Clonal Selection with Immune Dominance and Anergy Based Multiobjective Optimization

Licheng JIAO[1], Maoguo GONG[1], Ronghua SHANG[1], Haifeng DU[1, 2], Bin LU[1]

[1] Institute of Intelligent Information Processing, P.O. Box 224, Xidian University, Xi'an, 710071, P.R. China
lchjiao@mail.xidian.edu.cn maoguo_gong@hotmail.com
[2] School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an, 710049, P.R. China
Haifengdu72@163.com

Abstract. Based on the concept of Immunodominance and Antibody Clonal Selection Theory, we propose a new artificial immune system algorithm, Immune Dominance Clonal Multiobjective Algorithm (IDCMA). The influences of main parameters are analyzed empirically. The simulation comparisons among IDCMA, the Random-Weight Genetic Algorithm and the Strength Pareto Evolutionary Algorithm show that when low-dimensional multiobjective problems are concerned, IDCMA has the best performance in metrics such as Spacing and Coverage of Two Sets.

1 Introduction

In 1984, Schaffer put forward a vector evaluated genetic algorithm (VEGA) by modifying the fitness assignment and the individual selection strategy[1]. His work is regarded as the beginning of solving multiobjective optimization problems by genetic algorithm. Until the middle 1990s, the number of literatures on multiobjective evolutionary algorithms (MOEAs) increased greatly. Among them, Fonseca et al's Multiobjective Genetic Algorithm, Horn et al's Niched Pareto Genetic Algorithms and Srinivas et al's Nondominated Sorting in Genetic Algorithm attracted more attention[2]. These evolutionary algorithms show better performance in solving multiobjective problems than traditional algorithms. However, they didn't adopt elitism preserving strategy definitely, which was recognized and supported by experiments in the following years. In recent years, a lot of newly improved algorithms were proposed, such as Deb et al's A Fast Elitist Non-dominated Sorting Genetic Algorithm, Corne et al's Pareto Envelope based Selection Algorithm and Zitzler's the strength Pareto evolutionary algorithm (SPEA and SPEA2). In particular, Zitzler et al's SPEA and SPEA2 have shown many good performances[3,4]. At the same time, Coello Coello et al presented their own multiobjective evolutionary algorithms and proposed an multiobjective algorithm named by the Multiobjective Immune System Algorithm (MISA)[5] using the clonal selection principle. They set out that MISA was very promising based on a few simulations.

Artificial immune system (AIS) makes use of the mechanism of vertebrate immune system, and constructs new intelligent algorithms with immunology terms and fundamental. Artificial immune system provides the evolutionary learning mechanism like noise enduring, non-teacher learning, self-organization, and memory, thus it has the potential for providing novel method for solving problems, and its research production refers to many fields like control, data processing, optimization learning and trouble diagnosing, and it has been a research hot spot after the neural network, fuzzy logic and evolutionary computation.[6]

After defining several basic concepts of artificial immune system in Section 2, a novel multiobjective optimization algorithm, Immune Dominance Clonal Multiobjective Algorithm (IDCMA), is put forward in Section 3. The influences of main parameters are analyzed empirically, then five representative low-dimensional multiobjective problems and three famous multiobjective algorithms, the Random-Weight Approach proposed by Ishibuchi[7], and the Strength Pareto Evolutionary Algorithm proposed by Zitzler[3], and Multiobjective Immune System Algorithm proposed by Coello Coello [5] are selected for simulation tests in Section 4.

2 Basic Definitions

Although an antigen has many epitopes (antigenic determinants), only one works to induce a special immune response for the host cells. The phenomenon is called immunodominance, the epitope is called a dominant epitope, immunodominance is produced by the action of antibody and antigen[8]. The clonal selection theory (F. M. Burnet, 1959) is used in the immune system to describe the basic features of an immune response. Its main idea lies in that the antigens can selectively react to the antibodies, which are the native production and spread on the cell surface in the form of peptides. The reaction leads to cell proliferating clonally and the colony has the same antibodies. Some clonal cells divide into antibodies that produce cells, and others become immune memory cells to boost the second immune response. The clonal selection is a dynamic process of the immune system self-adapting antigen stimulation. From the viewpoint of the Artificial Intelligence, some biologic characters such as learning, memory and antibody diversity can be used in artificial immune system.

