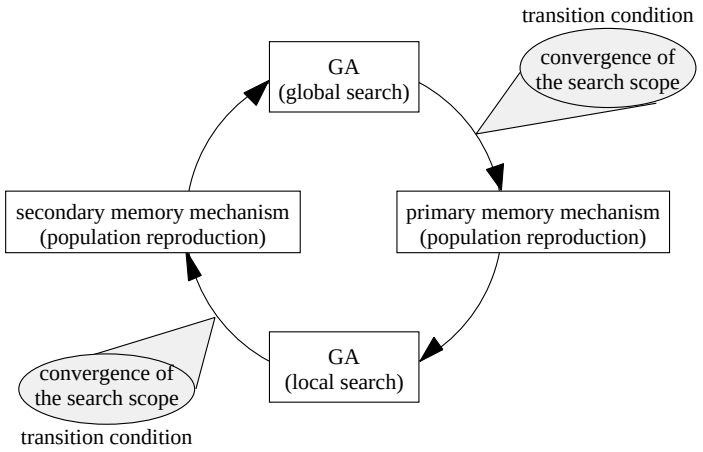


Figure 4.5: An illustration of the principle search behavior in the introduced memory mechanisms.

Figure4.6 shows the details algorithm of the memory mechanisms. In the determination processing for a local search in the Figure4.6, the processing is shifted to the primary

memory mechanism in the initial situation, and then a local search is carried out. When the processing was shifted to the same determination while a local search was running, the processing is shifted to the secondary memory mechanism, and then a global search is carried out. In this way, AMIA achieves the principle behaviors: the acquisition of superior solutions and the shift of the search scope, through the use of the primary and secondary memory mechanisms alternately.

The characteristics of the memory mechanisms

AMIA resolves the difficulty of the parameter adjustment in IA through establishment obtaining memory and restraining search individually by separation as two kinds of memory mechanisms. In addition, the primary memory mechanism to achieve a local search exploited the memorized solution in the act of first problem solving is introduced. The proposed AMIA with the two kinds of memory mechanisms can obtain multi optimal solutions through the principle behaviors:

- in the primary memory mechanism, an acquisition of a template as a partial superior solution has been achieved by the memory cell, and an enhancement of a local search has been exploited the memorized template, and
- in the secondary memory mechanism, an acquisition of superior solution has been achieved by the suppressor cells, a shift of the search scope and a restraint to search the memorized superior solution again are achieved.

According to these principles, AMIA is extended so as to get the easily parameter adjustment and the effective search ability.

Note that, the form and the content of the memorized solutions in AMIA is different between the memory cells and the suppressor cells, even IA memorizes the same formed and same content solution.

- **Memory cells:** The form is a partial solution (called template) of a superior one.
- **Suppressor cells:** The form is the same as a solution itself.

4.4.2 Primary memory mechanism

The following two functions are implemented to improve the search ability, in the primary memory mechanism.

refinement of memory:
This function abstracts a partial solution (i.e., schemata), which has a high quality, from the superior solutions.

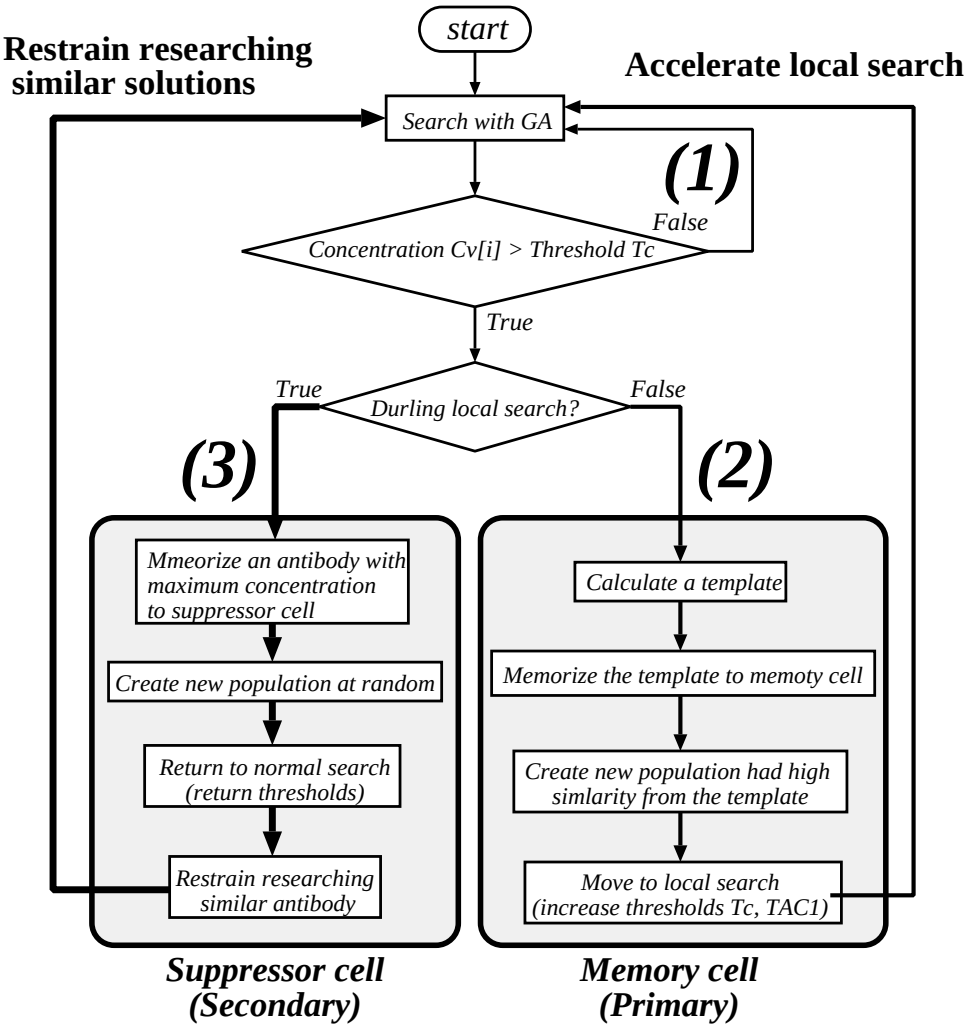


Figure 4.6: The memory mechanisms.

enhancement of a local search ability:

This function limits the search scope through generating solutions whose partial genes are same as the memorized partial solution. In this way, it is possible to enhance a local search ability, which is weak point of GA.

The algorithm based on the above-mentioned concepts is described in the followings. And the illustration is shown in Figure4.7. In this figure, we assume that each individual consists of two kinds of characteristics.

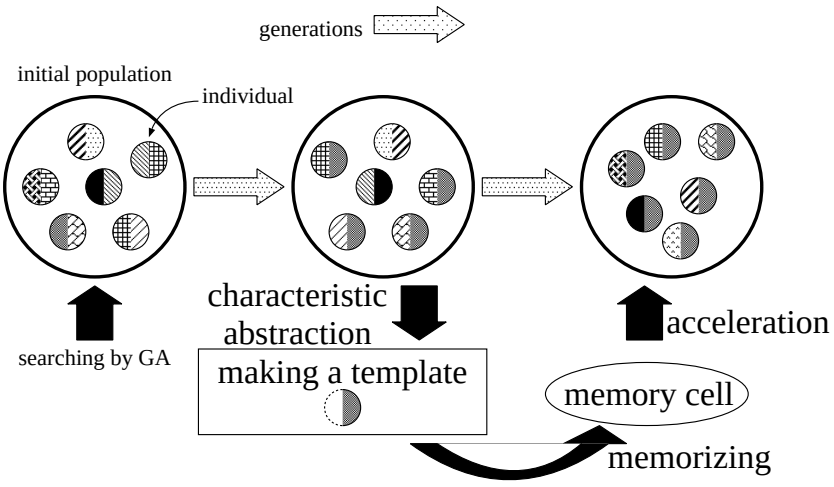


Figure 4.7: Primary memory mechanism.

[Step1. Abstract a template]
In a set of antibodies whose an affinity between an antibody and the maximum-concentration antibody exceeds a threshold $TAC1$, the common patterns are collected as a template.

[Step2. Memorize a template into memory cells]
When the number of memorized templates exceeds the upper limit, a template that has many common patterns against the new template is swapping to keep the diversity of the templates.

[Step3. Produce antibodies whose antibody similar to the template]
A set of solutions is produced so as to include the template. The lack of information as a solution is made at random.

[Step4. Shift to a local search]
The thresholds (TC and $TAC1$) are increased to search so as to become the high convergence rate using the population made in Step3.

4.4.3 Secondary memory mechanism

The objectives of the secondary memory mechanism are to acquire a candidate solution, shift the search scope and restrain to search after the local search, which is enhanced by the primary memory mechanism (Figure4.8).

[Step1. Memorize a antibody with maximum concentration into suppressor cells]
When the number of memorized antibodies exceeds the upper limit, a suppressor cell with the highest affinity is swapping.

[Step2. Produce population at random]
A shift of the search scope is carried out.

[Step3. Transit to a global search]
The thresholds, which are increased by the primary memory mechanism, are restored to transit to a general global search.

[Step4. Restrain to search the memorized antibodies]
An expectation whose solution is similar to the memorized antibody is decreased in order to restrain to search the memorized antibodies again (equation (3.14 and (3.15)).

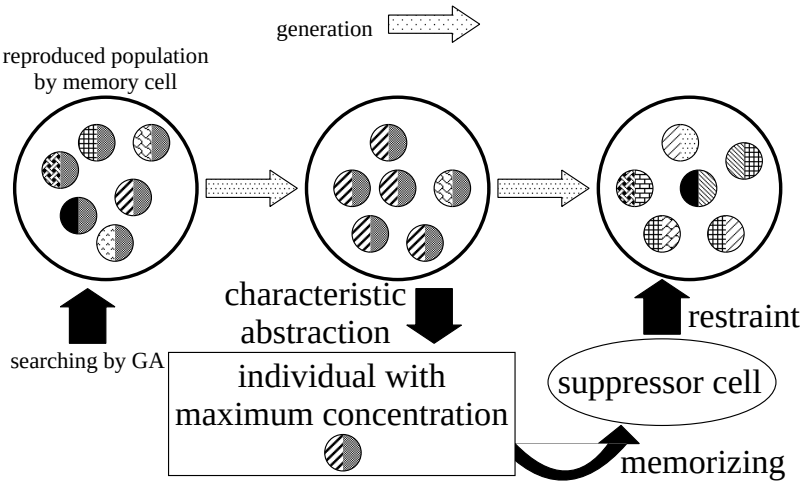


Figure 4.8: Secondary memory mechanism.

4.5 Conclusion

This chapter described about AMIA as a proposed method to obtain multi optimal solutions in the multimodal functions. The principle behaviors are

- a global search based on GA,
- a local search based on the primary memory mechanism, and
- a restraint to search the memorized one again based on the secondary memory mechanism.

The new points of AMIA are to resolve the parameter adjustment issue of IA, and to enhance the search ability. And, note that, the form and the content of the memorized solutions in AMIA are different between the memory cells and the suppressor cell, seven IA memorizes the same formed and same content solution.

- **Memory cells:** The form is a partial solution (called template) of a superior one.
- **Suppressor cells:** The form is the same as a solution itself.

In the next chapter, the computational experiments to investigate the efficiency of AMIA are described.

Chapter 5

Computational experiments

5.1 Introduction

In the previous chapter, AMIA with the two kinds of memory mechanisms as a new search method for the multimodal optimization problems were described. In order to evaluate the proposed method experimentally, this algorithm is investigated the efficiency through the computational experiments, which are applied to multimodal deceptive functions as follows.

target problem 1:
a deceptive problem in Traveling Salesman Problem (TSP) to confirm the basic performance.

target problem 2:
a bipolar deceptive function to investigate the efficiency to obtain multi optimal solutions.

In the followings sections, an overview of the problems are clarified at first, and then the results are reported.

5.2 Traveling Salesman Problem (TSP)

In order to confirm the basic performance and the behavior, AMIA is applied to a deceptive problem in TSP[Yamamura 1992] as one of the most typical combinatorial problems. The deceptive problems in TSP consists of the allocation of cities is a dual circle uniformly. In the deceptive problems, they have two characteristic superior solutions as shown in Figure5.1, and the optimal solution and the sub-optimal solution are swapped based on the ratio of the radii. Figure5.1 illustrates an example of 48-city TSP. The optimal solution

in the pattern (a) is composed of a traveling the cities on the one circle after the remaining cities on another circle travels (called as C-type). The optimal solution in the pattern (b) is composed of a traveling the cities on both circles alternatively (called as O-type).

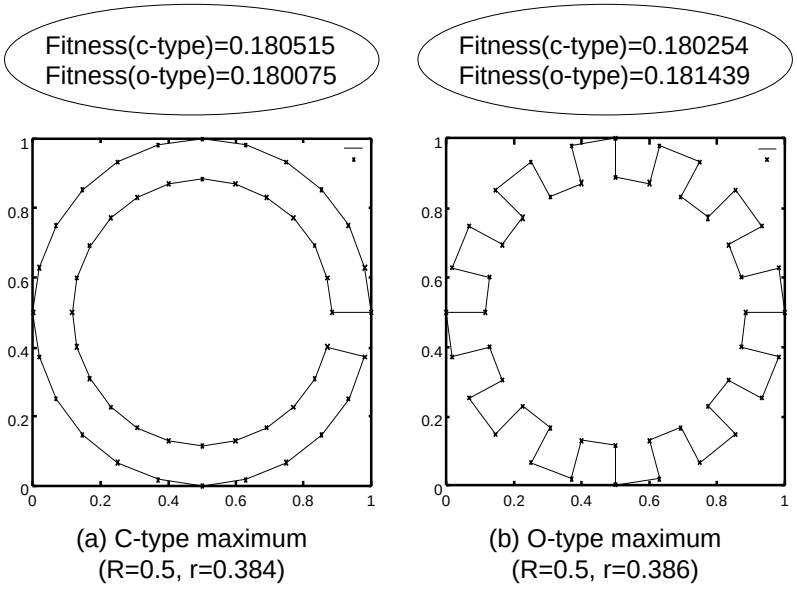


Figure 5.1: C-type and O-type.

Figure5.2 illustrates a conceptual behavior of AMIA on the deceptive problem. IA achieves the restraint only in the Figure, but AMIA can achieve the following things also because AMIA has the local search method.

- to search solutions effectively, which have higher fitness.
- to decrease the number of generations to take for problem solving.

5.2.1 Cording and genetic operators

For an application of immune algorithms to TSP, the solutions must be encoded, like GA. A concentration, which is one of the measurements in immune algorithms, is calculated based on an affinity. For the application to TSP, the affinity between two antibodies is defined as a common degree between the two tours as the strings. From the calculation's point of view, path representation, which has a structure that is easy to abstract common parts, is adopted as a cording. In the path representation, a specialized crossover operator

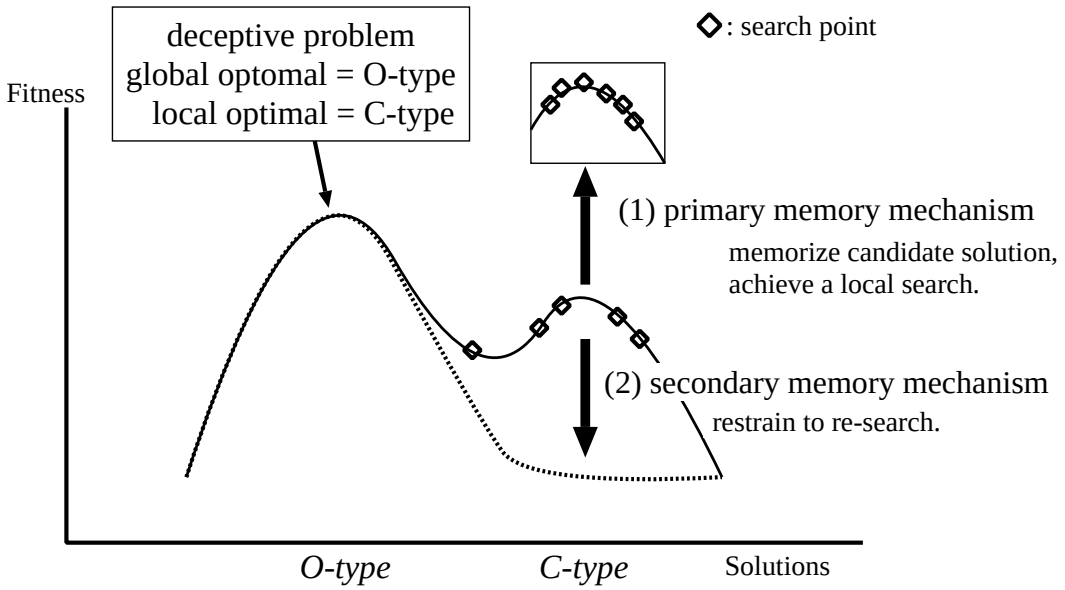


Figure 5.2: The conceptual behavior in a deceptive problem.

fails to produce any lethal genes is required. As a specialized operator, subtour exchange crossover[Yamamura 1992] corresponding to the pass representation method is adopted. In subtour exchange crossover, two subtours that 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. And mutation operator, which changes any two cities in the string, is adopted also.

