

PAPER

A Multi-Layered Immune System for Graph Planarization Problem

Shangce GAO[†], Student Member, Rong-Long WANG^{††}, Hiroki TAMURA^{†††}, and Zheng TANG^{†a)}, Members

SUMMARY This paper presents a new multi-layered artificial immune system architecture using the ideas generated from the biological immune system for solving combinatorial optimization problems. The proposed methodology is composed of five layers. After expressing the problem as a suitable representation in the first layer, the search space and the features of the problem are estimated and extracted in the second and third layers, respectively. Through taking advantage of the minimized search space from estimation and the heuristic information from extraction, the antibodies (or solutions) are evolved in the fourth layer and finally the fittest antibody is exported. In order to demonstrate the efficiency of the proposed system, the graph planarization problem is tested. Simulation results based on several benchmark instances show that the proposed algorithm performs better than traditional algorithms.

key words: artificial immune system, multi-layered, graph planarization, estimation, feature

1. Introduction

Extensive research has been done recently on applying the remarkable information processing performance of living organisms to computer systems. In particular, many studies have explored neural networks, which are related to the brain and nervous system. Nevertheless, the information processing performance of biological immune systems is extremely good, which is very significant in terms of future parallel distributed computing [1]. In recent years researchers have realized the potential revelation of biological immune mechanisms in developing new computational intelligence [2]. The computational models and applications based on artificial immune systems have established a research frontier. These include Immune Networks [3], [4], Negative Selection Algorithms [5], [6], and Clonal Selection Algorithms [7]. There have been attempts to apply these immune systems to the autonomous behavior of robots [8], pattern recognition [9], [10], noise removal [11], information security [12] and optimization problems [13]–[16].

In particular, the evolution-based immune optimization is a key area in the study of artificial immune systems. Motivated by the diversity of antibodies in the immune clonal system, Fukuda proposed multi-modal func-

tion optimization to overcome the shortcoming of Genetic Algorithm (GA), where GA easily gets into a local minimum [17]. For solving complex problems such as multi-peaks function optimization and combinatorial optimization, De Castro et al. [18] proposed the clonal selection algorithm. The algorithm tries to imitate the mechanisms in the clonal selection principle to better understand its natural processes and simulate its dynamical behavior such as the immune clone and affinity aberrance in the presence of antigens. Further developments of the clonal selection algorithm such as the introductions of receptor editing operator, chaotic dynamics, adaptive population size, expanded search space, immune memory and lateral interaction of different antibodies have been considered in the literature [19]–[23]. Based on the immune network theory [24], Yoo et al. put forward an immune network model to solve constrained optimization problems in which the objective function and the constrained function are combined linearly into a new function [25]. In addition, Coello presented a technique to deal with the constraints which adopted the binding of antigen and antibody, gene evolution and binary system code expression and can be used to cope with multi-objective optimization [26]. Inspired from the negative selection mechanism [27], [28], Cao et al. designed a model having the advantages of not only the maintenance of the diversity of antibodies but also the removal of poor genes [29]. The present studies mentioned above deal with the immunity mechanism behind the immune response (i.e. the adaptive immune response) and are focused on information processing based on various immune reactions caused by the immune response. In other words, these immune optimization models mainly make use of the antibody diversity and learning mechanisms during the immune response.

On the other hand, the immune system is a complex of cells, molecules and organs that has been proven to be capable of performing several tasks, like pattern recognition, learning, memory acquisition, generation of diversity, noise tolerance, generalization, distributed detection and optimization [30]. The powerful computation capability comes from the immune system's intrinsic multi-layered architecture [31]–[33]. Based on this consideration, the only usage of the immunity mechanisms behind the immune response leads to an incomplete and inefficient system design to some extent. As a result, new computational techniques can be developed, aiming not only at a better understanding of the whole immune system but also at solving engineering problems more efficiently.

Manuscript received April 6, 2009.

Manuscript revised July 21, 2009.

[†]The authors are with the Graduate School of Innovative Life Science, University of Toyama, Toyama-shi, 930–8555 Japan.

^{††}The author is with the Faculty of Engineering, University of Fukui, Fukui-shi, 910–8507 Japan.

^{†††}The author is with the Faculty of Engineering, University of Miyazaki, Miyazaki-shi, 889–2192 Japan.

a) E-mail: ztang@eng.u-toyama.ac.jp

DOI: 10.1587/transinf.E92.D.2498

In this paper, based on the immune system's intrinsic multi-layered architecture, we propose a new multi-layered immune system (MLIS) in which not only the adaptive immune response but also the physical and physiologic barriers, the innate immune response and the antigen presenting cells are taken into consideration. From an immunological standpoint, all these immunological components work together to protect us against various antigens such as bacteria and pathogens and each component has a unique function to resist the invaders. From an optimization perspective, all layers of the immune system lead to algorithms that improve candidate solutions to a given problem through a process of *estimation of search space*, *feature extraction*, and *evolution of antibodies*. In addition, it can be declared that the MLIS is immune-inspired because of its multi-layered architecture and uses several immune-related principles, such as the proliferation, clonal selection, and so on. In order to verify the applicability and efficiency of the proposed system, MLIS is used to solve the graph planarization problem as an example. Experimental results show that the proposed immune system provides better performance when compared to other traditional methodologies.

The rest of this paper is organized as follows: the proposed multi-layered immune system is presented in the next section. In Sect. 3, we validate our model by applying it to the graph planarization problem. Experimental results and discussions are presented in Sect. 4. Finally we give some general remarks to conclude this paper.

2. Multi-Layered Natural and Artificial Immune System

In order to make the paper self-explanatory, before actually proposing the multi-layered artificial immune system, the principles and mechanisms of the biological immune system are briefly explained in the following subsection.

2.1 Multi-Layered Natural Immune System

The immune system is a complex of cells, molecules and organs with the primary role of limiting damage to the host organism by pathogens, which elicit an immune response and thus called antigens (Ag). Immune system constitutes the defense mechanism of the body by means of physical barriers, physiologic barriers, innate immune response, and adaptive immune responses. Among these, adaptive immune response is more important for human being because it contains metaphors like recognition, memory acquisition, and etc. The main component of adaptive immune response is lymphocytes, which divide into two classes as T and B lymphocytes (cells), each having its own function. B cells have a great functionality because of their secreted antibodies (Ab) that take very critical roles in adaptive immune response. Figure 1 presents a simplified architecture of the basic immune mechanisms of defense.

