# Multiobjective Immune Algorithm with Nondominated Neighbor-Based Selection

**Maoguo Gong** gong@ieee.org
Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Institute of Intelligent Information Processing, Xidian University, Xi'an 710071, China

**Licheng Jiao** lchjiao@mail.xidian.edu.cn
Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Institute of Intelligent Information Processing, Xidian University, Xi'an 710071, China

**Haifeng Du** haifengdu@mail.xjtu.edu.cn
School of Public Policy and Administration, Xi'an Jiaotong University, Xi'an 710049, China

**Liefeng Bo** blf0218@hotmail.com
Institute of Intelligent Information Processing, Xidian University, Xi'an 710071, China

**Abstract**

Nondominated Neighbor Immune Algorithm (NNIA) is proposed for multiobjective optimization by using a novel nondominated neighbor-based selection technique, an immune inspired operator, two heuristic search operators, and elitism. The unique selection technique of NNIA only selects minority isolated nondominated individuals in the population. The selected individuals are then cloned proportionally to their crowding-distance values before heuristic search. By using the nondominated neighbor-based selection and proportional cloning, NNIA pays more attention to the less-crowded regions of the current trade-off front. We compare NNIA with NSGA-II, SPEA2, PESA-II, and MISA in solving five DTLZ problems, five ZDT problems, and three low-dimensional problems. The statistical analysis based on three performance metrics including the coverage of two sets, the convergence metric, and the spacing, show that the unique selection method is effective, and NNIA is an effective algorithm for solving multiobjective optimization problems. The empirical study on NNIA's scalability with respect to the number of objectives shows that the new algorithm scales well along the number of objectives.

**Keywords**

Multiobjective optimization, evolutionary algorithm, artificial immune system, crowding-distance, Pareto-optimal solution.

## 1 Introduction

Many real-word problems have several objectives to be optimized at the same time. The simultaneous optimization of multiple objectives is different from single objective optimization in that there is no unique solution to multiobjective optimization problems (MOPs), but instead, we aim at finding all of the good trade-off solutions which

 

must be considered equivalent in the absence of information concerning the relevance of each objective relative to the others. Evolutionary algorithms (EAs) have been recognized to be well suited to multiobjective optimization since early in their development because they deal simultaneously with a set of possible solutions. The ability to handle complex problems, involving features such as discontinuities, multimodality, disjoint feasible spaces, and noisy function evaluations, reinforces the potential effectiveness of EAs in multiobjective optimization (Fonseca and Fleming, 1995). The vector evaluated genetic algorithm (Schaffer, 1984) was probably the first multiobjective optimization EA (MOEA) to search for multiple Pareto-optimal solutions concurrently in a single run. Since the mid 1990s, the amount of literature about MOEAs increased greatly and many MOEAs were proposed one after another. The Niched Pareto Genetic Algorithm (Horn and Nafpliotis, 1993) and the Nondominated Sorting Genetic Algorithm (Srinivas and Deb, 1993) were representative of them. These MOEAs were characterized by the use of selection mechanisms based on Pareto ranking and fitness sharing to maintain diversity (Coello Coello, 2003). In the past few years, some MOEAs using the elitism strategy were presented, such as the Strength Pareto Evolutionary Algorithm (SPEA; Zitzler and Thiele, 1999), the Pareto Archived Evolution Strategy (PAES; Knowles and Corne, 2000), the Pareto Envelope based Selection Algorithm (PESA; Corne et al., 2000), the Multi-Objective Messy Genetic Algorithm (MOMGA; Van Veldhuizen and Lamont, 2000), the revised version of PESA with region-based selection (PESA-II; Corne et al., 2001), the Micro Genetic Algorithm (microGA; Coello Coello and Pulido, 2001), the improved version of NSGA (NSGA-II) with a more efficient nondominated sorting method, elitism, and a crowded comparison operator without specifying any additional parameters for diversity maintaining (Deb, Pratap, et al., 2002), and the improved version of SPEA (SPEA2) with a revised fitness assignment strategy, a nearest neighbor density estimation technique, and an enhanced archive truncation method (Zitzler et al., 2002). These MOEAs can be considered to be different MOEAs in the sense of their different ways to do selection (or fitness assignment) and population maintenance in multiobjective spaces. Coello Coello maintains an evolutionary multiobjective optimization repository at (www.lania.mx/˜ccoello/EMOO) in which almost all of these algorithms can be found.

The human immune system (HIS) is a highly evolved, parallel, and distributed adaptive system. The information processing abilities of the HIS provide important aspects in the field of computation. This emerging field is referred to as the Artificial Immune Systems (AIS) (Tarakanov and Dasgupta, 2000). In recent years, AIS have received a significant amount of interest from researchers and industrial sponsors. Applications of AIS include such areas as machine learning, fault diagnosis, computer security, and optimization (Nicosia et al., 2004; Jacob et al., 2005). Recently, Coello Coello proposed an artificial immune system algorithm MISA (Coello Coello and Cortes, 2002, 2005) based on the clonal selection principle (Burnet, 1959) to solve multiobjective optimization problems. We also proposed an immune algorithm IDCMA (Jiao et al., 2005) which is the groundwork for this paper. Both the two algorithms adopted binary representation. Freschi and Repetto (2005) proposed a Vector Artificial Immune System (VAIS) based on the multimodal optimization algorithm opt-aiNet (de Castro and Timmis, 2002).

In this paper, we propose a novel multiobjective algorithm, the Nondominated Neighbor Immune Algorithm (NNIA). In NNIA, the fitness value of each nondominated individual is assigned as the average distance of two nondominated individuals on either side of this individual along each of the objectives, namely, the crowding-distance defined by Deb, Pratap, et al. (2002). According to the fitness values, only partial nondominated individuals with greater crowding-distance values are selected

to do proportional cloning, recombination, and hypermutation. So in a single generation, NNIA pays more attention to the less-crowded regions (referred to in Section 2.3) in the current trade-off front.

The remainder of this paper is organized as follows: Section 2 describes related background including multiobjective optimization, immune system inspired optimization algorithms, and three terms used in this paper. Section 3 describes the main loop of NNIA. The fitness assignment, population evolution, and computational complexity of NNIA are also analyzed in Section 3. In Section 4, five DTLZ problems (Deb, Thiele, et al., 2002), five ZDT problems (Zitzler et al., 2000), and three low-dimensional problems are used to evaluate NNIA's effectiveness by comparing with NSGA-II, SPEA2, PESA-II, and MISA, based on three performance metrics, the coverage of two sets (Zitzler and Thiele, 1998), the convergence metric (Deb and Jain, 2002), and the spacing (Schott, 1995). The comparison of NNIA with and without recombination, and NNIA's scalability along the number of objectives, will also be investigated. In Section 5, concluding remarks are presented.

## 2 Related Background

### 2.1 Multiobjective Optimization

Multiobjective Optimization (Deb, 2001; Coello Coello et al., 2002) seeks to optimize a vector of functions,

$$\mathbf{F}(\mathbf{x}) = \big(f_1(\mathbf{x}),\, f_2(\mathbf{x}),\, \ldots,\, f_k(\mathbf{x})\big)^{\mathrm{T}} \tag{1}$$

subject to $\mathbf{x} = (x_1, x_2, \ldots, x_m) \in \Omega$, where $\mathbf{x}$ is the decision vector, and $\Omega$ is the feasible region in decision space.

Considering a maximization problem for each objective, it is said that a decision vector $\mathbf{x}_A \in \Omega$ dominates another vector $\mathbf{x}_B \in \Omega$ (written as $\mathbf{x}_A \succ \mathbf{x}_B$) if and only if

$$\forall i = 1, 2, \ldots, k \quad f_i(\mathbf{x}_A) \geq f_i(\mathbf{x}_B) \wedge \exists j = 1, 2, \ldots, k \quad f_i(\mathbf{x}_A) > f_i(\mathbf{x}_B) \tag{2}$$

We say that a vector of decision variables $\mathbf{x}^* \in \Omega$ is a Pareto-optimal solution or nondominated solution if there does not exist another $\mathbf{x} \in \Omega$ such that $\mathbf{x} \succ \mathbf{x}^*$.

Then the Pareto-optimal set is defined as

$$\mathbf{P}^* \triangleq \{\mathbf{x}^* \in \Omega | \neg \exists \mathbf{x} \in \Omega, \mathbf{x} \succ \mathbf{x}^*\} \tag{3}$$

So the Pareto-optimal set is the set of all Pareto-optimal solutions. The corresponding image of the Pareto-optimal set under the objective function space

$$\mathbf{PF}^* \triangleq \left\{\mathbf{F}(\mathbf{x}^*) = \big(f_1(\mathbf{x}^*),\, f_2(\mathbf{x}^*),\, \ldots,\, f_k(\mathbf{x}^*)\big)^{\mathrm{T}} \mid \mathbf{x}^* \in \mathbf{P}^*\right\} \tag{4}$$

is called the Pareto-optimal front. The aim of an MOEA is to find a set of Pareto-optimal solutions approximating the true Pareto-optimal front.

### 2.2 Immune System Inspired Optimization Algorithms

The immune system's ability to adapt its B-cells to new types of antigens is powered by processes known as clonal selection and affinity maturation by hypermutation (Garrett,

2005). The majority of immune system inspired optimization algorithms are based on the applications of the clonal selection and hypermutation (Hart and Timmis, 2005). The first immune optimization algorithm may be Fukuda et al. (1993), which included an abstraction of clonal selection to solve computational problems. But the clonal selection algorithm for optimization has been popularized mainly by de Castro and Von Zuben's CLONALG (de Castro and Von Zuben, 2002). CLONALG selects the fittest antibodies to clone proportionally to their antigenic affinities. The hypermutation operator performs an affinity maturation process inversely proportional to the fitness values generating the matured clone population. After computing the antigenic affinity of the matured clone population, CLONALG randomly creates new antibodies to replace the lowest fitness antibodies in the current population and to retain the best antibodies to recycle. de Castro and Timmis (2002) proposed an artificial immune network called opt-aiNet for multimodal optimization. In opt-aiNet, antibodies are part of an immune network and the decision about the individual which will be cloned, suppressed, or maintained depends on the interaction established by the immune network. Garrett (2004) has presented an attempt to remove all the parameters from the clonal selection algorithm. This method, which is called ACS for short, attempts to self-evolve various parameters during a single run. Cutello and Nicosia proposed an immune algorithm for optimization called opt-IA (Cutello et al., 2004; Cutello, Narzisi, Nicosia, et al., 2005). Opt-IA uses three immune operators, cloning, hypermutation, and aging. In the hypermutation operator, the number of mutations is determined by mutation potential. The aging operator eliminates old individuals to avoid premature convergence. Opt-IA also uses a standard evolutionary operator, the $(\mu + \lambda)$-selection operator. As far as multiobjective optimization is concerned, MISA (Coello Coello and Cortes, 2002, 2005) may be the first attempt to solve general multiobjective optimization problems using artificial immune systems. MISA encodes the decision variables of the problem to be solved by binary strings, clones the Pareto-optimal and feasible solutions, and applies two types of mutation to the clones and other individuals, respectively. More recently, Freschi and Repetto (2005) proposed a Vector Artificial Immune System (VAIS) for solving multiobjective optimization problems based on the opt-aiNet. VAIS adopted the flowchart of opt-aiNet and the fitness assignment method in SPEA2 with some simplification that for nondominated individuals the fitness is the strength defined in SPEA2 and for dominated individuals the fitness is the number of individuals which dominate them. Cutello, Narzisi, and Nicosia (2005) modified the (1+1)-PAES using the two immune inspired operators, cloning and hypermutation, and applied the improved PAES to solving the protein structure prediction problem. In order to compare these algorithms clearly, we summarize their key techniques in Table 1.

In Jiao et al. (2005), we proposed an immune algorithm for multiobjective optimization called the Immune Dominance Clonal Multiobjective Algorithm (IDCMA). IDCMA assigned the fitness values of current dominated individuals as the values of a custom distance measure, termed as Ab-Ab affinity, between the dominated individuals and one of the nondominated individuals found so far. According to the values of Ab-Ab affinity, all dominated individuals (antibodies) are divided into two kinds, subdominant antibodies and cryptic antibodies. A heuristic search only applies to the subdominant antibodies while the cryptic antibodies are redundant and have no function during search, but they can become subdominant (active) antibodies in the subsequent evolution. But recently, we found that the IDCMA, which adopts binary string representation and the Ab-Ab affinity based selection on dominated individuals, had difficulties in converging to the true Pareto-optimal front and obtaining the well-distributed solutions for

Table 1: Main Optimization Algorithms Inspired by Immune System

| Algorithm | Fitness assignment | Key components | Applications |
|---|---|---|---|
| CLONALG | The objective function to be optimized | Elitist selection, cloning, mutation, death | Single objective optimization problems (SOPs) |
| Opt-aiNet | The objective function to be optimized and the Euclidean distance between two individuals | Elitist selection, cloning, mutation, network suppression, death | SOPs |
| ACS | The objective function to be optimized | Elitist selection, cloning, mutation, death | SOPs |
| opt-IA | The objective function to be optimized | $(\mu + \lambda)$-selection, cloning, hypermutation, hypermacromutation, aging | SOPs |
| MISA | The objective functions and the Pareto dominance relationships, the Euclidean distance between two individuals | Elitist selection according to fitness and diversity, cloning, uniform mutation and nonuniform mutation, archive update | MOPs |
| VAIS | For Pareto-optimal individuals the fitness is the strength defined in SPEA2 and for dominated individuals the fitness is the number of individuals which dominate them | Cloning, mutation, clonal selection, suppression, death, archive update | MOPs |
| I-PAES | The Pareto dominance relationships | Cloning, mutation, clonal selection, (1+1)-selection, archive update | MOPs |

some complicated problems, for example, the DTLZ problems (Deb, Thiele, et al., 2002). So we modified the IDCMA by using real-coded representation, and a new selection technique and population maintenance strategy, the NNIA, was proposed accordingly.

### 2.3 Summary of Related Terms

In this paper, we follow the nomenclature of immunology and define the terms as follows.