Note that, the implemented subtour exchange crossover is arranged as follows.

Step1: decide a crossover point (subtour) between parent A and B at random.

Step2: replace two subtours when the subtours which have the same elements are looked up.

Figure5.3 shows an example of subtour exchange crossover. In the arranged crossover, there are many situations, which the decided subtours randomly don't much, and so, the part as a genetic operator cannot be achieved. In the experiments, an adjustment to become the number of execution to 20-30% is performed by multiple trials of Step1.

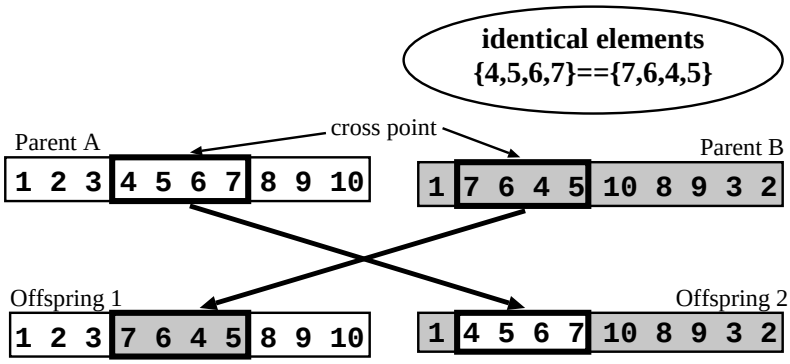


Figure 5.3: Subtour exchange crossover.

5.2.2 Fitness and affinity

The four measurements: fitness, affinity, concentration and expectation, must be designed to apply immune algorithms to the problem. The concentration and the expectation is calculated based on equation (3.12), (3.13), (3.14), (3.15) in section 3.3.4. The fitness value $fitness_i$ for TSP is formulated as an equation (5.1) using a reciprocal of the length of the solution i. The affinity $ay_{v,w}$ is formulated as an equation (5.2) using the number of the common cities between the two solution v, w.

[fitness]: an evaluation of solution i

$$fitness_i = 1/(Cost * Cost_W) \tag{5.1}$$

$Cost : length of the solution as a tour$
$Cost_W : weight for Cost$

[affinity]: a degree of common subtours between solutions v and w

$$ay_{v,\,w} = common\_path/m \tag{5.2}$$

$common\_path : the number of common subtours$
$m : the number of cities$

5.2.3 Design for experiments

In order to confirm the efficiency, two experiments are designed. In the first experiment, AMIA is applied to a deceptive problem to confirm to acquire the characteristics of solutions

at one problem solving. Specifically, the abstraction of the two superior solutions: C-type and O-type, is attempted. In the second experiment, the three methods: AMIA, IA and GA with subtour exchange crossover are applied to the same problem in the first experiment in order to compare their performances. Note that, for a comparison with GA implemented subtour exchange crossover, the number of cities is configured as 48, which is the same number in the experiments in the study[Yamamura 1992].

5.2.4 Experiment 1: design of simulation

In order to confirm the abstraction, AMIA solves on a situation without any initial memories. The definition of the problem is shown in Table5.1, and the parameter of AMIA is shown in Table5.2 In Figure5.2, the parenthetic value in TC is equal to $TC \times TC_{power}$. TC_{power} is a parameter for the modification of the threshold, which the modification is performed in the primary and secondary memory mechanisms.

Table 5.1: Problem definition

A type of optimal solution	C-type
The number of cities	48
A radius of outside circle	0.500
A radius of inside circle	0.384
Terminal genrations	100000

Table 5.2: Parameters of AMIA

Population size	200
Elite rate	0.05
Crossover rate	1.0
Mutation rate	0.01
The number of memory cells	5
The number of suppressor cells	5
$TC(thresholdformemorizing)$	0.5 (0.695)
$TAC1(thresholdforconentration)$	0.9
$TAC2(thresholdforrestraint)$	0.5
$MemoryT(thresholdformakingtemplate)$	0.9
TC_{power}	1.39

5.2.5 Experiment 1: results and considerations

A transition of maximum fitness per each generation, the memorized generations of memory cells ($MemN$) and supressor cells ($SupN$) are shown in Figure5.4. And the memory cells and the suppressor cells are shown in Figure5.5.

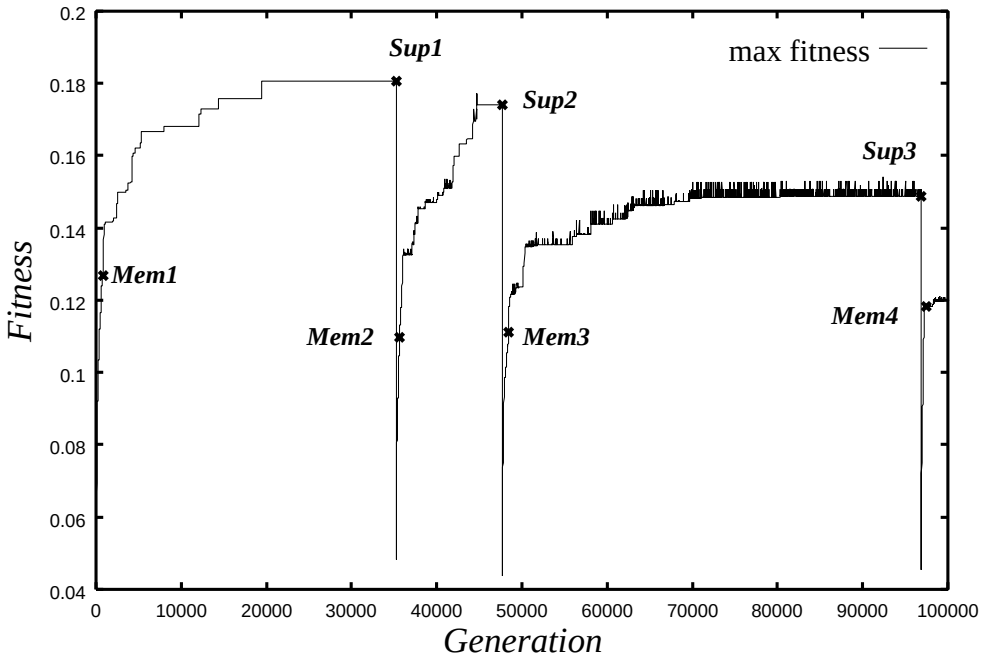


Figure 5.4: Transition of maximum fitness and memorized generations

In Figure5.4, the basic behavior, which acquires a template with common subtours into the memory cell and acquires superior solution into the suppressor cell after the local search, is confirmed. In addition, from the relations between the transition and memorysuppressor cell, the following two considerations are arose.

consideration 1:

The reproduction of the population based on the primary memory mechanism functions effectively as a local search. Because the maximum fitness rises stably after acquisitions of the memories such as $Mem1$ and $Mem2$.

consideration 2:

In the search stage after memorized suppressor cells such as $Sup1$, the search for $Sup1$ has been restrained by the secondary memory mechanism. And the suppressor

cells $Sup2, Sup3$ have lower fitness than $Sup1$. That is considered that the shift of the search scope is achieving because the search scope will converge to another point with lower fitness than $Sup1$.

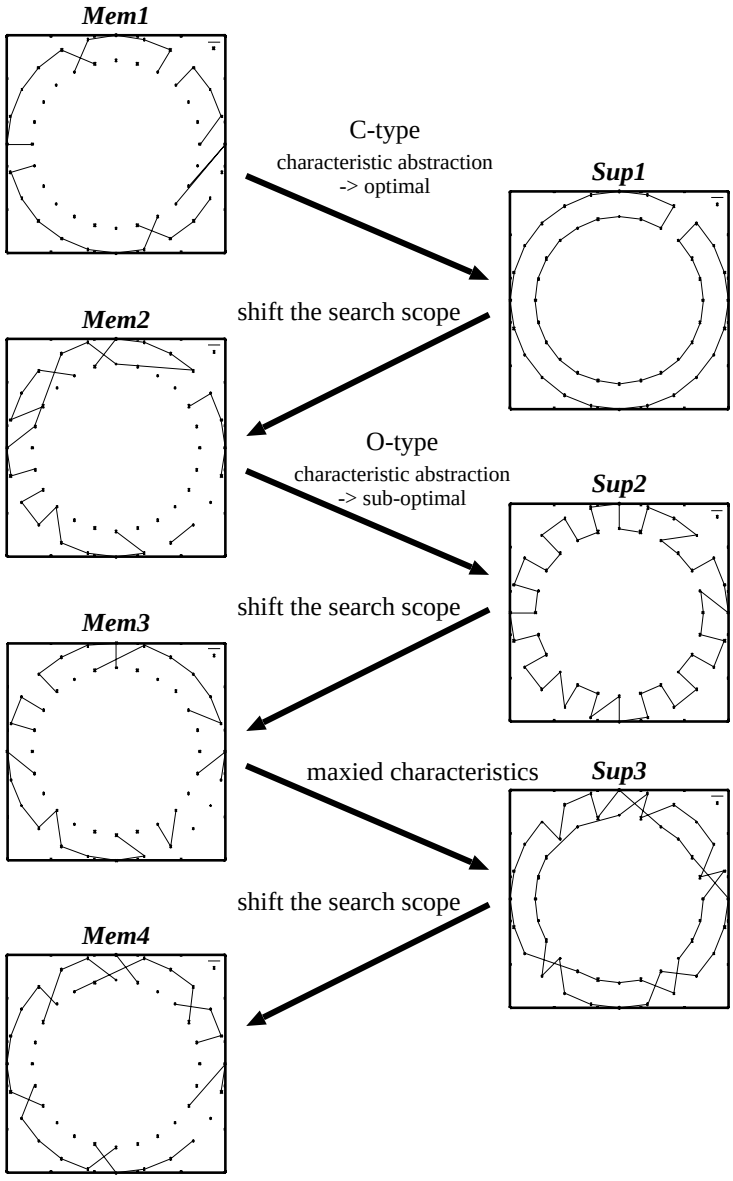


Figure 5.5: Memory cells and suppressor cells

Next, the memorized contents are investigated in order to validate the above considerations, from the acquired memory cells and suppressor cells are shown in Figure5.5.

[$Mem1$]
This memory has a characteristic similar to C-type which is the optimal solution in this problem.

[$Sup1$]
The suppressor cell acquired the optimal solution with C-type itself after the local search based on $Mem1$. In the search after this, the search for the solution that is similar to C-type has been restrained.

[$Mem2$]
In this memory, a characteristic is similar to O-type begins to arise. This means that the current situation is considered as the stage for shift from C-type to O-type.

[$Sup2$]
The suppressor cell acquired the solution is similar to sub-optimal O-type after the local search based on $Mem2$. In the search after this, the search for the solution that is similar to O-type is restrained.

By the way, a decrease of the number of generations to obtain suppressor cell is identified. For obtaining the optimal solution, it takes about 35000 generations. And for obtaining the sub-optimal solution, it takes about 12000 generations that the generation is lower than the first optimal's one. This reason is considered that there are two pressures to decide the search scope: one pressure is enhancing to search a solution with high expectation by a selection operator, and the other pressure is booting off the searched scope by the restraint.

[**after** $Sup2$]
As a result of the restraint for optimal and sub-optimal solutions, $Mem3$, $Mem4$ and $Sup3$ are not similar to both, or they have mixed characteristics.

When AMIA obtains $Sup2$, we observed a decrease of the number of generations. But in case of $Sup3$, the number of generations increases. In the search after obtained $Sup2$, the search for the solution that is similar to O-type and C-type is restrained. At this stage, the fitness landscape that consists of low peaks only is considered. Namely, in this situation, it is considered that the search scope will be easily shifting even though AMIA wants to converge one peak.

Such behavior continues until $Sup1$ or $Sup2$ is overwritten. And after the overwritten is occurring, the problem will have the fitness landscape with high fitness again.

According to the acquisition of characteristics by the primary memory mechanism and the acquisition of superior solution by the secondary memory mechanism, this behavior can be gained.

5.2.6 Experiment 2: design of simulation

In the second experiment, the three methods: AMIA, IA and GA, are applied to the problem defined as Table5.1 in order to compare their performances from the transition of fitness per generation. The GA implements the subtour exchange crossover and mutation described in section5.2.1, and the fitness function is same as an equation (5.1). The parameters (population size, elite rate, crossover rate and mutation rate) are configured as the same in Table5.2.

5.2.7 Experiment 2: results and considerations

The transitions of maximum fitness are shown in Figure5.6. By the way, there are some cases that IA cannot obtain any memory. So, an obtained situation is written as 'memory', and other situation without memory as 'no memory' in the Figure.

The transitions in the Figure5.6 are considered by a comparison between AMIA and other method.

consideration 3: AMIA and IA (memory)

IA (memory) obtained the memory at about 35200 generations. Following the search scope shifts to other scope because the fitness of antibodies is similar to the memory decreases immediately. For that purpose, IA cannot continue to search the solution with C-type characteristic, even though the obtained memory doesn't reach to the optimal solution.

On the other hand, AMIA continues to search the solution with C-type characteristic because this method carries out the local search based on $Mem1$ as mentioned in experiment 1. Namely, AMIA has a high possibility to search high fitness solutions.

consideration 4: AMIA and IA (no memory)

The behavior of IA (no memory) is similar to the monotonous upward-sloping tendency of GA. This means, to obtain memories is important for the search behavior. However, the adjustment so as to satisfy two conditions: acquires superior solutions and shifts the search scope, is the difficult issue in fact.

In AMIA, each threshold can be appropriate configured easily because the acquisition of memory and the restraint of search are implemented as independent mechanisms. For this purpose, AMIA can obtain memory stably.

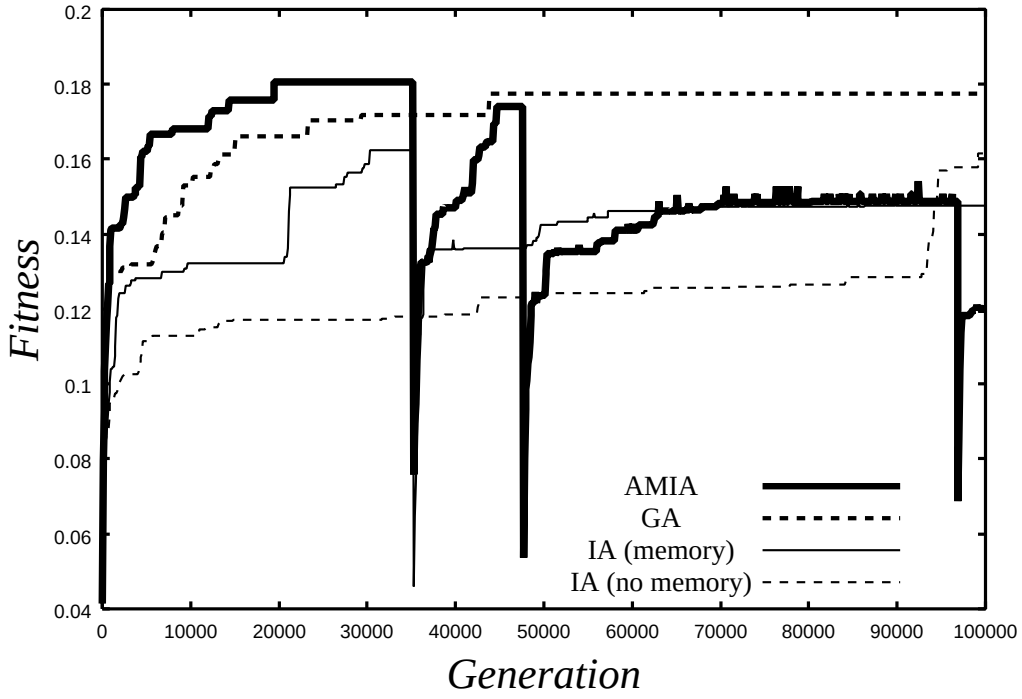


Figure 5.6: Comparison of the transitions of maximum fitness.

consideration 5: AMIA and GA

The behavior of GA is almost same as AMIA up until about 850 generations. But AMIA continues to increase the maximum fitness after memory obtained. So, the local search based on the memory is efficient approach in the optimization's point of view.

5.2.8 Summary of the experiments

In the experiment 1 and 2, AMIA was applied to a deceptive problem in TSP in order to confirm the basic performance and the principle behavior. The advantages are the followings.

1. **validity of a local search**

   It was demonstrated that a local search exploited the obtained memory in the act of searching was validity from the optimization's point of view.

2. **stability of obtaining memory**

The adjustment to obtain appropriate memory got easily through establishment obtaining memory and restraining re-search individually.

3. **possibility to search high fitness solutions**
   It is possible to search high fitness solutions because AMIA can continue to search the same scope after memory obtained.

In the section 5.3, the efficiency of AMIA is investigated about obtaining multi optimal solutions.

5.3 Bipolar Deceptive Function (BDF)

Bipolar Deceptive Function (BDF) is a kind of deceptive function provided by Goldberg to investigate the search ability in the multimodal functions, which the function has many global and local optimal solutions[Goldberg 1992]. In the computational experiments, 30-bit Bipolar Deceptive Function (called 30-bit BDF) made by binding five 6-bit BDF is used for the target problem.