In order to describe the algorithm for multiobjective optimization problems well, we just define the glossary as follows.

Definition 1 Antigen

In AIS, antigen usually means the problem and its constraints. Especially, for multiobjective optimization problems, we have

$$(P)\begin{cases}\min\ F(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots f_p(\boldsymbol{x}))^T \\ S.T.\quad g_i(\boldsymbol{x}) \le 0 \quad i = 1, 2, \cdots m\end{cases} \tag{1}$$

where, $\boldsymbol{x} = (x_1, x_2, \ldots x_n)$, $p \ge 2$, the antigen is defined as a function of objective function $F(\boldsymbol{x})$, namely, $G(\boldsymbol{x}) = g(F(\boldsymbol{x}))$. Similar to the function of antigen in

immunology, it is the initial factor for the artificial immune system algorithm. Usually, we let $G(\boldsymbol{x}) = F(\boldsymbol{x})$ when not mentioned especially.

Definition 2 Antibody

Antibodies represent candidates of the problem. The limited-length character string $\boldsymbol{a} = a_1 a_2 \cdots a_l$ is the antibody coding of variable $\boldsymbol{x}$, denoted by $\boldsymbol{a} = e(\boldsymbol{x})$, and $\boldsymbol{x}$ is called the decoding of antibody $\boldsymbol{a}$, expressed as $\boldsymbol{x} = e^{-1}(\boldsymbol{a})$. In practice, binary coding and decimal coding are often used. For example, an antibody of binary coding whose length is 8 can be written as '0-1-1-1-0-1-0-0'. Set $\boldsymbol{I}$ is called antibody space, namely $\boldsymbol{a} \in \boldsymbol{I}$. The antibody population $\boldsymbol{A} = \{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n\} \in \boldsymbol{I}^n$ is an n-dimensional group of antibody $\boldsymbol{a}$, namely,

$$\boldsymbol{I}^n = \left\{ \boldsymbol{A} : \boldsymbol{A} = (\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n), \ \ \boldsymbol{a}_k \in \boldsymbol{I}, \ \ 1 \le k \le n \right\} \tag{2}$$

where the positive integer n is the antibody population size.

Definition 3 Antibody-Antigen Affinity

Antibody-Antigen Affinity is the reflection of the total combination power locates between antigen and antibodies. In AIS, it generally indicates values of objective functions or fitness measurement of the problem.

Definition 4 Antibody-Antibody Affinity

Antibody-Antibody Affinity is the reflection of the total combine power locates between two antibodies. In this paper, we compute the antibody-antibody affinity as reference [5]. Namely, if the coding of an antibody $\boldsymbol{a_i}$ is '1 1 0 0 0 0 1 0', and the coding of another antibody $\boldsymbol{a}_{di}$ is '1 1 0 1 0 1 1 0', then the number of genes matched between the two antibodies is 6, the matched gene strings whose length are greater than 2 are '110' and '10', and the corresponding lengths are 3 and 2, so the antibody-antibody affinity between $\boldsymbol{a_i}$ and $\boldsymbol{a}_{di}$ is $6 + 3^2 + 2^2 = 19$. Of course, other distance measures are possible.

Definition 5 Immune Dominance

For problem (P), the antibody $\boldsymbol{a_i}$ is an immune dominance antibody in antibody population $\boldsymbol{A} = \{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n\}$, iff there is no antibody $\boldsymbol{a}_j$ $(\boldsymbol{j} = 1, 2, \cdots \boldsymbol{n} \wedge \boldsymbol{j} \ne \boldsymbol{i})$ in population $\boldsymbol{A}$ satisfied the formula (3)

$$(\forall k \in \{1, 2, \cdots p\} \ f_k(e^{-1}(\boldsymbol{a}_j)) \ge f_k(e^{-1}(\boldsymbol{a}_i))) \wedge (\exists l \in \{1, 2, \cdots p\} \ f_l(e^{-1}(\boldsymbol{a}_j)) > f_l(e^{-1}(\boldsymbol{a}_i))) \tag{3}$$

So the immune dominance antibodies are the Pareto-optimal individuals in the current population.

