A Novel Immune Clonal Algorithm for MO Problems
Ronghua Shang, Member, IEEE, Licheong Jiao, Senior Member, IEEE, Fang Liu, Senior Member, IEEE, and Wenping Ma, Member, IEEE

Abstract—Research on multiobjective optimization (MO) becomes one of the hot points of intelligent computation. Compared with evolutionary algorithm, the artificial immune system used for solving MO problems (MOPs) has showed many good performances in improving the convergence speed and maintaining the diversity of the antibody population. However, the simple clonal selection computation has some difficulties in handling some more complex MOPs. In this paper, the simple clonal selection strategy is improved and a novel immune clonal algorithm (NICA) is proposed. The improvements in NICA are mainly focus on four aspects.

1) Antibodies in the antibody population are divided into dominated ones and non-dominated ones, which is suitable for the characteristic of one multiobjective optimization problem has a series Pareto-optimal solutions.

2) The entire cloning is adopted instead of different antibodies having different clonal rate.

3) The clonal selection is based on the Pareto-dominance and one antibody is selected or not depending on whether it is a non-dominated one, which is different from the traditional clonal selection manner.

4) The antibody population updating operation after the clonal selection is adopted, which makes antibody population under a certain size and guarantees the convergence of the algorithm.

The influences of the main parameters are analyzed empirically. Compared with the existed algorithms, simulation results on MOPs and constrained MOPs show that NICA in most problems is able to find much better spread of solutions and better convergence near the true Pareto-optimal front.

Index Terms—Artificial immune system (AIS), multiobjective optimization (MO), pareto-optimal front, performance metric.

Manuscript received January 14, 2009; revised June 4, 2009, and October 23, 2009; accepted January 11, 2010. Date of publication December 15, 2011; date of current version January 31, 2012. This work was partially supported by the National Basic Research Program 973 of China, under Grant 2006CB705707, the National High Technology Research and Development Program 863 of China, under Grants 2008AA01Z125 and 2009AA12Z210, the National Natural Science Foundation of China, under Grants 60703107, 60703108, 60803106, 60803098, and 60804021, the National Research Foundation for the Doctoral Program of Higher Education of China, under Grant 2007070322, the China Post-Doctoral Science Foundation, under Grants 2009045128 and 20100451369, the Provincial Natural Science Foundation of Shangxi of China, under Grant 2009JQ00515, the China Post-Doctoral Science Foundation Special, under Grant 2008045128, and the Basic Science Research Fund in Xidian University, Xi’an, China, under Grant JY1000090239.

R. Shang, L. Jiao, and W. Ma are with the School of Electronic Engineering, Xidian University, Xi’an, Shaanxi, China (e-mail: rshang@mail.xidian.edu.cn; lichjiao@mail.xidian.edu.cn; wpma@mail.xidian.edu.cn). P. Liu is with the School of Computer Science, Xidian University, Xi’an, Shaanxi, China (e-mail: liu616@163.com). Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEVC.2010.2046328

I. INTRODUCTION

Many real world optimization problems involve optimization of several (conflicting) criteria [1], [2], [3] and these optimization problems are called multiobjective optimization (MO) problems. MO searches for not only a single value but also a set of optimal solutions, called Pareto front [4]. In MO, one solution often cannot be said to be better than another and there are four goals in MO: 1) to discover solutions as close to the true Pareto front as possible; 2) to find solutions as diverse as possible in the obtained nondominated front; 3) to find solutions as uniform as possible in the obtained nondominated front; and 4) to discover solutions to cover the true Pareto front as width as possible. Satisfying these four goals is a challenging task for any algorithm used for MO problems (MOP).

Since the mid-1990s, many algorithms for MOPs have been presented. In 1984, Schaffer put forward for multiple objective optimization with vector evaluated genetic algorithms [5]. In 1994, Horn et al. proposed the niched Pareto genetic algorithm [6], and Srinivas et al. presented the nondominated sorting genetic algorithm (NSGA) [7], which attracted more attentions. In 2002, the improved version of NSGA (NSGA-II) [8] is proposed, which uses the elitism strategy. The other two famous algorithms, the strength Pareto evolutionary algorithm (SPEA) [9], and SPEA2 [10] (the improved version of SPEA) proposed by Zitzler et al. are also based on elitism strategy. Tan et al. ‘s distributed cooperative coevolutionary algorithm [11] is a currently research, which uses a distributed cooperative coevolutionary algorithm for solving MO problems. The MOPs in dynamic cases are studied in [12] and [13]. Most of the algorithms originate in the field of evolutionary algorithms (EAs). Just like EAs, the artificial immune system (AIS) constructs new intelligent algorithms with immunology terms and fundamental. Coello Coello et al. [14], [15] and Jiao et al. [16], [17] designed algorithms for MOPs inspired by the AIS.

AIS is a new intelligent method simulating natural immune system. It has enormous potential to supply novel methods to solve complex problems and is becoming another research hot point in the artificial intelligent techniques after neural network, fuzzy system, and evolutionary computation [18]–[21].

Compared with EAs [22]–[28] and other MO approaches [29]–[33], AIS used for solving MOPs has shown many advantages in improving the convergence speed and maintaining the diversity of the antibody population. However, the simple clonal selection computation has some difficulties in handling
some more complex problems. In this paper, the simple clonal selection strategy is improved and a novel immune clonal algorithm (NICA) is proposed. The improvements in NICA are mainly focus on four aspects.

1) Antibodies in the antibody population are divided into dominated ones and non-dominated ones, which is suitable for the characteristic of one MOP has a series Pareto-optimal solutions.

2) The entire cloning is adopted instead of different antibodies having different clonal rate.

3) The clonal selection is based on the Pareto-dominance and one antibody is selected or not depending on whether it is a non-dominated one, which is different from the traditional clonal selection manner.

4) The antibody population updating (APU) operation after the clonal selection is adopted, which makes antibody population under a certain size and guarantees the convergence of the algorithm.

The influences of main parameters are analyzed empirically. Compared with the existed algorithms, simulation results on MOPs and constrained MOPs show that NICA in most problems is able to find much better spread of solutions, a better diverse set of solutions and better convergence near the true Pareto-optimal front.