The 6-bit BDF (Figure5.7's top) is a multimodal function made by using lower order Walsh coefficient, and has 2 global and 20 local optimal solutions. The 30-bit BDF, which is made by binding five 6-bit BDF, has $2^5 = 32$ global and $(20+2)^5 = about5million$ local optimal solution. The parameters used in equation (5.3) are configured as the same values in [Goldberg 1992] ($2l = 6, w_0 = 0.4350960, w_2 = -0.020048, w_4 = 0.060024$). By the way, in equation (5.3), when λ is set to $2l$, the equation becomes the fitness function $f(u)$ in the 6-bit BDF.

$$f(u, \lambda, 2l) = \sum_{i=0}^{\lambda} w_i \psi'_i(u, \lambda) \tag{5.3}$$

- u: unitation
- λ: order of the function
- $2l$: the length of the function (6-bit)
- w_i: Walsh coefficient
- ψ'_i: Walsh function created from Walsh coefficient w_i

$$\psi'_i(u, \lambda) = \sum_{j=0}^{i} (-1)^j \left(\begin{array}{c} u \\ j \end{array} \right) \left(\begin{array}{c} \lambda - u \\ i - j \end{array} \right) \tag{5.4}$$

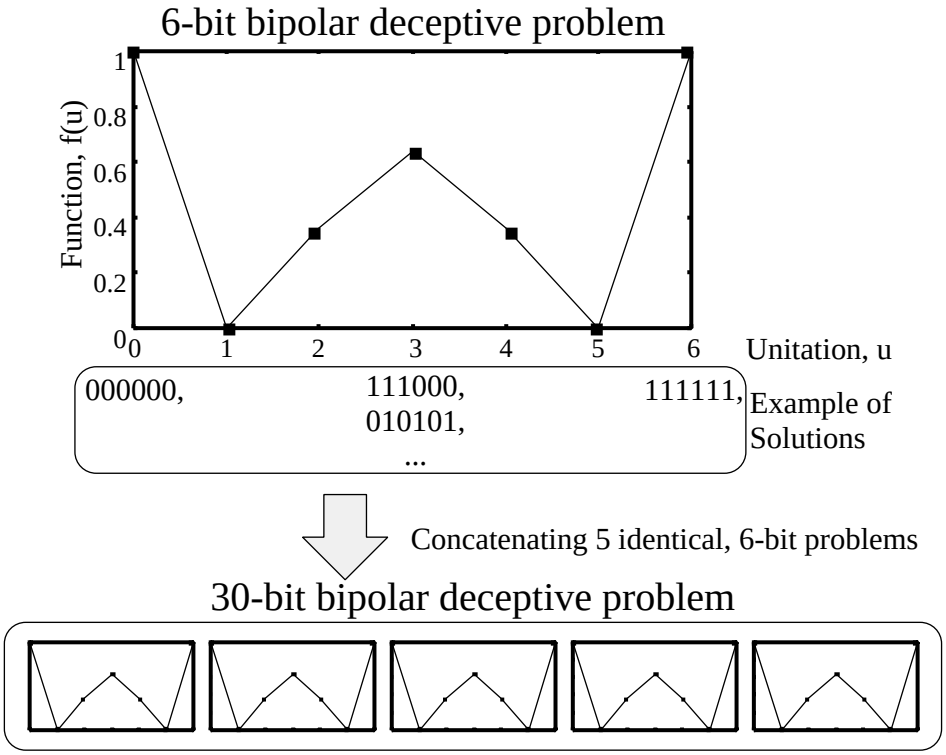


Figure 5.7: The 30-bit Bipolar Deceptive Function.

5.3.1 Cording and genetic operators

In order to apply immune algorithms to the 30-bit BDF, a binary string as a cording and one-point crossover and mutation as genetic operators are adopted.

$$solution_i = \sum_{j=0}^{N} sub\_solution_{i_j} \tag{5.5}$$

- $sub\_solution_{i_j}$ = 6-bit binary string
- N: the binding number of 6-bit BDF

5.3.2 Fitness and affinity

The fitness is calculated as a sum of the function value in the 6-bit BDF (equation (5.6)). The affinity is adopted a hamming distance (equation (5.7)). The concentration and the expectation is calculated by equation (3.12), (3.13), (3.14), (3.15) in section 3.3.4.

[fitness]: an evaluation of solution i

$$fitness_i = \sum_{j=0}^{N} f(u_{i_j}) \tag{5.6}$$

- u_{i_j}: unitation in $sub\_solution_{i_j}$.
- $f(u_{i_j})$: equation (5.3) with $\lambda = 2l$.

[affinity]: a degree of common subtours between solutions v and w

$$ay_{v,\,w} = 1/(1 + H(v, w)) \tag{5.7}$$

- $H(v, w)$: a hamming distance between solution v and w.

5.3.3 Experiment 3: design of simulation

In order to investigate the search ability in the multimodal functions, IA and AMIA are applied to the 30-bit BDF. The parameters of AMIA are configured in Table5.3.

Table 5.3: The parameters of AMIA.

Population size	100
Elite rate	0.05
Crossover rate	0.9
Mutation rate	0.01
The number of memory cells	1000
The number of suppressor cells	1000
$TC(thresholdformemorizing)$	0.25 (0.375)
$TAC1(thresholdforconcentration)$	0.6
$TAC2(thresholdforrestraint)$	0.96
$MemoryT(thresholdformakingtemplate)$	0.6
TC_{power}	1.5
Terminal generations	300000

5.3.4 Experiment 3: results and considerations

The average fitness in the global optimal solutions is lower than tthe fitness in the local optimal solutions in the 30-BDF. Namely, the 30-BDF is a deceptive function that is easy to

converge the search scope into the local optimal. Although, IA and AMIA obtained all 32 global optimal solutions, the average generations to take up for finding all global optimal solutions were about $140,000$ generations for AMIA and about $210,000$ generations for IA. In addition, the average number of the obtained solutions were about 200 solutions for AMIA and about 40 solutions for IA (Table5.4), and they obtained the solutions from higher fitness solutions one by one.

Table 5.4: The average generations to take up for finding all global optimal solutions and the average number of the obtained solutions

	AMIA	IA
The average generations	$140,000$	$210,000$
The average numbers	220	40

From these results. it is clear that AMIA is a superior algorithm as a search method to obtain multi optimal solutions in the multimodal functions.

Next, the considerations about the transition of maximum fitness in IA and AMIA and the fitness of obtained suppressor cell (Sup) are described.

consideration 6:

AMIA obtained second optimal solutions (the fitness is 4.640576) occasionally before obtained all global optimal solutions (the fitness is 5.0). Especially, AMIA obtained sub-optimal solutions in the generation 1400, 2300 and 3800, even though the population had a global optimal solution. On the other hand, IA obtained the global optimal solution as a suppressor cell when the population had the global optimal solution in the act of searching in general. However, AMIA found all global optimal solutions with lower generations than IA, even if the generations included a generation for sub-optimal solutions. Consequently, the behavior of AMIA has a possibility to converge to sub-optimal solutions occasionally. But the generation to take up for obtaining multi optimal solutions decreases, and the number of the obtained superior solutions is about five times of IA. Namely, AMIA is efficient algorithm as a search method in the multimodal functions.

5.3.5 Summary of experiment 3

In the experiment 3, IA and AMIA was applied to the 30-bit BDF in order to evaluate the search ability to obtain multi optimal solutions in the multimodal functions. The advantages of AMIA are the followings.

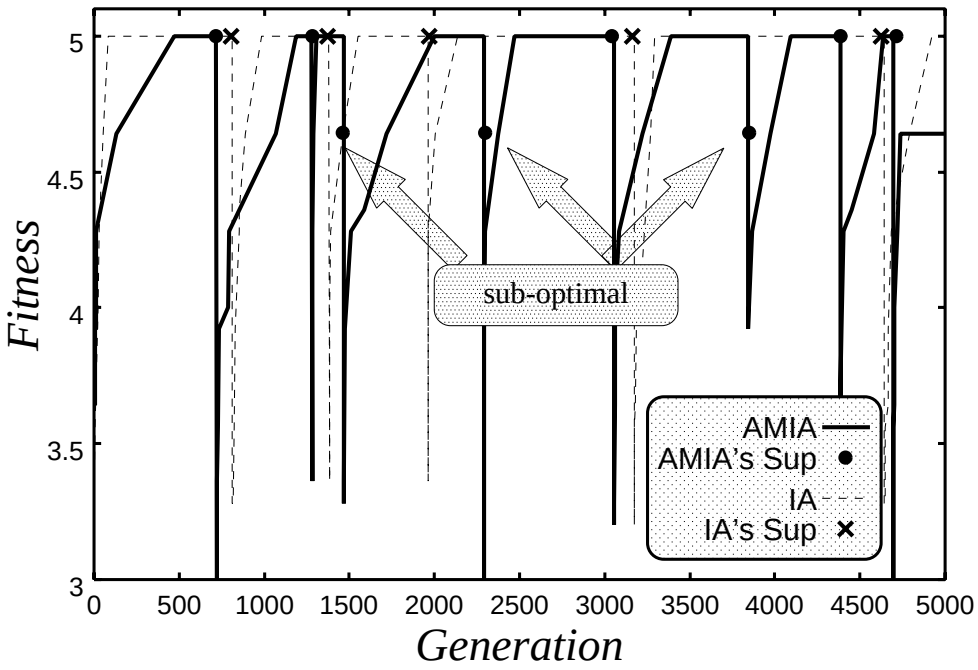


Figure 5.8: The fitness of the obtained memories and the transition of the fitness.

1. **improvement of the search ability**
   AMIA can find all optimal solutions with lower generations than IA.

2. **enhancement to obtain superior solutions**
   AMIA can obtain more superior solutions than IA.

In addition, the behavior of AMIA has a possibility to converge to sub-optimal solutions occasionally. But the generation to take up for obtaining multi optimal solutions decrease, and the number of the obtained superior solutions is about five times of IA. Namely, AMIA is efficient algorithm as a search method in the multimodal functions.

5.4 Conclusion

In order to investigate the principle behavior and the performances, AMIA was applied to a deceptive problem in TSP and Bipolar Deceptive Function. From these results:

1. **validity of a local search,**

2. **stability of obtaining memory,**
3. **possibility to search high fitness solutions,**
4. **improvement of the search ability, and**
5. **enhancement to obtain superior solutions**

were validated. As the results of the experiments, it confirmed that the search efficiency improved by a memory mechanism's working effectively.

Chapter 6

Application a Biological Immune Sytem to Multi-agent System

6.1 Introduction

The immune system in described in chapter2 is a model to focus on the primary and secondary immune responses and the restraint system based on immunological memory. In this model, (1) the immune responses were carried out on that condition, which was recognized the invaded antigen already, and (2) the construction of the antibodies was carried out by antibody reproduction cell. However, the originally behavior of the biological immune system is

"to recognize the antigen as a **nonself**, which the antigen invades into the living body (**self**), and to eliminate the nonself".

This means, it is important how to distinguish the self or the nonself, and major histocompatibility complex (**MHC**) is used to distinguish them when the nonself invades the self. In addition, the construction of the antibodies is achieved through the interactions between the immune cells such as macrophages, T cells and B cells. The system, which recognizes the nonself and constructs the antibodies through the interactions, is called **immune network** [Janeway 1997]. Furthermore, a framework, with the two 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.

This chapter attempts an application to multi-agent system (MAS) to verify the further potentialities of the methods based on the biological immune system. The main objective is to construct two kinds of immune based optimization algorithms for the division-of-labor problems, which attach importance to how to allocate the work-domains for agents. As the methods to solve the division-of-labor problems, two kinds of methods are constructed. The

first method is a competitive algorithm, which have plural immune agents are competes each other according to the models of MHC and immune network. The second method is a co-evolutionary algorithm, which have two kinds of agents: immune agents and antigen agents that they evolve so as to prevent other's evolution according to the model of cell-cooperation. In this chapter, the issues of the target problem are clarified at first, and the models for the first algorithm (compettitive algorithm) are described.

6.2 Muti-agent system and division-of-labor problem

MAS, which is a study in the field of distributed artificial intelligence. MAS is an information processing technique in which autonomous agents solve problems by interactions among the agents [Russell 1995], and can be expected that has robustness, adaptability and stability [Ishida 1996]. In order to achieve such characteristics, elucidating the following things is required.

1. how to allocate work-domains to multi-agent?
2. how organization is tough for environment changes and/or any troubles?
3. how to communicate between agents efficiency?

The key subject of division-of-labor problems is to focus on the issues of distribution of work for agents, and the definitions are described as follows.

division-of-labor problems optimization:

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). The objective of the division-of-labor problems optimization is to find a solution, which is satisfied the following three conditions.

(a) **equaling work assignment**: In the problem domain PD, each work domain WD_i for agent$_i$ must be divided so as to have equal work assignment.
(b) **optimizing a work-cost individually**: In the divided work domain WD_i, the cost for the work of agent$_i$ must be optimized (minimized).
(c) **optimizing all amount of work-costs**: All amount of the work costs must be optimized (minimized) with satisfying (a) and (b).

Basically, the division-of-labor problems optimization is a subject, which requires minimizing a total cost in a parallel distributed processing. As the applications, it is considered as scheduling problem, job scheduling in a networked parallel computers[Coffman 1976], TSP with some constraints[1] , vehicle routing problem (VRP) and the VRP with time constraints, and so on.

[1] http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/

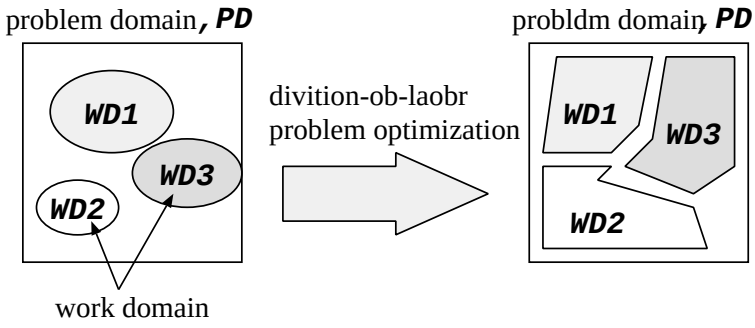


Figure 6.1: Division-of-labor problems optimization.

In the following sections, an overview of MHC and immune network, and the models based on the functions are described.

6.3 An immune system as an optimization method for division-of-labor problems

In order to construct an immune optimization system for the division-of-labor problems, we developed the models of the two immune functions: (1) MHC to recognize antigen and (2) immune network to produce appropriate antibodies through the interactions between the immune cells, in the first instance. Our objective models are shown in Figure6.2 and the details are described below in italics.

6.3.1 Antigens and Antibodies

In order to construct the biological immune system as an adaptive model in MAS, the antigens and the antibodies are defined as follows.

- 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 a problem solver, which is adapted to a framework of a biological immune system that performs self-preservation by searching specific antibodies that react against unknown vast antigens.

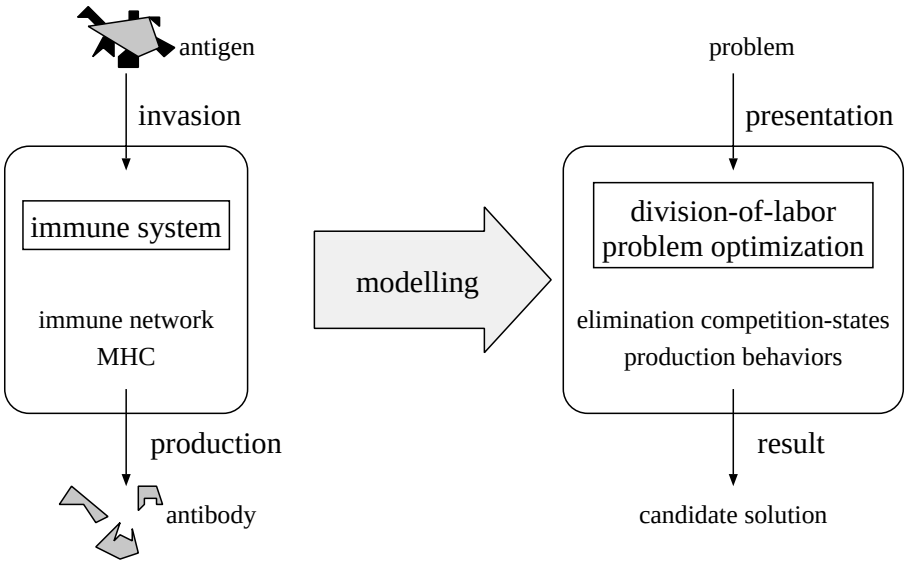


Figure 6.2: An immune system and an optimization method for division-of-labor problems

And then, the model exploited MHC to recognize the environment and the other agent, and the behavior of production system based on an immune network to adapt to the environment the optimization are described.

6.3.2 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 (Figure6.3).

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, invades 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 Figure6.3. The elimination of nonself means removal of the nonself system from the self-system, expressed in the lower part of Figure6.3. 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. So, in this paper, the elimination of the invaded antigen is defined as follows, and we use the model as a problem solver.

- MHC = a system to eliminate the competition states in the work domain for agent

In this way, one of the methods for division-of-labor problem optimization is achieving through an elimination of the competition-states. As a method to obtain good divided solutions, an immune network has been used.