(I) The first layer in the immune system is a diverse army of antigens. An antigen, which can cause diseases, is

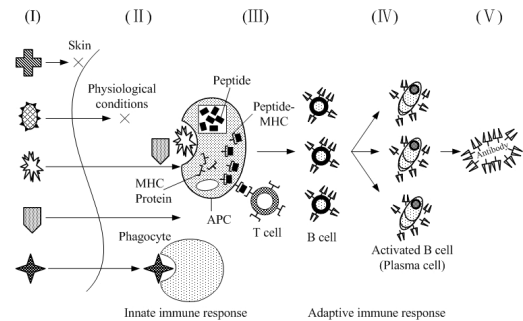


Fig. 1 Multi-layered immune system architecture.

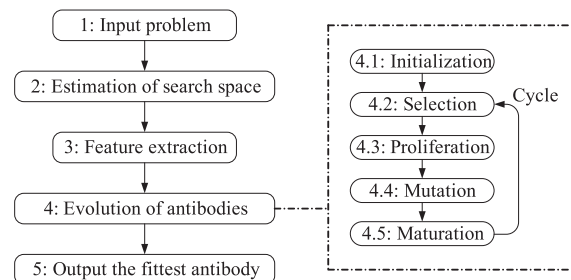


Fig. 2 The basic flow diagram of MLIS.

a foreign substance from the environment such as chemicals, viruses, pollen and so on. The ultimate target of all immune response is to prevent or eliminate the antigens. (II) The second layer is composed of three kinds of components involving the physical barriers such as the skin and mucous membranes, the physiologic barriers such as saliva, sweat and tears, and the innate immune response. Through the second layer, a certain portion of antigens are recognized and destroyed. (III) The remaining antigens are subjected to the specialized antigen presenting cells (APC) where those encountered antigens are ingested and fragmented into antigenic peptides. The pieces of these peptides are displayed on the cell surface by major histocompatibility complex (MHC) molecules existing in the digesting APC. The presented MHC-peptide combination on the cell surface is recognized by the T cells causing them to be activated. (IV) Activated T cells secrete some chemicals as alert signals to B cells in response to this recognition. Those stimulated B cells proliferate (divide) and eventually mature into terminal (non-dividing) antibody secreting cells, called plasma cells. This process is known as the adaptive immune response. (V) Finally, antibodies which are secreted on the surfaces of plasma cells bind the existing antigens and neutralize them signaling other components of immune system to destruct the antigen-antibody complex.

2.2 Multi-Layered Artificial Immune System (MLIS)

According to the natural immune mechanism (NIM) discussed in the previous subsection, we propose an multi-layered artificial immune system as illustrated in Fig. 2. Similar to NIM, MLIS is also composed of five layers. Ta-

Table 1 Relationship between the biological immune architecture and the corresponding functions in the artificial immune system.

Layer	Biological Immune Component	Corresponding Function
I	Antigen	Problem
II	Physical barriers Physiologic barriers Innate immune response	Estimation of search space
III	Antigen presenting cells	Feature extraction
IV	Adaptive immune response	Evolution of antibodies
V	Antibody	Solution

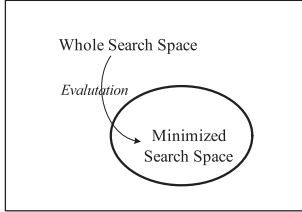


Fig. 3 The conceptual graph of the evaluation of search space.

ble 1 depicts the corresponding relationship between the biological immune system and the functions in the proposed model.

The first layer in MLIS is the problem input. In this layer the immune system encounters various antigens $AG = \{Ag_1, Ag_2, \dots, Ag_N\}$, i.e., the input optimization problem, where $|AG| = N$ is the size of search space and each antigen Ag_i ($i = 1, 2, \dots, N$) denotes a state in the search space. In order to solve the problem efficiently, the problem should be expressed in a suitable representation. Several components of the problem are analyzed, such as the objective of the problem, the constrain conditions, the search space and so on.

The key to find a good solution for an optimization problem lies in the estimation of search space $\Theta(AG)$. The estimation of search space indicates the analysis of topological structure or statistical state in the search space, for instance, the characteristics in the neighbourhood of the optimal solution. Several works have been done in the literature [34]. For the Traveling Salesman Problem (TSP), several techniques addressing this concern have been developed such as the “don’t look bits” technique [35]. For the Job Scheduling Problem (JSP), the search space are analyzed and divided into three groups involving semi-actives, actives and non-delay. Based on this observation, efficient algorithms have also been proposed [36], [37]. Obviously, the estimation is problem-oriented.

In this study, inspired from the function of the second layer in NIM that a certain portion of antigens are recognized and destroyed through this layer, we estimates the whole search space by using a minimized one to replace it, that is $\Theta(AG) = \{\dots, Ag_k, \dots, Ag_l, \dots\}$ ($k, l \in \{1, 2, \dots, N\}$) and $|\Theta(AG)| = M < N$. Figure 3 illustrates the conceptual graph of the basic idea. A minimized search space (MSS) indicates that not only fewer iterations of search are needed but the computational time are also reduced. However, there

is a main disadvantage of MSS that it can’t guarantee the existence of the global optimal solution within MSS. As a result, when solving problems using MLIS, MSS should balance the speed of solution acquirement with the quality of solution. In the next section, we will illustrate how to generate a minimized search space by solving the graph planarization problem as an example. Furthermore, it is worth pointing out that we use the sentence “Estimation of search space” rather than “Minimization of search space” in Table 1 is because that the conception of the former is broader, and this might lead us to divide the three components in the second layer more precise to conduct a more layered immune system. That might be a promising further research field.

The third layer in MLIS is the feature extraction. Before activating B cells to generate antibodies, the antigen is subjected to the APC layer in which some elements of antigen are recognized. Motivated by this, MLIS manages the optimization problem with extraction of features $\Gamma(AG)$ such as the basic and obvious characteristics or knowledge in the problem. These features assist MLIS with some heuristic information and therefore enable MLIS to find better solutions. For example, the distance between two cities in TSP and the processing time of each operation in JSP can be utilized as features [34], [38].