II. MOP

The MOP [34] is also called multicriteria optimization problem, multiperformance or vector optimization problem. A MOP can be formulated as follows:

\[
\begin{align*}
\min & \quad y = F(x) = (f_1(x), f_2(x), \ldots, f_m(x))^T \\
\text{s.t.} & \quad g_i(x) \leq 0, \quad i = 1, 2, \ldots, q \\
& \quad h_j(x) = 0, \quad j = 1, 2, \ldots, p \\
& \quad x = (x_1, \ldots, x_n) \in X \subset \mathbb{R}^n \\
& \quad y = (y_1, \ldots, y_m) \in Y \subset \mathbb{R}^m
\end{align*}
\]

where \( x = (x_1, \ldots, x_n) \in X \subset \mathbb{R}^n \) is called decision vector and \( X \) is \( n \)-dimensional decision space. The objective function \( F \) is an \( m \)-dimensional objective vector which defines the mapping function and \( m \) objectives are optimized simultaneously. \( g_i(x) \leq 0 \) (\( i = 1, 2, \ldots, q \)) defines \( q \) inequality constraints and \( h_j(x) = 0 \) (\( j = 1, 2, \ldots, p \)) defines \( p \) equality constraints. Furthermore, all the constraints determine the set of feasible solutions. Only those decision vectors \( x \in X \) satisfy all constraints \( g_i(x) \leq 0 \) (\( i = 1, 2, \ldots, q \)) and \( h_j(x) = 0 \) (\( j = 1, 2, \ldots, p \)) are in the set of feasible solutions. All the feasible solutions constitute a feasible set \( X_f \subseteq X \). The following four definitions are of great importance [35]:

**Definition 1 (Pareto-Dominance):** A solution \( x^* \) is said to dominate (Pareto-optimal) another solution \( x \) (denoted \( x^* \succ x \)) iff

\[
(\forall i \in \{1, \ldots, m\} : f_i(x^*) \leq f_i(x)) \land (\exists k \in \{1, \ldots, m\} : f_k(x^*) < f_k(x)) \quad (2)
\]

**Definition 2 (Pareto-Optimal):** A solution \( x^* \in X_f \) is said to be non-dominated (Pareto-optimal) iff

\[
(x^* \not\prec x \land x \not\prec x^*). \quad (3)
\]

**Definition 3 (Pareto-Optimal Set):** The set \( P_\gamma \) of all Pareto-optimal solutions is defined as

\[
P_\gamma = \{ x^* | \exists x \in X_f : x \succ x^* \} \quad (4)
\]

**Definition 4 (Pareto-Optimal Front):** The set \( P_F \) of all objective function values corresponding to the solutions in \( P_\gamma \)

\[
P_F = \{ F(x) = (f_1(x), f_2(x), \ldots, f_m(x))^T | x \in P_\gamma \} \quad (5)
\]

III. NICA FOR MOPS

A. Artificial Immune System (AIS)

The main goal of the immune system is to protect the human body from the attack of foreign organisms. The immune system is capable of distinguishing between the normal components of our organism and the foreign material that can cause harm. These foreign organisms are called *antigens*. The molecules called *antibodies* play a main role on the immune system response. The immune system response is specific to a certain foreign organism (antigen) [14]. When an antigen is detected, those antibodies that best recognize an antigen will proliferate by cloning. This process is called *clonal selection principle* [19], [20]. In order to describe the algorithm well, we define the terms as follows.

1) **Antigen:** In immunology, an antigen is any substance that causes the immune system to produce antibodies against it [21]. In this paper, for MOP

\[
\begin{align*}
\min & \quad y = F(x) = (f_1(x), f_2(x), \ldots, f_m(x))^T \\
\text{s.t.} & \quad g_i(x) \leq 0, \quad i = 1, 2, \ldots, q \\
& \quad h_j(x) = 0, \quad j = 1, 2, \ldots, p
\end{align*}
\]

where \( x = (x_1, \ldots, x_n) \in X \subset \mathbb{R}^n \) is called decision vector, \( m \geq 2 \) is the number of objective functions, \( q \) is the number of inequality constraints and \( p \) is the number of equality constraints, the antigen is defined as the objective vector \( F(x) \).

2) **Antibody:** In immunology, B cells, T cells, and antigen-specific lymphocytes are generally called antibodies. In this paper, an antibody is a representation of a candidate solution of an antigen [21]. The antibody \( a = (a_1, a_2, \ldots, a_m) \) with coding length \( n \) is the coding of antibody \( a \). In practice, binary coding and decimal coding are often used.

3) **Antibody Population:** Set \( X \) is called antibody space, namely \( a \in X \). An antibody population

\[
A = \{a_1, a_2, \ldots, a_N\} \quad (7)
\]

\( a_k \in X, 1 \leq k \leq N \) is an \( N \)-dimensional group of antibody \( a \), where the positive integer \( N \) is the antibody population.

B. Description of the Algorithm

In this section, we describe an NICA. In NICA, the initialization, immune clonal operation, immune gene operation, clonal selection operation (CSO), and antibody population updating operation (APU) are described as follows.
1) Initialization: Initialize the iterative number it := 0. Generate the initial population \( A (it) = \{ a_1 (it), a_2 (it), \ldots, a_N (it) \} \) with \( N (it) \) solutions randomly and for each solution calculate the value of the objective function. Then \( N (it) \) \( m \)-dimensional vectors are obtained with each dimension in each vector indicating one value of the objective function. And the objective-value matrix \( F (A (it)) \) with \( N (it)m \)-dimensional vectors is obtained, which is shown as follows:

\[
F (A (it)) = [ f_1 (A (it)), f_2 (A (it)), \ldots, f_m (A (it))] 
\]

where \( it \) indicates the current iteration.

2) Immune Clonal Operation: In immunology, clone means asexual propagation so that a group of identical cells can be descended from a single common ancestor, such as a bacterial colony whose members arise from a single original cell as the result of mitosis [21]. In artificial immune response, the immune clonal operation \( R^c_c \) on the antibody population \( A (it) \) is defined as

\[
\begin{align*}
A' (it) &= R^c_c (A (it)) \\
 &= R^c_c (a_1 (it), a_2 (it), \ldots, a_N (it)) \\
 &= \{ a^1_1 (it), a^1_2 (it), \ldots, a^1_N (it) \} + \ldots + \{ a^p_1 (it), a^p_2 (it), \ldots, a^p_N (it) \} \tag{9} \\
&\quad + \{ a^q_1 (it), a^q_2 (it), \ldots, a^q_N (it) \}
\end{align*}
\]

where \( R^c_c (a_i (it)) = \{ a^1_i (it), a^2_i (it), \ldots, a^p_i (it) \}, a^q_i (it) = a_i (it), j = 1, 2, \ldots, q_i, i = 1, 2, \ldots, N (it), q_i \in [1, m_c] \) is a self-adaptive parameter, or set as a constant, \( m_c \) is a given value related to the upper limit of clone scale, \( q_i = 1 \) represents that there is no clonal proliferation on antibody \( a_i (it) \). One MOP has a series of Pareto-optimal solutions, so the same \( m_c \) is used to each antibody \( a_i (it) (i = 1, 2, \ldots, N (it)) \).