Definition 6 Clonal Operation

In the immunology, clone is the process of antibody proliferation. In AIS, the clonal operation to the antibody population is defined as:

$$\boldsymbol{Y}(k) = T_{\mathrm{c}}^{\mathrm{C}}(\boldsymbol{A}(k)) = [T_{\mathrm{c}}^{\mathrm{C}}(\boldsymbol{a}_1(k)) \quad T_{\mathrm{c}}^{\mathrm{C}}(\boldsymbol{a}_2(k)), \quad \cdots \quad , T_{\mathrm{c}}^{\mathrm{C}}(\boldsymbol{a}_n(k))]^{\mathrm{T}} \tag{4}$$

where $T_{\mathrm{c}}^{\mathrm{C}}(\boldsymbol{a}_{ci}(k)) = \boldsymbol{I}_{ci} \times \boldsymbol{a}_{ci}(k)$ $i = 1, 2 \cdots n$, $\boldsymbol{I_{ci}}$ is a q_{ci}-dimensional identity row vector. The process is called the q_{ci} clone of antibody a_i, namely $q_{ci}(k) = \hbar(n_{\mathrm{c}}, \boldsymbol{\Theta}_i)$, where

$\boldsymbol{\Theta}_i$ stands for the affinity function of antibody a_i and other antibody, and n_c is the clonal scale.

Definition 7 Immune Differential Degree

In this paper, the Immune Differential Degree denotes the relative distribution of an immune dominance antibody. Namely, assuming that there are N immune dominance antibodies (Pareto-optimal solutions) in current population, f_{kl} is the value of the K-th objective function of the l-th antibody. The Immune Differential Degree of the l-th antibody $\boldsymbol{a}_l$ can be calculated as follow,

$$d_l^* = \min\left\{ d_l(m) = \sqrt{\sum_{k=1}^{q}\left(\frac{\phi(f_{kl})-\phi(f_{km})}{\phi(f_{kl})}\right)^2} \,\middle|\, l=1,2,\cdots N;\ m=1,2,\cdots N \wedge m \neq l \right\} \tag{5}$$

where $\phi(\bullet)$ is an incremental function without the value of zero.

3 Algorithm Description

Inspired from the immuodominace of the biology immune system and the clonal selection mechanism, we designed a novel artificial immune system algorithm based on clonal selection with immune dominance and clone anergy for multiobjective optimization problems which can be implemented as follow:

Step 1: Give the termination generation $G_{\max}$, the size of Immune Dominance Antibody population n_d, the size of Generic Antibody population n_b, the size of Dominance Clonal Antibody population n_t, and clonal scale n_c. Set the mutation probability p_m, recombination probability p_c and coding length c. Randomly generate the original antibody population $\boldsymbol{A}(0)=\{\boldsymbol{a}_1(0),\boldsymbol{a}_2(0),\cdots\boldsymbol{a}_{n_b}(0)\}\in \boldsymbol{I}^{n_b}$, k:=0;

Step 2: Compute the antibody-antigen affinities of all the antibodies in $\boldsymbol{A}(k)$;

Step 3: According to the affinities, select all the immune dominance antibodies to constitute the population $\boldsymbol{DT}(k)$, if the number of antibodies in $\boldsymbol{DT}(k)$ is no larger than n_d, let Immune Dominance Antibody population $\boldsymbol{D}(k)$=$\boldsymbol{DT}(k)$, go to Step6; otherwise go to Step4;

Step 4: Compute the Immune Differential Degrees of all the antibodies in population $\boldsymbol{DT}(k)$;

Step 5: Sort all the antibodies in $\boldsymbol{DT}(k)$ by descending of their Immune Differential Degrees, and select the first n_d antibodies to constitute the current Immune Dominance Antibody population $\boldsymbol{D}(k)$;