#### 2.3.1 Antibody and Antibody Population

For the MOP

$$\begin{cases} \max \mathbf{F}(\mathbf{x}) = \big(f_1(\mathbf{x}),\ f_2(\mathbf{x}), \ldots,\ f_k(\mathbf{x})\big)^{\mathrm{T}} \\ \text{subject to } \mathbf{x} \in \Omega \end{cases} \tag{5}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_m)$, $\Omega$ is the feasible region, and $k \geq 2$, an antibody $\mathbf{b} = (b_1, b_2, \ldots, b_l)$ is the coding of variable $\mathbf{x}$, denoted by $\mathbf{b} = e(\mathbf{x})$, and $\mathbf{x}$ is called the decoding of antibody $\mathbf{b}$, expressed as $\mathbf{x} = e^{-1}(\mathbf{b})$. In this study, we adopt real-valued presentation, that is, $\mathbf{b} = e(\mathbf{x}) = \mathbf{x}$, so $l = m$, and $\mathbf{b} \in \Omega$. An antibody population

$$\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}, \quad \mathbf{b}_i \in \Omega, \quad 1 \leq i \leq n \tag{6}$$

is an $n$-dimensional group of antibody $\mathbf{b}$, where the positive integer $n$ is the size of antibody population $\mathbf{B}$.

#### 2.3.2 Dominant Antibody

For the MOP in equation (5), the antibody $\mathbf{b}_i$ is a dominant antibody in the antibody population $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$, if and only if there is no antibody $\mathbf{b}_j \in \mathbf{B}$ that satisfies

$$\begin{aligned} &\forall p = 1, 2, \ldots, k \quad f_p\big(e^{-1}(\mathbf{b}_j)\big) \geq f_p\big(e^{-1}(\mathbf{b}_i)\big) \wedge \\ &\exists q = 1, 2, \ldots, k \quad f_q\big(e^{-1}(\mathbf{b}_j)\big) > f_q\big(e^{-1}(\mathbf{b}_i)\big) \end{aligned} \tag{7}$$

So the dominant antibodies are the nondominated individuals in population $\mathbf{B}$. In this paper, we denote the set of dominant antibodies as $\mathbf{D}$, and denote dominant antibodies as $\mathbf{d}$ with different suffixes.

In order to explain the concepts of antibody, antibody population, and dominant antibody, we give a simple example as follows. For the MOP in equation (5), if a vector $\mathbf{x}_1 = (0.5, 0.2, 4, 5)$ belongs to the feasible region $\Omega$, then $\mathbf{x}_1$ is a candidate solution of the MOP, and the corresponding antibody is denoted by $\mathbf{b}_1 = (0.5, 0.2, 4, 5)$. If $\mathbf{b}_1 = (0.5, 0.2, 4, 5)$, $\mathbf{b}_2 = (0.7, 0.6, 4, 7)$, and $\mathbf{b}_3 = (0.2, 0.6, 6, 1)$ are three antibodies, then the set $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\}$ is an antibody population with size 3. In the antibody population $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\}$, if both $\mathbf{b}_2$ and $\mathbf{b}_3$ do not dominate $\mathbf{b}_1$, then $\mathbf{b}_1$ is a dominant antibody in the antibody population $\mathbf{B}$.

#### 2.3.3 Crowding-Distance

In our multiobjective algorithm, the dominant antibodies in $\mathbf{D}$ are ranked according to how much they contribute to the diversity of objective function values. This can be measured by the crowding-distance (Deb, Pratap, et al., 2002). For the MOP in equation (5), the crowding-distance of a dominant antibody $\mathbf{d} \in \mathbf{D}$ is given by

$$\zeta(\mathbf{d}, \mathbf{D}) \triangleq \sum_{i=1}^{k} \frac{\zeta_i(\mathbf{d}, \mathbf{D})}{f_i^{\max} - f_i^{\min}} \tag{8}$$

where $f_i^{\max}$ and $f_i^{\min}$ are the maximum and minimum value of the $i$th objective and

$$\zeta_i(\mathbf{d}, \mathbf{D}) = \begin{cases} \infty, \text{ if } f_i(\mathbf{d}) = \min\{f_i(\mathbf{d}') | \mathbf{d}' \in \mathbf{D}\} \text{ or } f_i(\mathbf{d}) = \max\{f_i(\mathbf{d}') | \mathbf{d}' \in \mathbf{D}\} \\ \min\{f_i(\mathbf{d}') - f_i(\mathbf{d}'') | \mathbf{d}', \mathbf{d}'' \in \mathbf{D} : f_i(\mathbf{d}'') < f_i(\mathbf{d}) < f_i(\mathbf{d}')\}, \qquad \text{otherwise} \end{cases} \tag{9}$$

Based on the crowding-distance $\zeta(\mathbf{d}, \mathbf{D})$, we can estimate the density of dominant antibodies surrounding $\mathbf{d}$ in the population $\mathbf{D}$. If $\zeta(\mathbf{d}, \mathbf{D}) > \zeta(\mathbf{d}', \mathbf{D})$, $\mathbf{d}, \mathbf{d}' \in \mathbf{D}$, then $\mathbf{d}$ is a less-crowded individual, and $\mathbf{d}$ lies in a less-crowded region of the trade-off front, in contrast to $\mathbf{d}'$.

## 3 Description of the Algorithm

In this section, we describe a novel multiobjective optimization algorithm, the NNIA. NNIA stores nondominated individuals found so far in an external population, called the dominant population. Only partial less-crowded nondominated individuals, called

active antibodies, are selected to do proportional cloning, recombination, and static hypermutation (Cutello et al., 2004). Furthermore, the population storing clones is called the clone population. The dominant population, active population, and clone population at time $t$ are represented by time-dependent variable matrices $\mathbf{D}_t$, $\mathbf{A}_t$, and $\mathbf{C}_t$, respectively. The main loop of NNIA is as follows.

***Algorithm1**: Nondominated Neighbor Immune Algorithm*

***Input**:* $G_{\max}$ *(maximum number of generations)*
$n_{\mathrm{D}}$ *(maximum size of dominant population)*
$n_{\mathrm{A}}$ *(maximum size of active population)*
$n_{\mathrm{C}}$ *(size of clone population)*
***Output**:* $\mathbf{D}_{G\max+1}$ *(final approximate Pareto-optimal set)*

*Step1:* **Initialization***: Generate an initial antibody population* $\mathbf{B_0}$ *with size* $n_{\mathrm{D}}$*. Create the initial* $\mathbf{D}_0 = \phi$, $\mathbf{A}_0 = \phi$, *and* $\mathbf{C}_0 = \phi$*. Set* $t = 0$.

*Step2:* **Update Dominant Population***: Identify dominant antibodies in* $\mathbf{B}_t$*. Copy all the dominant antibodies to form the temporary dominant population (denoted by* $\mathbf{DT}_{t+1}$*). If the size of* $\mathbf{DT}_{t+1}$ *is not greater than* $n_{\mathrm{D}}$*, let* $\mathbf{D}_{t+1} = \mathbf{DT}_{t+1}$*. Otherwise, calculate the crowding-distance values of all individuals in* $\mathbf{DT}_{t+1}$*, sort them in descending order of crowding-distance, and choose the first* $n_{\mathrm{D}}$ *individuals to form* $\mathbf{D}_{t+1}$.

*Step3:* **Termination***: If* $t \geq Gmax$ *is satisfied, export* $\mathbf{D}_{t+1}$ *as the output of the algorithm, Stop; Otherwise,* $t = t + 1$.

*Step4:* **Nondominated Neighbor−Based Selection***: If the size of* $\mathbf{D}_t$ *is not greater than* $n_A$*, let* $\mathbf{A}_t = \mathbf{D}_t$*. Otherwise, calculate the crowding-distance values of all individuals in* $\mathbf{D}_t$*, sort them in descending order of crowding-distance, and choose the first* $n_{\mathrm{A}}$ *individuals to form* $\mathbf{A}_t$.

*Step5:* **Proportional Cloning***: Get the clone population* $\mathbf{C}_t$ *by applying proportional cloning to* $\mathbf{A}_t$.

*Step6:* **Recombination and Hypermutation***: Perform recombination and hypermutation on* $\mathbf{C}_t$ *and set* $\mathbf{C}'_t$ *to the resulting population.*

*Step7: Get the antibody population* $\mathbf{B}_t$ *by combining the* $\mathbf{C}'_t$ *and* $\mathbf{D}_t$*; go to Step2.*

When the number of dominant antibodies is greater than the maximum limitation and the size of dominant population is greater than the maximum size of active population, both the reduction of dominant population and the selection of active antibodies use the crowding-distance based truncation selection. The proportional cloning, recombination, and hypermutation operators are described as follows.

### 3.1 Proportional Cloning

In immunology, cloning 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. Following the works summarized in Section 2.2, in this study, the proportional cloning $\mathbf{T}^C$ on the active

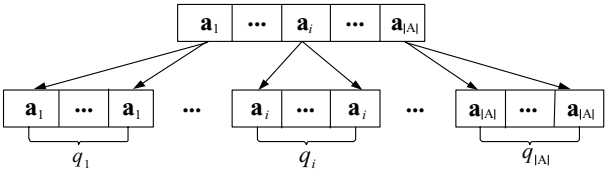


Figure 1: Illustration of the proportional cloning.

population $\mathbf{A} = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_{|\mathbf{A}|}\}$ is defined as

$$\begin{aligned}\mathbf{T}^C(\mathbf{a}_1 + \mathbf{a}_2 + \cdots + \mathbf{a}_{|\mathbf{A}|}) &= \mathbf{T}^C(\mathbf{a}_1) + \mathbf{T}^C(\mathbf{a}_2) + \cdots + \mathbf{T}^C(\mathbf{a}_{|\mathbf{A}|}) \\ &= \left\{\mathbf{a}_1^1 + \mathbf{a}_1^2 + \cdots + \mathbf{a}_1^{q_1}\right\} + \left\{\mathbf{a}_2^1 + \mathbf{a}_2^2 + \cdots + \mathbf{a}_2^{q_2}\right\} \\ &\quad + \cdots + \left\{\mathbf{a}_{|\mathbf{A}|}^1 + \mathbf{a}_{|\mathbf{A}|}^2 + \cdots + \mathbf{a}_{|\mathbf{A}|}^{q_{|\mathbf{A}|}}\right\} \end{aligned} \tag{10}$$

where $\mathbf{T}^C(\mathbf{a}_i) = \{\mathbf{a}_i^1 + \mathbf{a}_i^2 + \cdots + \mathbf{a}_i^{q_i}\}, \mathbf{a}_i^j = \mathbf{a}_i, i = 1, 2, \ldots, |\mathbf{A}|, j = 1, 2, \ldots, q_i$. $q_i$ is a self-adaptive parameter. The representation $+$ is not the arithmetical operator, but only separates the antibodies here. $q_i = 1$ denotes that there is no cloning on antibody $\mathbf{a}_i$.

In this study, the individual with greater crowding-distance value is reproduced more times, therefore, the individual with greater crowding-distance value has a larger $q_i$. Because the crowding-distance values of boundary solutions are positive infinity, before computing the value of $q_i$ for each active antibody, we set the crowding-distance values of the boundary individuals (in objective space) to be equal to the double values of the maximum value of active antibodies except the boundary individuals. Then the values of $q_i$ are calculated as

$$q_i = \left\lceil n_C \times \frac{\zeta(\mathbf{a}_i, \mathbf{A})}{\sum_{j=1}^{|\mathbf{A}|} \zeta(\mathbf{a}_j, \mathbf{A})} \right\rceil, \tag{11}$$

where $\zeta(\mathbf{a}_j, \mathbf{A})$ denotes the crowding-distance value of the active antibodies $\mathbf{a}_j$, $n_C$ is an expectant value of the size of the clone population. For example, suppose that there are five antibodies in an active population for solving a bi-objective optimization problem and the corresponding values of objective functions are (1.0, 0), (0.9, 0.2), (0.6, 0.4), (0.2, 0.7), and (0, 1.0). Therefore, the crowding-distance values of the five individuals are 2.4, 0.8, 1.2, 1.2, and 2.4, respectively. If $n_C = 40$, then $q_1 = \lceil 40 \times \frac{2.4}{2.4+0.8+1.2+1.2+2.4} \rceil = 12$ and $q_2 = 4, q_3 = 6, q_4 = 6, q_5 = 12$. Note that we use the ceil function here, therefore the clone population size $\sum_{j=1}^{|\mathbf{A}|} q_i$ is sometimes greater than the expectant value $n_C$. However, the subsequent update of the dominant population and nondominated neighbor-based selection make sure the size of dominant population and the active population are not greater than $n_D$ and $n_A$, respectively.

Figure 1 illustrates the procedure of proportional cloning. All the antibodies in subpopulation $\{\mathbf{a}_i^1, \mathbf{a}_i^2, \ldots, \mathbf{a}_i^{q_i}\}$ are the result of the cloning on antibody $\mathbf{a}_i$, and have the same property as $\mathbf{a}_i$. In fact, cloning on antibody $\mathbf{a}_i$ is to make multiple identical copies of $\mathbf{a}_i$. The aim is that the greater the crowding-distance value of an individual, the more times the individual will be reproduced. So there exist more chances to search in less-crowded regions of the trade-off front.

### 3.2 Recombination and Hypermutation

If $\mathbf{C} = (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \ldots, \mathbf{c}_{|\mathbf{C}|})$ is the resulting population from applying proportional cloning to $\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \ldots, \mathbf{a}_{|\mathbf{A}|})$, then the recombination $\mathbf{T}^R$ on the clone population $\mathbf{C}$ is defined as

$$\begin{aligned}
&\mathbf{T}^R(\mathbf{c}_1 + \mathbf{c}_2 + \cdots + \mathbf{c}_{|\mathbf{C}|}) \\
&\quad = \mathbf{T}^R(\mathbf{c}_1) + \mathbf{T}^R(\mathbf{c}_2) + \cdots + \mathbf{T}^R(\mathbf{c}_{|\mathbf{C}|}) \\
&\quad = \mathit{crossover}(\mathbf{c}_1, \mathbf{A}) + \mathit{crossover}(\mathbf{c}_2, \mathbf{A}) + \cdots + \mathit{crossover}(\mathbf{c}_{|\mathbf{C}|}, \mathbf{A})
\end{aligned} \tag{12}$$

where $\mathit{crossover}(\mathbf{c}_i, \mathbf{A})$, $i = 1, 2, \ldots, |\mathbf{C}|$, denotes selecting equiprobably one individual from the two offspring generated by a general crossover operator on clone $\mathbf{c}_i$ and an active antibody selected randomly from $\mathbf{A}$.