In NICA, the entire cloning is adopted instead of cloning for one antibody by the fitness value. After clonal operation, the antibody population \( A' (it) \) is as follows:

\[
\begin{align*}
A' (it) &= \{ a^1_1 (it), a^2_1 (it), \ldots, a^1_N (it) \} + \ldots + \{ a^p_1 (it), a^p_2 (it), \ldots, a^p_N (it) \} \\
&\quad + \{ a^q_1 (it), a^q_2 (it), \ldots, a^q_N (it) \} \tag{10} \\
\end{align*}
\]

where \( q = R \) is the clone rate.

3) Immune Gene Operation: Clone provides multistrategy conditions for recombination and mutation [21]. Taking different strategies on different clonal antibodies of the same antibody, can promote the cooperation and information exchanging among antibodies, increase the diversity of population and speed up the convergence. Clonal recombination \( R^c_c \) and clonal mutation \( R^c_m \) are the main operators in immune gene operation \( R^c_c \).

In NICA, the clonal mutation is the main operation and the clonal recombination is not used. In immunology, hyper-mutation is the main operation of affinity maturation. The essential content of hyper-mutation is changing genetic values in some genetic positions of antibodies in \( A (it) \) with probability 1. As far as the binary representation is concerned, hyper-mutation makes some genetic positions inverse with a certain probability (i.e., \( 1 \rightarrow 0 \) or \( 0 \rightarrow 1 \)). For decimal representation, uniform mutation, Gaussian mutation, Cauchy mutation and some stochastic mutations can be used [21].

Cutello et al. [36–41] proposed three well known hypermutation methods: 1) static hypermutation, in which the number of mutations is independent of the fitness values; 2) proportional hypermutation, in which the number of mutations is proportional to the fitness value; and 3) inversely proportional hypermutation, in which the number of mutations is inversely proportional to the fitness value [16]. In NICA, the dominated individuals are not assigned fitness, so the static hypermutation operation is used on the clone population in our algorithm.

\( A' (it) \) is the clone population and the static hypermutation operation \( R^c_m \) on \( A' (it) \) is defined as

\[
\begin{align*}
A'' (it) &= R^c_m (A' (it)) \\
 &= R^c_m (\{ a^1_1 (it), a^2_1 (it), \ldots, a^q_1 (it) \} + \ldots + \{ a^1_N (it), a^2_N (it), \ldots, a^q_N (it) \}) \\
&= \{ a^{1'}_1 (it), a^{2'}_1 (it), \ldots, a^{q'}_1 (it) \} + \ldots + \{ a^{1'}_N (it), a^{2'}_N (it), \ldots, a^{q'}_N (it) \} \\
&\quad + \{ a^{q''}_1 (it), a^{q''}_2 (it), \ldots, a^{q''}_N (it) \} \tag{11}
\end{align*}
\]

where \( R^c_m (a^j_i (it)) = a^{j'}_i (it), j = 1, 2, \ldots, q, i = 1, 2, \ldots, N (it), j_i \in [1, m_m] \) is a self-adaptive parameter, or set as a constant, \( m_m \) is a given value related to the upper limit of mutation scale, \( q_i = 1 \) represents that there is no hypermutation on antibody \( a^j_i (it) \). One MOP has a series of Pareto-optimal solutions, so the same \( m_m \) is used to each antibody \( a^j_i (it) (i = 1, 2, \ldots, N (it)) \).

In NICA, antibodies in the population are divided into dominated ones and non-dominated ones. An antibody is selected or not depends on whether it is a non-dominated solution, and only the non-dominated antibodies are selected in the antibody population.

a) Divide the Antibodies into Dominated Ones and Non-dominated Ones: In NICA, before clonal selection the antibodies in the antibody population are divided into dominated ones and non-dominated ones.
For any antibody, \( a^*(ii) \in A^*(ii) \), \( a^*(ii) \) is called a nondominated antibody in the current iteration iff
\[
\neg \exists a^*_j(ii) \neq a^*(ii) (i = 1, 2, \ldots, N; j = 1, 2, \ldots, q) \in A^*(ii) : (\forall i \in \{1, \ldots, m\} : f_i(a^*(ii)) \\
\geq f_i(a^*_j(ii))) \land (\exists k \in \{1, \ldots, m\} : f_k(a^*(ii)) \\
> f_k(a^*_j(ii))).
\]
(12)

Or else, \( a^*(ii) \) is called a dominated antibody.

b) Select the Nondominated Antibodies With Size \( \text{N}_{\text{nond}} \).

Based on section a), the antibodies in \( A^*(ii) \) are also divided into two parts: \( A_{\text{nond}}(ii) \) with \( N_{\text{nond}}(ii) \) nondominated antibodies and \( A_{\text{dom}}(ii) \) with \( N_{\text{dom}}(ii) \) dominated antibodies. Note that \( N_{\text{nond}}(ii) + N_{\text{dom}}(ii) = q \cdot N(ii) \).

So, for antibody population \( A^*(ii) \), the clonal selection operation \( R^C_C \) is defined as
\[
A^*(ii) = R^C_C \left( A^*(ii) \right) = R^C_C \left( \{a^*_1(ii), a^*_2(ii), \ldots, a^*_q(ii) \} \right)
\]
\[
= R^C_C \left( \{a^*_1(ii), a^*_2(ii), \ldots, a^*_q(ii), a^*_q(1)(ii), a^*_q(2)(ii), \ldots, a^*_q(N_{\text{nond}}(ii))(ii) \} \right)
\]
\[
= \{a^*_1(ii), a^*_2(ii), \ldots, a^*_q(ii), a^*_q(1)(ii), a^*_q(2)(ii), \ldots, a^*_q(N_{\text{nond}}(ii))(ii) \}
\]
(13)

where \( a^*_i(ii) (i = 1, \ldots, N_{\text{nond}}(ii)) \)are the nondominated antibodies in the antibody population \( A^*(ii) \). \( N_{\text{nond}}(ii) \) is the number of the nondominated antibodies in \( A^*(ii) \).