Then MLIS generates a variety of antibodies (solutions) for the problem. Each antibody is regarded as a candidate solution. Initially P antibodies $\{Ab_1, Ab_2, \dots, Ab_P\}$ are generated randomly or based on some heuristic mechanisms. Each antibody is evaluated by the antigen with a certain affinity, mathematically, the affinity function $A(\cdot)$ is calculated by $A(Ab_i) = E(AG, Ab_i)$ where the function $E(\cdot)$ is related with the objective of the problem. Then Q ($Q \leq P$) antibodies with higher affinity are selected as the elites and, without loss of generality, $\{Ab_1, Ab_2, \dots, Ab_Q\}$ are selected and arranged in an ascending order of the affinity. Subsequently, the selected antibodies will be stimulated to proliferate, creating a set of clones identical to the parent antibody, with a rate proportional to its fitness, i.e., the fitter the antibody, the more clones it will have. The amount of clones generated for these antibodies is given as $s_i = \text{round}((Q - i) * F/Q)$ where i is the ordinal number of the selected antibody, F is a multiplying factor which determines the scope of the clone and $\text{round}(\cdot)$ is the operator that rounds its argument towards the closest integer. Thereafter all the clones mutate by changing the genes within themselves. The hypermutation and receptor editing are

two main operators in the mutation, either of which is carried out on each antibody in 50% probability to realize a complementary role for improving the affinity of antibodies [13]. After mutation, the mutated antibodies with higher affinity replace their competitors (i.e. parent antibodies) and this process is known as the affinity maturation. The above procedures are iterated until meeting a pre-specified termination criterion.

Finally, the fittest antibody is exported as the final solution for the problem.

3. Application to the Graph Planarization Problem (GPP)

3.1 Introduction to GPP

A graph is said to be planar or embeddable in a plane if it can be drawn in the plane so that no two edges intersect except at a common endpoint. Given a n -vertex m -edge nonplanar graph $G = (V, E)$ with vertex set V and edge set E , the objective of graph planarization is to find a spanning planar subgraph with a maximum number of edges. The problem has applications in circuit board layout, facility layout, automatic graph drawing, VLSI circuit routing, and so on [39].

Since finding a maximum planar subgraph is NP-hard [40], the planarization problem has been widely studied for many years. Several graph planarization heuristics, neural network learning methods and genetic algorithms have been proposed in the literature. Based on the PQ-tree technique [41], a near-maximal planarity testing algorithm with computational complexity $O(n^2)$ was proposed by Jayakumar et al. [42]. Reference [43] presented a corrected and more generalized version of Jayakumar's algorithm. Two efficient algorithms [44], [45] with the same complexity bound of $O(m \log n)$ can be derived from the Hopcroft-Tarjan planarity testing algorithm [46] and the incremental planarity testing algorithm [45], respectively. Linear-time $O(m+n)$ algorithms can be considered in [47], [48]. A two-phrase graph planarization heuristic and a greedy randomized adaptive search procedure are presented in [49] and [50] respectively. Moreover, a few parallel algorithms have been proposed to solve the GPP. Using the neural network techniques, Takefuji and Lee presented a parallel planarization algorithm for generating a near-maximal planar subgraph within $O(1)$ time [51], [52]. A Hopfield network learning algorithm with a gradient ascent technique for the problem is proposed in [53]. Furthermore, the applicability of GAs in solving GPP has been verified in [54]. An effective genetic algorithm performing crossover and mutation operators conditionally instead of probability can be also referred to as in [55].

3.2 MLIS for Solving GPP

In this section, MLIS is used for solving GPP. The general steps and interpretations of the algorithm are illustrated in the following.

3.2.1 The 1st Layer: Representation of GPP

The objective of GPP is to find a maximum planar subgraph from a general nonplanar graph. Consider the simple undirected graph composed of four vertices and six edges as shown in Fig. 4 (a). The graph is planar as long as two edges, $e_5 = (v_1, v_3)$ and $e_6 = (v_2, v_4)$, do not cross each other. Figure 4 (b) shows a planar graph. In this study, we adopt the single-row routing [51] to represent GPP. First, the vertices in the graph are placed on a line and then the connection is established by either an upper edge or a lower edge. Figure 4 (c) shows a possible planar graph based on the single-row routing representation. The two-edge-crossing violation condition can be easily determined from the single-row representation used. The existence of a crossing between two upper edges (v_i, v_j) and (v_p, v_q) (or two lower edges) is determined by the following conditions as shown in Fig. 4 (d):

$$\text{if } v_i < v_p < v_j < v_q \text{ or } v_p < v_i < v_q < v_j \quad (1)$$

In this representation, the search space is $|G| = n! \cdot 3^m$. Here, $n!$ denotes the total number of possible permutations of the vertex sequence. Each of the edge has three possible states according to whether, in accordance with a determined vertex sequence the edge is a lower edge, is not considered or is an upper edge. Therefore, the total number of possible combinations of the edges is 3^m .

3.2.2 The 2nd Layer: MSS for GPP

In order to minimize the search space of GPP, the *Hamiltonian cycle* based vertex sequence is considered. As noticed by Goldschmidt [49], the vertex sequence influences strongly the size of the planar subgraph which can be drawn and Hamiltonian cycle based vertex sequence does actually produce good solutions. As a result, in this study only P different Hamiltonian cycle based vertex sequences (abbreviated as HVS) are generated for solutions and the search

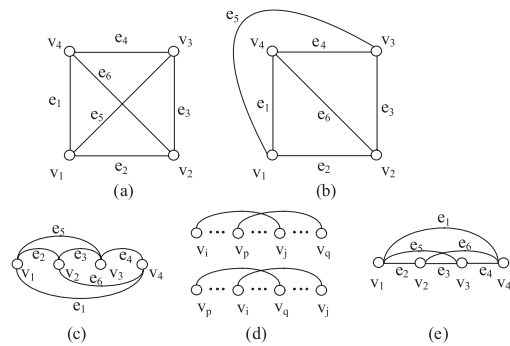


Fig. 4 (a) A graph with four vertices and six edges. (b) A planar graph. (c) A possible planar graph based on the single-row routing representation. (d) Violation conditions in the single-row routing representation. (e) The feature of an input graph for each edge: its crossing-number φ . $\varphi(e_1) = \varphi(e_2) = \varphi(e_3) = \varphi(e_4) = 0$, $\varphi(e_5) = \varphi(e_6) = 1$.

space is therefore reduced to $|\Theta(G)| = P \cdot 3^m (\ll |G|)$. It should be noted that the proposed method is an approximate algorithm for getting near-optimal solutions. Although it can't guarantee to find the optimal solution, near-optimal solutions can be easily found by using much less computational times and, on the other hand, it is well-known that near-optimal but less-cost solutions also have engineering-importance.