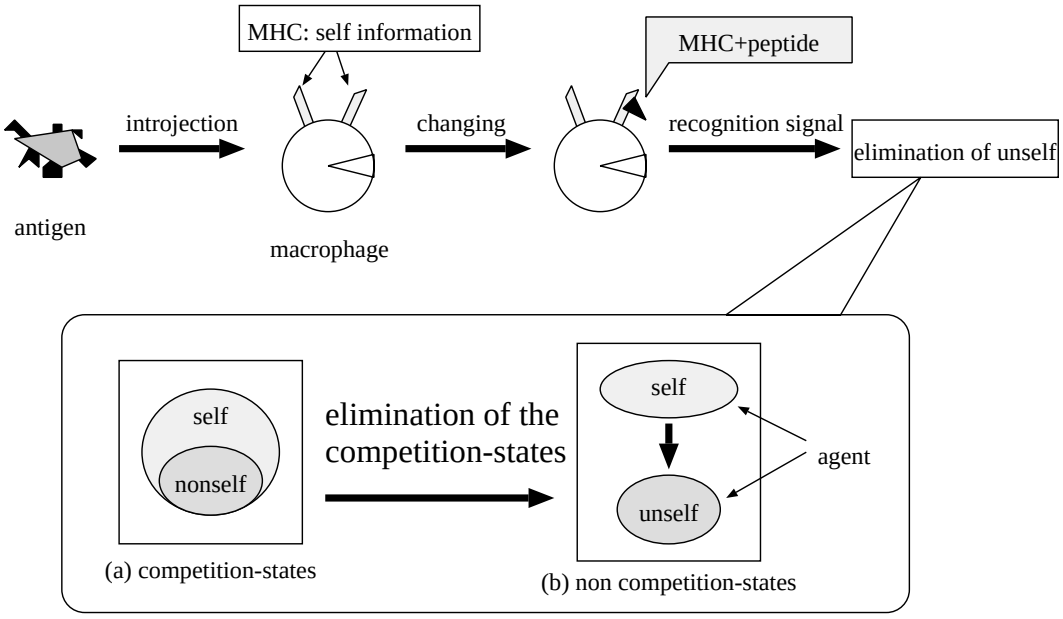


Figure 6.3: Elimination of competition-states is based on the elimination of nonself using MHC.

6.3.3 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 (see, left side of Figure6.4). The one piece of information transmitted through the network is MHC. The procedures of the immune system producing a specific antibody are described below.

Macrophage:

One of the information transmitted on the network is MHC, which has information about the invaded antigen and also the role as a signal that denotes an invasion of the antigen. Macrophage recognizes antigens and transmits MHC (with the information about the antigens) to T cell.

T cell:

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.

B cell:

The activated B cell makes a memory related the antigen, and produces specific antibodies.

Antibody:

The antibodies thus produced begin to eliminate the invaded antigens.

To apply such procedures as a production model of behavior of agents, the following procedures were constructed (see, right side of Figure6.4).

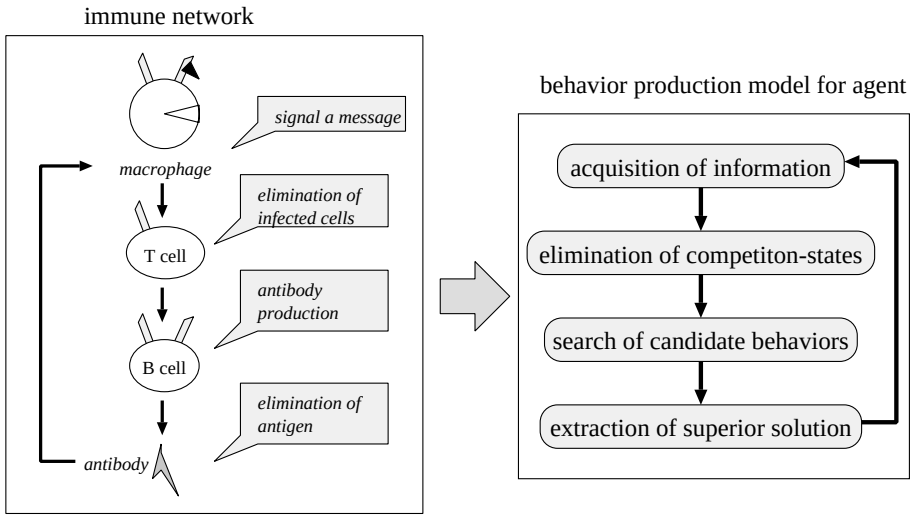


Figure 6.4: Production of behaviors by immune network.

Acquisition of information:

The agent acquires the information about the environment and internal-state in the agent.

Elimination of competition-states:
If the states of competition exist in the internal-state, the agent eliminates these states of competition.

Search of candidate behaviors:
The agent searches for a set of candidate behaviors in order to increase the fitness of internal-state in the environment.

Extraction of superior solution:
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. Our algorithm solves the division-of-labor problems through the elimination of competition by MHC and production of behaviors by Immune Network.

6.4 Conclusion

In this chapter, an overviews of MHC and immune network were described, and the model as a division-of-labor problems optimization was made. By exploiting the model, we can make a system as follows.

1. elimination of competition-states among agents by MHC, and
2. production of behaviors by immune network.

Chapter 7

An Immune Distributed Competitive Problem Solver

7.1 Introduction

The immune system described in chapter6 was a model that recognizes and distinguishes self and nonself by MHC, produces appropriate antibodies by immune network, and eliminates the invaded antigen. This chapter describes an Immune Distributed Competitive Problem Solver based on the models using MHC and immune network.

7.2 An immune optimization for division-of-labor problems

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 a part of 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 such competition-states this algorithm uses MHC model. and uses immune network model to search good state solutions exploited by the state without any competition. The proposed algorithm obtains superior divided solutions by using their functions.

7.2.1 An Immune Distributed Competitive Problem Solver

The details of our method are described below (Figure7.1).

[Step1. Definition of antigen and MHC]
A problem area where agents exist is defined as antigen and the internal state (for example, a/partial solution or behavior against the problem) 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.).

[Step4. Competition processing]
If any state 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 has been consisted of 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 (in other words, the solution is set as new solution and the other solutions are deleted). 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.

7.2.2 Competition processing

The competition processing has been executed at the fourth Step in the algorithm described in previous section. 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 Table7.1. Figure7.2 illustrates a situation (equation 7.1), which a nonself agent B invades to the work domain A_{MHC} of a self agent A, is void of competition-states based on the following two comparisons.

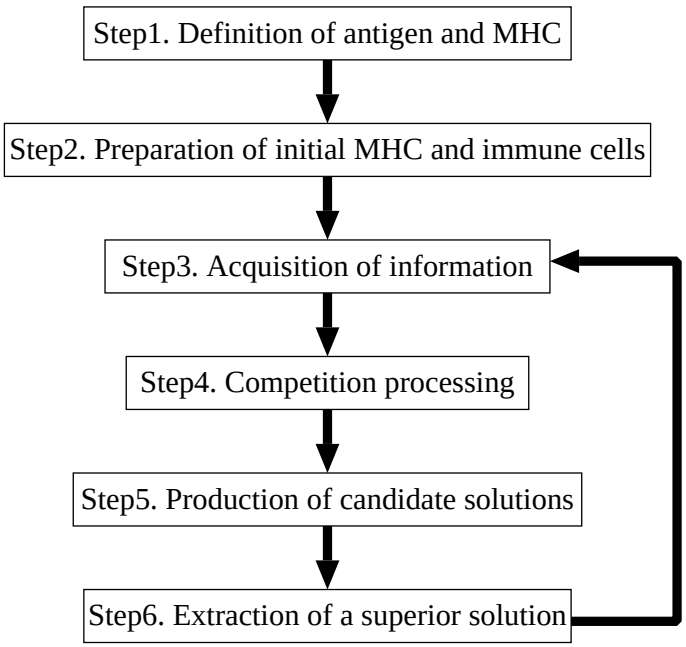


Figure 7.1: The algorithm of Immune Distributed Competitive Problem Solver

$$A_{MHC} \cap B_{MHC} \neq \emptyset \tag{7.1}$$

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

7.2.3 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 as follows (also, see Figure7.3).

Table 7.1: 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	$fitness(A)$	$fitness(B)$
fitness, after	$fitness(A')$	$fitness(B')$

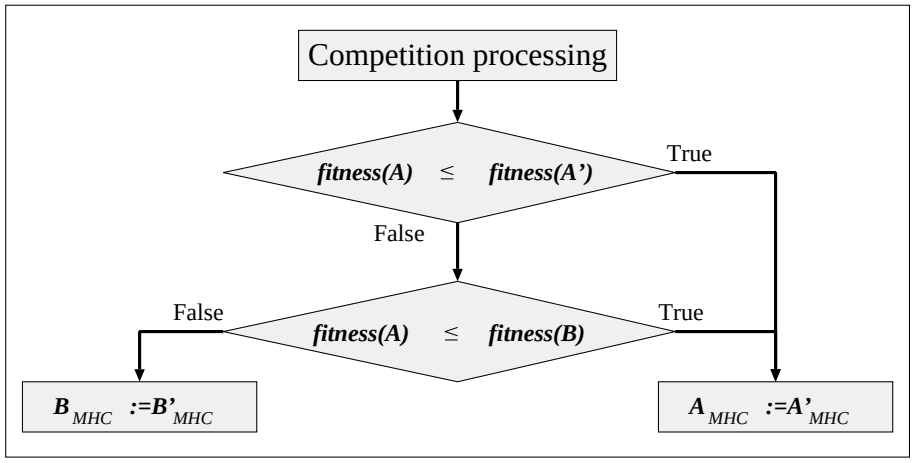


Figure 7.2: An example of competition processing.

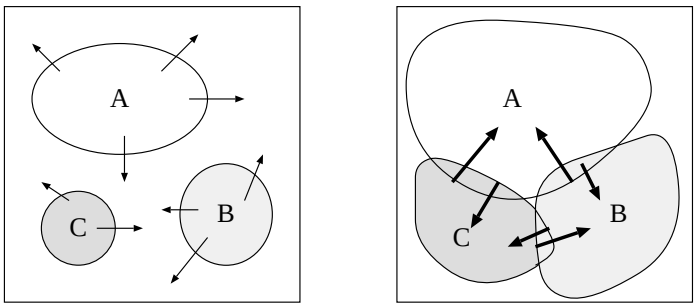
1. (Expansion of the work domain) Each agent adapts in the direction of raising the level of fitness on the inside of itself through production utilizing immune network.

2. (Reduction of the work domain) Each agent adapts in the direction of elimination of inefficient region in each work domain through elimination of competition.

According to these interactions, our method is capable of solving the division-of-labor problems efficiently.

7.3 Conclusion

In this chapter, an Immune Distributed Competitive Problem Solver based on the models using a competition processing based on MHC and a behavior production based on immune



(a) Expansion of the work domain (b) Reduction of the work domain

Figure 7.3: Basic concept of the proposed method.

network was described. The proposed algorithm has antithetic adaptations:

1. (Expansion of the work domain) Each agent adapts in the direction of raising the level of fitness on the inside of itself through production utilizing immune network, and

2. (Reduction of the work domain) Each agent adapts in the direction of elimination of inefficient region in each work domain through elimination of competition.

According to these interactions, the proposed method is capable of solving the division-of-labor problems efficiently.

In the next chapter, the computational experiments to investigate the efficiency of the proposed method are described.

Chapter 8

Computational experiments for Immune Distributed Competitive Problem Solver

8.1 Introduction

In the chapters from 6 and 7, the models using MHC and immune network were constructed, and an Immune Distributed Competitive Problem Solver based on the models were described.

In this chapter, the basic performances are investigated to evaluate the proposed method through an application to nth-agent's Traveling Salesman Problem as a typical case problem in the MAS. At first, the definition and the characteristics of the target problem are clarified.

8.2 Nth agent's Traveling Salesman Problem (n-TSP)

8.2.1 TSP and n-TSP

As a typical case problem of the division-of-labor, this chapter deals with the n-th agent's Traveling Salesman Problem (n-TSP) to investigate the adaptability of our model. Traveling salesman problem (TSP) is one of the most typical combinatorial optimization problems. The TSP's objective is to find a minimal roundtrip route visiting each node exactly once. TSP is investigated by many kinds of method because TSP has a complexity classified in NP-hard and is of wide applications. But, a more powerful system, which solves a problem through the interactions between agents, is required to deal with a huge complicated problem.

And so, n-TSP (Figure8.1), which is extended the definition of the number of salesman from singular to plural number, is set as a standard problem to evaluate the performances in MAS[Nakamura 1994]. The objective is to find the minimal tour by division among salesmen

Note that two conditions have been set.

condition 1: the number of salesmen is fixed while the system runs to solve.

condition 2: all salesmen use the same city as the starting point.

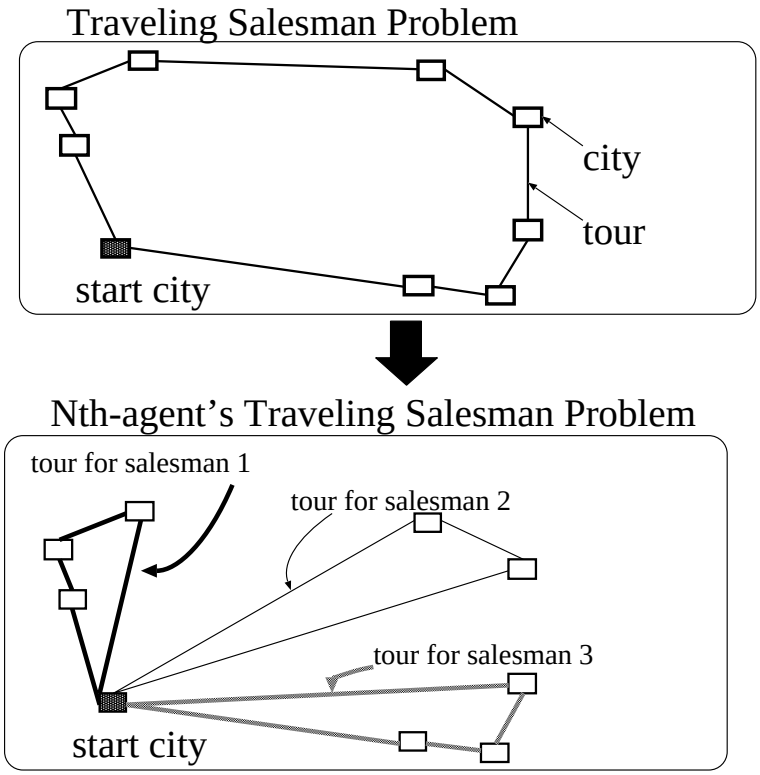


Figure 8.1: Notion of n-TSP.

8.2.2 Division-of-labor and optimization

n-TSP has the following two aspects by nature.

(1) a distribution for each salesman, and

(2) a search for the shortest tour in each distribution.

These two aspects have a difficult characteristic to solve them respectively because the aspects are linked with each other. This means, a solver is required to have any functions to resolve these aspects at a time.

8.2.3 The behaviors of the Immune Distributed Competitive Problem Solver in n-TSP

The example behaviors of the proposed method are shown in Figure8.2 and Figure8.3.

The immune network

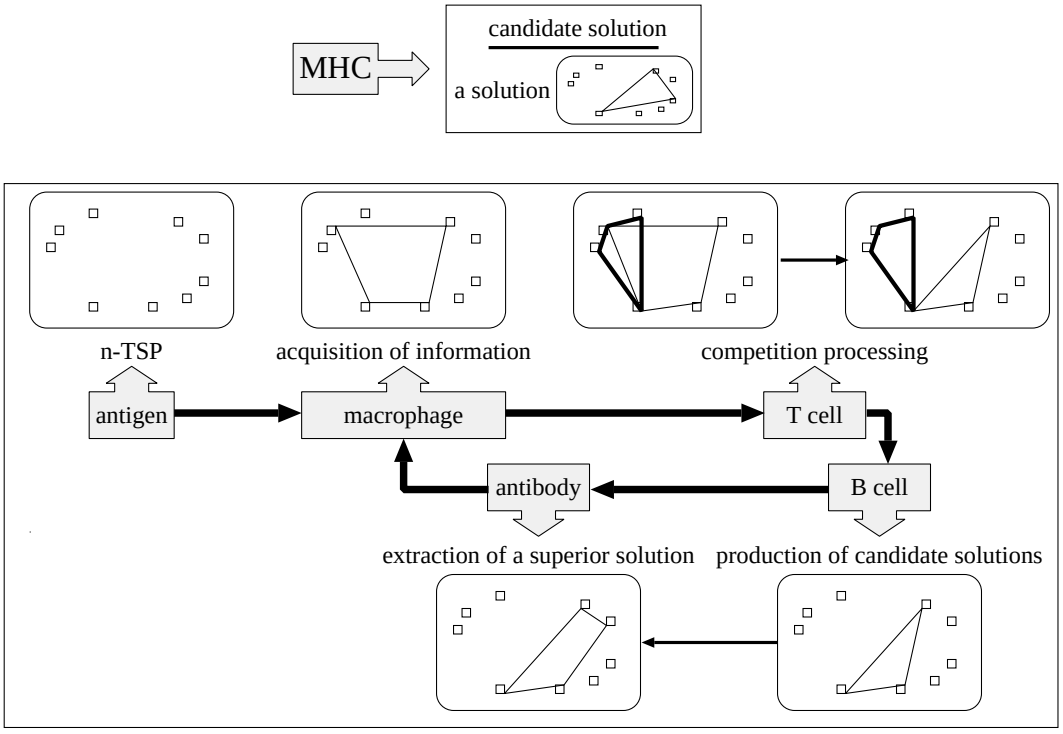


Figure 8.2: An example behavior in n-TSP.