5) APU: The selection strategy in NICA is specially designed for MO, which is different from the selection manner based on the traditional affinity values. An antibody is selected or not depends on whether it is a nondominated solution. If the number of the nondominated antibodies in one generation is very large, all of them are selected to the next generation based on the selection strategy in NICA and they are cloned by entire clonal strategy. Therefore, the size of antibody population is very large and after the gene operation, the number of the nondominated antibodies may be much larger. As the progress of the generation, more and more nondominated antibodies will be generated, which influences the speed of the algorithm seriously.

In order to void this case, the APU strategy is adopted in NICA. Namely, when the number of the selected nondominated antibodies is larger than the pre-defined number \( N_h \), one antibody which is the most crowded in the Pareto front is deleted, which guarantees the algorithm’s convergent speed and maintaining the diversity of antibody population. The antibody population updating operation is shown in Algorithm 1, where \( F \left( A^*(ii) \right) \) represents the objective value matrix of antibody population \( A^*(ii) \); \( \max \left( \{F \left( A^*(ii) \right) \} (; i) \right) \) and \( \min \left( \{F \left( A^*(ii) \right) \} (; i) \right) \) indicates the maximum and the minimum of the \( i \) th objective value, respectively. \( \delta \) is a small integer, which avoid the denominator being 0 when \( \max \left( \{F \left( A^*(ii) \right) \} (; i) \right) = \min \left( \{F \left( A^*(ii) \right) \} (; i) \right) \).

Algorithm 1 the APU operation in NICA

Step 1: Give the antibody population size number \( N_{\text{nond}}(ii) \), the dimension \( m \) of the objective vector, the antibody population size \( N_h \) expected to maintain; initialize \( i := 1 \), \( j := 1 \);

Step 2: Give the ascending order of the antibody population based on the \( i \) th objective value:
\[
\left[ F \left( A^*(ii) \right) \right] (i, i) = \left[ f \left( A^*(ii) \right) \right] (i, i) = \left[ \left[ f_1 \left( A^*(ii) \right) \right], \left[ f_2 \left( A^*(ii) \right) \right], \ldots, \left[ f_m \left( A^*(ii) \right) \right] \right] (i, i) = \left[ f_1 \left( A^*(ii) \right) \right] ;
\]
Step 2.1: Give an large enough value \( N_h \) to the boundary antibodies; \( c_1 = NN_h, c_2, c_{\text{nond}} = NN_h \);

Step 2.2: Give an value as:
\[
c_{ij} = \frac{\left( F \left( A^*(ii) \right) \right) (i, i) - \left( F \left( A^*(ii) \right) \right) (j, j)}{(i < j < N_{\text{nond}}(ii), \delta \text{ is a little positive quantity})};
\]
each other antibody and \( \left( F \left( A^*(ii) \right) \right) (j + 1, i) \) indicates the \( i \) th objective value of \( a^*_j(ii) \);

Step 3: If \( i = m \), go to Step 4; or else, \( i := i + 1 \), return Step 2;

Step 4: If \( N_{\text{nond}}(ii), \text{go to Step 5} \); or else, \( j := j + 1 \), \( i := i + 1 \), return Step 2;

Step 5: Calculate the \( j \)th antibody’s fitness function:
\[
f \left( c_j(k) \right) = c_{ij} + c_{j2} + \cdots + c_{mj},
\]
\( j = 1, 2, \ldots, N_{\text{nond}}(ii) \);

Step 6: If \( N_{\text{dom}}(ii) = N_h \), stop; or else, go to Step 7;

Step 7: Delete the antibody with the least fitness and get the new \( A^*(ii) \) and \( F \left( A^*(ii) \right) \):
\[
\text{Give} N_{\text{gen}}(ii) := N_{\text{nond}}(ii) - 1; A^*(ii) = A^*(ii) \setminus \{ \text{抗体} \};
\]
\[
F \left( A^*(ii) \right) = F \left( A^*(ii) \right) // i := 1, j := 1; \text{return Step 2.}
\]

After the APU operation \( R^C_C \) to \( A^*(ii) \), a new \( A^{\prime\prime}(ii) \) will be obtained
\[
A^{\prime\prime}(ii) = R^C_C \left( A^*(ii) \right) = R^C_C \left( \{a^*_1(ii), a^*_2(ii), \ldots, a^*_N(\text{gen})(ii) \} \right)
\]
\[
= \{a^*_1(ii), a^*_2(ii), \ldots, a^*_N(\text{gen})(ii) \}.
\]
(14)

6) NICA for MO: In the light of the above considerations, the proposed algorithm NICA integrating the improved immune clonal operation, the immune gene operation, the CSO and the APU operation can be summarized by the chart of Fig. 1 and can be described as Algorithm 2.

IV. SIMULATION RESULTS

In this section, we describe ten benchmark problems and five performance metrics for comparing the performance of NICA with other six popular multiobjective optimization evolutionary algorithms (MOEAs). They are Pareto archived evolution strategy (PAES) [42], Pareto envelope-based selection algorithm (PESA) [43], NSGA-II [8], SPEA2 [10], MOEA/D [44], and adaptive multiobjective optimization algorithm using the clonal selection principle (ACSMO) [45].
Algorithm 2 NICA for MO

Step 1: Give the termination criterion: gmax, antibody population size N, and clonal rate R; Generate the initial antibody population:
A (0) = \{a_1 (0), a_2 (0), \ldots, a_N (0)\} ⊆ I^N; Set the initial generation it := 0;

Step 2: Implement the immune clonal operation with R to A (it); A' (it) = R^c_E (A (it));

Step 3: Implement the immune gene operation to A'' (it); A'' (it) = R^c_E (A' (it));

Step 4: Implement the CSO to A''' (it):
A''' (it) = R^c_E (A'' (it));

Step 4.1: Calculate the objectives values of all the antibodies in A'' (it) and the objective value matrix F (A'' (it)) is obtained;

Step 4.2: Divide the antibodies in A'' (it) into dominated ones and nondominated ones based on the Pareto-optimal theory; and the dominated antibody population A_d (it) and the nondominated antibody population A_n (it) will be obtained;

Step 4.3: Select the nondominated antibodies and the new antibody population A''' (it) = A_n (it) and the new related objective-value matrix F (A''' (it)) are obtained;

Step 5: Implement the APU operation to A''' (it) and the new A'''' (it) and the related objective-value matrix F (A'''' (it)) are obtained;

Step 6: Give A (it + 1) = A'''' (it); F (A (it + 1)) = F (A'''' (it)); it := it + 1; If it > gmax, stop; output A (it) and F (A (it)); or else, return Step 2.