In this study, we use static hypermutation operator (Cutello et al., 2004) on the clone population after recombination. Cutello et al. (2004) designed three hypermutation methods, namely, static hypermutation (the number of mutations is independent of the fitness values), proportional hypermutation (the number of mutations is proportional to the fitness value), and inversely proportional hypermutation (the number of mutations is inversely proportional to the fitness value). We chose static hypermutation in our algorithm for the following reasons:

(1) The hypermutation operator is implemented on the clone population after recombination. If we chose proportional hypermutation or inversely proportional hypermutation, we have to calculate the fitness values for all the individuals of the clone population. However, in other phases of NNIA, the dominated individuals are not assigned fitness. Therefore, we have to define a fitness assignment strategy to dominated individuals only for the hypermuation operator unless we use a mutation operator independent from the fitness values.

(2) In order to reduce complexity, the number of fitness evaluations is as low as possible. Suppose we were to define a fitness assignment strategy suitable for proportional hypermutation or inversely proportional hypermutation, then we would have to calculate the fitness values of all recombined clones before mutation.

(3) The experimental study in Cutello et al. (2004) showed that inversely proportional hypermutation did only slightly better than static and proportional hypermutation in solving the Trap function problems. But if the combination of multiple operators was not considered, static hypermutation achieved the best results among the three hypermutation operators in solving protein structure prediction problems.

Following from the above reasons, we adopt the static hypermutation operator. If $\mathbf{R} = (\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \ldots, \mathbf{r}_{|\mathbf{R}|})$ is the clone population after recombination, then the static hypermutation operator $\mathbf{T}^H$ on the population $\mathbf{R}$ is defined as

$$\begin{aligned}
\mathbf{T}^H(\mathbf{r}_1 + \mathbf{r}_2 + \cdots + \mathbf{r}_{|\mathbf{R}|}) &= \mathbf{T}^H(\mathbf{r}_1) + \mathbf{T}^H(\mathbf{r}_2) + \cdots + \mathbf{T}^H(\mathbf{r}_{|\mathbf{R}|}) \\
&= \mathit{mutate}(\mathbf{r}_1) + \mathit{mutate}(\mathbf{r}_2) + \cdots + \mathit{mutate}(\mathbf{r}_{|\mathbf{R}|})
\end{aligned} \tag{13}$$

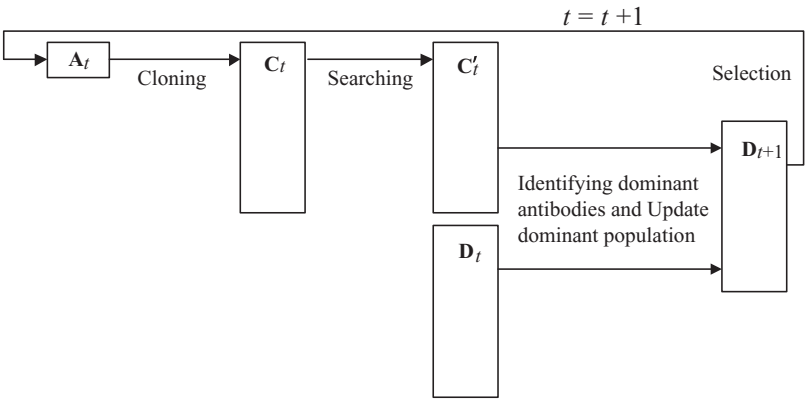


Figure 2: Population evolution of NNIA.

where $mutate(\mathbf{r}_i), i = 1, 2, \ldots, |\mathbf{R}|$, denotes changing each element of the variable vector $\mathbf{r}_i$ by a general mutation operator with probability $p_m$, so each individual in the clone population $\mathbf{R}$ at each time step will undergo about $m \times p_m$ mutations, where $m$ is the dimension of the variable vector.

### 3.3 Fitness Assignment and Population Evolution

It can be seen that NNIA uses some well-known techniques such as storing the nondominated individuals previously found externally and reducing the number of nondominated individuals stored without destroying the characteristics of the trade-off front. Furthermore, NNIA adopts a novel selection technique. In NNIA, the dominant population is set as an external population for elitism. The fitness values of the individuals in the dominant population are assigned as the values of the crowding-distance, which serves as an estimate of the perimeter of the cuboid formed by the nearest neighbors as the vertices in objective space. The selection is therefore biased toward individuals with a high isolation value. Only partial nondominated individuals (much less than nondominated individuals found so far) with high crowding-distance values are selected. And cloning, recombination, and mutation only apply to the selected individuals (active antibodies). So in a single generation, only less-crowded individuals perform heuristic search in order to obtain more solutions in the less-crowded regions of the trade-off fronts. In contrast to NSGA-II, the nondominated neighbor-based selection and the proportional cloning makes the less-crowded individuals have more chances to do recombination and mutation. The population evolution in a single generation at time $t$ is shown in figure 2.

### 3.4 Computational Complexity

Analyzing NNIA's computational complexity is revealing. In this section, we only consider population size in computational complexity. Assuming that the maximum size of the dominant population is $n_{\mathrm{D}}$, the maximum size of active population is $n_{\mathrm{A}}$, and the clone population size is $n_{\mathrm{C}}$, then the time complexity of one generation for the algorithm can be calculated as follows:

The time complexity for identifying nondominated individuals in the population is $O((n_D + n_C)^2)$; the worst time complexity for updating the dominant population is $O((n_D + n_C)\log(n_D + n_C))$; the worst time complexity for nondominated neighbor-based selection is $O(n_D \log(n_D))$; the time complexity for cloning is $O(n_C)$; and the time complexity for recombination and mutation is $O(n_C)$. So the worst total time complexity is

$$O((n_D + n_C)^2) + O((n_D + n_C)\log(n_D + n_C)) + O(n_D \log(n_D)) + 2O(n_C). \quad (14)$$

According to the operational rules of the symbol $O$, the worst time complexity of one generation for NNIA can be simplified as

$$O((n_D + n_C)^2). \quad (15)$$

So the cost of identifying the nondominated individuals in the population dominates the computational complexity of NNIA. We will do some empirical study on NNIA's runtime complexity in Section 4.4.

## 4 Evaluation of NNIA's Effectiveness

In this section, we compare PESA-II (Corne et al., 2001), NSGA-II (Deb, Pratap, et al., 2002), SPEA2 (Zitzler et al., 2002), and MISA (Coello Coello and Cortes, 2005) with NNIA in solving 13 well-known multiobjective function optimization problems including three low-dimensional bi-objective problems, five ZDT problems (Zitzler et al., 2000), and five DTLZ problems (Deb, Thiele, et al., 2002). The NNIA toolbox for Matlab 7.0 designed by the authors for solving the 13 problems and 20 other problems is available at the first author's homepage (http://see.xidian.edu.cn/iiip/mggong/Projects/NNIA.htm). All the simulations were run on a personal computer with P-IV 3.2G CPU and 2G RAM.

### 4.1 Experimental Setup

First, we describe the 13 test problems used in this study. The first three low-dimensional bi-objective problems, named SCH, DEB, and KUR, were defined by Schaffer (1984), Deb (1999), and Kursawe (1991), respectively. The next five ZDT problems were developed by Zitzler et al. (2000). The last five DTLZ problems were developed by Deb, Thiele, et al. (2002). These MOPs have been cited in a number of significant studies in this area. The first three MOPs are either simple or not scalable, where the number of decision variables is no more than three. The five ZDT problems have 30 or 10 decision variables. All the above eight problems have two objectives. The five DTLZ problems can be scaled to any number of decision variables and objectives. The Pareto-optimal fronts illustrated by a set of 200 uniform points are shown in figure 3. The Pareto-optimal fronts of the five ZDT problems and the five DTLZ problems have been mathematically defined (Zitzler et al., 2000; Deb, Thiele, et al., 2002). The three low-dimensional problems, SCH, DEB, and KUR, do not have mathematically defined Pareto-optimal fronts. However, their approximate Pareto-optimal fronts have been illustrated (Deb, 1999; Van Veldhuizen, 1999; Deb, 2001). Here we derive their approximate Pareto-optimal fronts deterministically as follows,

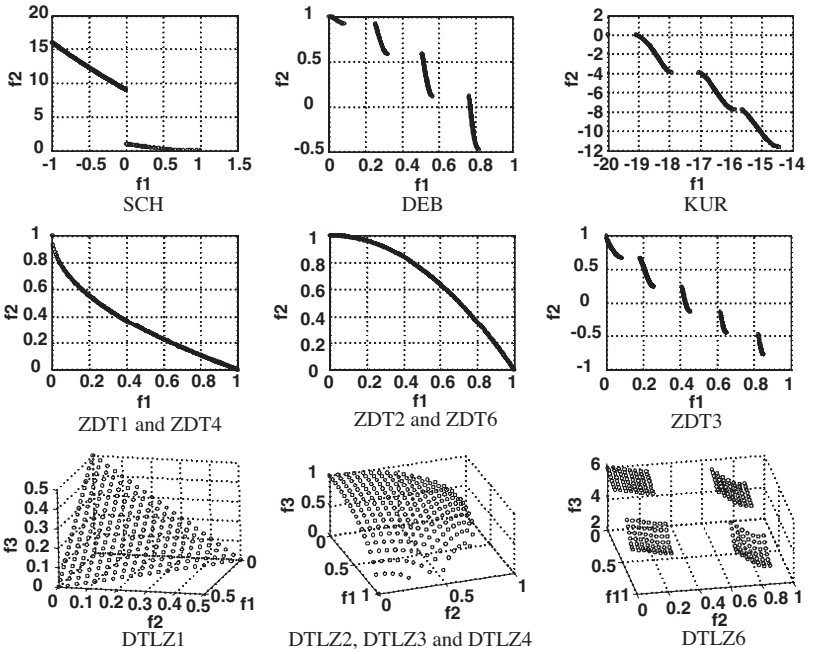


Figure 3: The Pareto-optimal fronts of the thirteen test problems illustrated by a set of 200 uniform points on them.

***Algorithm2**: The deterministic algorithm of approximating Pareto-optimal fronts*

*Step1: Generate finite discrete points uniformly distributed in the decision space.*

*Step2: Calculate the objective function values of all the points.*

*Step3: Select all the nondominated individuls from these points as the approximate Pareto-optimal set, and the corresponding image under the objective space constitutes the approximate Pareto-optimal front; Stop.*

In order to approximate the Pareto-optimal fronts with sufficient resolution, here we set the number of discrete points at $10^6$. The running time for getting the approximate Pareto-optimal fronts of SCH, DEB, and KUR is about 18 hours using a personal computer with P-IV 3.2G CPU and 2G RAM.

More details of the test problems can be found (Deb, 2001; Zitzler et al., 2000; Deb, Thiele, et al., 2002). It is necessary to note that the performance of an MOEA in tackling multiobjective constrained optimization problems may be largely dependent on the constraint-handling technique used (Van Veldhuizen, 1999), so we did not include side-constrained problems in this study. For the DTLZ problems, to be discussed in Section 4.2, we set the values of $k$ and $|\mathbf{x}_k|$ to be the values suggested by Deb et al. (2001, 2002), that is, $k = 3$ and $|\mathbf{x}_k| = 5$ for DTLZ1, $k = 3$ and $|\mathbf{x}_k| = 10$ for DTLZ2, DTLZ3, and DTLZ4, $k = 3$ and $|\mathbf{x}_k| = 20$ for DTLZ6. In Section 4.4, we will study the scalability of NNIA along the number of objectives based on two of the five DTLZ problems with $k$ increasing from 2 to 9.

Zitzler et al. (2003) suggested that for a $k$-objective optimization problem, at least $k$ performances are needed to compare two or more solutions and an infinite number of metrics to compare two or more sets of solutions. Deb and Jain (2002) suggested a running performance metric for measuring the convergence to the reference set at each generation of an MOEA run. As the reference Khare et al. (2003) describes, in Section 4.2, we apply this metric only to the final Pareto-optimal set obtained by an MOEA to evaluate its performance. Zitzler et al. (2003) and Knowles et al. (2006) have suggested that the power of unary quality indicators was restricted. So we choose a binary quality metric, the coverage of two sets (Zitzler and Thiele, 1998). We also adopt the spacing metric (Schott, 1995) to measure the diversity in the population. The three metrics are summarized as follows.

**Coverage of Two Sets**: Let $\mathbf{A}$, $\mathbf{B}$ be two approximate Pareto-optimal sets. The function $I_C$ maps the ordered pair ($\mathbf{A}$, $\mathbf{B}$) to the interval [0, 1]:

$$I_C(\mathbf{A}, \mathbf{B}) \triangleq \frac{|\{\mathbf{b} \in \mathbf{B}; \exists \mathbf{a} \in \mathbf{A} : \mathbf{a} \succeq \mathbf{b}\}|}{|\mathbf{B}|} \tag{16}$$

where $\succeq$ means dominate or equal (also called weakly dominate). The value $I_C(\mathbf{A}, \mathbf{B}) = 1$ means that all decision vectors in $\mathbf{B}$ are weakly dominated by $\mathbf{A}$. $I_C(\mathbf{A}, \mathbf{B}) = 0$ implies no decision vector in $\mathbf{B}$ is weakly dominated by $\mathbf{A}$. Note that both directions always have to be considered because $I_C(\mathbf{A}, \mathbf{B})$ is not necessarily equal to $1 - I_C(\mathbf{B}, \mathbf{A})$ .

**Convergence Metric**: Let $\mathbf{P}^* = (\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \ldots, \mathbf{p}_{|\mathbf{P}^*|})$ be the reference or target set of points on the true Pareto-optimal front and $\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \ldots, \mathbf{a}_{|\mathbf{A}|})$ be the final approximate Pareto-optimal set obtained by an MOEA. Then for each point $\mathbf{a}_i$ in $\mathbf{A}$, the smallest normalized Euclidean distance to $\mathbf{P}^*$ will be:

$$d_i = \min_{j=1}^{|\mathbf{P}^*|} \sqrt{\sum_{m=1}^{k} \left( \frac{f_m(\mathbf{a}_i) - f_m(\mathbf{p}_j)}{f_m^{\max} - f_m^{\min}} \right)^2}. \tag{17}$$

Here, $f_m^{\max}$ and $f_m^{\min}$ are the maximum and minimum values of the $m$th objective function in $\mathbf{P}^*$. The convergence metric is the average value of the normalized distance for all points in $\mathbf{A}$.

$$C(\mathbf{A}) \triangleq \frac{\sum_{i=1}^{|\mathbf{A}|} d_i}{|\mathbf{A}|} \tag{18}$$

The convergence metric represents the distance between the set of approximate Pareto-optimal solutions and the true Pareto-optimal fronts. Hence lower values of the convergence metric represent good convergence ability. Similar metrics were proposed by Schott (1995), Rudolph (1998), Zitzler et al. (2000), and Van Veldhuizen and Lamont (2000).