The way of generating the Hamiltonian cycle based vertex sequence π is shown as follows. The first vertex in the sequence is $\pi(1) = v_\beta$ where v_β is a randomly selected vertex in the graph G . Different vertex sequence can be generated by selecting different β . After the first k ($k = 1, 2, \dots, n-1$) vertices of the sequence have been determined, say $\pi(1), \pi(2), \dots, \pi(k)$, the next vertex $\pi(k+1)$ is selected from the vertices adjacent to $\pi(k)$ in G having the least adjacencies in the subgraph G_k of G induced by $V \setminus \{\pi(1), \pi(2), \dots, \pi(k)\}$. If $\pi(k)$ has no neighbors in G_k , we select $\pi(k+1)$ as a vertex of minimum degree in G_k . Correspondingly, we also define an inverse function $\pi^{-1}(\cdot)$, having $\pi^{-1}(v_s) = r$ when the vertex v_s is located on the r -th element of π . Then, we place the vertices of G on a line in accordance with the determined vertex sequence π . In this condition, the violation conditions of Eq. (1) in the single-row routing representation should be expressed as in Eq. (2). Let (v_i, v_j) and (v_p, v_q) be two upper (or lower) edges of the input graph G , and let us assume without loss of generality that $\pi^{-1}(v_i) < \pi^{-1}(v_j)$ and $\pi^{-1}(v_p) < \pi^{-1}(v_q)$. Edge (v_i, v_j) intersects with edge (v_p, v_q)

$$\begin{aligned} & \text{if } \pi^{-1}(v_i) < \pi^{-1}(v_p) < \pi^{-1}(v_j) < \pi^{-1}(v_q) \\ & \text{or } \pi^{-1}(v_p) < \pi^{-1}(v_i) < \pi^{-1}(v_q) < \pi^{-1}(v_j) \end{aligned} \quad (2)$$

3.2.3 The 3rd Layer: Crossing-Number Extraction

In GPP, the crossing-number is an obvious feature of the edge and it is used as $\Gamma(G)$ in this paper. The crossing-number of edge e_i (expressed as $\varphi(e_i)$) is calculated by placing all edges above the vertex's line and then counting the number of edges that cross with e_i according to Eq. (2). Figure 4 (e) illustrates an example for the six edges in the graph.

The reason that we choose the crossing-number as the $\Gamma(G)$ is based on the fact that the edge with smaller crossing-number has higher probability of embedding into the planar graph as a solution component and vice-versa. Hereby the crossing-number can be utilized as heuristic information or guided knowledge to construct solutions.

3.2.4 The 4th Layer: Solution Evolution

For solving GPP, a suitable antibody representation is important. In this paper, we propose a permutation encoding with greedy decoding mechanism for each antibody. An antibody is expressed as an ordering (a_1, a_2, \dots, a_m) of all m edges in the input graph, for instance, $(e_2, e_4, e_6, e_1, e_5, e_3)$ is an antibody in Fig. 4. A greedy decoding strategy as shown in Fig. 5 is utilized to manipulate the antibody. The number of edges in the output planar subgraph is regarded as the

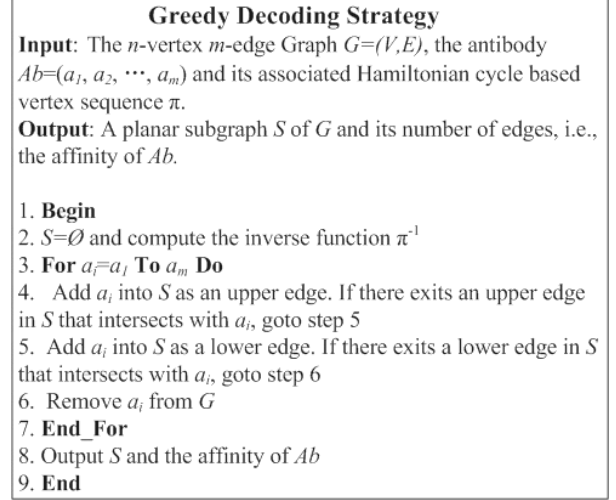


Fig. 5 The greedy decoding strategy.

affinity of the antibody. The characteristics of the strategy is that it can maintain the feasibility of the antibody. That is to say, the constraints of the problem are dealt with during the encoding-decoding procedure. As a result, some traditional but efficient operators which can manage the permutation individuals can be easily and naturally incorporated into the algorithm. In this paper, we use an improved simple point-mutation and a single point asexual crossover method [56] to act as the hypermutation and receptor editing respectively. The improved simple point-mutation operator which makes use of the crossing-number of edges is illustrated as follows:

$$\begin{aligned} & a_1 \rightarrow a_2 \dots \rightarrow a_i \rightarrow a_{i+1} \dots \rightarrow a_j \rightarrow a_{j+1} \dots \rightarrow a_m \\ & \quad \quad \quad \downarrow \\ & a_1 \rightarrow a_2 \dots \rightarrow a_i \rightarrow a_j \rightarrow a_{i+1} \dots \rightarrow a_{j+1} \dots \rightarrow r_m \end{aligned}$$

After mutation, the edge a_j is inserted behind the edge a_i while the others remains the same. Furthermore, a_i is randomly selected while a_j ($j > i$) is selected based on the following probability rule:

$$p_j = \frac{\varphi_{\max} - \varphi(a_j) + c}{\sum_{k=i+1}^m (\varphi_{\max} - \varphi(a_k) + c)} \quad (3)$$

Here $\varphi_{\max} = \max\{\varphi(a_k)\}$, $k = i+1, \dots, m$ indicates the maximum crossing-number during all the edges behind a_i . c is a constant in order to avoid producing zero probability and usually we set $c = 1$. According to this rule, the edge with smaller crossing-number has a higher probability to be selected and therefore an earlier appearance in the antibody's permutation. After decoding, it will have a better chance of embedding into the planar graph.

Moreover, the termination criterion is fulfilled until the iteration number reaches M_k .

4. Experiments and Discussions

In this section, we verified the proposed multi-layered im-

mune system (MLIS) through applying it to a total of 19 benchmark instances of the graph planarization problem (GPP). These instances consisted of graphs ($G_1 \sim G_{12}$) arising from various areas of applications such as coding theory and geometry as well as random graphs ($G_{13} \sim G_{19}$) whose existing edge between any pair of vertices were independently generated with a certain probability.

Table 2 lists the associated parameters adopted in the MLIS for GPP. These parameters were determined through numerical experiments after multiple simulation runs. In addition, it should be noted that the values of P , Q and F were set as relative small values ($P = 20$, $Q = 10$, and $F = 5$) rather than large values ($P = 150$, $Q = 100$, and $F = 50$) as suggested in the clonal-selection based algorithms [13], [19]. These parameters strongly influence the computational cost of the algorithm. To illustrate the trade-off between performance and computational cost obtained with different parameter settings (i.e. using small values or large values for P , Q and F), we used the example studied in this paper, G_1 , to make a comparison. The average computational time was increased from 1.359 to 11.906 seconds, a drastic lost in computational cost, while the average performance remained the same. On the other hand, too small values (such as $P = 2$, $Q = 1$, and $F = 2$) for these parameters made the diversity of population too quickly lost and thus can not find satisfying solutions. Furthermore, similar phenomena can be found when larger scale problems are tested, suggesting that relative small values for P , Q and F are appropriate for MLIS. As for the parameter M_k , the same value used in [13], [19] was adopted.