A. Test Problems

In order to examine the performance of NICA, ten test problems that were frequently used as benchmark problems in [1], [8], [11], and [46]–[50] are selected. These well-known problems are KUR (proposed by F. Kursawe) [50], ZDT1–ZDT4 and ZDT6 (proposed by E. Zitzler, K. Deb, and L. Thiele) [47], and DTLZ1–DTLZ4 (proposed by K. Deb, L. Thiele, M. Laumanns, and E. Zitzler) [46].

B. Performance Measures

Generally, an MOP has a set of Pareto-optimal solutions, instead of one single optimal solution, so the goal in an MOP is different from that in a single-objective optimization. In MOPs, we desire to find as many different Pareto-optimal or near Pareto-optimal solutions as possible, as minimal distance of the solutions to the Pareto-optimal front as possible, as good distribution of the solutions as possible and as maximum spread (MS) of the solutions as possible.

In this paper, we use five quantitative measures to evaluate all of the three objectives of the tradeoff fronts produced by various MO algorithms [49]–[52]. These measures are capable of evaluating nondominated antibodies in several nontrivial aspects and have been widely used in the studies of MO algorithms [11]. They are: 1) the generational distance (GD) metric, which represents how “far” the known Pareto front is away from the true Pareto front; 2) the spacing (S) metric, which measures how “evenly” members in the known Pareto front are distributed; 3) the MS metric, which measures how “well” the true Pareto front is covered by the known Pareto front through hyperboxes formed by the extreme function values observed in the true Pareto front and the known Pareto front; 4) the hypervolume ratio (HVR) metric defined by Van Veldhuizen and Lamont [49], which is defined as a ratio of the hypervolume of the known Pareto front and the hypervolume of the true Pareto front and here the hypervolume calculates the “volume” in the objective domain covered by the set of nondominated antibodies for an MO minimization problem; and 5) the spread Δ, which measures the extent of spread achieved among the obtained solutions [45]. The expressions of GD, S, MS, and HVR please see [11] and the expression of Δ please see reference [8] or [45] for details.

C. Parameters Analysis

In NICA, the main parameters are the size of the antibody population N, the termination generation gmax, the dimension of the objective vector m, the size of the nondominated antibody population Nnon, the number of the nondominated antibodies Nn expected to maintain, the clonal rate R, and the mutation probability pm. The influences of the parameters N, gmax, and Nnon to the algorithm’s performance are obviously and if not taking the complexity into account, the larger of their values are, the better of the results. The parameter r depends on the problems and Nnon depends on the problems and the antibody population. The influences of the parameters R and pm to the algorithm’s performance are more complex. In NICA, let the mutation probability pm = 1/n and the following is the empirical analysis results of parameter R.
The clonal rate $R$ is sampled by the same interval of 1 between 1 and 5 and runs 20 times to get the average values and the deviation of four Performance Measures. A common problem ZDT1 is selected to show the influences of the parameter $R$ which are shown in Table I.

It can be seen from the results that the influence of clonal scale on the performance of the algorithm is notable. Table I shows that the average value of GD decreases approximate linearly with the increasing of $R$, which represents the larger $R$ is, the smaller generation distance of the known Pareto front away from the true Pareto front. The metric S changes obviously with $R$ sample by the same interval of 1 between 1 and 5, which shows as the increases of $R$, the distribution of the obtained solutions on the Pareto front becomes more and more evenly. The metric MS changes not obviously with $R$. However, the metric GD increases with $R$ becoming larger. And the influence of $R$ on the metric HVR is obviously.

The average of HVR becomes larger when the clonal rate $R$ becomes from 1 to 4, but as the clonal rate $R$ changes from 4 to 5, the average of HVR becomes smaller. So for the metric HVR, the coverage ratio of the set of nondominated antibodies for an MO minimization problem doesn’t become lager and lager with the clonal rate $R$ becoming lager. Take all the factors into account, clonal rate $R = 4$ is selected in NICA.

### D. Comparisons of NICA With PAES, PESA, NSGA-II, and SPEA2

In this section, some simulation results and comparisons that demonstrate the potential of NICA are presented, and the comparisons mainly focus on four aspects: 1) the convergence of the known Pareto front to the true Pareto front; 2) the uniformity of the obtained solutions in the objective domain; 3) the coverage of the known Pareto front to the true Pareto front; and 4) the diversity of the nondominated antibodies.

Although such comparisons are not exhaustive, they provide a good basis to assess the performance of NICA.

In order to provide a fair comparison, all approaches are real-coded and run for a maximum of 25,000 function evaluations. For SPEA2, we use a population of size 100 and an external population of size 100. For PESA, the internal population size is 100, the archive size is 100, and the number of hyper-grid cells per dimension is 10. For NSGA-II and PAES, the population size is 100. The mutation probability for all the algorithms is $p_m = 1/n$. For PAES, PESA, NSGA-II and SPEA2, the distribution index for polynomial mutation is 20. In the simulation results, 30 independent runs (with random initial populations) on the five algorithms are performed on each of the test functions in order to study the statistical performance, such as consistency and robustness of the algorithms.

The nondominated solutions obtained with PESA, NSGA-II, SPEA2, and NICA on ZDT6 and DTLZ3 shown in Figs. 2 and 3. The problem ZDT6 is nonconvex has nonuniform distribution of solutions and this problem is more difficult than ZDT1, ZDT2, and ZDT3, which makes ZDT6 a difficult problem to tackle by most MO algorithms.

For ZDT6, Fig. 2 demonstrates that both PESA and NSGA-II get stuck at different local Pareto-optimal sets and NSGA-II has difficulties in converging near the global Pareto-optimal front. However, NSGA-II has the ability to find a diverse set of solutions. SPEA2 has found a better spread in the entire Pareto-optimal region and a better convergence than PESA and NSGA-II. NICA has the abilities in maintaining the uniformity of the obtained solutions, in finding diverse solutions in the front and in converging to the Pareto-optimal front. In the aspects of uniformity, distribution and convergence of solutions, NICA performed best.