In Figure8.2, the information of antigen and the MHC are defined as follows.

- information of antigen = the definitions for n-TSP (the number of cities, the arrangement of cities and the number of salesmen).

- MHC = a solution in n-TSP.

The MHC defined in the above-mentioned is transmitted on the network, and the MHC is reproduced to an appropriate solution. And the processings on the immune network are running out as follows.

1. Macrophage recognizes the antigen and the MHC, and binds a solution with information about n-TSP as a MHC. And then, macrophage transmits the MHC to T cell, which includes the competition-states in the solution.

2. T cell uses the competition processing to modify so as to travel the overlapped city by one salesman when the MHC has any competition-states.

3. B cell produces N number of candidate (similar) solutions that are approximated to the MHC (i.e., exploited search based on the MHC).

4. Finally, the immune agent extracts superior solution in the candidate behaviors, and sets the solution as a candidate solution in the next step.

The competition processing

Figure8.3 shows an example situation, which has a competition-states in the tours between $SalesmanA$ and B.

1. In the first comparison, the fitness of $SalesmanA$ itself are compared. If the $f(A)$ with the competition-city T is high, go to the following next comparison. Oppositely, if the $f(A')$ without the competition-city T is high, the $SalesmanA$ removes the T from own tour, and finishes this processing.

2. In the second comparison, a comparison between $SalesmanA$ and B is carried out. If the $f(A)$ is high, the opponent $SalesmanB$ removes the T. Otherwise, the $SalemanA$ itself removes the T.

8.2.4 Designs for fitness function and production method for similar solutions

In order to apply the proposed method, to design (1) fitness function and (2) production method for similar solutions are required.

The fitness function in formulated as equation (8.1) aims to optimize the efficiency for divisions. So, the fitness function is made so as to include the following two terms.

- In the first term, a solution, which has lower cost for traveling many cities, has high fitness.

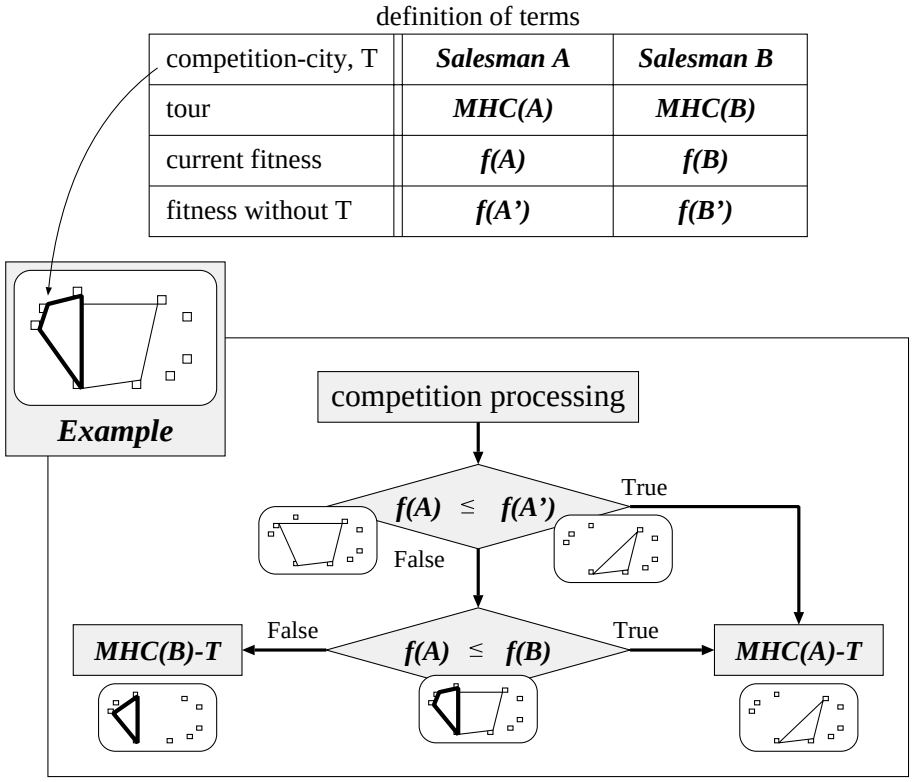


Figure 8.3: An example of competition processing.

- In the second term, a solution, which has higher cost, has lower fitness.

And, as a method for production of similar solutions, the two methods are implemented: (a) produce a solution, which includes a un-visited city in the current tour, and (b) produce a solution, which swaps any two cities in the current tour.

[fitness]: evaluation for agent$_i$

$$fitness_i = Cities(MHC(i))/Cost(MHC(i)) + 1/Cost(MHC(i)) \tag{8.1}$$

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

8.2.5 Design of simulation

In order to investigate the basic performance of our method, we applied this algorithm to the following n-TSP (Table8.1). The parameters of proposed method are set as Tables8.2. In the Table8.2, the increment rate and the swap rate are used in the production of antibodies.

Table 8.1: Problem definition.

The number of cities	25
The arrangement of cities	circle
Start city	Centered in the circle
Radius	0.4
The number of salesmen	3
Terminal steps	100

Table 8.2: Parameters.

The number of B cells	25
The increment rate for the number of traveled cities	0.5
The swap rate for the traveled order	0.5

8.2.6 Results and considerations

Figure8.4 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:

1. (a) the number of cities visited by each agent is 8 and the even division of work domain is computed and

2. (b) the solution of each agent is considered as the optimum solution.

Namely, the proposed method can solve the division-of-labor problems appropriately. Next, it considers the transition of the fitness and the distributions.

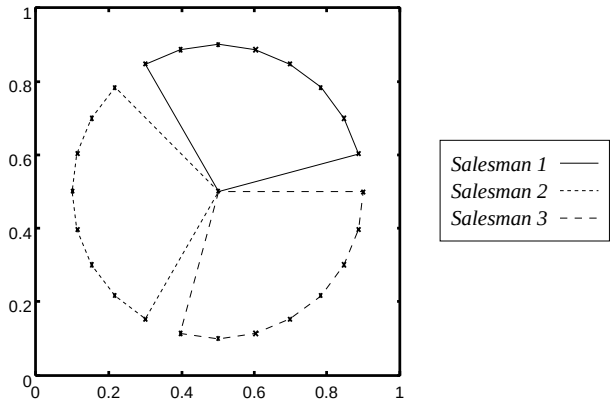


Figure 8.4: The obtained solution.

The transition of the fitneses

The transition of the fitness per step are shown in Figure8.5. The *System* in Figure8.5 means the efficiency for the total of the system, which the efficiency is calculated as a sum of all fitness. From Figure8.5, the following things are validated:

- the all fitness denotes a transition upper right simply, even though there are some decrease points.

In the proposed method, the reproduction of candidate solutions is carried out in the two cases: when (1) any eliminations are archieved for the competition-state or (2) the fitness increases by the current reproduction of solutions. In other words, these situations appear as the increase of the fitness. In the same way, when the competition processing is achieved, there will be decrease of the fitness. But there are the rarely situations in Figure8.5. So, in order to confirm the elimination appropriately, the transition of the number of the traveled cities in the tour and the tour itself are investigated.

Transition of the size of the work domains

Figure 8.6 shows a transition of the number of the cities (in other words, the size of the work domain) in the tour who each agent has.

From Figure8.6, the following thins are validated:

- the work domains are fluctuated greatly up to step 30, as contrasted to the transition of the fitness.

- the work domains are converged to the 8-city after step 60.

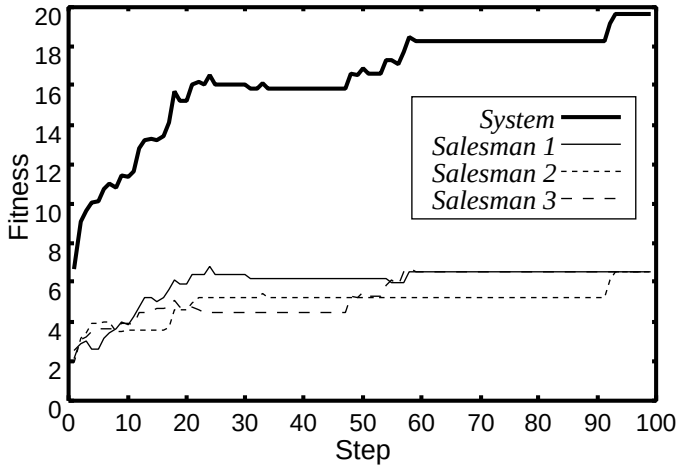


Figure 8.5: Transition of the fitnesss of the system and the each agent.

In this way, the proposed method is considered that shifts the work domains with maintain the fitness because the fitness are increased stably, even though the changes of the work domains are carried out aggressively. Especially, the steps that the margin of fluctuation are large are (1) from step 0 to 10, (2) in the vicinity of step 20, and (3) from step 50 to 60. The process of the changes are shown in Figure8.7.

(1) from step 0 to 10:
The initial work domains are made at random. This means, there are any overlapped work domains and any un-worked domains, so the reproduction of solutions are carried out frequently. The effect were arose as the changes of the work domains in Figure8.6 and the changes of solution in Figure8.6.

(2) in the vicinity of step 20:
In the vicinity in Figure8.5, the increase degree of the fitness is high comparably. And the large changes of the work domains in Figure8.6 are occurred by the decrease of salesman 3's one. Salesman works on the large work domain, but the distribution is not even in the problem domain. It is considered that the large changes is occurred to reduce the work domain for acquisition of even distributions. In fact, the work domain of salesman 3 shifted to the work domains of salesman 1 and salesman 2 in Figure8.7-(2).

(3) from step 50 to 60:
Again, in the Figure8.5, the increase of the fitness is good from step 50 to 60. And

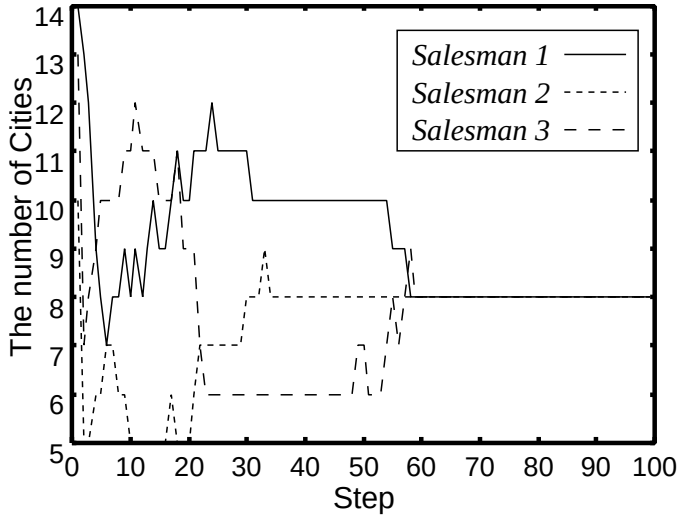


Figure 8.6: The transition of the traveled cities.

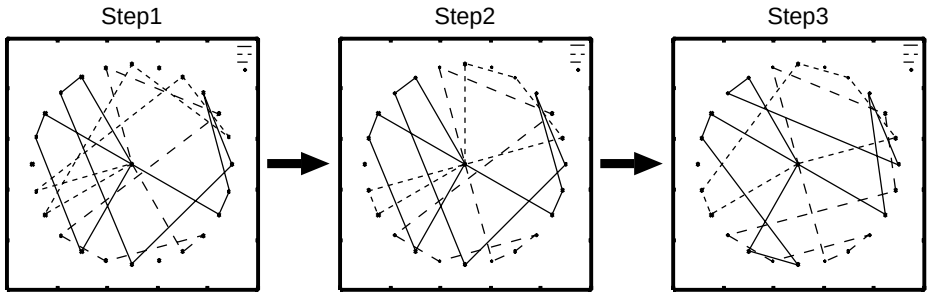
in Figure8.6, the decrease of the salesman 1's work domain and the increase of the salesman 3's work domain are carried out. After the step 40, the salesman 2 is converging already, but the changes through a competition processing between salesman 1 and salesman 3 are validated in Figure8.7-(3).

8.3 Conclusion

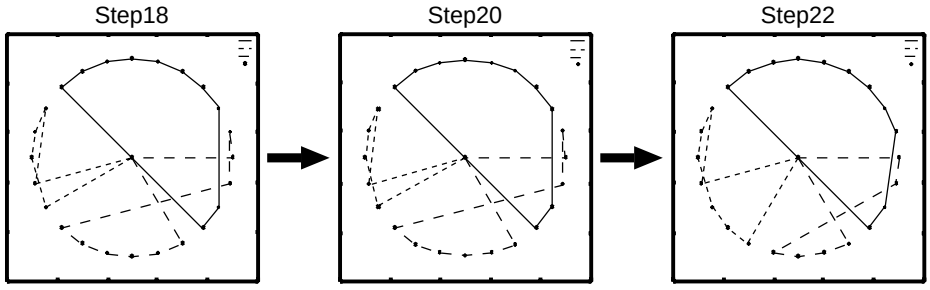
The principle behaviors and the basic performances to optimize the division-of-labor problems in MAS were validated through computational simulations. According to the above results and Toma et al (2000-b), our method has two advantages:

- (1) modification of the work-domain is changed while maintaining the fitness, and
- (2) the average costs are very small compared with GA.

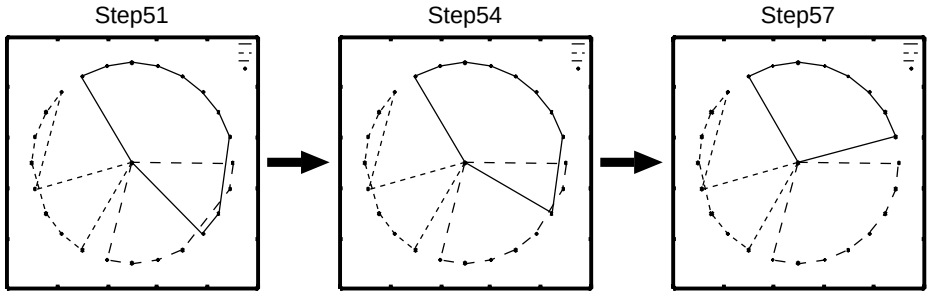
By contrast, the downside is, if the effect of maintaining is too strong, dividing evenly becomes difficult. This means, any improvements for the equality in the task of division-of-labor problems are required. In this instance, our algorithm has been improved so that the immune cell-cooperation which is a framework including the MHC and the immune network may be applied rather than having it applied directly to the local functions.



(1) from step 0 to 10



(2) in the vicinity of step 20



(3) from step 50 to 60

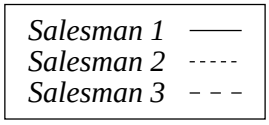


Figure 8.7: The process of the changes of each tours.

Chapter 9

An Immune Optimization based on an immune co-evolutionary phenomenon and cell-cooperation

9.1 Introduction

In this section, we propose an immune optimization algorithm, which is based on the immune cell-cooperation, to solve the division-of-labor problems. In the previous work for MAS (from chapter 6 to chapter:IDCPS, and [Toma 2000-b]), the immune distributed competitive problem solver had two problems; the even work domain and divergence could not be computed as mentioned in chapter8. The previous algorithm has been improved so that the immune cell-cooperation, which is a framework in a broad sense of elimination of antigens, will be applied rather than having it applied directly to the local functions.

The purposes of this chapter are to propose and evaluate an immune optimization algorithm using a biological immune co-evolutionary phenomenon and cell-cooperation. The co-evolutionary model searches the solution through the interactions between two kinds of agents, one kind of the agents is called immune agent, which optimizes the cost of its own work. The other is called antigen agent, which realizes 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, *escape processing* is used to perform equal work assignment as a result of evolving the antigen agents.

The immune agent optimizes own cost using division as well as integration processing based on the immune cell-cooperation which is considered as a kind of parallel-distributed system with role differentiation. 'Splicing' is one of the re-combination operator of genes, whose function is used for forming the role. The division as well as integration processing

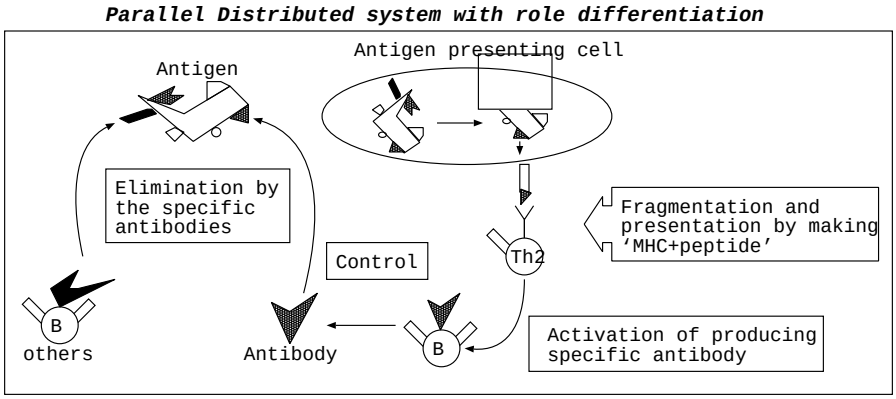


Figure 9.1: Concept of immune cell-cooperation.