Step 6: If k=$G_{\max}$, export $\boldsymbol{D}(k)$ as the output of the algorithm, Stop. Otherwise, replace the immune dominance antibodies in $\boldsymbol{A}(k)$ by new antibodies generated randomly. Then marked the antibody population as $\boldsymbol{B}(k)$;

Step 7: Select an immune dominance antibody $\boldsymbol{a}_{di}$ randomly from $\boldsymbol{D}(k)$. Compute the antibody-antibody affinities between the antibodies in $\boldsymbol{B}(k)$ and the antibody $\boldsymbol{a}_{di}$.

Step 8: Sort all the antibodies in $\boldsymbol{B}(k)$ by descending of their antibody-antibody affinities, select the first n_t antibodies to constitute the Dominance Clonal Antibody

population ***TC***(k), and other antibodies to constitute the Immune Anergy Antibody population ***NR***(k).

Step 9: Compute the clonal proportion $q_{ci}(k)$ of each antibody $\boldsymbol{a}_{ci}$ in ***TC***(k) according to antibody-antibody affinity and the clonal scale.

Step 10: Implement the Antibody Clonal Operation $T_{\mathrm{c}}^{\mathrm{C}}$ at ***TC***(k) and get the antibody population ***CO***(k) after clonal operation.

Step 11: Implement the recombination operation at ***CO***(k) with the probability p_{c} and get the antibody population $\boldsymbol{CO'}(k)$, $\boldsymbol{CO'}(k) = T_{\mathrm{r}}^{\mathrm{C}}(\boldsymbol{CO}(k))$; Namely, for the antibody $\boldsymbol{y}_i(k)$ in ***CO***(k), implement the following operation:

$$\boldsymbol{y}'_i(k) = T_{\mathrm{r}}^{\mathrm{C}}(\boldsymbol{y}_i(k), \boldsymbol{a}_{di}(k)), \quad \boldsymbol{y}_i(k) \in \boldsymbol{CO}(k), \quad \boldsymbol{a}_{di}(k) \in \boldsymbol{D}(k) \tag{6}$$

and $\boldsymbol{CO'}(k) = \{\boldsymbol{y}'_i(k)\}\ i = 1, 2, \cdots n_{\mathrm{c}}$.

For binary coding, the recombination operation in this paper is as follow:

$$\boldsymbol{y}'_i(k) = T_{\mathrm{r}}^{\mathrm{C}}(\boldsymbol{y}_i(k), \boldsymbol{a}_{di}(k)) = \boldsymbol{y}_i(k)_{a \sim b \to 0} + \boldsymbol{a}_{di}(k)_{1 \sim a | b \sim c \to 0}, \quad \boldsymbol{y}_i(k) \in \boldsymbol{CO}(k), \quad \boldsymbol{a}_{di}(k) \in \boldsymbol{D}(k) \tag{7}$$

where c is the coding length, a, b are random integers between 1 and c. $a \sim b \to 0$ means the bits from a to b set zeros, $1 \sim a \,|\, b \sim c \to 0$ means the bits from 1 to a and the bits from b to c set zeros, + means 'or' operation by bits.

Step 12: Implement the mutation operation at $\boldsymbol{CO'}(k)$ with the probability p_{m} generating the antibody population ***COT***(k), $\boldsymbol{COT}(k) = T_{\mathrm{m}}^{\mathrm{C}}(\boldsymbol{CO'}(k))$:

For binary coding, the clonal antibody population is mutated as follow:

$$\boldsymbol{COT}(k) = (-1)^{random \le p_m} \boldsymbol{CO'}(k) \tag{8}$$

$(-1)^{random \le p_m} \boldsymbol{CO'}(k)$ means each element of $\boldsymbol{CO'}(k)$ multiplies -1 with probability of p_{m}.

Step 13: Combine the populations ***COT***(k), ***D***(k) and ***NR***(k) to form the antibody population ***A***(k+1), k:=k+1, go to Step 2.