In DTLZ3, the $g$ function introduces $(2^k - 1)$ local Pareto-optimal fronts, and one global Pareto-optimal front. All local Pareto-optimal fronts are parallel to the global Pareto-optimal front and an MO algorithm can get stuck at any of these local Pareto-optimal fronts, before converging to the global Pareto-optimal front [43].

The nondominated solutions obtained by the four algorithms on DTLZ3 are shown in Fig. 3, parts (a1), (b1), (c1), and (d1) are the side views of (a), (b), (c), and (d), respectively. Fig. 3 shows that PESA and NSGA-II cannot maintain a good diversity of solutions on the Pareto-optimal front and cannot quite converge on to the Pareto-optimal front. SPEA2 can maintain a good diversity of solutions on the Pareto-optimal
Fig. 3. Nondominated solutions with PESA, NSGA-II, SPEA2, and NICA on DTLZ3.
front but cannot quite converge on to the Pareto-optimal front; however, NICA can maintain a good diversity of solutions on the Pareto-optimal front and can converge on to the Pareto-optimal front.

The simulation results of five algorithms for each test problem with respect to four performance metrics GD, S, MS, and HVR are summarized in Figs. 4–7. The distribution of simulation data for the 30 independent runs is presented in the box plot format [53], which has been applied by Zitzler and Thiele [9] and Tan et al. [11] to visualize the distribution of simulation data in their studies of MOBAs.

In this paper, the notched boxes are used to illustrate the comparison result. The notches represent a robust estimate of the uncertainty about the medians for box-to-box comparison. Each box plot represents the distribution of a sample set where a thick horizontal line within the box and each box has lines at the lower quartile, median, and upper quartile values. Dashed appendages illustrate the spread and shape of distribution. The whiskers are lines extending from each end of the box to show the extent of the rest of the data. Outliers are data with values beyond the ends of the whiskers [11].

On the metric GD, as shown in Fig. 4, for the problems ZDT1, ZDT2 and ZDT3, NICA performs poorer than PESA and better than PAES, NSGA-II, and SPEA2. So for the three problems, PESA is able to converge best and NICA is able to converge better than PAES, NSGA-II, and SPEA2. However, for the other seven problems, NICA performs absolutely better than the other four algorithms. Therefore, NICA have the best convergence in the five algorithms.

On the metric S, Fig. 5 shows that for the ten problems NICA has the absolutely advantage in finding a uniform set of solutions than the other four algorithms.

TABLE II

<table>
<thead>
<tr>
<th>Instance</th>
<th>KUR</th>
<th>ZDT1</th>
<th>ZDT2</th>
<th>ZDT3</th>
<th>DT LZ1</th>
<th>DT LZ2</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOEA/D</td>
<td>0.037707</td>
<td>0.00565</td>
<td>0.001417</td>
<td>0.000058</td>
<td>0.0179841</td>
<td>0.024651</td>
</tr>
<tr>
<td>NICA</td>
<td>0.023465</td>
<td>0.002112</td>
<td>0.001654</td>
<td>0.003156</td>
<td>0.00213201</td>
<td>0.002539</td>
</tr>
<tr>
<td></td>
<td>1.22e-004</td>
<td>2.24e-005</td>
<td>2.12e-004</td>
<td>6.15e-005</td>
<td>4.53e-004</td>
<td>1.65e-003</td>
</tr>
</tbody>
</table>

TABLE III

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>ZDT1</th>
<th>ZDT2</th>
<th>ZDT3</th>
<th>ZDT4</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACSAMO</td>
<td>Mean</td>
<td>2.32e-4</td>
<td>9.70e-5</td>
<td>6.28e-4</td>
</tr>
<tr>
<td>Variance</td>
<td>3.36e-5</td>
<td>9.30e-6</td>
<td>2.65e-5</td>
<td>4.75e-5</td>
</tr>
<tr>
<td>NICA</td>
<td>Mean</td>
<td>1.29e-4</td>
<td>1.25e-4</td>
<td>5.39e-4</td>
</tr>
<tr>
<td>Variance</td>
<td>2.14e-6</td>
<td>6.35e-5</td>
<td>1.14e-5</td>
<td>5.89e-4</td>
</tr>
</tbody>
</table>

On the metric MS, for the problems KUR, ZDT1, ZDT3, ZDT6, and DT LZ2, it can be seen from Fig. 6, all the algorithms achieve a similar performance except PAES, which shows that the solutions obtained by the algorithms NICA, PESA, NSGA-II, and SPEA2 cover the true Pareto front well. However, for the other test problems, NICA performs excellently, which shows that the solutions obtained by NICA covers the true Pareto front very well.

On the metric HVR, it can be seen from Fig. 7, except for the problems KUR, ZDT3, and ZDT6, on the other seven problems, NICA achieves an absolutely better performance than the other four algorithms and the other four algorithms have a very similar performance. Especially for problem ZDT4. It can be concluded from the simulation results that NICA manages to find the global Pareto front, while the other four algorithms get stuck at different local Pareto-optimal sets, e.g., the solutions of the other four algorithms do not cover the entire global Pareto front well, which results in the poor performance.

The simulation results of four metrics show that NICA in most problems is able to find much better spread of solutions, a better diverse set of solutions and better convergence near the true Pareto-optimal front.
Fig. 5. Box plots for the metrics of $S$ on ten test problems.

Fig. 6. Box plots for the metrics of MS on ten test problems.

Fig. 7. Box plots for the metrics of HVR on ten test problems.
Fig. 8. Nondominated solutions obtained with MOEA/D and NICA on KUR and ZDT4.

E. Comparisons of NICA With MOEA/D

MOEA/D is a famous and effective algorithm for MOPs [44], [54]–[57]. It first uses a decomposition method to decompose the MOPs into a number of scalar optimization problems. In order to evaluate our algorithm NICA and the algorithm MOEA/D, we used KUR, ZDT1–ZDT4, ZDT6, DTLZ1, and DTLZ2 test problems. The nondominated solutions obtained with MOEA/D and NICA on KUR and ZDT6 are shown in Fig. 8.

Fig. 8 demonstrates that for test problems KUR and ZDT4, both the algorithms can maintain a good convergence and a widely distribution. However, NICA gets a better uniform distribution than MOEA/D.

The spacing measures $S$ is used to evaluate the uniformity of the algorithm MOEA/D and the algorithm NICA, which is shown in Table II.