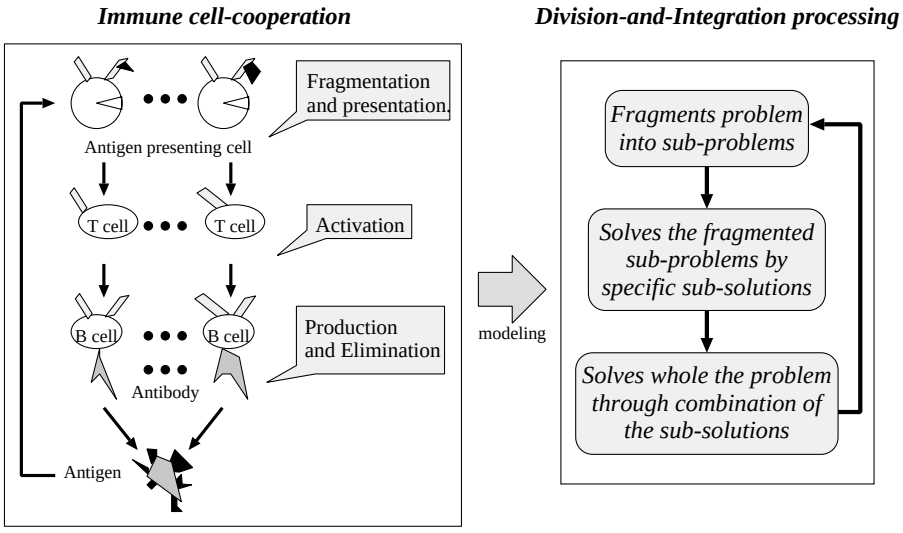


Figure 9.2: Analogy from immune cell-cooperation.

in our method is based on the splicing. And the antigen agent computes even division of work domain using escape processing based on a phenomenon that the antigen evolves to escape (survive) from the elimination of immune system.

9.2 Modeling of immune cell-cooperation

In the biological immune system, the two mechanisms (MHC and immune network) are considered important in eliminating invaded antigens. Thereupon, a framework, which together with the two functions eliminates the unknown vast antigens in parallel, is called

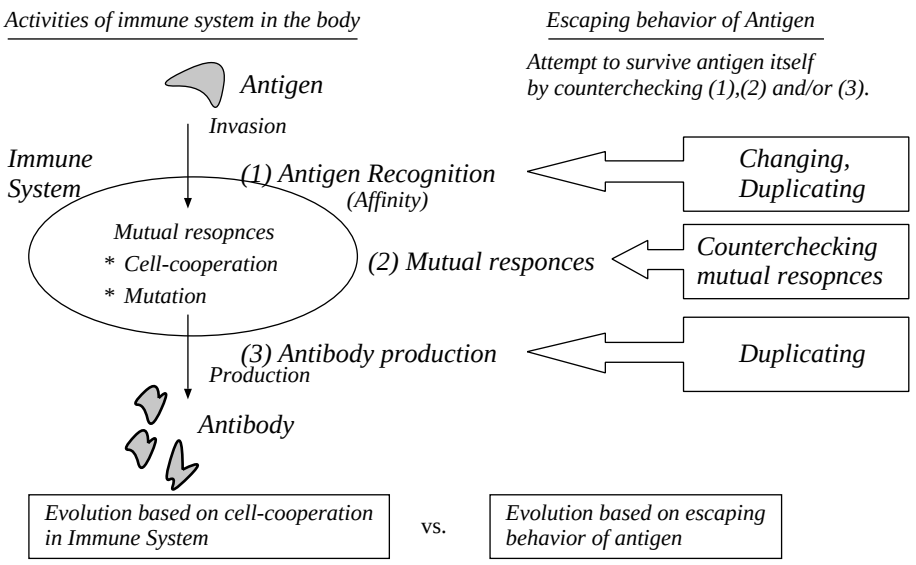


Figure 9.3: Concept of immune co-evolutionary phenomenon

'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 the functions (see Figure9.1). Note that, the cell-cooperation has a set of rules not only as 'function' (fragmentation, activation) but also as 'work domain' (as defined in section6.2). The object of the model is to implement and to apply them to the multi-agent system.

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* (Figure9.2).

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 for survival itself (Figure 9.3). 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 (Figure9.4).

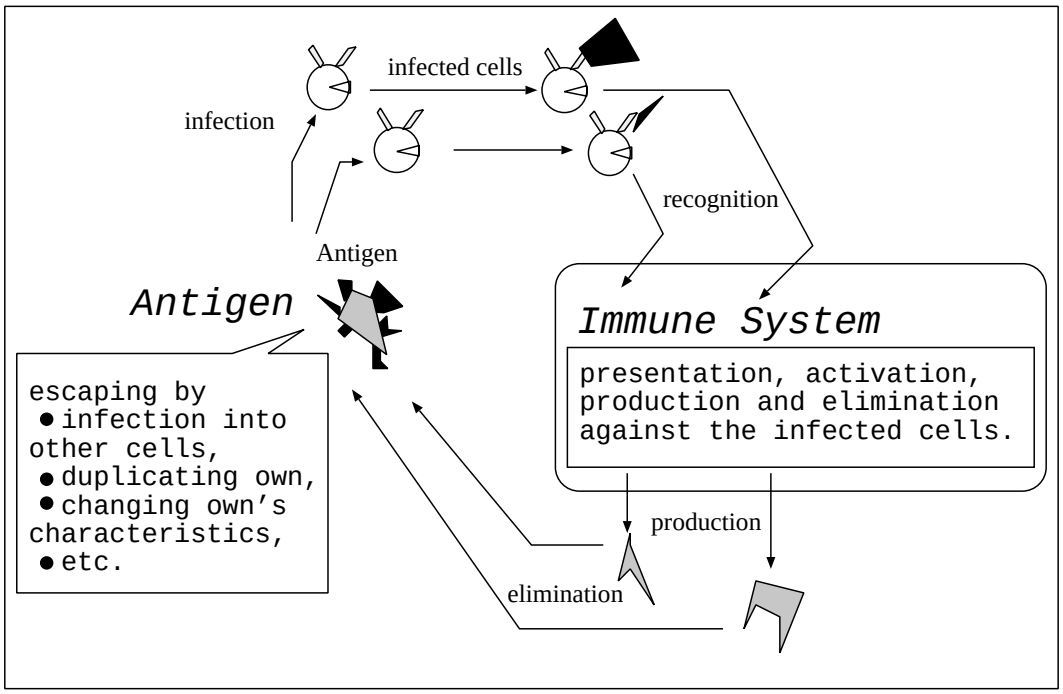


Figure 9.4: Illustration of a co-evolutionary like concept in the case of antigen vs. immune system.

9.3 An immune co-evolutionary algorithm for n-TSP

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 9.5).

[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 also. 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.

$$Cost(S_i) = \sum distance(tour_{S_i}) \tag{9.1}$$

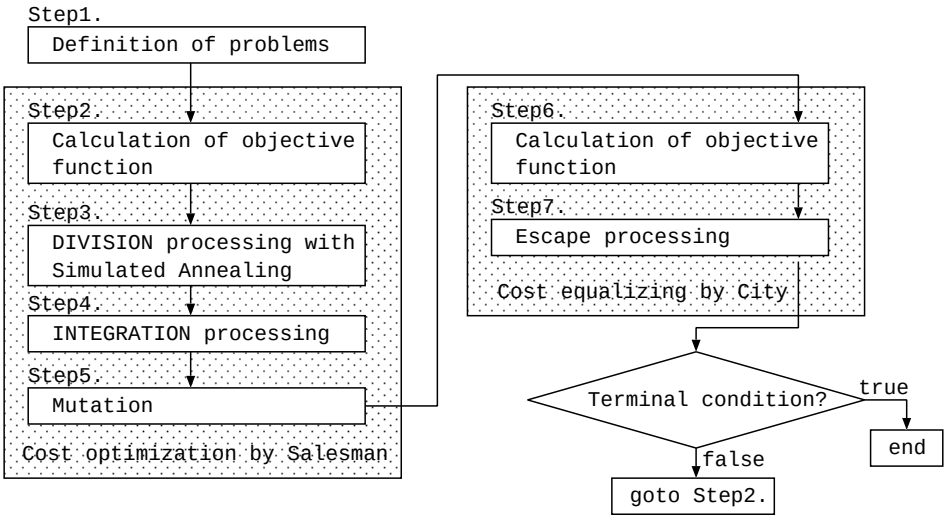


Figure 9.5: Flowchart of proposed method.

- 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 9.6): decide of any one $subtour_j$, make new $tour_i$ except the $subtour_j$, and calculate costs of both tours. Then, if the cost is improved, the division will be performed (as a consequence, a new agent is generated). Note that, in order to adjust the number of execution times of division processing, 'Simulated Annealing' (SA) 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 9.7): decide of any two tours, make new tour by binding, and calculate cost of the new tour. Then, if the cost is improved, the integration performed (as a consequence, a original salesman is deleted).

[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

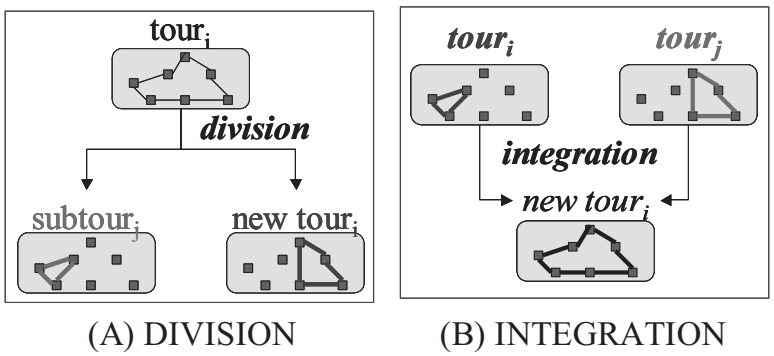


Figure 9.6: Examples of Division and Integration.

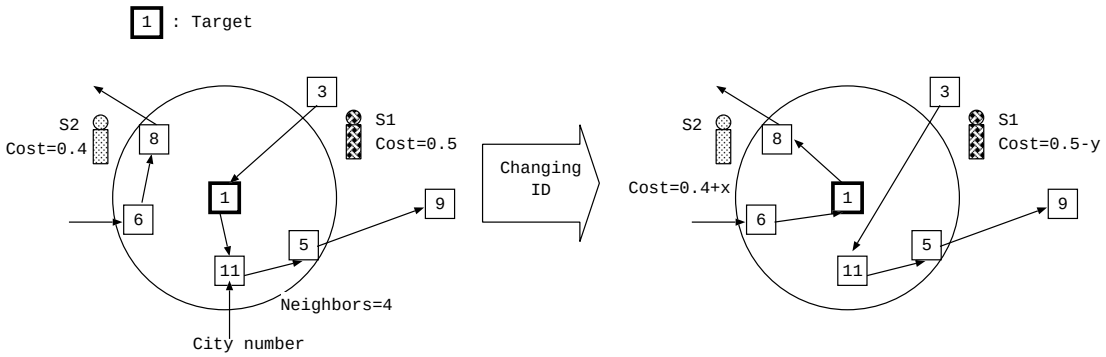


Figure 9.7: Escape processing.

9.1.

[Step7. Escape processing.]

This approach changes ID of a city depending on neighbors cost to process following steps (see also figure 9.7). 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 9.7, the ID of target city is changed to $S2$, and then, the target city is visited by $S2$ like the right side of figure 9.7.

In the computational experiments, we adjust the number of execution times of division processing by using of SA. Assuming that T for a temperature parameter and α for the

temperature adjustment parameter, the temperature is calculated by equation (9.2). The n denotes the number of repeated steps for algorithm.

$$T_n = \alpha \times T_{n-1} \tag{9.2}$$

9.4 Conclusion

In this chapter, an immune optimization algorithm for division-of-labor problems was proposed. This algorithm is based on a biological immune co-evolutionary phenomenon and cell-cooperation. The co-evolutionary models searches the solution through the three kinds of interactions: *division-and-integration processing* is used for optimization of the work-cost of immune agents and, *escape processing* is used to perform equal work assignment as a result of evolving the antigen agents.

In the next chapter, the computer simulations to investigate the performances are described.

Chapter 10

Computational experiments for a type of immune co-evolutionary optimization

10.1 Introduction

In the previous chapter, an immune optimization algorithm, which is based on a biological immune co-evolutionary phenomenon and cell-cooperation, for division-of-labor problems was propoased. In this chapter, the following three computer experiments are performed to investigate the basic performance of the algorithm.

experiment 1:
In the first experiment, the algorithm is applied to a problem in n-TSP and then the results are considered.

experiment 2:
In the second experiment, in order to confirm the availability to the general problems, this algorithm is applied to two testbed problems.

experiment 3:
In the third experiment, GA is applied to the same problem in the simulation 1, and the two performances are compared.

experiment 4:
As a additional investigation about the performances, a comparison wigh some conventional methods (GA, Saving method, and the hybrid method) is achieved.

10.2 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 10.1. The parameters of the proposed algorithm are shown in Table 10.2.

Table 10.1: 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 10.2: Parameters of the algorithm.

the number of neighbors	4
the number of initial agents	under 24
terminal steps	2000
the initial temperatureT_0	50
temperature adjustment parameterα	0.99

10.3 Experiment 1: system behavior

Figure10.1 shows an acquired solution. The best solution, which has been led to equal divisions and optimized the costs, is obtained. Distribution of the number of searching solutions is shown in Table10.3. 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.

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 10.2. 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

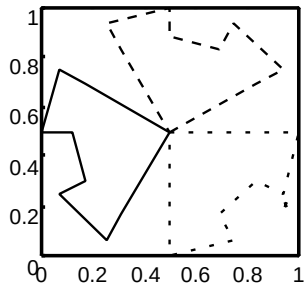


Figure 10.1: An obtained solution.

Table 10.3: Distribution of the number of searching solutions. (parenthetic value denots a value without the simulated annealing)

	Trials	Times of execution
Division processing	6,242	221 (6)
Integration processing	546	225 (16)
Mutation	6,037	50
Escape processing	47,876	47,876
Total	60,701	48,372

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.

10.4 Experiment 2: Verification through two testbed problems

The proposed algorithm is applied to two testbed problems (*eil51.tsp, gr202.tsp*), which provides on the TSBLIB [1] . The problem in the experiment 1 is a specialized problem that is a kind of deceptive problem, which has highly common characteristics in the optimal and

[1] http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/

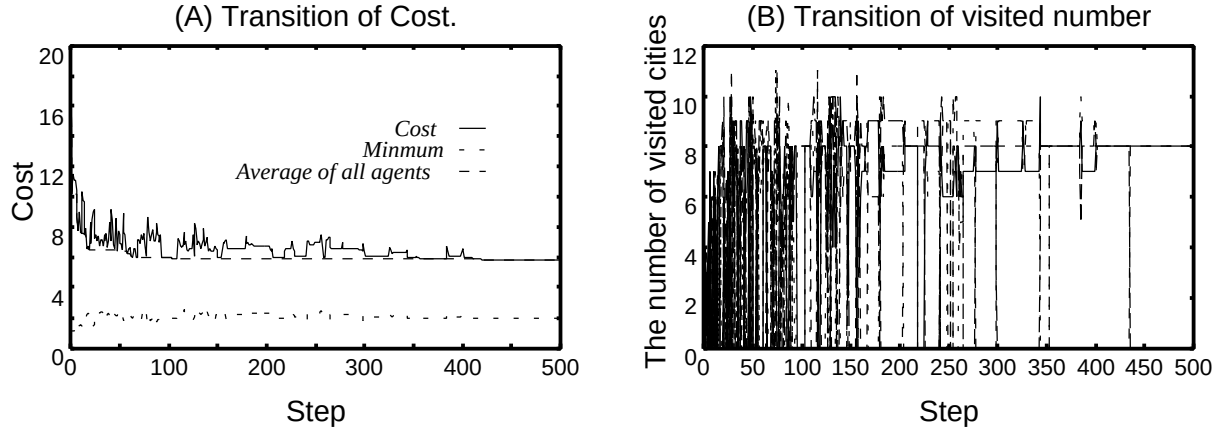


Figure 10.2: Transition of cost and the number of visited cities.

local optimum. Experiment 2 shows that our algorithm will function well in the general problems also.

10.4.1 Example 1: *eil51.tsp*

First example is *eil51.tsp* that has 51 cities whose distribution has no bias. And the optimal solution in case of 1 salesman is shown in Figure10.3.