From the description above we can see that the new algorithm divides all the antibodies into three sorts, and stores them in three populations. Different evolutionary strategies are adopted at different populations, but they are not isolated. The combination of the three populations helps to increase the global search ability. The operation of immune dominance antibody population based on the Immune Differential Degree retains the diversity of the population. The operation of dominance clonal antibody population based on the antibody-antibody affinity can select the effective local in antibody space and assure the validity of the search in next generation. The existence of the immune anergy antibody population assures the diversity of populations and simulates the immune response process more meticulously. In additional, the worst time complexity of one generation for immune dominance clonal multiobjective optimization algorithm is $\mathrm{O}(n_{\mathrm{d}} + n_{\mathrm{b}})^2$, where n_{d} is the size of Immune Dominance Antibody population, and n_{b} is the size of Generic Antibody population.

4 Simulation Analyses

In this section, we adopt two popular metrics, Coverage of Two Sets and Spcing, which are defined as follows:

Coverage of Two Sets[9]: Let $\boldsymbol{A}', \boldsymbol{A}'' \subseteq \boldsymbol{X}$ be two sets of decision vectors. The finction ς maps the ordered pair $(\boldsymbol{A}', \boldsymbol{A}'')$ to the interval [0, 1]:

$$\varsigma(\boldsymbol{A}', \boldsymbol{A}'') \triangleq \frac{\left|\{\boldsymbol{a}'' \in \boldsymbol{A}''; \exists \boldsymbol{a}' \in \boldsymbol{A}' : \boldsymbol{a}' \rhd \boldsymbol{a}''\}\right|}{\left|\boldsymbol{A}''\right|} \tag{9}$$

Where $\rhd$ means Pareto dominate or equal. The value $\varsigma(\boldsymbol{A}', \boldsymbol{A}'') = 1$ means that all decision vectors in $\boldsymbol{A}''$ are weakly dominated by $\boldsymbol{A}'$. $\varsigma(\boldsymbol{A}', \boldsymbol{A}'') = 0$ implies the opposite. Note that always both directions have to be considered because $\varsigma(\boldsymbol{A}', \boldsymbol{A}'')$ is not necessarily equal to $1 - \varsigma(\boldsymbol{A}'', \boldsymbol{A}')$.

A metric called Spacing was proposed by Schott[10] as a way of measuring the range variance of neighboring vectors in the Pareto front known. This metric is defined as:

Spacing[10]: Let $\boldsymbol{A}' \subseteq \boldsymbol{X}$ be a set of decision vectors. The function S

$$S \triangleq \sqrt{\frac{1}{\left|\boldsymbol{A}'\right| - 1} \sum_{i=1}^{\left|A'\right|} (\bar{d} - d_i)^2} \tag{10}$$

Where $d_i = \min_j \left\{ \sum_{k=1}^{p} \left| f_k(\boldsymbol{x}_i) - f_k(\boldsymbol{x}_j) \right| \right\} \quad \boldsymbol{x}_i, \boldsymbol{x}_j \in \boldsymbol{A}' \quad i, j = 1, \cdots \left|\boldsymbol{A}'\right| \quad \bar{d}$ is the mean of all d_i, and p is the number of objective functions.

4.1 Analysis of the influences of main parameters

In IDCMA, the main parameters are $G_{\max}$, n_{d}, n_{b}, n_{t}, n_{c}, p_{m}, p_{c} and the coding length c. The influences of the parameters $G_{\max}$, n_{d}, n_{b}, n_{t} and c to the performance are obvious, if not take the complexity into account, the larger their values, the better the results. The influences of n_{c}, p_{m} and p_{c} to the algorithm performance is more complex. The following is the empirical analysis results.