Table II shows that for test problems ZDT2, DTLZ1, and DTLZ2, the mean values of $S$ obtained by the algorithm NICA are slightly larger than that obtained by the algorithm MOEA/D, which indicates that for these test problems, the nondominated solutions obtained by the algorithm MOEA/D distribute more uniform than that obtained by the algorithm NICA. However, for test problems KUR, ZDT1, and ZDT3, the mean values of $S$ obtained by the algorithm NICA are smaller than that obtained by the algorithm MOEA/D, which indicates that for these test problems, the nondominated solutions obtained by the algorithm NICA distribute more uniform.

F. Comparisons of NICA With ACSAMO

The algorithm ACSAMO is an effective algorithm which uses adaptive clonal selection principle for MO optimization [45]. In this section, In order to validate our approach NICA and the algorithm ACSAMO, we used four ZDT test problems.

In the objective space, the known Pareto-optimal fronts are shown as continuous lines in NICA. The obtained solutions are shown with "*" in NICA and with "·" in ACSAMO. The simulation results of NICA and ACSAMO on ZDT1, ZDT2, ZDT3, and ZDT4 are shown in Fig. 9.

Fig. 9 shows that for ZDT1, ZDT2, ZDT3, and ZDT4, both NICA and ACSAMO have the ability in converging to the true front. But NICA has better ability in finding diverse solutions in the front and can maintain a more uniform distribution than ACSAMO.

In order to compare NICA with ACSAMO quantitatively, we use two performance metrics, namely, the GD and the spread $\Delta$.

Table III shows the mean and variance of the convergence metric GD obtained by NICA and ACSAMO. NICA is able to converge better in all problems expect ZDT2. Especially for ZDT1 and ZDT4, the mean of the convergence metric GD obtained by NICA is far smaller than that of ACSAMO. This metric shows that NICA has better performance in the convergence to the Pareto-optimal front than ACSAMO.

Table IV shows that the mean and variance of the diversity metric $\Delta$ obtained by NICA and ACSAMO.
It can be seen from Table IV that NICA is able to find a better spread of solutions than ACSAMO in all problems. In all the cases with NICA, the variance in thirty runs is small. This metric shows that NICA has better performance in finding the spread of solutions.

V. CONSTRAINT HANDLING

In this section, we treat the constraints of the constrained multiobjective problem (CMOP) as an objective and the constrained NICA is used to solve these problems. The constrained NICA could find the Pareto-optimal solutions from the feasible region and the edge of the infeasible region, which
assures both the convergence and diversity of the obtained solutions. Simulation results show that the constrained NICA has much better performance in finding a much better spread of solutions, in maintaining a better uniformity of the solutions and in obtaining a better convergence.

A. Proposed Constraint-Handling Approach—Constrained NICA

Formally, a CMOP with \( n \) decision variables, \( k \) objective functions and \( m \) constraints is shown as follows [8]:

\[
\begin{align*}
\min \quad & y = f(x) = (f_1(x), \ldots, f_k(x)) \\
\text{subject to} \quad & g(x) = (g_1(x), \ldots, g_m(x)) \leq 0 \\
& x = (x_1, x_2, \ldots, x_n) \in X \\
& I = (l_1, l_2, \ldots, l_n), \quad u = (u_1, u_2, \ldots, u_n) \\
& y = (y_1, y_2, \ldots, y_n) \in Y
\end{align*}
\]  

(15)

where \( x \) is decision vector, \( y \) is the objective function, \( X \) represents the decision space, \( I \) and \( u \) represent the lower bound and the upper bound, respectively, \( Y \) is the decision space.

This constraint-handling method treats the constraints as an objective. Given that

\[
G_j(x) = \max \left\{ 0, g_j(x) \right\}, \quad 1 \leq j \leq m.
\]  

(16)

A new objective function is obtained as

\[
f_{k+1}(x) = G(x) = \sum_{j=1}^{m} G_j(x). \tag{17}
\]

So, the CMOP becomes an unconstrained MOP

\[
\begin{align*}
\min \quad & y = f(x) = (f_1(x), \ldots, f_k(x), f_{k+1}(x)) \\
\text{where} \quad & x = (x_1, x_2, \ldots, x_n) \in X \\
& X = \{(x_1, x_2, \ldots, x_n) | l_i \leq x_i \leq u_i \} \\
& I = (l_1, l_2, \ldots, l_n), \quad u = (u_1, u_2, \ldots, u_n) \\
& y = (y_1, y_2, \ldots, y_n) \in Y
\end{align*}
\]  

(18)

In this situation, the feasible set \( X_f \) can be defined as

\[
X_f = \{x = (x_1, x_2, \ldots, x_n) | f_{k+1}(x) = 0, l_i \leq x_i \leq u_i \}.
\]  

(19)

B. Simulation Results

Four constrained test problems CONSTR, SRN, TNK, and WATER (see [8] and [58]) are used to test the algorithm NICA.

For CONSTR, a part of the unconstrained Pareto-optimal region is not feasible. Thus, the resulting constrained Pareto-optimal region is a concatenation of the first constraint boundary and some part of the unconstrained Pareto-optimal region [8].

Fig. 10(a) and (b) shows the obtained nondominated solutions with NSGA-II and NICA. Although the Pareto-optimal is discontinuous, both NSGA-II and NICA do not have any difficulty in finding a wide spread of solutions. However, Fig. 10(b) demonstrates the abilities of NICA in maintaining the uniformity of the obtained solutions, in finding diverse solutions in the front. In both the aspects of uniformity and distribution of the obtained solutions, NICA has a better performance than NSGA-II.

For the problem SRN, the constrained Pareto-optimal set is a subset of the unconstrained Pareto-optimal set. Fig. 11(a) and (b) shows the obtained nondominated solutions with NSGA-II and NICA on SRN. Both NSGA-II and NICA have the abilities in maintaining the uniformity of the obtained solutions, in finding diverse solutions in the front and in converging to the Pareto-optimal front.

Fig. 12(a) and (b) shows the obtained nondominated solutions with NSGA-II and NICA on TNK. Fig. 12(b) shows the abilities of NICA in maintaining the uniformity of the obtained solutions, in finding diverse solutions in the front. In both the aspects of uniformity and distribution of solutions, NICA also gives a better performance than NSGA-II.