**Spacing**: Let $\mathbf{A}$ be the final approximate Pareto-optimal set obtained by an MOEA. The function $S$

$$S \triangleq \sqrt{\frac{1}{|\mathbf{A}| - 1} \sum_{i=1}^{|\mathbf{A}|} (\bar{d} - d_i)^2} \tag{19}$$

where

$$d_i = \min_j \left\{ \sum_{m=1}^{k} |f_m(\mathbf{a}_i) - f_m(\mathbf{a}_j)| \right\} \qquad \mathbf{a}_i, \mathbf{a}_j \in \mathbf{A} \quad i, j = 1, 2, \ldots, |\mathbf{A}|, \tag{20}$$

$\bar{d}$ is the average value of all $d_i$, and $k$ is the number of objective functions. A value of zero for this metric indicates all the nondominated solutions found are equidistantly spaced in objective space.

### 4.2 Comparison of NNIA with PESA-II, SPEA2, NSGA-II, and MISA

In order to solve MOPs, many MOEAs have been proposed as mentioned in Section 1. Following Coello Coello's recent review of the evolutionary multiobjective optimization field (Coello Coello, 2005, 2006), NSGA-II, SPEA2, PESA-II, and some others can be considered as representative of the state-of-the-art in multiobjective optimization. In this section, NNIA was compared with PESA-II, NSGA-II, SPEA2, and MISA. NSGA-II was proposed by Deb, Pratap, et al. (2002) as an improvement of NSGA by using a more efficient nondominated sorting method, elitism, and a crowded comparison operator without specifying any additional parameters for diversity maintainance. SPEA2 was proposed by Zitzler et al. (2002) as a revised version of SPEA by incorporating a revised fitness assignment strategy, a nearest neighbor density estimation technique, and an enhanced archive truncation method. The revised fitness assignment strategy takes into account the number of individuals it dominates and that it is dominated by. PESA-II was proposed by Corne et al. (2001) as a revised version of PESA by introducing a new selection technique, region-based selection. In region-based selection technique, selective fitness is assigned to the hyperboxes (Corne et al., 2001) in objective space instead of the Pareto-optimal individuals. Its update of auxiliary population (external population) also used the hyperboxes division. MISA (Coello Coello and Cortes, 2002, 2005) may be the first attempt to solve general multiobjective optimization problems using artificial immune systems. MISA encodes the decision variables of the problem to be solved by binary strings, clones the nondominated and feasible solutions, and applies two types of mutation to the clones and other individuals. MISA updates its external population by using the grid-based techniques used in PAES (Knowles and Corne, 2000).

We use the simulated binary crossover (SBX) operator and polynomial mutation (Deb and Beyer, 2001) for NNIA, PESA-II, NSGA-II, and SPEA2. The SBX and polynomial mutation has been adopted in many MOEA papers (Zitzler et al., 2002; Deb, Pratap, et al., 2002; Deb and Jain, 2002; Khare et al., 2003; Igel et al., 2007). Before the actual experimentation, some tuning of the parameters involved was required. Finding the values of parameters for which an MOEA works best is a difficult MOP in itself. We tuned the parameter values within DTLZ2 and DTLZ3 as Khare et al. (2003) for the best obtained value of the convergence metric described in Section 4.1. The tuned parameter values are listed in Table 2 where $n$ is the number of variables. Because MISA is a binary-coded algorithm, the mutation is bit-flip with the probability described in Coello Coello and Cortes (2005). For SPEA2, we use a population of size 100 and an external population of size 100. For NSGA-II, the population size is 100. For PESA-II, the internal population size is 100, the archive size is 100, and the number of hyper-grid cells per dimension is 10. For NNIA, the maximum size of dominant population $n_{\mathrm{D}} = 100$, the

Table 2: Tuned Parameter Values

| Parameter | PESA-II | SPEA2 | NSGA-II | NNIA |
|---|---|---|---|---|
| Crossover probability $p_c$ | 0.8 | 0.8 | 0.8 | 1 |
| Distribution index for SBX | 15 | 15 | 15 | 15 |
| Mutation probability $p_m$ | $1/n$ | $1/n$ | $1/n$ | $1/n$ |
| Distribution index for polynomial mutation | 20 | 20 | 20 | 20 |

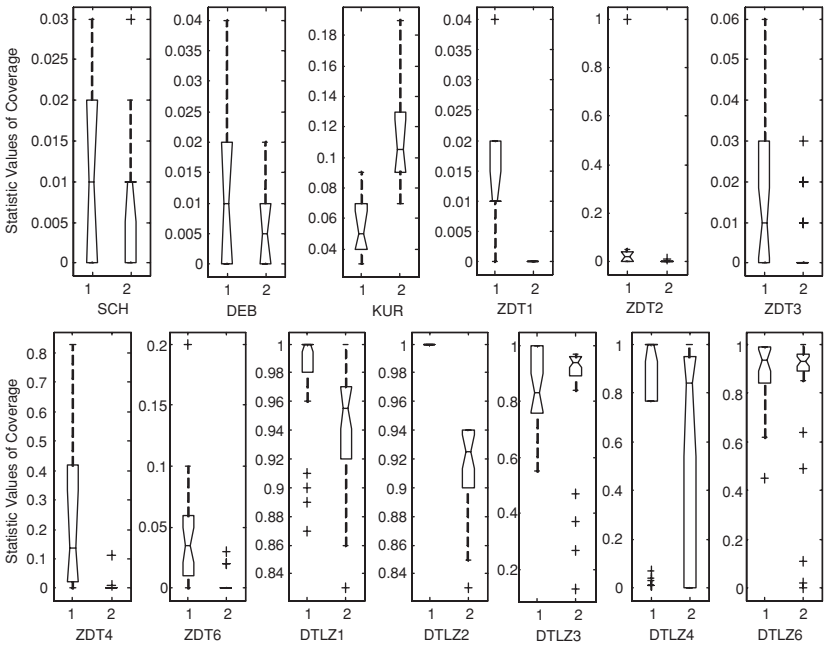


Figure 4: Statistical values of the coverage of the two sets obtained by NNIA and PESA-II in solving the 13 problems. Here, box plots are used to illustrate the distribution of these samples. In a notched box plot the notches represent a robust estimate of the uncertainty about the medians for box-to-box comparison. Symbol + denotes outliers. The 13 plots denote the results of the 13 problems respectively. In each plot, the left box represents the distribution of $I_C(\mathbf{I}, \mathbf{P})$ and the right box represents the distribution of $I_C(\mathbf{P}, \mathbf{I})$.

maximum size of active population $n_{\mathrm{A}} = 20$, and the size of clone population $n_{\mathrm{C}} = 100$. For MISA, the population size is 100, the size of the external population is 100, the total number of clones is 600, and the number of grid subdivisions is 25; these four values are suggested by Coello Coello and Cortes (2005), and the coding length for each decision variable is 30. It is difficult to formulate the optimal and evidential stop criterion for an MOEA (Coello Coello, 2005). Researchers usually stop the algorithm when it reaches a given number of iterations or function evaluations. In this section the number of function evaluations is kept at 50,000 (not including the function evaluations during initialization) for all the five algorithms.

In the following experiments, we performed 30 independent runs on each test problem. Figures 4 to 7 show the box plots (McGill et al., 1978) of NNIA against PESA-II,

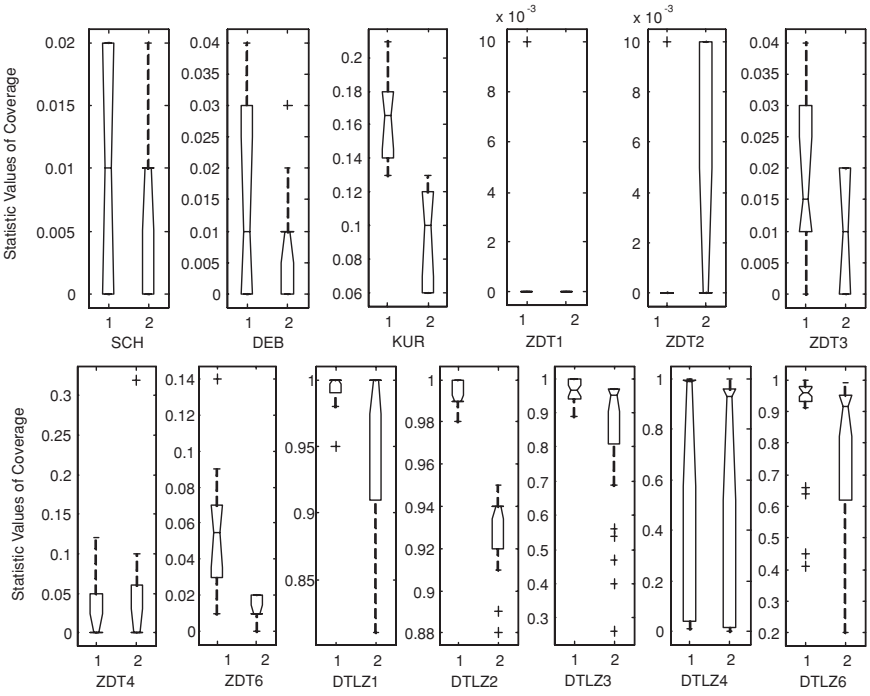


Figure 5: Box plots of the coverage of the two sets obtained by NNIA and NSGA-II in solving the 13 problems. The 13 plots denote the results of the 13 problems, respectively. In each plot, the left box represents the distribution of $I_C(\mathbf{I}, \mathbf{N})$ and the right box represents the distribution of $I_C(\mathbf{N}, \mathbf{I})$.

NSGA-II, SPEA2, and MISA based on the coverage of two sets. In the following, **I** denotes the solution set obtained by NNIA, **P** denotes the solution set obtained by PESA-II, **S** denotes the solution set obtained by SPEA2, **N** denotes the solution set obtained by NSGA-II, and **M** denotes the solution set obtained by MISA.

As the analysis in Zitzler et al. (2003) shows, $0 < I_C(\mathbf{A}, \mathbf{B}) < 1$ and $0 < I_C(\mathbf{B}, \mathbf{A}) < 1$ demonstrate that **A** does not weakly dominate **B** nor does **B** weakly dominate **A**, that is, **A** and **B** are incomparable. But if only the values of coverage are considered, the box plots of $I_C(\mathbf{I}, \mathbf{P})$ are higher than the corresponding box plots of $I_C(\mathbf{P}, \mathbf{I})$ in SCH, DEB, the five ZDT problems, and the five DTLZ problems, while the box plot of $I_C(\mathbf{P}, \mathbf{I})$ is higher than the corresponding box plot of $I_C(\mathbf{I}, \mathbf{P})$only in KUR. $I_C(\mathbf{I}, \mathbf{P})$ denotes the ratio of the number of solutions obtained by PESA-II which are weakly dominated by the solutions obtained by NNIA to the total number of the solutions obtained by PESA-II in a single run. So in a sense (but neither $\triangleright-$compatible nor $\triangleright-$complete), NNIA did better than PESA-II in SCH, DEB, the five ZDT problems, and the five DTLZ problems, while PESA-II did better than NNIA in KUR as far as the coverage is concerned.

The comparison between NNIA and NSGA-II and the comparison between NNIA and SPEA2 in terms of the coverage are similar to the comparison between NNIA and PESA-II. But NNIA did better than NSGA-II in SCH, DEB, KUR, ZDT1, ZDT3, ZDT6, and the five DTLZ problems, while NSGA-II did better than NNIA in ZDT2, and ZDT4.

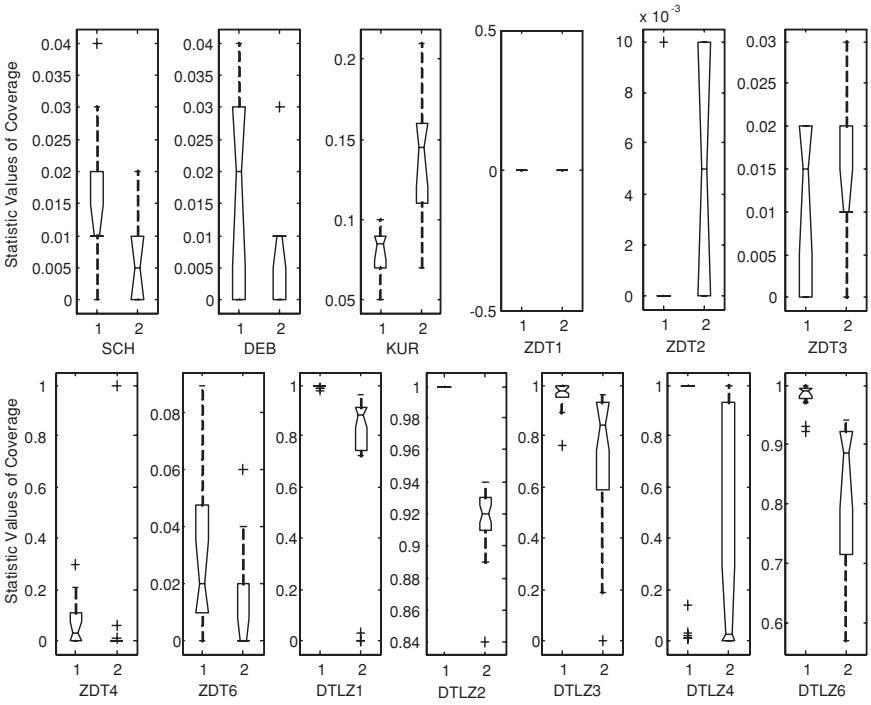


Figure 6: Box plots of the coverage of the two sets obtained by NNIA and SPEA2 in solving the 13 problems. The 13 plots denote the results of the 13 problems respectively. In each plot, the left box represents the distribution of $I_C(\mathbf{I}, \mathbf{S})$ and the right box represents the distribution of $I_C(\mathbf{S}, \mathbf{I})$.

Meanwhile, NNIA did better than SPEA2 in SCH, DEB, ZDT4, ZDT6, and the five DTLZ problems, while SPEA2 did better than NNIA in KUR, ZDT2, and ZDT3 as far as the coverage is concerned.