4.1 Effect of the “Estimation of Search Space” Layer

As we declared that the powerful searching ability of MLIS should come from its intrinsic multi-layered architecture, we analyzed the effects of the “Estimation of Search Space” layer and “Feature Extraction” layer respectively.

In order to demonstrate the effect of the “estimation of search space” layer to the proposed algorithm, we compared MLIS with its variants involving “MLIS/WSS”,

“MLIS/RVS”, and “MLIS/FVS”. The difference during them was the method of generating candidate vertex sequences.

As described in Sect. 3.2.2, MILS used a heuristic method that was based on the Hamiltonian cycle to generate P vertex sequences, and these sequences were expected to be promising candidates. “MLIS/WSS” used an exhaustive method. It generated all $n!$ possible permutations of the vertex sequence, and then conducted an antibody for each vertex sequence to optimize the edge order in the affinity evolution process. A point worth emphasizing was that “MLIS/WSS” used $n!$ rather than P antibodies in the population, thus needing vast computational time even to implement a search iteration. Without a doubt, the termination criterion of “MLIS/WSS” can not be fulfilled within reasonable times. “MLIS/RVS” used a stochastic method to generate P vertex sequences, while “MLIS/FVS” only generated a fixed vertex sequence which was the same as the sequence of the input vertices [51]. By doing so, we can find out that whether the minimized search space (MSS) by using multiple heuristically generated vertex sequences (HVS) was better than that by using multiple randomly generated vertex sequence (RVS), that by using single vertex sequence (FVS), or the whole search space (WSS).

The four algorithms mentioned above were run 20 times to make a statistical comparison. The simulation results based on four typical instances involving G_3 , G_8 , G_{13} and G_{16} were shown in Table 3. The results that we recorded for each graph were the average size of the planar subgraphs (Ave.), the best size of the planar subgraphs (Best), and the computational times (Time(s)) produced by each algorithm. It can be observed from Table 3 that the proposed algorithm generated the best solutions than the other variants. This can be attributed to the fact that the MSS generated by HVS enabled the system to carry out searching around an attractive area (i.e., the solutions generated based on HVS were better than those generated based on RVS and FVS). This also confirmed the statement in [49] that the vertex sequence actually influenced strongly the size of the final planar subgraph that can be drawn by the algorithm. In Fig. 6, we illustrated an example based on the nonplanar graph G_3 . The results also showed that HVS was a better choice for generating vertex sequences of a graph. In addition, “MLIS/WSS” can not acquire any solution within reasonable computational times since the termination criterion can not be fulfilled. As a result, it can be concluded that it was necessary and useful to apply the estimation of search space in the algorithm.

Table 2 Parameters used in the experiments.

Initial population number	$P=20$
Elite population number	$Q=10$
Expectant clone size	$F=5$
Probability rule constant	$c=1$
Maximum iteration number	$M_k=1000$

Table 3 Effect of the minimized search space which was composed of Hamiltonian cycle based vertex sequences.

	G_3			G_8			G_{13}			G_{16}		
	Best	Ave.	Time(s)	Best	Ave.	Time(s)	Best	Ave.	Time(s)	Best	Ave.	Time(s)
MLIS/WSS	—	—	—	—	—	—	—	—	—	—	—	—
MLIS/RVS	23	21.5	1.675	56	48.8	11.542	103	87.5	138.269	94	86.9	223.541
MLIS/FVS	22	22	1.663	61	58.3	11.513	84	82.5	137.423	101	98.6	222.368
MLIS	24	24	1.688	69	67.5	11.687	131	125.3	138.188	192	183.5	224.437

Note: The symbol “—” means the termination criterion can not be fulfilled within reasonable computational times.

4.2 Effect of the “Feature Extraction” Layer

On the other hand, we demonstrated the effect of the feature extraction (FE) in the algorithm. In Table 4, we compared the simulation results between the algorithms with and without FE. Without utilizing the information of the crossing-number of each edge in the graph, the algorithm without FE made use of a simplified version of the hypermutation operator, that is both of the two mutated edges were randomly selected and then exchanged the positions between them. From Table 4, it can be found that the algorithm with FE can produce better solutions than that without FE. That is to say, by using some basic features of the problem the search performance of the algorithm can be improved. As a result, the feature extraction layer was also necessary and useful for constructing an efficient algorithm.

Furthermore, as the computational time (or computational complexity) was another important metric for evaluating algorithms, we depicted the comparative results during the three layers in Fig. 7, where the horizontal axis in a linear scale denotes the tested instances and the vertical axis in a logarithmic scale represents the computational times. From Fig. 7, we can observe that running times for the estimation of search space layer and the feature extraction layer were negligible compared to the times taken by the evolution of antibodies layer. Therefore, we can say that the multi-layered architecture of the immune system (incorporating two layers before the immune response) resulted in a better searching ability but a non-increasing computational complexity of the algorithm.

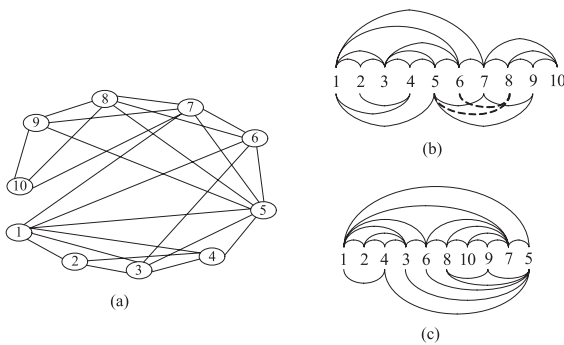


Fig. 6 (a) A nonplanar graph (G_3) composed of 10 vertices and 24 edges. (b) The maximal planar subgraph of G_3 found by FVS. The dashed lines indicated the removal from the original graph. (c) The maximal planar subgraph of G_3 found by HVS.