Deb et al. [8] have used the problem WATER in their study and normalized the objective functions in the following manner:

\[
f_1/8 \left(10^6\right), \quad f_2/1500, \quad f_3/3 \left(10^6\right), \quad f_4/6 \left(10^5\right), \quad f_5/8000.
\]

We observe the range of the normalized objective function values of the obtained nondominated solutions. Table V shows the comparison of NICA with NSGA-II. In most objective functions, NICA has found a similar spread of solutions with NSGA-II.
TABLE V
LOWER AND UPPER BOUNDS OF THE FUNCTION VALUES OBSERVED IN THE OBTAINED NONDOMINATED SOLUTIONS

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$f_1$</th>
<th>$f_2$</th>
<th>$f_3$</th>
<th>$f_4$</th>
<th>$f_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>NICA</td>
<td>0.798–0.920</td>
<td>0.027–0.900</td>
<td>0.095–0.951</td>
<td>0.031–1.096</td>
<td>0.001–3.125</td>
</tr>
<tr>
<td>NSGA-II</td>
<td>0.798–0.920</td>
<td>0.027–0.900</td>
<td>0.095–0.951</td>
<td>0.031–1.110</td>
<td>0.001–3.124</td>
</tr>
</tbody>
</table>

Fig. 12. Nondominated solutions with NSGA-II and NICA on TNK.

Fig. 13. Upper diagonal plots are for NSGA-II and lower diagonal plots are for NICA. Compare $(i, j)$ plot (NICA with $i > j$) with $(j, i)$ plot (NSGA-II). Label and ranges used for each axis are shown in the diagonal boxes.
Inspired by [8], in order to show the pairwise interactions among these five normalized objective functions, we plot ten interactions in Fig. 13 for NSGA-II and NICA. In Fig. 13, the results obtained by NSGA-II are shown in the upper diagonal portion of the figure and the results obtained by NICA are shown in the lower diagonal portion. The axes of any plot can be obtained by looking at the corresponding diagonal boxes and their ranges. For example, the plot at the first row and third column has its vertical axis as $f_1$ and horizontal axis as $f_3$. Since this plot belongs in the upper side of the diagonal, this plot is obtained using NSGA-II. In order to compare this plot with a similar plot using NICA, we look for the plot in the third row and first column. For this figure, the vertical axis is plotted as $f_3$ and horizontal axis is plotted as $f_1$. To get a better comparison between these two plots, we observe NICA's plot as it is, but turn the page no. 90 in the clockwise direction for NSGA-II results. This would make the labeling and ranges of the axes same in both cases.

It can be seen from Fig. 13 that NICA plots have better formed patterns than NSGA-II plots. For example, figures $f_2-f_3$, $f_3-f_2$, $f_1-f_2$, $f_1-f_3$, and $f_2-f_4$ interactions are very clear from NICA results. Although similar patterns exist in the results obtained using NSGA-II algorithm, the convergence to the Pareto-optimal front is not adequate. The figures $f_3-f_1$, $f_2-f_3$, and $f_2-f_4$ interactions show the abilities of NICA in maintaining the uniformity of the obtained solutions, in finding diverse solutions in the front. In both the aspects of uniformity and distribution of solutions, NICA performed better than NSGA-II.

VI. CONCLUSION

In this paper, an NICA was proposed. The improvements in NICA are mainly focused on four aspects and the influences of main parameters were analyzed empirically. Compared with the existed algorithms, simulation results on MOPs and constrained MOPs showed that NICA in most problems is able to find much better spread of solutions and better convergence near the Pareto-optimal front. Therefore, NICA seems promising and is able to produce results similar to or better than those generated by other algorithms that represents the state-of-the-art in evolutionary MO. However, how to produce a highly competitive algorithm (based on AIS) and how to evaluate an MO algorithm more efficiently are our further work and such work is currently under way.

REFERENCES


---

**Ronghua Shang** (M’09) received the B.S. degree in information and computation science and the Ph.D. degree in pattern recognition and intelligent systems from Xidian University, Xi’an, China, in 2003 and 2008, respectively. Since 2008, she has been a Lecturer with the School of Electronic Engineering, Xidian University. Her current research interests include optimization problems, evolutionary computation, artificial immune systems, and data mining.

**Licheng Jiao** (SM’97) was born in Shannxi, China, in October 1959. He received the B.S. degree in circuits and systems from Shanghai Jiaotong University, Shanghai, China, in 1982, and the M.S. and Ph.D. degrees in circuits and systems from Xi’an Jiaotong University, Xi’an, China, in 1984 and 1990, respectively. From May 1990 to May 1992, he was a Post-Doctoral Fellow with the National Key Laboratory of Radar Signal Processing, Xidian University, Xi’an, China, where he is an Assistant Professor. From June 1992 to March 2003, he was a Professor with the National Key Laboratory of Radar Signal Processing, Xidian University. Since 2003, he has been the Dean of the School of Electronic Engineering and the Director of the Institute of Intelligent Information Processing, Xidian University. His current research interests include signal and image processing, natural computation, and intelligent information processing.

Dr. Jiao is the Chairman of the Awards and Recognition Committee of the IIEE Xi’an Section, the Chairman of IEEE Computational Intelligence Society Xi’an Section, an executive committee member of the Chinese Association of Artificial Intelligence, the Councilor of the Chinese Institute of Electronics, a committee member of the Chinese Committee of Neural Network, and an Expert of the Academic Degrees Committee of the State Council. He is also an editorial board member of Memetic Computing, the Chinese/English version of the Acta Electronica Sinica, and the Journal of Electronics and Information Technology. He was a member of the 8th National People’s Congress. He received the Youth Science and Technology Award, the Fok Ying Tung Young Teacher First Class Award, the Cross-Century Specialists Fund, the Fok Ying Tung Young Teacher Fund, the first group of the first level and the second level of Millions of Talents Project, the first level of Shannxi Three, Five Talents Project, and the Shannxi Contributive Specialists.
Fang Liu (M'07–SM'07) received the B.S. degree in computer science and technology from Xi'an Jiaotong University, Xi'an, China, in 1984, and the M.S. degree in computer science and technology from Xidian University, Xi'an, China, in 1995. She is currently a Professor with the School of Computer Science, Xidian University. Her current research interests include synthetic aperture radar image processing, multiscale geometry analysis, optimization problems, and data mining.

Wenping Ma (M'07) received the B.S. degree in computer science and technology, and the Ph.D. degree in pattern recognition and intelligent systems from Xidian University, Xi'an, China, in 2003 and 2008, respectively. Since 2006, she has been a Lecturer with the School of Electronic Engineering, Xidian University. Her current research interests include pattern recognition, artificial immune systems, and data mining.