The comparison between NNIA and MISA shows that for the five ZDT problems and the five DTLZ problems, the majority of the solutions obtained by MISA are weakly dominated by the solutions obtained by NNIA. We estimate that NNIA did better than MISA in all the problems as far as the coverage is concerned because the box plots of $I_C(\mathbf{I}, \mathbf{M})$ are higher than the corresponding box plots of $I_C(\mathbf{M}, \mathbf{I})$ for all the test problems. The main limitation of MISA may be its binary representation. Since the test problems that we are dealing with have continuous spaces, real encoding should be preferred to avoid problems related to Hamming cliffs and to achieve arbitrary precision in the optimal solution (Khare et al., 2003). We think it is not appropriate to compare the performance of a binary-coded algorithm with respect to a real-coded algorithm, but MISA's special operators (two types of mutation in Step 8 and Step 9 of MISA) were designed for antibodies represented by binary strings. However, in a sense, MISA has the ability to approximate these real-coded algorithms by using the binary representation with enough coding length.

Figures 8 and 9 illustrate the box plots based on the convergence metric over 30 independent runs for the 13 problems. In this section, for calculating the values of the

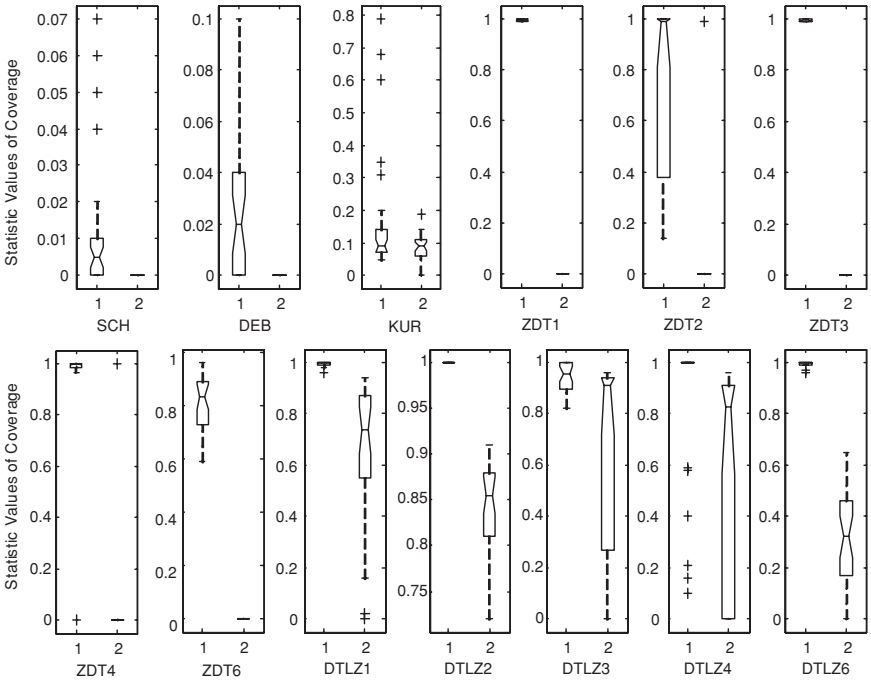


Figure 7: Box plots of the coverage of the two sets obtained by NNIA and MISA in solving the 13 problems. The 13 plots denote the results of the 13 problems respectively. In each plot, the left box represents the distribution of $I_C(\mathbf{I}, \mathbf{M})$ and the right box represents the distribution of $I_C(\mathbf{M}, \mathbf{I})$.

convergence metric, we use the set of 200 uniform points on the Pareto-optimal front as shown in Figure 3.

Figures 8 and 9 show that, for the three low-dimensional problems, all the five algorithms can obtain values less than $10^{-2}$ in almost all the 30 independent runs. For the five ZDT problems, DTLZ2, DTLZ4, and DTLZ6, the differences between the corresponding values obtained by NNIA, PESA-II, NSGA-II, and SPEA2 are small. Hereinto, NNIA did a little better than the others in ZDT1, ZDT4, ZDT6, and DTLZ2. PESA-II did a little better than the others in ZDT3 and DTLZ6. NNIA and PESA-II obtained similar values in ZDT2. NNIA, NSGA-II, and SPEA2 obtained similar values in DTLZ4. For DTLZ1 and DTLZ3, NNIA did much better than the other four algorithms even though DTLZ1 and DTLZ3 have $(11^{|\mathbf{x}_k|-1})$ and $(3^{|\mathbf{x}_k|-1})$ local Pareto-optimal fronts, respectively. Deb et al. (2001) and Khare et al. (2003) claimed that for DTLZ3, both NSGA-II and SPEA2 could not quite converge onto the true Pareto-optimal fronts in 500 generations (50,000 function evaluations). We have found that PESA-II also did badly in solving DTLZ3, but NNIA did very well. Overall, as far as the convergence metric is concerned, NNIA did best in DEB, ZDT1, ZDT4, ZDT6, DTLZ1, DTLZ2, and DTLZ3 (7 out of the 13 problems).

Figures 10 and 11 illustrate the box plots based on the spacing metric over 30 independent runs for the 13 problems.

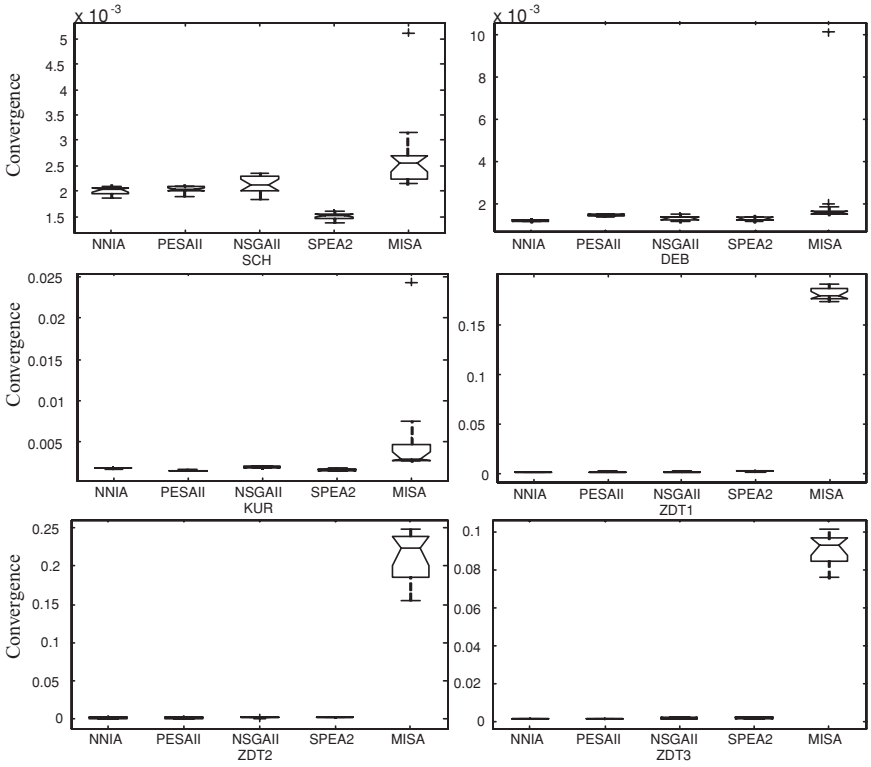


Figure 8: Box plots of the convergence metric obtained by NNIA, PESA-II, NSGA-II, SPEA2, and MISA in solving the SCH, DEB, KUR, ZDT1, ZDT2, and ZDT3.

Figures 10 and 11 show that SPEA2 did best in nine problems in terms of the diversity metric spacing. SPEA2 used an expensive archive truncation procedure whose worst runtime complexity is $O(N^3)$, where $N$ is the number of nondominated individuals. PESA-II and MISA reduced their nondominated individuals using a hyper-grid based scheme, whose grid size was a crucial factor (Khare et al., 2003). NNIA reduced nondominated solutions using the crowded comparison procedure (Deb, Pratap, et al., 2002), whose worst runtime complexity is only $O(N \log(N))$. Except for SPEA2, the box plots of spacing obtained by NNIA are lower than those obtained by NSGA-II, PESA-II, and MISA in DEB, ZDT1, ZDT3, ZDT4, ZDT6, DTLZ1, DTLZ3, and DTLZ4 (8 out of the 13 problems), while PESA-II did the best (excepting SPEA2) in ZDT2, DTLZ2, and DTLZ6, NSGA-II did the best (excepting SPEA2) in KUR, and MISA did the best (excepting SPEA2) in SCH. For DTLZ1 and DTLZ3, NNIA did the best in all the five algorithms because the other four algorithms could not quite converge onto the true Pareto-optimal fronts. Overall, SPEA2 is the best algorithm in diversity maintenance, but the differences between the values of spacing obtained by NNIA, PESA-II, and NSGA-II are inconspicuous.

Overall, considering the experimental results, we can conclude that

(1) For the three low-dimensional problems, all the five algorithms were capable of approximating the true Pareto-optimal fronts.

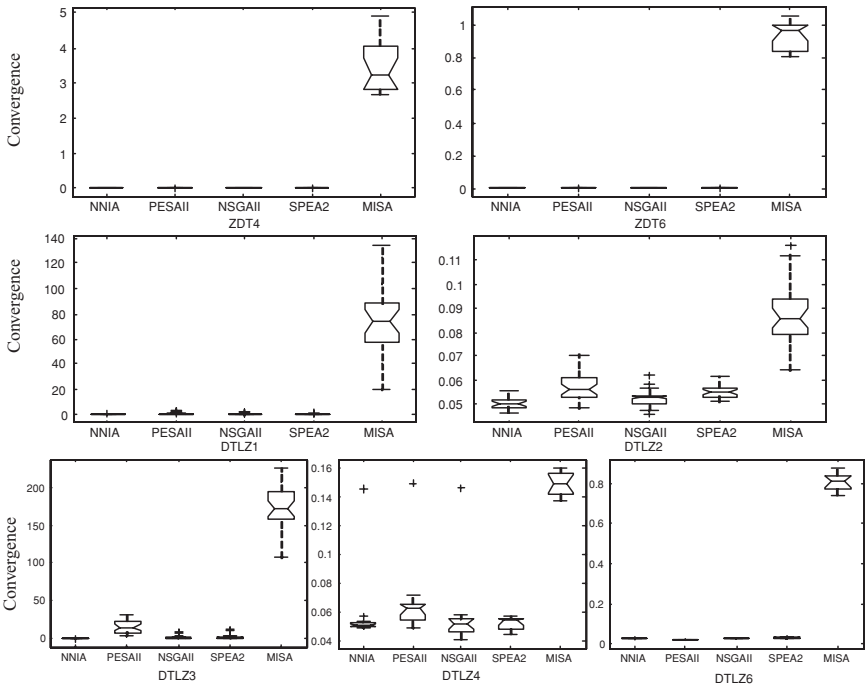


Figure 9: Box plots of the convergence metric obtained by NNIA, PESA-II, NSGA-II, SPEA2, and MISA in solving the ZDT4, ZDT6, and DTLZ problems.

(2) For DTLZ2, DTLZ4, DTLZ6, and the five ZDT problems, NNIA did the best in four out of the eight problems in terms of the convergence metric. SPEA2 did the best in seven out of the eight problems in terms of diversity maintenance. Except for SPEA2, NNIA did the best in six out of the eight problems in terms of diversity maintenance.

(3) For DTLZ1 and DTLZ3 problems, NSGA-II, SPEA2, PESA-II, and MISA could not quite converge on the true Pareto-optimal fronts in 50,000 function evaluations, but NNIA were capable of approximating the true Pareto-optimal fronts.

In the experiments, NNIA and NSGA-II adopt the same heuristic search operators (simulated binary crossover and polynomial mutation) and archive maintenance method. NNIA is different from NSGA-II only in its individual selection and cloning before heuristic search. Therefore, the better performance of NNIA in contrast to NSGA-II results from the unique nondominated neighbor-based selection method cooperating with the proportional cloning, which causes the less-crowded individuals to have more chances to do a heuristic search. NNIA is different from PESA-II, SPEA2, and MISA both in its way to do selection and in archive maintenance. NNIA also outperformed PESA-II, SPEA2, and MISA in convergence for the majority of the test problems. Depending on these empirical comparisons, we conclude that the nondominated neighbor-based selection technique is effective, and NNIA is an effective algorithm for solving MOPs.

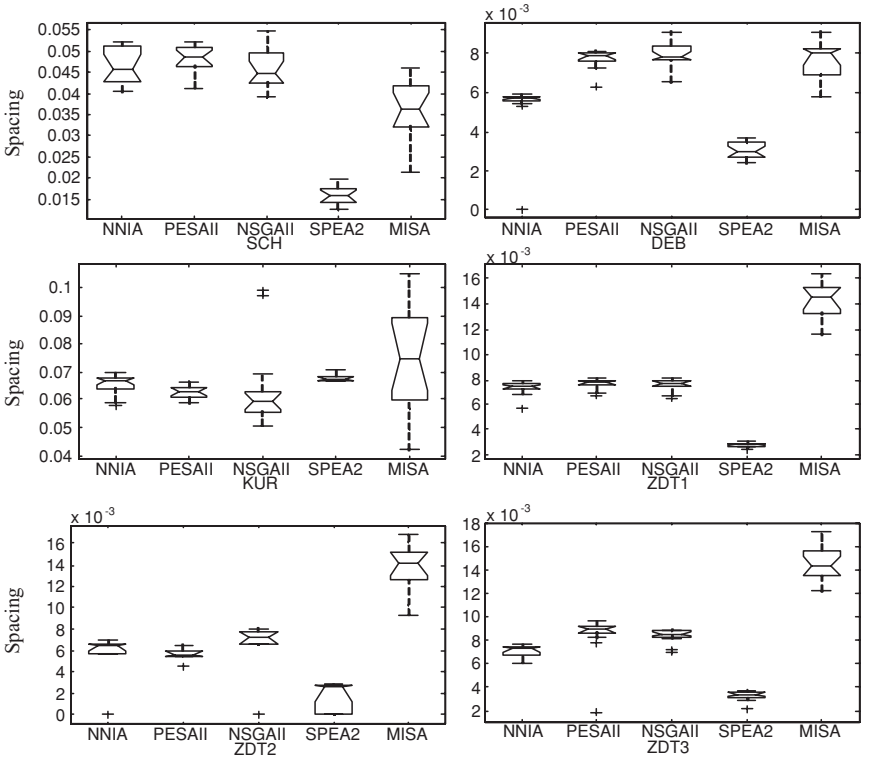


Figure 10: Box plots of the spacing metric obtained by NNIA, PESA-II, NSGA-II, SPEA2, and MISA in solving the SCH, DEB, KUR, ZDT1, ZDT2, and ZDT3.