4.3 Comparison with Other Algorithms

To assess the search performance of the proposed MLIS, we used four other traditional algorithms to make a comparison. These algorithms included two neural network based algorithms (T-L [51] and HNL [53]), a meta-heuristic algorithm (UT [49]) and an evolutionary algorithm (IGA [55]). The algorithm T-L which was proposed by Takefuji/Lee has been claimed to be one of the most popular parallel algorithm and to have superior performance to that of previously published algorithms [52]. Aiming at solving T-L's inherent local minimum problem, HNL which used a gradient ascent learning method was proposed and showed better performance than T-L. UT was a two-phase heuristic for effectively solving GPP and the basic idea of how to generate a minimized search space for GPP in MLIS was directly inspired from it. IGA was an efficient generic algorithm, performing crossover and mutation conditionally instead of probability.

The comparative results of the five algorithms were summarized in Table 5. The results recorded for each graph (from G_1 to G_{19}) were the number of vertices, the number of edges, the upper bound on the optimal solution (i.e., $3n - 6$), the size of the planar subgraphs produced by each algorithm and the corresponding computational times. Here, the results of all algorithms were based on 20 runs. For all tested 19 graphs with an exception of the instance G_{12} , the best solutions (even the average solutions) produced by the proposed algorithm were much better than those obtained by

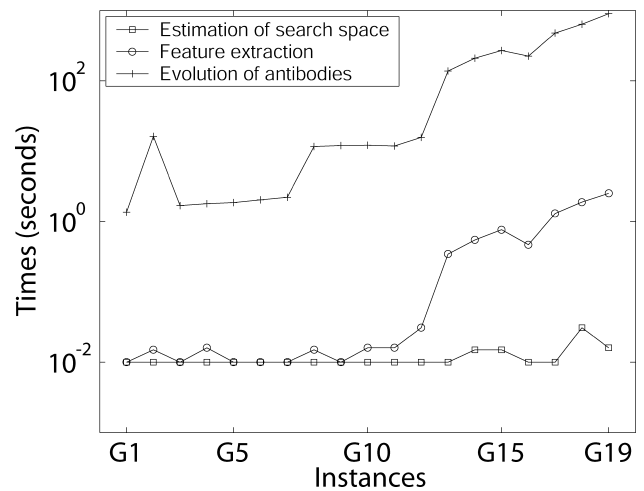


Fig. 7 Average computational time as a function of the input instances of the graph: estimation of search space, feature extraction, and evolution of antibodies.

Table 4 Effect of the feature extraction.

	G_3			G_8			G_{13}			G_{16}		
	Best	Ave.	Time(s)	Best	Ave.	Time(s)	Best	Ave.	Time(s)	Best	Ave.	Time(s)
MLIS without FE	24	24	1.265	69	66.6	10.859	125	117.6	125.132	178	164.3	190.874
MLIS with FE	24	24	1.688	69	67.5	11.687	131	125.3	138.188	192	183.5	224.437

Table 5 Simulation results during five algorithms.

Graph	Vertices	Edges	Upper Bound	T-L	HNL		UT		IGA		Proposed		
				Best	Best	T(s)	Best	Ave.	Best	T(s)	Best	Ave.	T(s)
G1	10	22	20*	20	20	0.3	20	19.00	20	18.66	20	20	1.359
G2	45	85	82*	80	80	1.09	82	81.45	80	46.54	82	80.8	16.281
G3	10	24	24*	21	22	0.24	24	22.95	22	20.52	24	24	1.688
G4	10	25	24*	22	22	0.12	24	23.20	22	30.56	24	24	1.813
G5	10	26	24*	22	22	0.16	24	22.45	22	23.51	24	24	1.86
G6	10	27	24*	22	22	0.26	24	23.00	22	21.33	24	24	2.031
G7	10	34	24*	23	23	0.40	24	23.55	23	19.56	24	24	2.215
G8	25	69	69*	58	61	0.81	68	65.15	61	36.25	69	67.5	11.687
G9	25	70	69*	59	61	1.15	69	65.85	61	29.27	69	67.3	12.016
G10	25	71	69*	58	61	0.28	68	64.00	61	38.11	69	66.8	12.156
G11	25	72	69*	60	61	0.30	68	62.65	61	42.36	68	65.8	11.875
G12	25	90	69*	61	63	0.32	67	60.65	63	35.26	65	63.5	15.75
G13	50	367	144	70	82	82.17	129	124.55	84	261.25	131	125.3	138.188
G14	50	491	144	100	109	47.71	138	135.30	114	325.31	143	137.6	209.797
G15	50	582	144	101	115	107.43	142	140.20	119	445.36	142	138.1	270.281
G16	100	451	294	92	100	122.62	183	178.15	101	398.42	192	183.5	224.437
G17	100	742	294	116	126	246.98	215	209.20	127	452.01	225	218.8	478.01
G18	100	922	294	115	135	504.4	234	227.00	138	556.54	237	229.4	639.25
G19	150	1064	444	127	138	680.07	291	284.45	145	701.36	311	288.5	901.625

* Actual known optimal size of the planar subgraph.

The best solution during all algorithms were shown in bold fonts.

the other algorithms. Compared with the neural network based algorithms (T-L and HNL), MLIS can find better solutions while costing more computational times. Although the termination conditions of T-L, HNL and MLIS were different, the comparison can be said to be sufficient and fair since the results of T-L and HNL had trapped into local minima and can not be improved any further. The evident can be found in [53] that the learning epochs of HNL were always smaller than the pre-defined maximal number of learning times. Besides, the extra time of MLIS was clearly compensated by the quality of the solutions produced. Compared with UT and IGA, MLIS acquired better performances in terms of the solution quality and computational times. Consequently, we can say that the proposed methodology was efficient for solving the graph planarization problem within reasonable computational times.

5. Conclusions

In this paper, we proposed a multi-layered immune system for solving combinational optimization problems. Inspired from the intrinsic multi-layered architecture of the biological immune system, the proposed methodology is composed of five layers involving the problem expression, the estimation of search space, the feature extraction, the evolution of antibodies and the output. All five layers work together to constitute a powerful computational tool. In order to show the efficiency of the proposed methodology, the graph planarization problem was tested. Simulation results exposed the effects of the estimation of search space and the feature extraction layers and further the effectiveness of the proposed method when compared with four other traditional algorithms.

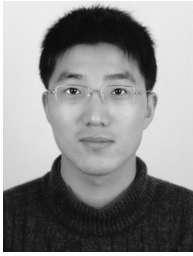
Furthermore, the proposed MLIS is similar to mutation-based evolutionary algorithms (such as GA) and

has good features for optimization and searching. These features include: (1) exploitation and exploration of the search space; (2) capability of allocating multiple optima and maintaining local optimal solutions, thus especially effective for highly multimodal problems. However, there are some important differences between MLIS and the typical evolutionary algorithms. The MLIS is capable of incorporating traditional techniques (such as estimation of search space and feature extraction) into the algorithm to improve its search performance. In addition, the MLIS performs not only proportionate proliferation, but also a combination of mutation operators involving hypermutation and receptor editing, and there is no crossover operation. As a result, the effectiveness of MLIS for GPP is due to its characteristics mentioned above. In order to further verify the effectiveness and robustness of the proposed algorithm, in the future, we plan to apply the MLIS to solve other optimization problems, such as the job shop scheduling problems (JSP), the traveling salesman problems (TSP) and the quadratic assignment problems (QAP).