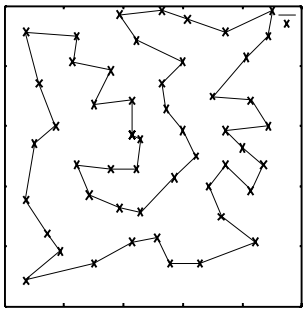
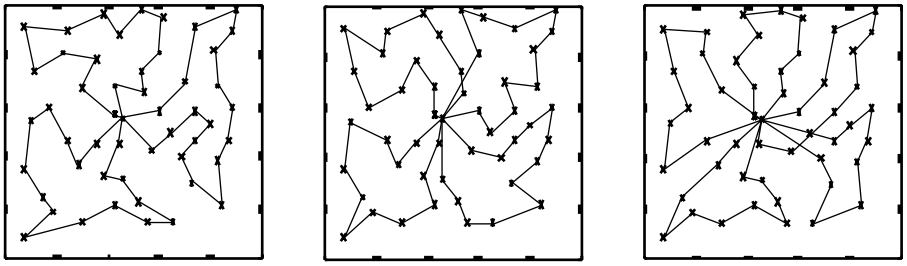


Figure 10.3: Optimal solution in case of 1 salesman in *eil51.tsp*.

For applying to the *eil51.tsp*, the city allocated to near central coordinate is set as a start city, and the minimum number of salesman is set from 3 to 5. The result is shown in the Figure10.4 and the cost for each configuration is shown in Table10.4. In the Table,

cost increases based on the increment of the number of salesman because the tour cost for return to the start city is required. In case of 4 or 5 salesmen, there are a few parts of overlapping tour. But (1) some parts of tours show similar common solution in Figure10.3, and (2) the increased cost for each salesman is about 0.3 that is small compared with total cost. So, we think our algorithm get good quality solution.



(A) 3-salesman (B) 4-salesman (C) 5-salesman

Figure 10.4: Solutions with 3, 4 and 5 salesmen in *eil51.tsp*.

Table 10.4: A comparison of cost.

	The number of salesmen			
	1	3	4	5
Cost	6.1471	6.7937	7.2458	7.8205
Increased rate	100%	about 111%	about 118%	about 127%
Increase cost for each salesman		0.215	0.274	0.334

Figure10.5 shows a transition of cost for each salesman per a step, in case of 5 salesmen. In the initial steps, the cost changes dramatically up to about step 600, and many vertical lines arise. This is an effect of cost optimizations by division-and-integration processing, and these processings cause making or deleting salesmen. And the other cost modifications in case of without the vertical lines have been achieved by mutation or escape processing. According to these processings's interactions, a best solution that consists of 8 cities with 3 agents obtained.

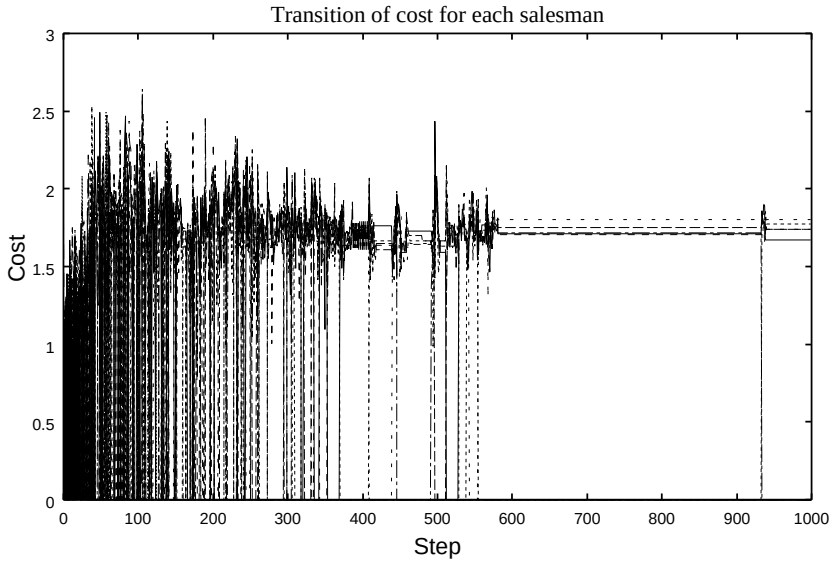


Figure 10.5: A transition of cost for each salesman in $eil51.tsp$ with 5 salesmen.

10.4.2 Example 2: *gr202.tsp*

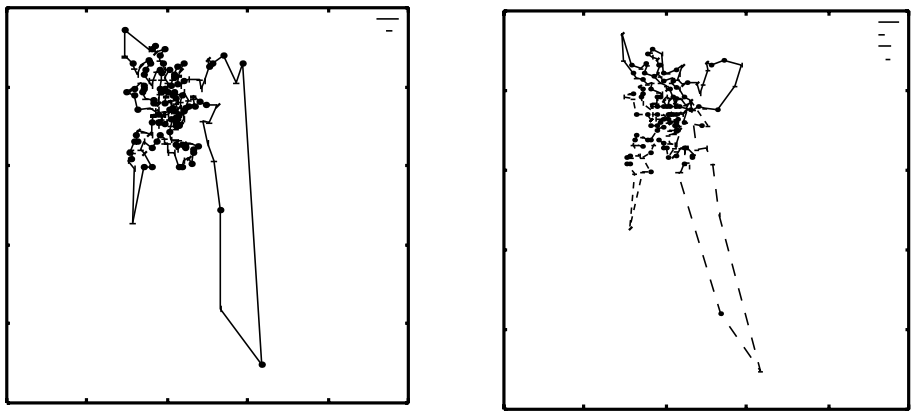
Example 2 applies our algorithm to *gr202.tsp* that has 202 cities whose distribution has a bias. The optimal solution with 1 salesman is shown in Figure10.6-(A).

The configuration of algorithm is set like this: the minimum number of salesman is 3, the number of neighboring cities is 10 and terminal step is 2000. The results are shown in Figure10.6-(B). The increased cost is 7% compared with 1 salesman's result. The average cost for each agent is 1.597939 and the sum of absolute error is 0.008215. In addition, the cost error is only 2% of the total cost, so this solution is considered as a superior solution.

10.5 Experiment 3: Comparison with GA

10.5.1 Design of GA

Experiment 3 compares 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[Yamamura 1992]. Furthermore, subtour crossover corresponding to the pass representation method is adopted. In subtour crossover, two subtours that have the same elements are looked up, and those subtours are replaced. It becomes possible that the destruction of the important subtour is prevented



(A) 1-salesman (cost=4.447073)

(B) 3-salesman (cost=4.793818)

Figure 10.6: The results for $gr202.tsp$ with 1 and 3 salesmen.

by adopting this method. The parameters of GA are set in Table10.5. The fitness_i, which is the fitness of agent_i is expressed in equation (10.1).

Table 10.5: Parameters of GA.

Population	100
Crossover rate	1.0
Mutation rate	0.01
Terminal generations	5000

$$fitness_i = 1/(Cost + Penalty) \tag{10.1}$$

$Cost : Lengthofthetour$
$Penalty : Penaltyfortheequality$

10.5.2 Results

The searching performances are compared through simulation results of 20 trials. Table 10.6 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 10.6: 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	Min(best)	74,100	1,616
of searched	Ave	167,100	8,488
solutions	Max(worst)	227,100	16,580
Run time		50sec	10sec

10.6 Experiment 4: Comparison experiment 2

Experiment 4 compares the performances of Saving method, GA and the hybrid method Saving-GA[Yokoyama 2001]. In order to evaluate the performance of our method, we apply this algorithm to the following n-TSP. One of the reason to adopt the n-TSP is that n-TSP provides two peaks deceptive problem. In this deceptive problem, each city is arranged on two-fold concentric circles. Two kinds of problems can be made by changing two circular radius ratios. As for one problem, c-type is the answer whose tours visit other circle's cities after the one circumferential's ones. O-type is the answer whose tours visit each circumferential's city alternatively. Definition of problem is set in Table10.7.

10.6.1 Design of Saving method

Saving method is a kind of deterministic approximate method which is a famous method in the filed of vehicle routing problem in the front of very fast algorithm. Assuming a vehicle routing problem that agents with lading size b distribute baggage size a_i to city$_i$($i = 1, 2, ..., n$).

1. Set initial distributing routes.
   The initial distributing routes consist of a set of route that distributes to just one city (see Figure10.7).

Table 10.7: Definiton of problem.

The number of cities	49
The number of salesmen	3
The arrange ment of cities Dual circle	
Start city	Center of circle
Radius (out, in)	
Case1 (O-type)	Out=0.5, In=0.384
Case2 (C-type)	Out=0.5, In=0.386

2. Calculate saving-value.
   The saving-value that is a saved distance by binding any two routes is calculated by following equation.
   $$s_{i,j} = d_{i,0} + d_{0,j} - d_{i,j}, \tag{10.2}$$
   where $d_{i,j}$ is a distance between city$_i$ and city$_j$.

3. Choose two routes for binding.
   The highest saving-value from all possible combination is set as a binding target.

4. Bind the routes.
   The binding target must be bound and then return to process 3. Note that, when the highest saving-value is minus or the loading value of the bound route is over the b, Saving method terminates.

In the computational experiment, $a_i = 1$ and $b = (the number of cities)/(the number of salesmen)$ are set.

10.6.2 Design of Saving-GA

Saving-GA is a hybrid method[Yokoyama 2001] that GA searches by using the result of saving method (and this GA uses the subtour-crossover also). Note that, in our computer simulations, Saving-GA doesn't use the improving by simulated annealing because it takes excessive run time in accordance with the authors saying in the report. It is important that a high quality solution with divided cost evenly among agents finds fast in the best way possible.

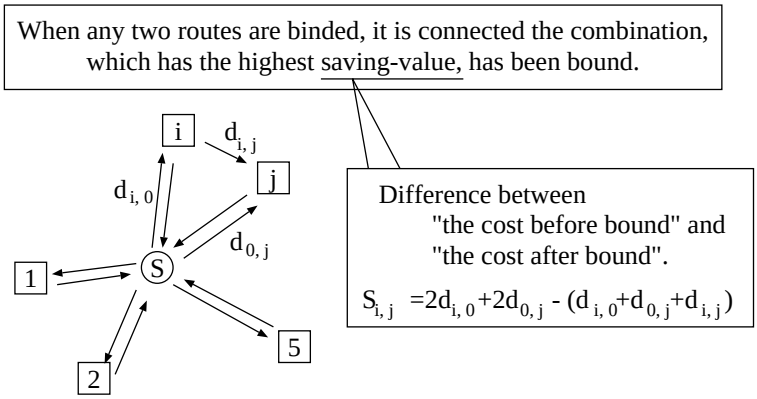


Figure 10.7: Saving method.

10.6.3 Results and discussions

We applied the four method to the problem defined in Table 10.7: (1) Genetic Algorithm using subtour crossover (population = 400, terminal generation = 10,000), (2) Saving method, (3) Saving-GA (population = 100, terminal generation = 5,000), and (4) our proposed method (neighboring cities = 4, initial salesmen = 48, terminal steps = 1,000).

Figure 5 shows the obtained solutions (Tour and Cost), the run time for obtaining the solutions, and the degree of division (sum error: calculated by equation 10.3).

$$sum\_error = \sum_{i}^{N} |\bar{cost} - x_i| \tag{10.3}$$

$$\bar{cost} = (\sum_{i}^{N} x_i)/N \tag{10.4}$$

The solutions obtained by GA and Saving method are 5-10% poor costs in relation to best solution, thus these methods are no sufficient ability from point of view's optimization for whole working's cost. And these sum errors have differences more than 20% among salesmen's cost, so the dividing ability is insufficiently also.

On the other hand, Saving method is possible to obtain a good quality solution with short run time. In the dual circled n-TSP which is a kind of deceptive problem, however, there is a deceptive case which Saving method obtains the same or similar solution although the problems are different. In fact, the dual circled problems have two typically solutions: c-type and o-type, but Saving method couldn't obtain the o-type's one. Therefore, Saving-GA which uses the solution of Saving method as initial population is possible to obtain

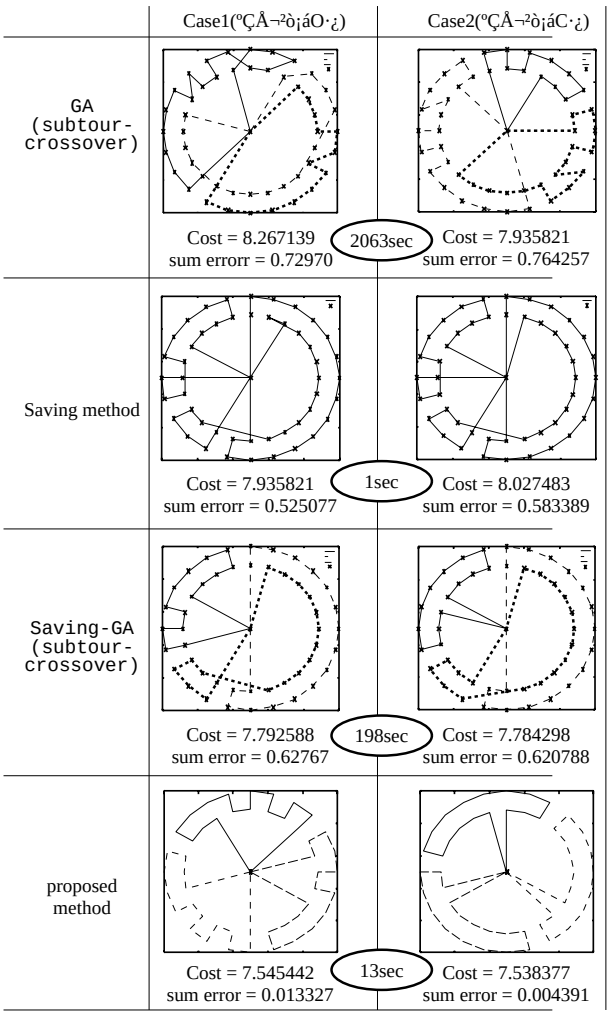


Figure 10.8: The results of GA, Saving method, Saving-GA and our method.

better (lower cost) solution with short run time than GA, but it converges to the c-type solution even though the type of problem is o-type.

The proposed method is possible to get a good solution in a short time extremely comparison with GA and Saving-GA, although it needs more run time against the saving method. The solutions which are very lower costs (proposed method's cost is lowest!), in addition, the sum error has little difference (only 1-2% among selesmen's cost) also. From these results, it is considered that our method is very efficient algorithm for division-of-labor optimization.

10.7 Conclusion

In this chapter, we proposed and evaluated an immune optimization algorithm 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.

Chapter 11

Conclusion

This paper described the potentialities of the engineering models inspired from a biological immune system through making some models from analogies, constructing algorithms on the models and then investigating on the computational simlations. Especially, we dealt with multimodal functions and division-of-labor problems as the target problems that issue included strongly required demands from users.

In the first target problem, the objective of multimodal function optimizations was to obtain robust solutions, which could provide some rooms to choose carrying solution for users. The optimization methods for that purpose were required to obtain multi optimal solutions at a problem solving. In other wards, we resolved the demands through ready for multi alternatives.

As a method for multimodal function optimization, we proposed an Adaptive Memorizing Immune Algorithm (AMIA) with two memory mechanisms, which AMIA was an extended algorithm of Mori's IA[Mori 1993, Mori 1997]. IA based on the biological immune system can obtained multi optimal solutions without the constraint in sharing[Goldberg 1987, Goldberg 1992] and niche method[Shima 1995] that the number of individuals in the population (i.e., the population size) should be greater than the number of optimal solutions. However, there are two issues; (1) the appropriate parameters adjustment is a critical point for apropriate behavior, but any method for the adjustment didn't support, and (2) the use of obtained good solutions carries out only for the second solving time.

This paper provided the two memory mechanisms to resolve the above-mentioned two issues. The basic idea is how to use a secondary immune response on the first problem solving. The secondary immune response, which can carry out effective problem solving as antigen elimination, exploits the obtained memory on the after second elimination for the same antigen only, in the biological immune system. But, if the use of the secondary immune response on the first problem solving is implemented, the search method will get superior performance. So, the models of the three subsystems (the primary and secondary immune responses and the restraint system) in the biological immune system were constructed and

applied as an approach for a search system, and then the two memory mechanisms made based on the models. As results, in point of the first issue, the parameters adjustment got easily though establishment obtaining memory and restraining re-search individually. About the second issue, the search ability was enhanced through use of obtaining memory and secondary immune response at the first solving time also. The introduced memory mechanisms consist of following two information processing mechanisms.

primary memory mechanism:
It memorizes common characteristics of the candidate group are memorized as a template in the memory cell. And then, it carries out narrowing down of search space through a candidate group is reproduced based on the template.

secondary memory mechanism:
It memorizes a search point who dominates whole population in the suppressor cell. And then, it carries out restraint to search same point again using the memory.

In the computational experiments, AMIA was applied to a deceptive problem in TSP and Bipolar Deceptive Function to investigate the principle behavior and the performances. From these results:

1. **validity of a local search**,
2. **stability of obtaining memory**,
3. **possibility to search high fitness solutions**,
4. **improvement of the search ability**, and
5. **enhancement to obtain superior solutions**

were validated, and we concluded the search efficiency of AMIA improved by a memory mechanism's working effectively.

In the second target problem, we attempted an application to division-of-labor problems in MAS to verify the further potentialities. The objective of division-of-labor problem optimization in MAS was to make best distributions of the work domains for agents, which the distributions were satisfied (1) equaling work assignment, (2) optimizing a work-cost individually, and (3) optimizing all amount of work-costs. And the problem was required to minimize a total cost in a parallel distributed processing such as scheduling problem, job scheduling in a networked parallel computers[Coffman 1976], TSP with some constraints[1] , vehicle routing problem (VRP) and the VRP with time constraints, and so on.