### 4.3 Comparison of NNIA With and Without Recombination

Most of the existing immune inspired optimization algorithms, especially some pure clonal selection algorithms, did not use recombination. However, we think that recombination should not be forbidden in AIS community. Some immunologists have claimed that recombination is a mode of receptor editing used by the B cell antibodies to improve their affinity (George and Gray, 1999). de Castro and Von Zuben (2002) also claimed that genetic recombination and mutation are the two central processes involved in the production of antibodies, even though they did not use recombination in their pure clonal selection algorithm CLONALG according to the clonal selection theory (Burnet, 1959).

In Section 4.2, the simulated binary crossover (SBX) was introduced as the recombination operator. SBX has been adopted in many MOEA papers (Zitzler et al., 2002; Deb, Pratap, et al., 2002; Deb and Jain, 2002; Khare et al., 2003; Igel et al., 2007). In order to identify the improvement produced by SBX, we performed the proposed algorithm with SBX (denoted by NNIA) and without SBX (denoted by NNIA-X) for 30 independent runs on each test problem, respectively. The parameter settings are the same as Section 4.2. Figure 12 illustrates the box plots based on the coverage of the two sets obtained by NNIA and NNIA-X. Here, $\mathbf{I}$ denotes the solution set obtained by NNIA, and $\mathbf{I}^{-\mathbf{X}}$ denotes the solution set obtained by NNIA-X.

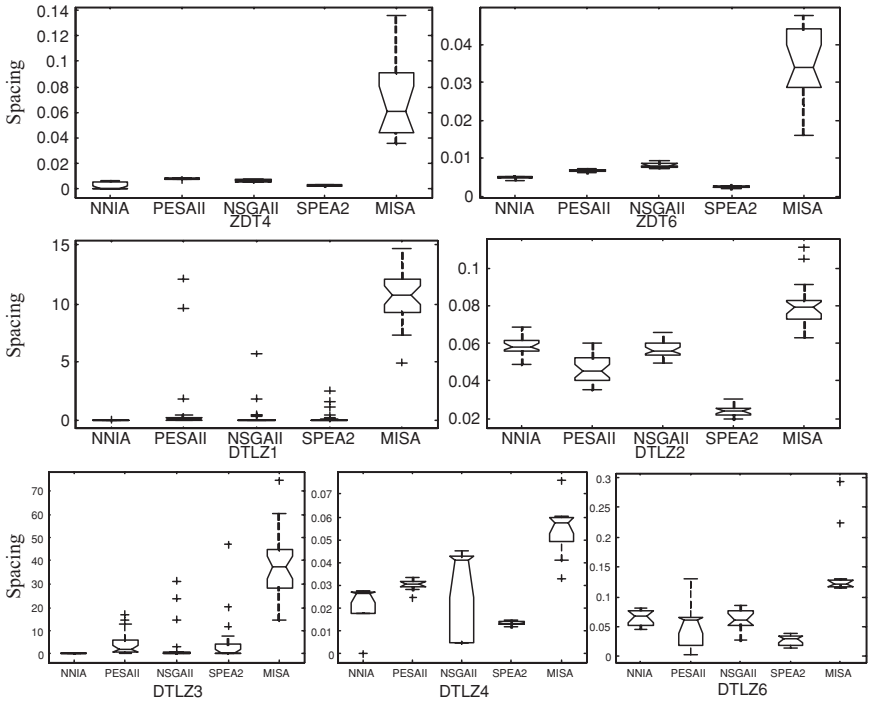


Figure 11: Box plots of the spacing metric obtained by NNIA, PESA-II, NSGA-II, SPEA2, and MISA in solving the ZDT4, ZDT6, and DTLZ problems.

The comparison between NNIA and NNIA-X in coverage shows the following.

For ZDT1, ZDT2, ZDT3, ZDT4, DTLZ1, DTLZ2, DTLZ3, and DTLZ4, the box plots of $I_C(\mathbf{I}, \mathbf{I}^{-\mathbf{X}})$ are obviously higher than the corresponding box plots of $I_C(\mathbf{I}^{-\mathbf{X}}, \mathbf{I})$. Therefore, NNIA did much better than NNIA-X in solving these eight problems as far as the coverage is concerned.

For SCH, ZDT6, and DTLZ6, the box plots of $I_C(\mathbf{I}, \mathbf{I}^{-\mathbf{X}})$ are slightly higher than the corresponding box plots of $I_C(\mathbf{I}^{-\mathbf{X}}, \mathbf{I})$. Therefore, NNIA did a little better than NNIA-X in solving these three problems.

For DEB and KUR, the box plots of $I_C(\mathbf{I}^{-\mathbf{X}}, \mathbf{I})$ are slightly higher than the corresponding box plots of $I_C(\mathbf{I}, \mathbf{I}^{-\mathbf{X}})$. Therefore, NNIA-X did a little better than NNIA in solving these two problems.

Overall, NNIA did better than NNIA-X in solving 11 out of the 13 problems. NNIA-X only did a little better than NNIA in solving the two low-dimensional problems DEB and KUR.

Based on the above empirical results, we did not abandon the recombination in our algorithm because our aim was to construct a useful algorithm for multiobjective optimization rather than a pure clonal selection algorithm. Furthermore, the recombination of genes involved in the production of antibodies differs somewhat from the recombination of parental genes in sexual reproduction used in genetic algorithms. In the former, the recombination on a clone is performed as a crossover between the clone and a randomly selected active antibody. This method achieved replacement of some gene

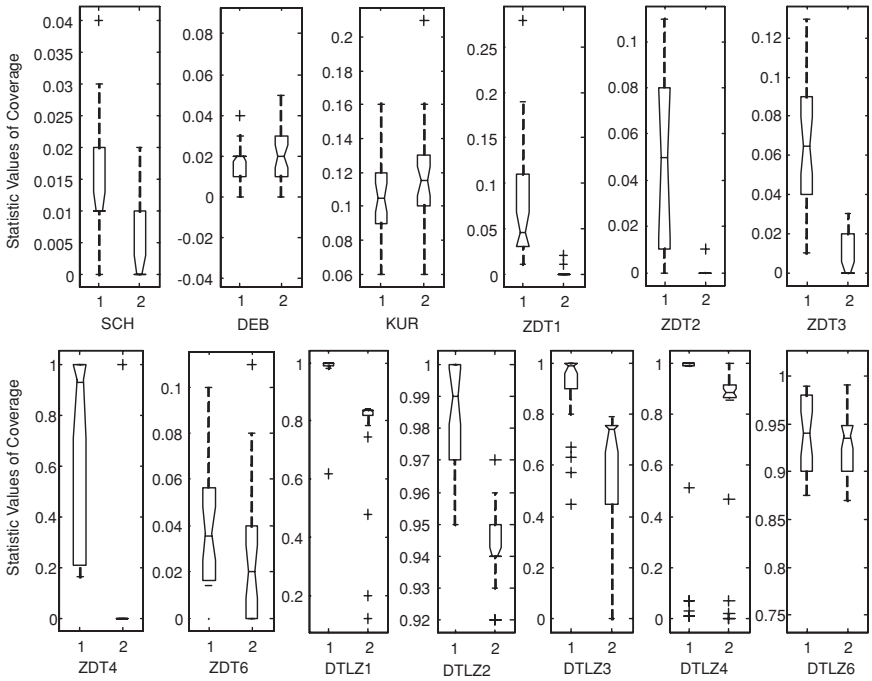


Figure 12: Box plots of the coverage of the two sets obtained by NNIA with and without SBX, in solving the 13 problems. The 13 plots denote the results of the 13 problems, respectively. In each plot, the left box represents the distribution of $I_C(\mathbf{I}, \mathbf{I}^{-\mathbf{X}})$ and the right box represents the distribution of $I_C(\mathbf{I}^{-\mathbf{X}}, \mathbf{I})$.

segments of the clone at random by the corresponding gene segments of the selected active antibody. But most genetic algorithms involve the crossover of two parents to generate offspring. However, the similarity between them is also obvious, because the similarity between biological evolution and the production of antibodies is even more striking as claimed in de Castro and Von Zuben (2002).

### 4.4 Scalability of NNIA Along the Number of Objectives

In this section, we will study the NNIA's scalability in the convergence metric, spacing, and runtime with respect to the number of objectives. Among the 13 test problems, only the five DTLZ problems are scalable to have any number of objectives. As suggested by Deb, Thiele, et al. (2002), DTLZ1 can be used to investigate an MOEA's ability to scale up its performance in a large number of objectives, and DTLZ3 can be used to investigate an MOEA's ability to converge to the true Pareto-optimal front. Here, DTLZ1 and DTLZ3 with two to nine objectives are used for the empirical study. In order to investigate the change of these metrics along the number of generations, we set the number of generations as a considerably larger value of 1,000. Some references have implied that MOEAs with small population size have difficulty in converging to the Pareto-optimal front with well-distributed solutions for complicated problems, especially for the MOPs with more than three objectives (Khare et al., 2003; Tan et al., 2001; Deb, 2001). In order to abate the influence of population size, here we double the values of population

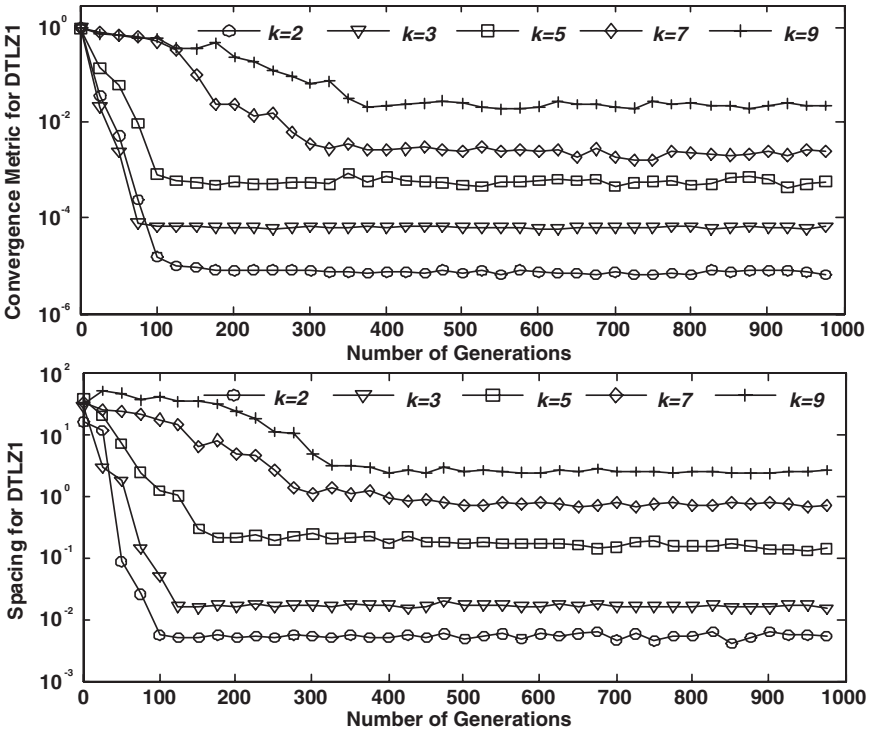


Figure 13: Mean values of the normalized convergence metric (the upper plot) and the spacing (the lower plot) versus number of generations over 30 independent runs when NNIA solves DTLZ1 with various numbers of objectives. A logarithmic (base 10) scale is used for the $Y$-axis.

size, namely, the maximum size of the dominant population $n_{\mathrm{D}} = 200$, the maximum size of active population $n_{\mathrm{A}} = 50$, and the size of clone population $n_{\mathrm{C}} = 200$. Other parameters are the same as those in Section 4.2. The number of points on the true Pareto-optimal front for calculating the values of the convergence metric is 200 when $k = 2$ and 3,500 when $k = 5$, and 1000 when $k = 7$ and $k = 9$. Figures 13 and 14 show the mean values of the normalized convergence metric (Deb and Jain, 2002) and spacing over 30 independent runs in 1 to 1,000 generations when NNIA solves DTLZ1 and DTLZ3 with 2, 3, 5, 7, and 9 objectives, respectively.

For DTLZ1, when $k = 2$, 3, and 5, NNIA obtains the approximate minimum values of the convergence metric and spacing in no more than 150 generations. When $k = 7$ and $k = 9$, NNIA needs to iterate about 350 generations to obtain the approximate minimum values of the convergence metric and spacing. For DTLZ3, when $k = 2, 3, 5,$ and 7, NNIA obtains the approximate minimum values of the convergence metric and spacing in no more than 300 generations. When $k = 9$, NNIA does not get acceptable values of the convergence metric and spacing with the population size used here because DTLZ3 introduces more difficulties to an MOEA in converging to Pareto-optimal front and in finding a diverse set of solutions (Khare et al., 2003).

Figure 15 shows the running time of NNIA in solving DTLZ1 and DTLZ3 using the parameters mentioned above. All the values of time are recorded using Matlab 7.0

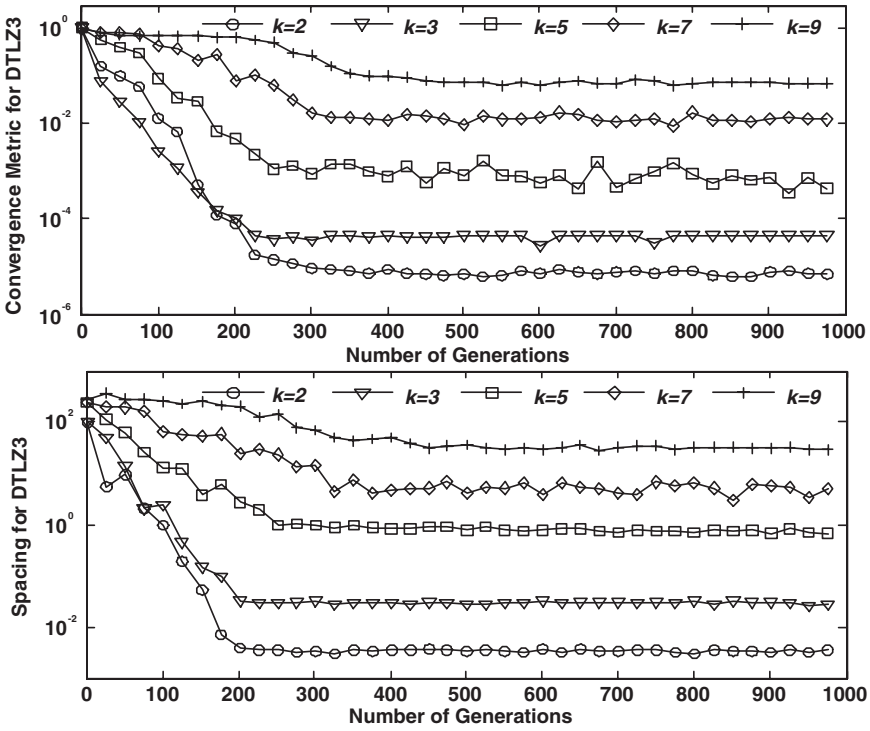


Figure 14: Mean values of the normalized convergence metric (the upper plot) and the spacing (the lower plot) versus number of generations over 30 independent runs when NNIA solves DTLZ3 with various numbers of objectives. A logarithmic (base 10) scale is used for the $Y$-axis.