References

- [1] C.G. Langton, ed. *Artificial Life*, MIT Press, 1995.
- [2] D. Dasgupta, An overview of artificial immune systems and their applications, pp.3–21, Springer-Verlag, 1999.
- [3] Y. Ishida, "The immune system as a prototype of autonomous decentralized systems: An overview," *Proc. International Symposium on Autonomous Decentralized Systems*, pp.85–92, April 1997.
- [4] L. de Castro and F.V. Zuben, "An evolutionary immune network for data clustering," *Proc. IEEE SBRN*, pp.84–89, Nov. 2000.
- [5] S. Forrest, A. Perelson, L. Allen, and R. Cherukuri, "Self-nonspecific discrimination in a computer," *IEEE Symposium on Research in Security and Privacy*, pp.202–212, 1994.
- [6] M.K. Urszula and B. Kordas, "Multi-class iteratively refined negative selection classifier," *Applied Soft Computing*, vol.8, no.2, pp.972–984, March 2008.
- [7] L. de Castro and F.J.V. Zuben, "Learning and optimization using

- clonal selection principle," *IEEE Trans. Evol. Comput.*, vol.6, no.3, pp.239–251, 2002.
- [8] Y. Ishida, "An immune network approach to sensor-based diagnosis by self-organization," *Complex Systems*, vol.10, pp.73–90, 1996.
 - [9] D. Dasgupta, Y. Cao, and C. Yang, "An immunogenetic approach to spectra recognition," *Proc. GECCO'99*, pp.149–155, 1999.
 - [10] W. Sun, Z. Tang, H. Tamura, and M. Ishii, "An artificial immune system architecture and its applications," *IEICE Trans. Fundamentals*, vol.E86-A, no.7, pp.1858–1868, July 2003.
 - [11] Y. Ishida and N. Adachi, "Active noise control by an immune algorithm: Adoption in immune system as a convolution," *Proc. ICEC*, pp.150–153, 1996.
 - [12] D. Dasgupta, "Immuno-inspired autonomic system for cyber defense," *Tech. Rep. 12/4, Information Security Technical Report*, Elsevier Ltd., 2007.
 - [13] S.C. Gao, H.W. Dai, G. Yang, and Z. Tang, "A novel clonal selection algorithm and its application to traveling salesman problems," *IEICE Trans. Fundamentals*, vol.E90-A, no.10, pp.2318–2325, Oct. 2007.
 - [14] W.J. Shyr and T.J. Su, "Optimal design using clonal selection algorithm," *Knowledge-Based Intelligent Information and Engineering Systems*, pp.604–610, 2005.
 - [15] A. Doyen, O. Engin, and C. Ozkan, "A new artificial immune system approach to solve permutation flow shop scheduling problems," *Turkish Symposium on Artificial Immune Systems and Neural Networks TAINN'03*, pp.1–3, 2003.
 - [16] O. Alonso, F.A. Gonzalez, F. Nino, and J. Galeano, "Search and optimization: A solution concept for artificial immune networks: A coevolutionary perspective," *Proc. 6th International Conference on Artificial Immune Systems*, Santors, Brazil, 2007.
 - [17] M. Fukuda, *Parallel search for multi-modal function optimization with diversity and learning of immune algorithm*, ch. 11, Springer-Verlag, 1999.
 - [18] L. de Castro and F.J.V. Zuben, eds., *Recent Developments in Biologically Inspired Computing*, Idea Group Incorporation, 2004.
 - [19] S.C. Gao, Z. Tang, H. Dai, and J. Zhang, "An improved clonal algorithm and its application to traveling salesman problems," *IEICE Trans. Fundamentals*, vol.E90-A, no.12, pp.2930–2938, Dec. 2007.
 - [20] H. Meng, X. Zhang, and S. Liu, "A novel clonal selection for multi-model function optimization," *Advances in Natural Computation*, pp.63–72, 2006.
 - [21] Y. Sun, M. Gong, L. Hao, and L. Jiao, "Clonal selection algorithm with search space expansion scheme for global function optimization," *Advances in Natural Computation*, pp.838–847, 2006.
 - [22] N. Khilwani, A. Prakash, R. Shankar, and M.K. Tiwari, "Fast clonal algorithm," *Engineering Applications of Artificial Intelligence*, vol.21, pp.106–128, 2008.
 - [23] S.C. Gao, H. Dai, J. Zhang, and Z. Tang, "An expanded lateral interactive clonal selection algorithm and its application," *IEICE Trans. Fundamentals*, vol.E91-A, no.8, pp.2223–2231, Aug. 2008.
 - [24] N.K. Jerne, "Towards a network theory of the immune system," *Ann. Immunol.*, vol.125C, pp.373–389, 1974.
 - [25] J. Yoo and P. Hajela, "Immune network modeling in design optimization," *Structural and Multidisciplinary Optimization*, vol.18, pp.85–94, 1999.
 - [26] C.A. Coello and N.C. Cores, "A parallel implementation of the artificial immune system to handle constraints in genetic algorithms: preliminary results," *Proc. 2002 Congress on Evolutionary Computation*, Honolulu, 2002.
 - [27] G.J.V. Nossal, "Negative selection of lymphocytes," *Cell*, vol.76, pp.229–239, 1994.
 - [28] M. Ayara, J. Timmis, L.N. de Lemos, R. de Castro, and R. Duncan, "Negative selection: How to generate detectors," *Proc. 1st International Conference on Artificial Immune Systems (ICARIS)*, pp.89–98, 2002.
 - [29] X. Cao, H. Qiao, and Y. Xu, "Negative selection based immune optimization," *Advances in Engineering Software*, vol.38, no.10, pp.649–656, 2007.
 - [30] L. de Castro and F.V. Zuben, "Artificial immune systems: Part 1—basic theory and application," *Tech. Rep. TR-DCA 01/99*, 1999.
 - [31] B. Rensberger, In *Self-Defense*, pp.212–228, Oxford University Press, New York, 1996.
 - [32] W. Sun, Z. Tang, H. Tamura, and M. Ishii, "An artificial immune network with multi-layered b cells architecture," *IEEJ Trans. EIS*, vol.C123, no.11, pp.2036–2045, 2003.
 - [33] C.A. Janeway Jr and P. Travers, *Immunobiology. The Immune System in Health and Disease*, Artes Medicas, 1997.
 - [34] D.S. Johnson and L.A. McGeoch, *The Traveling Salesman Problem: A Case Study in Local Optimization*, pp.215–310, John Wiley & Sons, 1997.
 - [35] J.L. Bentley, "Fast algorithms for geometric traveling salesman problems," *ORSA J. Comput.*, vol.4, pp.387–411, 1992.
 - [36] B.J. Park, H.R. Choi, and H.S. Kim, "A hybrid genetic algorithm for the job shop scheduling problems," *Computers & Ind. Eng.*, vol.45, pp.597–613, 2003.
 - [37] J.F. Gonçalves, J.J.M. Mendes, and M.G.C. Resende, "A hybrid genetic algorithm for the job shop scheduling problem," *Eur. J. Oper. Res.*, vol.167, pp.77–95, 2005.
 - [38] R. Haupt, "A survey of priority rule-based scheduling," *OR Spektrum*, vol.11, pp.3–16, 1989.
 - [39] T. Nishizeki and N. Chiba, *Planar Graphs: Theory and Algorithms*, North Holland, 1988.
 - [40] P. Liu and R. Geldmacher, "On the deletion of nonplanar edges of a graph," *Proc. 10th SE Conference on Combinatorics, Graph Theory and Computing*, pp.727–738, Boca Raton, FL, 1977.
 - [41] K. Booth and G. Lueker, "Testing for the consecutive ones property, interval graphs, and graph planarity using pq-tree algorithm," *J. Comput. Syst. Sci.*, vol.13, pp.335–379, 1976.
 - [42] R. Jayakumar, K. Thulasiraman, and M.N.S. Swamy, " $O(n^2)$ algorithms for graph planarization," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol.8, no.3, pp.257–267, 1989.
 - [43] G. Kant, "An $o(n^2)$ maximal planarization algorithm based on pq-tree," *Tech. Rep. RUU-CS-92-03*, Dept. of Computer Science, Utrecht University, Utrecht, The Netherlands, 1992.
 - [44] J. Cai, X. Han, and R.E. Tarjan, "An $o(m \log n)$ -time algorithm for maximal planar subgraph problem," *SIAM J. Comput.*, vol.22, pp.1142–1162, 1993.
 - [45] G.D. Battista and R. Tamassia, "Incremental planarity testing," *Proc. IEEE Symp. Foundation of Computer Science*, pp.436–441, 1989.
 - [46] J. Hopcroft and R.E. Tarjan, "Efficient planarity testing," *J. ACM*, vol.21, pp.549–568, 1974.
 - [47] H. Djidjev, "A linear algorithm for the maximal planar subgraph problem," *Lect. Notes Comput. Sci.*, vol.955, pp.369–380, 1995.
 - [48] W.L. Hsu, "A linear time algorithm for finding maximal planar subgraphs," *Lect. Notes Comput. Sci.*, vol.1004, pp.352–362, 1995.
 - [49] O. Goldschmidt and A. Takvorian, "An efficient graph planarization two-phase heuristics," *Networks*, vol.24, pp.69–73, 1994.
 - [50] M.G.C. Resende and C.C. Ribeiro, "A GRASP for graph planarization," *Networks*, vol.29, pp.173–189, 1997.
 - [51] Y. Takefuji and K.C. Lee, "A near-optimum parallel planarization algorithm," *Science*, vol.245, no.4923, pp.1221–1223, 1989.
 - [52] Y. Takefuji, K.C. Lee, and Y.B. Cho, "Comments on 'o(n^2) algorithm for graph planarization'," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol.10, no.12, pp.1582–1583, 1991.
 - [53] R.L. Wang, Z. Tang, and Q.P. Cao, "An efficient parallel algorithm for planarization problem," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol.49, no.3, pp.397–401, 2002.
 - [54] F. Comellas, *Using genetic algorithms for planarization problems*, pp.93–100, Elsevier Science Publishers B.V., North-Holland, 1992.
 - [55] R.L. Wang and K. Okazaki, "Solving the graph planarization problem using an improved genetic algorithm," *IEICE Trans. Fundamentals*, vol.E89-A, no.5, pp.1507–1512, May 2006.
 - [56] S. Chatterjee, C. Carrera, and L.A. Lynch, "Genetic algorithms and traveling salesman problems," *Eur. J. Oper. Res.*, vol.93, pp.490–510, 1996.