IDCMA can get different $\varsigma(\boldsymbol{P}_{\mathrm{ag}}, \boldsymbol{P}_{\mathrm{ture}}^{\Im})$ values with different parameter settings, where $\boldsymbol{P}_{\mathrm{ture}}^{\Im}$ is a set of solutions equidistantly spaced at the Pareto-optimal fronts and $\boldsymbol{P}_{\mathrm{ag}}$ is a set of decision vectors. We take the following test function for example:

$$\begin{aligned} &\min F(x, y) = \left(f_1(x, y), f_2(x, y) \right), \\ &f_1(x, y) = \frac{1}{x^2 + y^2 + 1}, \quad f_2(x, y) = x^2 + 3y^2 + 1, \\ &S.T. -3 \le x, y \le 3 \end{aligned} \tag{11}$$

We code x and y with binary string of 10 bits long. The main parameters are as follows: $G_{\max}$=150, n_{d}=100, $n_{\mathrm{b}}=100$, $n_{\mathrm{t}}=50$. We analyze the parameters n_{c}, p_{m} and p_{c} one after one by sampling a parameter with the same interval while fixing the other parameters. Choose 5000 solutions equidistantly spaced at the ideal Pareto-optimal front to compose $\boldsymbol{P}_{\mathrm{ture}}^{\Im}$. The data are the statistical results obtained from 10 times of random running.

i Influence of Clonal Scale

Let p_{m}=1/c and p_{c}=1. The clonal scale n_{c} is sampled by the same interval of 50 between 100 and 500 and runs 10 times to get the maximum, the minimum, the average values and the deviation of $\varsigma(\boldsymbol{P}_{\mathrm{ag}}, \boldsymbol{P}_{\mathrm{ture}}^{\Im})$ which are shown in Figure 1(a). It can be seen from the results that the influence of clonal scale to the algorithm is notable and the average value of $\varsigma(\boldsymbol{P}_{\mathrm{ag}}, \boldsymbol{P}_{\mathrm{ture}}^{\Im})$ increases by approximate linearity with the increasing of n_{c}. Practically, under the given experiment condition, the value of ς will increase approximately by 0.0122 when n_{c} increases by 50, but the computational complexity will increase $o((50-n_{\mathrm{t}})^2+50\times c)$ accordingly. Similar results are obtained by a great deal of experiments to other test problems.

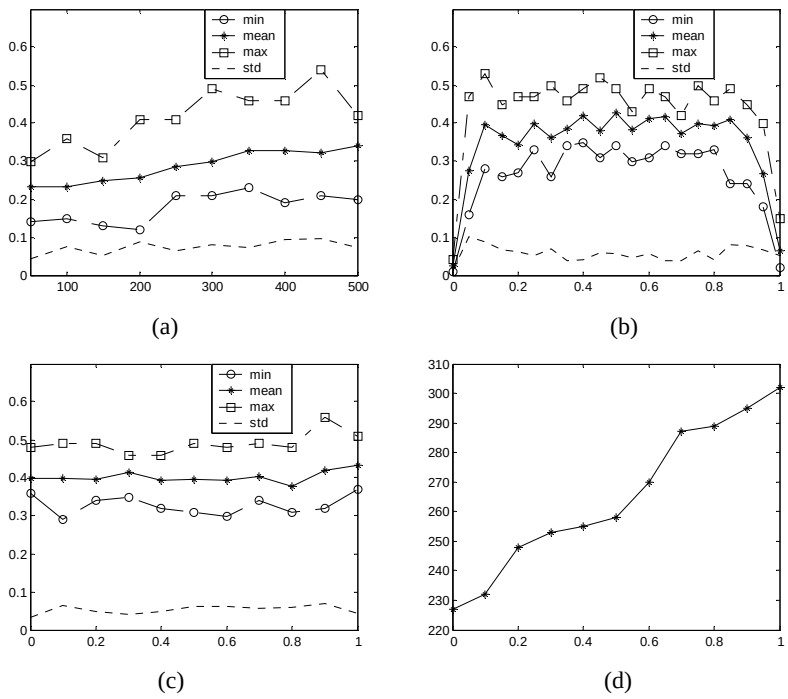

Fig. 1. The influence of the main parameters to the performance of the algorithm

ii Influence of mutation probability

Let $n_{\mathrm{c}}=300$, p_{c}=1. Sample the mutation probability by the interval of 0.05 from 0 to 1 and run 10 times to get the maximum, the minimum, the average values and the

deviation of $\varsigma(\boldsymbol{P}_{\mathrm{ag}}, \boldsymbol{P}_{\mathrm{ture}}^{\mathfrak{I}})$ which are shown in Figure 1(b). It can be seen that the change of ς is not obviously when the immune probability gets values from 0.1 to 0.8, but ς will decrease obviously when the mutation probability doesn't get the values from 0.1 to 0.8, which is obviously different from the influence of the mutation operation in genetic algorithm.