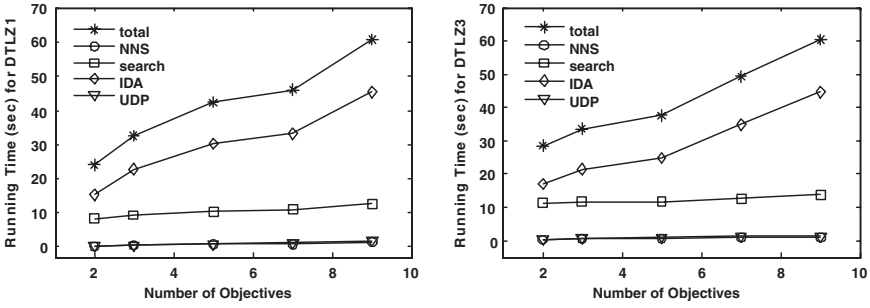


Figure 15: Mean values of each operator's running time and the total running time versus number of objectives when NNIA solves DTLZ1 and DTLZ3 with various numbers of objectives.

running on a personal computer with P-IV 3.2G CPU and 2G RAM. The runtime of the operators including nondominated neighbor-based selection (NNS), cloning, recombination and mutation (search), update dominant population (UDP), and

identifying dominant antibodies (IDA), are given, too. It can be seen that the cost of identifying the nondominated individuals in the population dominates the runtime complexity of NNIA. It is obvious that the complexity of NNIA along the number of objectives is approximately linear when the number of generations is the same.

## 5 Concluding Remarks

We have proposed a novel multiobjective algorithm based on a nondominated neighbor-based selection technique, an immune inspired operator, two heuristic search operators, and elitism. In NNIA, the fitness value of nondominated individuals is assigned as the crowding-distance. The selection technique only performs on nondominated individuals and selects minority isolated individuals to clone proportionally to crowding-distance values, recombine, and mutate. Some immunologists claimed that the repertoire diversity of an antibody underlies the immune response; however, the majority of antibodies do not play any active role during the immune response (Pannetier et al., 1995; Cziko, 1995; Baranzini et al., 1999; Parkin and Cohen, 2001). NNIA simulated this mechanism by selecting only the minority of nondominated individuals with greater crowding-distance values as active antibodies, and performing proportional cloning, recombination, and hypermutation only on these active antibodies. It realized the enhanced local search around the active antibodies that are the less-crowded individuals in objective space.

NSGA-II, SPEA2, PESA-II, and some other MOEAs can be considered as different MOEAs because they adopt different ways to do selection and population maintenance. In contrast to NSGA-II, the nondominated neighbor-based selection and proportional cloning make the less-crowded individuals have more chances to do recombination and mutation. So in a single generation, NNIA pays more attention to the less-crowded regions of the current trade-off front. The essential difference between NNIA and MISA is in their different selection methods and population maintenance strategies as well as individual representation methods. MISA adopts binary representation, clones all the nondominated individuals (and feasible individuals for constraint problems), and applies two types of mutation to the clones and other not so good individuals, respectively. MISA updates its external population by utilizing the grid based techniques used in PAES which need a crucial parameter, the number of grid cells. The difference between NNIA and VAIS also lies in their selection and population maintenance strategies. VAIS adopts the flowchart of opt-aiNet and the fitness assignment method in SPEA2 with some simplification. VAIS maintains its external population (memory population) using suppression mechanism like opt-aiNet based on the Euclidean distance in objective space and a threshold for suppression. We can also find some similar points between NNIA and PESA-II, PAES, SPEA2, for example, storing the nondominated individuals previously found externally and reducing the number of nondominated individuals stored without destroying the characteristics of the trade-off front, because our algorithm is inspired from them.

Most of the immune system inspired optimization algorithms essentially evolve solutions to problems via repeated application of a cloning, mutation, and selection cycle to a population of candidate solutions and remaining good solutions in the population. Just as Hart and Timmis (2005) said, anyone familiar with the EA literature will recognize all of these features as equally applicable to an EA. It may be due to the striking similarity between the functioning of the immune system and adaptive biological evolution. In particular, the central processes involved in the production of antibodies, genetic

recombination and mutation, are the same ones responsible for the biological evolution of species (de Castro and Von Zuben, 2002).

The main contribution of this study to MO field may be its unique selection technique. The selection technique only selects minority isolated nondominated individuals based on their crowding-distance values. The selected individuals are then cloned proportionally to their crowding-distance values before heuristic search. By using nondominated neighbor-based selection and proportional cloning, the new algorithm realizes enhanced local search in the less-crowded regions of the current trade-off front. Depending on the enhanced local search, NNIA can solve MOPs with a simple procedure. The experimental study of NNIA, SPEA2, NSGA-II, PESA-II, and MISA in solving three low-dimensional problems, five ZDT problems, and five DTLZ problems has shown that NNIA was able to converge to the true Pareto-optimal fronts in solving most of the test problems. More important, for the complicated problems DTLZ1 and DTLZ3, NNIA did much better than the other four algorithms. Depending on these empirical comparisons, we concluded that the nondominated neighbor-based selection method is effective, and NNIA is an effective algorithm for solving multiobjective optimization problems.

An issue that should be addressed in future research is the population size self-adaptive dynamic strategy. Some references have implied that MOEAs with small population size have difficulty in converging to the true Pareto-optimal front and obtaining the well-distributed solutions for some complicated problems (Deb, 2001; Tan et al., 2001; Khare et al., 2003). The dependence between problem complexity and population size of NNIA should be solved correctly. However, estimating the correct population size for a given problem cannot be done a priori as the complexity of the problem generally cannot be inferred ahead of time. The population size self-adaptive dynamic strategy using the rate of nondominated individuals or the relation between dominated and nondominated individuals may be an interesting direction for future research.

The formulation of an efficient stopping criterion for MOEAs has been claimed as one of the fundamental topics that must be properly addressed in the MOEA area (Rudenko and Schoenauer, 2004; Coello Coello, 2005). However, it is difficult to formulate the optimal and evidential stop criterion because judging the advance of the optimization can become as complex as the MOP itself. Furthermore, single-objective EAs can stop when the fitness does not improve during a given number of generations, but such a stopping criterion does not easily extend to the multiobjective framework. Therefore, researchers usually stop the algorithm when the algorithm reaches a given number of iterations or function evaluations as used in this study. A more appropriate stopping criterion should be addressed in future research.

Applying an effective constraint-handling technique to the proposed algorithm for solving constrained MOPs is also planned for our future work.

Most of the existing artificial immune system algorithms for multiobjective optimization are inspired by the clonal selection principle. Some immunological theories are not well explored in the field of multiobjective optimization, such as Negative Selection (Zhou and Dasgupta, 2007) and Danger Theory (Aickelin and Cayzer, 2002). An investigation of their potential for multiobjective optimization may be another direction of future research.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 60703107), the National High Technology Research and Development Program (863

Program) of China (Grant No. 2006AA01Z107), the National Basic Research Program (973 Program) of China (Grant No. 2006CB705700) and the Graduate Innovation Fund of Xidian University (Grant No. 05004). The authors wish to thank the anonymous reviewers for their valuable comments and helpful suggestions which greatly improved the paper's quality. The authors also wish to thank Prof. Coello Coello for his maintenance of the EMO repository.

## References

Aickelin, U., and Cayzer, S. (2002). The danger theory and its application to artificial immune systems. *Proceedings of the First International Conference on Artificial Immune Systems* (pp. 141–148). University of Kent at Canterbury, UK.

Baranzini, S. E., Jeong, M. C., Butunoi, C., Murray, R. S., Bernard, C. C. A., and Oksenberg, J. R. (1999). B cell repertoire diversity and clonal expansion in multiple sclerosis brain lesions. *The Journal of Immunology*, 163:5133–5144.

Burnet, F. M. (1959). *The Clonal Selection Theory of Acquired Immunity*. Cambridge University Press, Cambridge, UK.

Coello Coello, C. A. (2003). Evolutionary multiobjective optimization: Current and future challenges. In *Advances in Soft Computing-Engineering, Design and Manufacturing* (pp. 243–256). Springer-Verlag, Berlin.

Coello Coello, C. A. (2005). Recent trends in evolutionary multiobjective optimization. In A. Abraham, L. Jain, and R. Goldberg (Eds.), *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications* (pp. 7–32). Springer-Verlag, Berlin.

Coello Coello, C. A. (2006). Evolutionary multiobjective optimization: A historical view of the field. *IEEE Computational Intelligence Magazine*, 1(1):28–36.

Coello Coello, C. A., and Cortes, N. C. (2002). An approach to solve multiobjective optimization problems based on an artificial immune system. *Proceedings of the First International Conference on Artificial Immune Systems* (pp. 212–221). University of Kent at Canterbury, UK.

Coello Coello, C. A., and Cortes, N. C. (2005). Solving multiobjective optimization problems using an artificial immune system. *Genetic Programming and Evolvable Machines*, 6:163–190.

Coello Coello, C. A., and Pulido, G. T. (2001). Multiobjective optimization using a micro-genetic algorithm. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)* (pp. 274–282). Morgan Kaufmann, San Mateo, CA.

Coello Coello, C. A., Van Veldhuizen, D., and Lamont, G. B. (2002). *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic, New York.

Corne, D. W., Jerram, N. R., Knowles, J. D., and Oates, M. J. (2001). PESA-II: Region-based selection in evolutionary multiobjective optimization. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)* (pp. 283–290). Morgan Kaufmann, San Mateo, CA.

Corne, D. W., Knowles, J. D., and Oates, M. J. (2000). The pareto-envelope based selection algorithm for multiobjective optimization. *Parallel Problem Solving from Nature-PPSN VI*, volume 1917 of *Lecture Notes in Computer Science* (pp. 839–848).

Cutello, V., Narzisi, G., and Nicosia, G. (2005). A class of Pareto archived evolution strategy algorithms using immune inspired operators for ab-initio protein structure prediction. *Third European Workshop on Evolutionary Computation and Bioinformatics, EvoWorkshops 2005-EvoBio 2005*, volume 3449 of *Lecture Notes in Computer Science* (pp. 54–63).

Cutello, V., Narzisi, G., Nicosia, G., and Pavone, M. (2005). Clonal selection algorithms: A comparative case study using effective mutation potentials. *Proceedings of 4th International*

*Conference on Artificial Immune Systems, ICARIS 2005*, volume 3627 of *Lecture Notes in Computer Science* (pp. 13–28).

Cutello, V., Nicosia, G., and Pavone, M. (2004). Exploring the capability of immune algorithms: A characterization of hypemutation operators. *Proceedings of Third International Conference on Artificial Immune Systems, ICARIS 2004*, volume 3239 of *Lecture Notes in Computer Science* (pp. 263–276).

Cziko, G. (1995). The immune system: Selection by the enemy. In *Without Miracles: Universal Selection Theory and the Second Darwinian Revolution* (pp. 39–48). MIT Press, Cambridge, MA.

de Castro, L. N., and Timmis, J. (2002). An artificial immune network for multimodal function optimization. *Proceedings of the 2002 Congress on Evolutionary Computation, CEC2002* (pp. 699–704).

de Castro, L. N., and Von Zuben, F. J. (2002). Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation*, 6(3):239–251.

Deb, K. (1999). Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation*, 7(3):205–230.

Deb, K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley and Sons, Chichester, UK.

Deb, K., and Beyer, H. G. (2001). Self-adaptive genetic algorithms with simulated binary crossover. *Evolutionary Computation*, 9(2):197–221.

Deb, K., and Jain, S. (2002). *Running performance metrics for evolutionary multiobjective optimization*. Technical Report 2002004, KanGAL, Indian Institute of Technology, Kanpur 208016, India.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2001). *Scalable multi-objective optimization test problems.* Technical Report 112, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland.

Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2002). Scalable multi-objective optimization test problems. *Proceedings of Congress on Evolutionary Computation, CEC2002* (pp. 825–830).

Fonseca, C. M., and Fleming, P. J. (1995). An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 1(1):1–16.

Freschi, F., and Repetto, M. (2005). Multiobjective optimization by a modified artificial immune system algorithm. *Proceedings of the Fourth International Conference on Artificial Immune Systems, ICARIS 2005*, volume 3627 of *Lecture Notes in Computer Science* (pp. 248–261).

Fukuda, T., Mori, K., and Tsukiyama, M. (1993). Immune networks using genetic algorithm for adaptive production scheduling. *15th IFAC World Congress*, volume 3 (pp. 57–60).

Garrett, S. M. (2004). Parameter-free, adaptive clonal selection. *Proceedings of IEEE Congress on Evolutionary Computing, CEC 2004* (pp. 1052–1058).

Garrett, S. M. (2005). How do we evaluate artificial immune systems? *Evolutionary Computation*, 13(2):145–178.

George, A. J. T., and Gray, D. (1999). Receptor editing during affinity maturation. *Immunology Today*, 20(4):196.

Hart, E., and Timmis, J. (2005). Application areas of AIS: The past, the present and the future. *Proceedings of the Foruth International Conference on Artificial Immune Systems, ICARIS 2005*, volume 3627 of *Lecture Notes in Computer Science* (pp. 483–497).

Horn, J., and Nafpliotis, N. (1993). Multiobjective optimization using the niched Pareto genetic algorithm. IlliGAL Report 93005, Illinois Genetic Algorithms Laboratory, University of Illinois, Urbana-Champaign, Illinois.

Igel, C., Hansen, N., and Roth, S. (2007). Covariance matrix adaptation for multi-objective optimization. *Evolutionary Computation*, 15(1):1–28.

Jacob, C., Pilat, M. L., Bentley, P. J., and Timmis, J., Eds. (2005). *Artificial Immune Systems: Proceedings of The Fourth International Conference on Artificial Immune Systems, ICARIS 2005*, volume 3627 of *Lecture Notes in Computer Science*.