Shangce Gao received a B.S. degree from Southeast University, Nanjing, China in 2005 and an M.S. degree from University of Toyama, Toyama, Japan in 2008. Now, he is working toward the D.E. degree at University of Toyama, Toyama, Japan. His main research interests are multiple-valued logic, artificial immune system and artificial neural networks.



Rong-Long Wang received a B.S. degree from Hangzhe teacher's college, Zhejiang, China and an M.S. degree from Liaoning University, Liaoning, China in 1987 and 1990, respectively. He received his D.E. degree from University of Toyama, Toyama, Japan in 2003. From 1990 to 1998, he was an Instructor in Benxi University, Liaoning, China. In 2003, he joined University of Fukui, Fukui, Japan, where he is currently an associate professor in Department of Electrical and Electronics Engineering.

His current research interests include genetic algorithm, neural networks, and optimization problems.



Hiroki Tamura received the B.E. and M.E. degrees from Miyazaki University in 1998 and 2000, respectively. From 2000 to 2001, he was an Engineer in Asahi Kasei Corporation, Japan. In 2001, he joined University of Toyama, Toyama, Japan, where he was currently a Technical Official in Department of Intellectual Information Systems. In 2006, he joined Miyazaki University, Miyazaki, Japan, where he is currently an Assistant Professor in the Department of Electrical & Electronic Engineering.

His main research interests are neural networks and optimization problems.



Zheng Tang received the B.S. degree from Zhejiang University, Zhejiang, China in 1982 and an M.S. degree and a D.E. degree from Tshinghua University, Beijing, China in 1984 and 1988, respectively. From 1988 to 1989, he was an Instructor in the Institute of Microelectronics at Tshinhua University. From 1990 to 1999, he was an associate professor in the Department of Electrical and Electronic Engineering, Miyazaki University, Miyazaki, Japan. In 2000, he joined University of Toyama, Toyama,

Japan, where he is currently a professor in the Department of Intellectual Information Systems. His current research interests include intellectual information technology, neural networks, and optimizations.