[1] http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/

As the methods for division-of-labor problem optimizations, we constructed two kinds of immune based optimization algorithms. As first algorithm for division-of-labor problems, we proposed an immune distributed competitive problem solver with MHC and immune network. This algorithm solved the division-of-labor problems for each agent's work domain. The MHC was used for eliminations of the states of competition among agents. The immune network was used to produce adaptive behaviors for agents. Through the implementation of such models, we could construct an adaptive algorithm that solves division-of-labor problems, and from the computational simulations, this method had two advantages:

- (1) modification of the work-domain is achieved while maintaining the fitness, and
- (2) the average costs are very small compared with GA.

By contrast, in the work[Toma 2000-b], the immune distributed competitive problem solver had two problems; we were not able to compute even work domain and the appropriate divergence in the case of more complicated problems. And so, as second algorithm, we proposed a new immune optimization algorithm so that the immune co-evolutionary phenomenon and 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 purposes of the algorithm were to propose and evaluate an immune optimization algorithm using a biological immune co-evolutionary phenomenon and cell-cooperation. The co-evolutionary models searched the solution through the interactions between two kinds of agents, one of the agents was called immune agent, which optimizes the cost of its own work. The other was called antigen agent, which realizes the equal work assignment. This algorithm solved division-of-labor problems in MAS through the three kinds of interactions: *division-and-integration processing* was used for optimization of the work-cost of immune agents and, *escape processing* was used to perform equal work assignment as a result of evolving the antigen agents. And the antigen agent computes even division of work domain using escape processing based on a phenomenon that the antigen evolves to escape from the elimination of immune system.

In the computational experiments, the immune co-evolutionary method was applied to n-TSP to investigate the principle behavior and the performances. From these results:

1. **validity of an immune co-evolutionary algorithm**,
2. **stability of the obtaining superior solutions**,
3. **quickness of the running time**, and
4. **applicability to solve general problems**

were validated. 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. In this way, we concluded the optimization ability for division-of-labor problems was pretty good approach.

Finally, we described the mainly three approaches: AMIA, immune competitive algorithm, and immune co-evolutionary algorithm before now. They showed many interesting characteristics and superior results as engineering models (although the competitive method could obtain a good solution in simple problem only) through the computational experiments. This means, it will expect that the models exploited a biological immune system have many important considerations as engineering applications. Especially, an immunological memory is most interesting functions we think, which the memory will be considered that is acting as not only 'a just memory' but also 'a catalyst' for appropriate effective communications between immune cells. And I hope that this or the other immunological methods arouse interest among other researchers.

Acknowledgements

This doctoral thesis is the one that the author summarized the study that went the base of the guidance of Prof. Hayao Miyagi, Assistant Prof. Endo Satoshi and Assistant Prof. Yamada Koji while a graduate student in the university of the Ryukyus. In summarizing this research the mind of appreciation is expressed to Prof. Hayao Miyagi, Assistant Prof. Endo Satoshi and Assistant Prof. Yamada Koji in the Faculty of Engineering, University of the Ryukyus., who were given significant hand-holding guidance and encourages.

Also, I express the mind of appreciation to Prof. Shiro Tamaki and Prof. Tomohisa Wada who were given significant hand-holding guidance and encourages in completing this paper.

And, I thank for Japan Society for the Promotion of Science which asked to adopt as a research fellow (DC1).

Furthermore, I express the mind of appreciation to Dr. G. P. Yeh who was given significant guidance and encourages during the training to Fermi National Accelerator Laboratory in USA, and to the staffs that received the effort: Mr. Takashi Sunagawa (Executive Director) and Mr. Toshirou Kiyokawa (Human Resources Adviser) in Industry Promotion Public Corporation, Okinawa Prefectural Government, Mr. Yoritaka Hanashiro (Director General) and Ms. Mutsumi Nakasone in Dept. of Commerce. Industry & Labor, and Ms. Mari Rita Tobaru and Mr. David Coleman in Antenna, and all staffs for the training.

In addition, I 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 discussions with them were given valuable ideas.

Finally, I would like to appreciate to Ph. D. Moeko Nerome, the colleagues in complex systems lab.: Mr. Yuhei Akamine (D1), Mr. Nobuto Iji (M2), Mr. Tokinari Matayoshi (M2) and undergraduate students, and the seniors: Mr. Tadashi Tamashiro, Mr. Satoru Yonaha, Mr. Hirotaka Yamashiro who were given valuable comments and supports.

References

[Ballet 1998] P. Ballet, J. O. Pers, Y. Rodin, J. Tisseau : *A Multi-agent System to Simulate an Apoptosis Model of B-CD5 Cells*, IEEE International Conference on Systems, Man, and Cybernetics (SMC) '98 Conference Proceedings, pp.3799-3803 (1998).

[Coffman 1976] E. G. Coffman : *Computer and Job-shop Scheduling Theory*, John Willey & Sons (1976).

[Celada 1998] Franco Celada and Philip Seiden : *Modeling Immune Cognition*, IEEE International Conference on Systems, Man, and Cybernetics (SMC) '98 Conference Proceedings, pp.3787-3792 (1998).

[Dasgupta 1998] Dipankar Dasgupta : *An Artificial Immune Systems as a Multi-Agent Decision Support System*, IEEE International Conference on Systems, Man, and Cybernetics (SMC) '98 Conference Proceedings, pp.3816-3820 (1998).

[Dasgupta 1999] Dipankar Dasgupta (editor) : *Artificial Immune Systems and Their Applications* (1999).

[Forrest 1990] S. Forrest, A.A. Perelson. : *Genetic algorithm and the Immune system*, Proc. of 1st Workshop on Parrallel Problem Solving from Nature (PPSN), pp.320-325 (1990).

[Goldberg 1987] D. E. Goldberg and J.Richardson : *Genetic Algorithms with Sharing for Multimodal Function Optimization*, Proc. of the Second Intern. Conf. on Genetic Algorithms, pp.41-49 (1987).

[Goldberg 1989] D. E. Goldberg : *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley (1989).

[Goldberg 1992] Devid E. Goldberg, Kalyanmoy Deb, and Jeffrey Horn : *Massive Multimodality, Deception, and Genetic Algorithms*, PPSN 2 (1992).

[Grefenstette 1985] J.Grefenstette, R.Gopal, B.J.Rosmaita and D.Van Gucht : *Genetic Algorithms for the Traveling Salesman Problem*, Proc. of ICGA '85, 16/168 (1985).

[Hirota 1996] K. Hirota : *Introduction to intelligence engineering*, Syoko-do (1996).

[Holland 1992] John H. Holland : *Adaptation in Natural and Artificial Systems*, MIT Press (1992).

[Ishida 1996] Toru Ishida, Yasuhiro Katagiri, and Kazuhiro Kuwabara: *Distributed Artificial Intelligence*, Corona Publishing (1996).

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

[Ishiguro 1997] A. Ishiguro, T. Kondo, Y. Watanabe, Y. Shirai, Y. Uchikawa : *An Evolutionary Construction of Immune Network-Based Behavior Arbitration Mechanism for Autonomous Mobile Robot*, The Transactions of The Institute of Electrical Engineers of Japan, Vol.117-C, No.7, July (1997).

[Ishigro 1998] Akio Ishiguro, Shingo Ichikawa, Takanori Shibata, Yoshiki Uchikawa : *Moderationism in the Immune System: Gait Acquisition of a Legged Robot Using the Metadynamics Function*, IEEE International Conference on Systems, Man, and Cybernetics (SMC) '98 Conference Proceedings, pp.3827-3832 (1998).

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

[Kennedy 1998] Catriona M. Kennedy : *Evolution of Self-Definition*, IEEE International Conference on Systems, Man, and Cybernetics (SMC) '98 Conference Proceedings, pp.3810-3815 (1998).

[Kitano 1993] H. Kitano : *Genetic Algorithm*, Sangyo-Tosho (1993).

[Mori 1993] K. Mori, M. Tsukiyama, T. Fukuda : *Immune algorithm with searching diversity and its application to resource allocation problem*, T. IEE Japan, Vol. 133-C, No. 10, pp.872-878 (1993).

[Mori 1997] K. Mori, M. Tsukiyama, T. Fukuda : *Application of an immune algorithm to multi-optimization problems*, T. IEE Japan, Vol. 177-C, No. 5, pp.593-593 (1997).

[Mori 1998] Kazuyuki Mori, Makoto Tsukiyama and Toyoo Fukuda : *Adaptive Scheduling System Inspired by Immune System*, IEEE International Conference on Systems, Man, and Cybernetics (SMC) '98 Conference Proceedings, pp.3833-3837 (1998).

[Nakamura 1994] T. Nakamura, T. Tsunoda, H. Tanaka : *Solution of n-Traveling Salesman Problem Using Neural Network Dynamics*, Artificial Intelligence 94-2 (1994).

[Russell 1995] Stuart Russell and Peter Norvig : *Artificial Intelligence: A Modern Approach*, Prentice-Hall (1995).

[Sakawa 1995] M. Sakawa, M. Tanaka : *Genetic Algorithm*, Asakura Book Co. Ltd. (1995).

[Sasaki 1999] M. Sasaki, M. Kawafuku, and K. Takahashi : *An Immune Feedback Mechanism Based Adaptive Learning of Neural Network Controller*, 6th International Conference on Neural Information Processing, Perth, Australia, pp.502-507 (1999).

[Shima 1995] T. Shima : *Global Optimization by a Niche Method for Genetic Algorithm*, Journal of the Institute of Systems, Control and Information Engineers, Vol. 8, No. 5, pp.233-235 (1995).

[Thomas 1997] Thomas Back (editor) : *Proceedings of The Seventh International Conference on Genetic Algorithms*, Morgan Kaufmann (1997).

[Toma 2000-a] N. Toma, E. Satoshi, K. Yamada : *The Proposal and Evaluatino of an Adaptive Memorizing Immune Algorithm with Two Memory Mechanisms*, Journal of Japanese Society for Artificial Intelligence, Vol.15, No.6, pp.1097-1106 (2000).

[Toma 2000-b] N. Toma, E. Satoshi, K. Yamada, H. Miyagi : *The Immune Distributed Competitive Problem Solver with MHC and Immune Network*, INTELLIGENT ENGINEERING SYSTEMS THROUGH ARTIFICIAL NEURAL NETWORKS, Vol.10 (Editors C.H. Dagli *et al.*), ASME PRESS, pp.317-322 (2000).

[Toma 2001] N. Toma, E. Satoshi, K. Yamada, H. Miyagi : *A Proposal of an Immune Optimization Inspired by Cell-cooperation*, INTELLIGENT ENGINEERING SYSTEMS THROUGH ARTIFICIAL NEURAL NETWORKS, Vol.11 (Editors C.H. Dagli *et al.*), ASME PRESS, pp.423-428 (2001).

[Toma 2002-a] N. Toma, S. Endo, K. Yamada, H. Miyagi : *The Immune Distributed Competitive Problem Solver Using Major Histocompatibility Complex and Immune Network*, OPERATIONS RESEARCH / MANAGEMENT SCIENCE AT WORK (editor Erhan Kozan and Azuma Ohuchi), Kluwer's INTERNATIONAL SERIES, pp.129-147 (2002).

[Toma 2002-b] N. Toma, S. Endo, K. Yamada, H. Miyagi : *The Proposal and Evaluation of an Extended Immune Optimization Algorithm using the Immune Cell-cooperation and the Co-evolutionary-like Approach for the Division-of-labor Problems*, Journal of Japan Society for Fuzzy Theory and Systems, 2002. (in press)

[Watanabe 1998] Y. Watanabe, A. Ishiguro, Y. Shirai and Y. Uchikawa : *Emergent Construction of Behavior Arbitration Mechanism Based on the Immune System*, Advanced Robotics, Vol.12, No.3, pp.227-242 (1998).

[Yamamura 1992] M. Yamamura, T. Ono, S. Kobayashi : *Character-Preserving genetic algorithm for Traveling salesman problem*, Journal of JSAI, Vol. 7, No. 6, No.v, pp.117-127 (1992).

[Yokoyama 2001] H. Yokoyama, Y. Shirai, N. Matsumoto and T. Kawanago : it Combination Method using Saving Method, Genetic Algorithms, and Simulated Annealing for

Vehicle Routing Problem, Fuzzy, Artificial Intelligence, Neural Networks and Computational Intelligence (FAN Symposium '01), pp.55-60 (2001).

List of publications

1. Journals

1. Naruaki Toma, Satoshi Endo, Koji Yamada : *The Proposal and Evaluation of an Adaptive Memorizing Immune Algorithm with Two Memory Mechanisms*, Journal of Japanese Society for Artificial Intelligence, Vol.15, No.6, pp.1097-1106, 2000.

2. Naruaki Toma, Satoshi Endo, Koji Yamada, Hayao Miyagi : *The Immune Distributed Competitive Problem Solver with MHC and Immune Network*, INTELLIGENT ENGINEERING SYSTEMS THROUGH ARTIFICIAL NEURAL NETWORKS, Vol.10 (Editors C.H. Dagli *et al.*), ASME PRESS, pp.317-322, 2000.

3. Naruaki Toma, Satoshi Endo, Koji Yamada, Hayao Miyagi : *A Proposal of an Immune Optimization Inspired by Cell-cooperation*, INTELLIGENT ENGINEERING SYSTEMS THROUGH ARTIFICIAL NEURAL NETWORKS, Vol.11 (Editors C.H. Dagli *et al.*), ASME PRESS, pp.423-428, 2001.

4. Naruaki Toma, Satoshi Endo, Koji Yamada, Hayao Miyagi : *The Immune Distributed Competitive Problem Solver Using Major Histocompatibility Complex and Immune Network*, OPERATIONS RESEARCH / MANAGEMENT SCIENCE AT WORK (editor Erhan Kozan and Azuma Ohuchi), Kluwer's INTERNATIONAL SERIES, pp.129-147, 2002.

5. Naruaki Toma, Satoshi Endo, Koji Yamada, Hayao Miyagi : *The Proposal and Evaluation of an Extended Immune Optimization Algorithm using the Immune Cell-cooperation and the Co-evolutionary-like Approach for the Division-of-labor Problems*, Journal of Japan Society for Fuzzy Theory and Systems, 2002. (in press)

2. International conferences

1. Naruaki Toma, Satoshi Endo, Koji Yamada : *Immune algorithm with immune network and Major Histocompatibility Complex*, INTERNATIONAL SYMPOSIUM ON ARTIFICIAL LIFE AND ROBOTICS(AROB 4th '99), pp.391-394 Vol.2, Oita, JAPAN (19-22 Jan. 1999).

2. Naruaki Toma, Satoshi Endo, Koji Yamada : *: Immune algorithm with Immune Network and MHC for adaptive problem solving*, 1999 IEEE International Conference on Systems, Man and Cybernetics, pp.IV-271-276(CD-ROM), Tokyo JAPAN (12-15 Oct 1999).

3. Naruaki Toma, Satoshi Endo, Koji Yamada, Hayao Miyagi : *The Immune Distributed Competitive Problem Solver Using MHC and Immune Network*, ASOR2000 (ORSJ Hokkaido Chapter and ASOR Queensland Branch The 2nd Joint International Workshop), Proceedings of The 2nd Joint International Workshop Between ORSJ-The Operations Research Society of Japan, Hokkaido Chapter and ASOR-The Australian Society for Operations Research Inc. Queensland Branch, pp.82-89, Sapporo, JAPAN (26-28 June 2000).

4. Naruaki Toma, Satoshi Endo, Koji Yamada, Hayao Miyagi : *Evolutionary Optimization Algorithm using MHC and Immune Network*, SEAL2000 (The Third Asia-Pacific Conference on Simulated Evolution And Learning. SEAL sessino in IECON-2000.) , CD-ROM, pp.2849-2854, Nagoya, JAPAN (25-27 Oct. 2000).

5. Naruaki Toma, Satoshi Endo, Koji Yamada, Hayao Miyagi : *A proposal of an immune optimization for division-of-labor problems using immune cell-cooperation and tolerance*, ASOR2001, Proceedings of ASOR2001, 16th National Conference, (CD-ROM), Adelaide, Australia (23-26 Sep. 2001).

6. Naruaki Toma, Satoshi Endo, Koji Yamada, Hayao Miyagi : *An Immune Optimization Inspired by Biological Immune Cell-cooperation for Division-and-labor Problem*, ICCIMA2001, Proceedings of Fourth International Conference on Computational Intelligence and Multimedia Applications, pp.153-157, Hokkaido Chapter and ASOR-The Australian Society for Operations Research Inc. Queensland Branch, pp.18-25, Sapporo, JAPAN (26-28 June 2000).

7. Naruaki Toma, Satoshi Endo, Koji Yamada, Hayao Miyagi : *An Immune Optimization using Biological Immune Co-evolutionary Phenomenon and Cell-cooperation for Division-and-labor Problem*, Knowledge-Based Intelligent Information Engineering Systems and Allied Technologies, KES 2002, Volume 82 Frontiers in Artificial Intelligence and Applications (Edited by: E. Damiani , L.C. Jain and R.J. Howlet), pp.718-722, Crema, ITALY (16-18 Sep 2002).