Jiao, L. C., Gong, M. G., Shang, R. H., Du, H. F., and Lu, B. (2005). Clonal selection with immune dominance and energy based multiobjective optimization. *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization, EMO 2005*, volume 3410 of *Lecture Notes in Computer Science* (pp. 474–489).

Khare, V., Yao, X., and Deb, K. (2003). Performance scaling of multi-objective evolutionary algorithms. *Proceedings of the Second International Conference on Evolutionary Multi-Criterion Optimization, EMO 2003*, volume 2632 of *Lecture Notes in Computer Science* (pp. 376–390).

Knowles, J., Thiele, L., and Zitzler, E. (2006). *A tutorial on the performance assessment of stochastic multiobjective optimizers.* Technical Report 214, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland.

Knowles, J. D., and Corne, D. W. (2000). Approximating the nondominated front using the Pareto archived evolution strategy. *Evolutionary Computation*, 8(2):149–172.

Kursawe, F. (1991). A variant of evolution strategies for vector optimization. *Parallel Problem Solving from Nature-PPSN I*, volume 496 of *Lecture Notes in Computer Science* (pp. 193–197).

McGill, R., Tukey, J., and Larsen, W. (1978). Variations of boxplots. *The American Statistician*, 32:12–16.

Nicosia, G., Cutello, V., Bentley, P. J., and Timmis, J., Eds. (2004). *Artificial Immune Systems, Proceedings of The Third International Conference on Artificial Immune Systems, ICARIS 2004*, volume 3239 of *Lecture Notes in Computer Science*.

Pannetier, C., Even, J., and Kourilsky, P. (1995). T-cell repertoire diversity and clonal expansions in normal and clinical samples. *Immunology Today*, 16(4):176–181.

Parkin, J., and Cohen, B. (2001). An overview of the immune system. *The Lancet*, 357(9270):1777–1789.

Rudenko, O., and Schoenauer, M. (2004). A steady performance stopping criterion for Pareto-based evolutionary algorithms. *Proceedings of the Sixth International Multi-Objective Programming and Goal Programming Conference*.

Rudolph, G. (1998). On a multi-objective evolutionary algorithm and its convergence to the Pareto set. In *Proceedings of the Fifth IEEE Congress on Evolutionary Computation, CEC 1998* (pp. 511–516).

Schaffer, J. D. (1984). *Multiple objective optimization with vector evaluated genetic algorithms*. PhD thesis, Vanderbilt University, Nashville, TN.

Schott, J. R. (1995). Fault tolerant design using single and multicriteria genetic algorithm optimization. Masters thesis, Massachusetts Institute of Technology, Cambridge, MA.

Srinivas, N., and Deb, K. (1993). Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248.

Tan, K. C., Lee, T. H., and Khor, E. F. (2001). Evolutionary algorithms with dynamic population size and local exploration for multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 5(6):565–588.

Tarakanov, A., and Dasgupta, D. (2000). A formal model of an artificial immune system. *BioSystems*, 55(1/3):151–158.

Van Veldhuizen, D. A. (1999). *Multiobjective Evolutionary Algorithms: Classification, Analyses, and New Innovation*. PhD Thesis, Air Force Institute of Technology, Wright Patterson Air Force Base, Dayton, Ohio.

Van Veldhuizen, D. A., and Lamont, G. B. (2000). On measuring multiobjective evolutionary algorithm performance. *Proceedings of the 2000 IEEE Congress on Evolutionary Computation, CEC 2000* (pp. 204–211).

Zhou, J., and Dasgupta, D. (2007). Revisiting negative selection algorithms. *Evolutionary Computation*, 15(2):223–251.

Zitzler, E., Deb, K., and Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195.

Zitzler, E., Laumanns, M., and Thiele, L. (2002). SPEA2: Improving the strength Pareto evolutionary algorithm. *Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems* (pp. 95–100).

Zitzler, E., and Thiele, L. (1998). Multiobjective optimization using evolutionary algorithms—A comparative case study. *Parallel Problem Solving from Nature-PPSN V*, volume 1498 of *Lecture Notes in Computer Science* (pp. 292–301).

Zitzler, E., and Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., and da Fonseca, V. G. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132.

## The Authors' Annotation

By checking the experimental results seriously and thoroughly, to our regret, we have to admit that the obtained statistical values of the Coverage of the two sets in solving the five DTLZ problems are wrong.

From Figure 4 of (Gong et al., 2008), we can see that most of the values of $I_C(\mathbf{I}, \mathbf{P})$ and $I_C(\mathbf{P}, \mathbf{I})$ for the five DTLZ problems are greater than 0.5. Therefore, the majority solutions obtained by PESA-II are weakly dominated by the solutions obtained by NNIA, while the majority solutions obtained by NNIA are also weakly dominated by the solutions obtained by PESA-II. Obviously, it could not be true unless most solutions obtained by NNIA are equal to the solutions obtained by PESA-II. But after examining the solution sets obtained by NNIA and PESA-II for each trial, we found that identical solution in both sets is rare. The same situation exists in figure 5, 6, 7, and 12. We have examined our original codes for calculating the coverage metric in solving the five three-objective problems, an inadvertent error is found in them. However, for the two-objective problems, the codes are right.

We have corrected our codes and the right box plots are presented in the following figures. Note that the errors are only in the five DTLZ problems in terms of the Coverage of the two sets. The corrected five figures in this study correspond to the results of DTLZ problems in Figure 4, 5, 6, 7, and 12 of (Gong et al., 2008) respectively.

The corrected figures show that the box plots of $I_C(\mathbf{I}, \mathbf{P})$, $I_C(\mathbf{I}, \mathbf{N})$, $I_C(\mathbf{I}, \mathbf{S})$, $I_C(\mathbf{I}, \mathbf{M})$ and $I_C(\mathbf{I}, \mathbf{I}^{-\mathbf{X}})$ are higher than the corresponding box plots of $I_C(\mathbf{P}, \mathbf{I})$, $I_C(\mathbf{N}, \mathbf{I})$, $I_C(\mathbf{S}, \mathbf{I})$, $I_C(\mathbf{M}, \mathbf{I})$ and $I_C(\mathbf{I}^{-\mathbf{X}}, \mathbf{I})$ in all the five DTLZ problems, respectively. They are consistent with the original conclusions.

## References

Gong, M., Jiao, L., and Yang, D. (2008). Multiobjective immune algorithm with nondominated neighbor-based selection. *Evolutionary Computation*, 16(2):225–255.

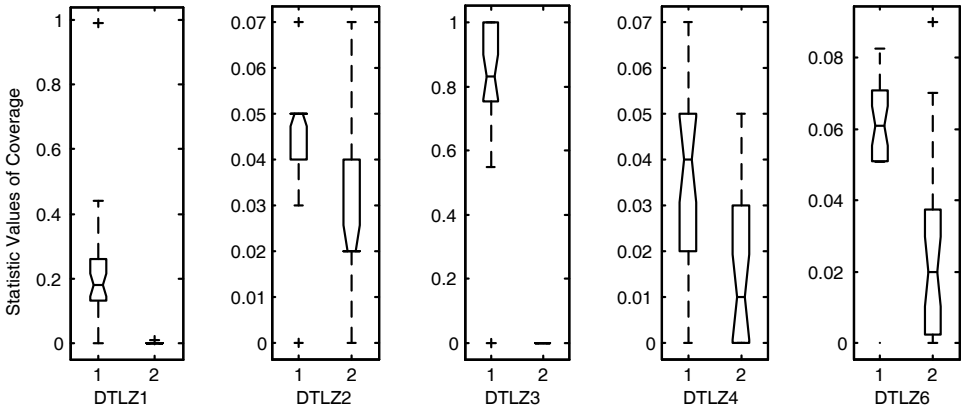


Figure 1: Statistical Values of the Coverage of the two sets obtained by NNIA and PESA-II in solving the five DTLZ problems. The five plots denote the results of the five problems respectively. In each plot, the left box represents the distribution of $I_C(\mathbf{I}, \mathbf{P})$ and the right box represents the distribution of $I_C(\mathbf{P}, \mathbf{I})$.

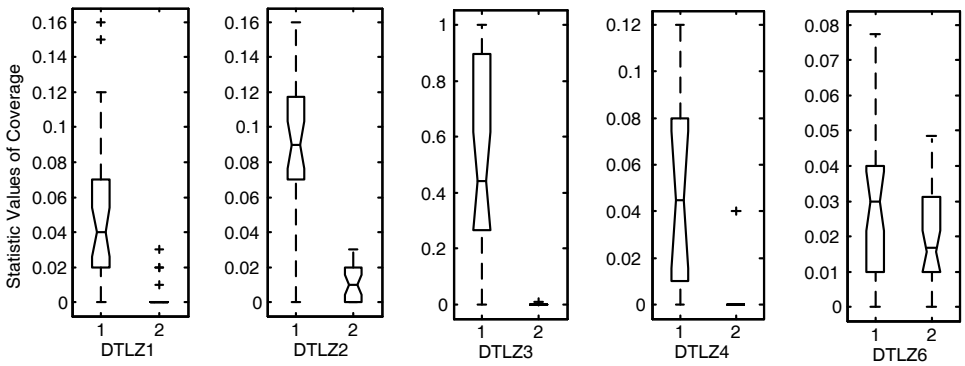


Figure 2: Statistical Values of the Coverage of the two sets obtained by NNIA and NSGA-II in solving the five DTLZ problems. The five plots denote the results of the five problems respectively. In each plot, the left box represents the distribution of $I_C(\mathbf{I}, \mathbf{N})$ and the right box represents the distribution of $I_C(\mathbf{N}, \mathbf{I})$.

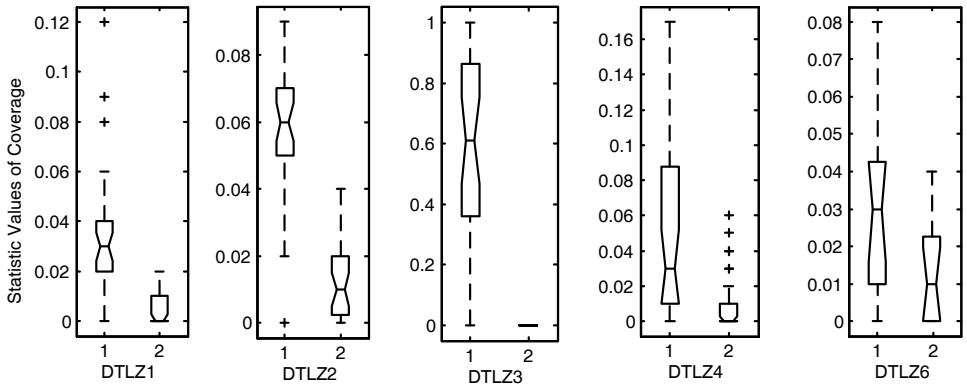


Figure 3: Statistical Values of the Coverage of the two sets obtained by NNIA and SPEA2 in solving the five DTLZ problems. The five plots denote the results of the five problems respectively. In each plot, the left box represents the distribution of $I_C(\mathbf{I}, \mathbf{S})$ and the right box represents the distribution of $I_C(\mathbf{S}, \mathbf{I})$.

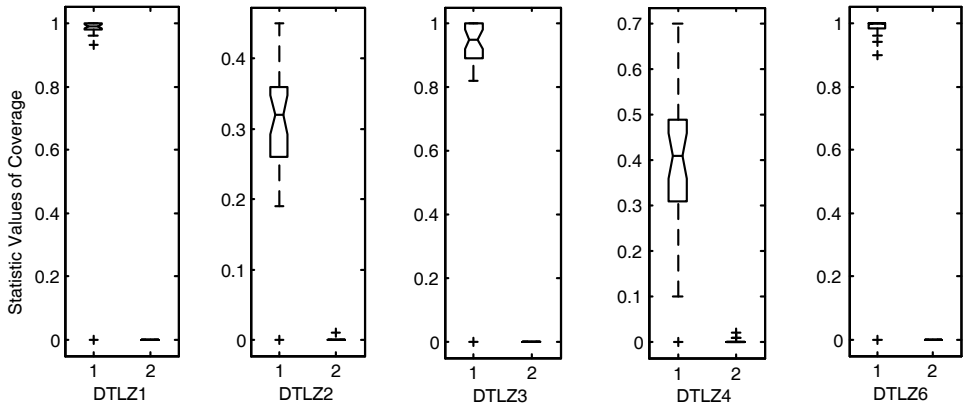


Figure 4: Statistical Values of the Coverage of the two sets obtained by NNIA and MISA in solving the five DTLZ problems. The five plots denote the results of the five problems respectively. In each plot, the left box represents the distribution of $I_C(\mathbf{I}, \mathbf{M})$ and the right box represents the distribution of $I_C(\mathbf{M}, \mathbf{I})$.

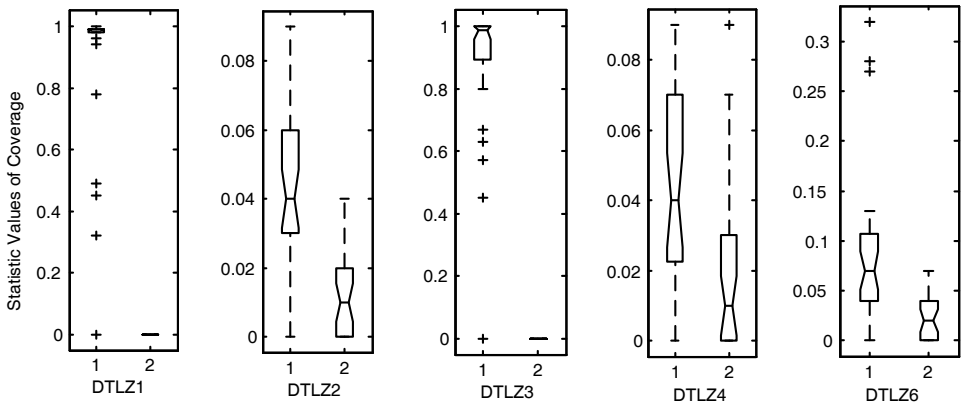


Figure 5: Statistical Values of the Coverage of the two sets obtained by NNIA with and without SBX in solving the five DTLZ problems. The five plots denote the results of the five problems respectively. In each plot, the left box represents the distribution of $I_C(\mathbf{I}, \mathbf{I}^{-\mathbf{X}})$ and the right box represents the distribution of $I_C(\mathbf{I}^{-\mathbf{X}}, \mathbf{I})$.