iii Influence of recombination probability

Let $n_{\mathrm{c}} = 300$, p_{m}=0.3. Sample the recombination probability by the interval of 0.1 from 0 to 1 and run 10 times to get the maximum, the minimum, the average values and the deviation of $\varsigma(\boldsymbol{P}_{\mathrm{ag}}, \boldsymbol{P}_{\mathrm{ture}}^{\mathfrak{I}})$ which are shown in Figure 1(c). It can be seen that the influence of recombination probability to ς is not obviously. Actually, p_{c} influences mainly the convergent speed. In order to explain the influence of p_{c} quantitatively, we adopt the following estimate manner:

Let the terminal generation be 10 and compare the influences of p_{c} to the amount of the nondominated solutions obtained from IDCMA. In order to eliminate the influence of other parameters, we set the parameters as follows:

$G_{\max}$=10, the immune dominance antibody population size is not confined, $n_{\mathrm{b}} = 100$, $n_{\mathrm{t}} = 50$, $n_{\mathrm{c}} = 300$, and p_{m}=0.3. The changes of the amount of nondominated solutions obtained are shown in Figure 1(d). It can be seen that the influence of p_{c} to the amount of nondominated solutions is obviously.

4.2 Test and results analysis

In order to validate the algorithm, we compare the algorithm with another three algorithms. They are Ishibuchi's Random-Weight genetic algorithm (RWGA)[7], Zitzler's Strength Pareto Evolutionary Algorithm (SPEA)[3] and Coello Coello's Multiobjective Immune System Algorithm (MISA)[5]. We design the software emulator of IDCMA using Matlab 6.1, and simulate RWGA and SPEA as exactly as we can under the same conditions. It is necessary to note that the performance of a MOEA in tackling multiobjective constrained optimization problems maybe largely depend on the constraint-handling technique used, so we don't consider side-constrained problems in this Paper. The parameters setting are as follows.

In IDCMA, the halt generation $G_{\max}$=150, immune dominance antibody population size $n_{\mathrm{d}} = 100$, antibody population size $n_{\mathrm{b}} = 100$, dominance clonal antibody population size $n_{\mathrm{t}} = 50$, clonal scale $n_{\mathrm{c}} = 300$, coding length $c = 8 \times n$ where n is the number of variables, mutation probability $p_{\mathrm{m}}=2/c$, recombination probability p_{c}=1.

In RWGA, the terminal generation is 150, population size is 300, the number of elite solutions is 10, crossover probability is 0.9, mutation probability is 0.6, and coding length is $8 \times n$ where n is the number of variables.

In SPEA, the terminal generation is 150, population size is 200, the number of elite solutions is 100, crossover probability is 0.9, mutation probability is 0.6, and coding length is $8 \times n$ where n is the number of variables.

For finding the reference solution set of each test problem, IDCMA needs to evaluate function values 45,100 times, RWGA needs 45,600 times, and SPEA needs

45,500 times. But the computation time of RWGA is the shortest, and that of SPEA is the longest. When MISA be concerned, we only take the comparison of the results shown in reference [5]. All the following results are the statistical data obtained from 30 times of random running.

Test 1

We consider a multiobjective problem having a Pareto-optimal front that is discontinuous and concave [11].

$$\min F(x,y) = (f_1(x), f_2(x)), \tag{12}$$

$$f_1(x) = \begin{cases} -x & if\ x \le 1 \\ -2+x & if\ 1 < x < 3 \\ 4-x & if\ 3 < x \le 4 \\ -4+x & if\ x > 4 \end{cases}, \quad f_2(x) = (x-5)^2,$$

$$S.T. \quad -5 \le x \le 10$$

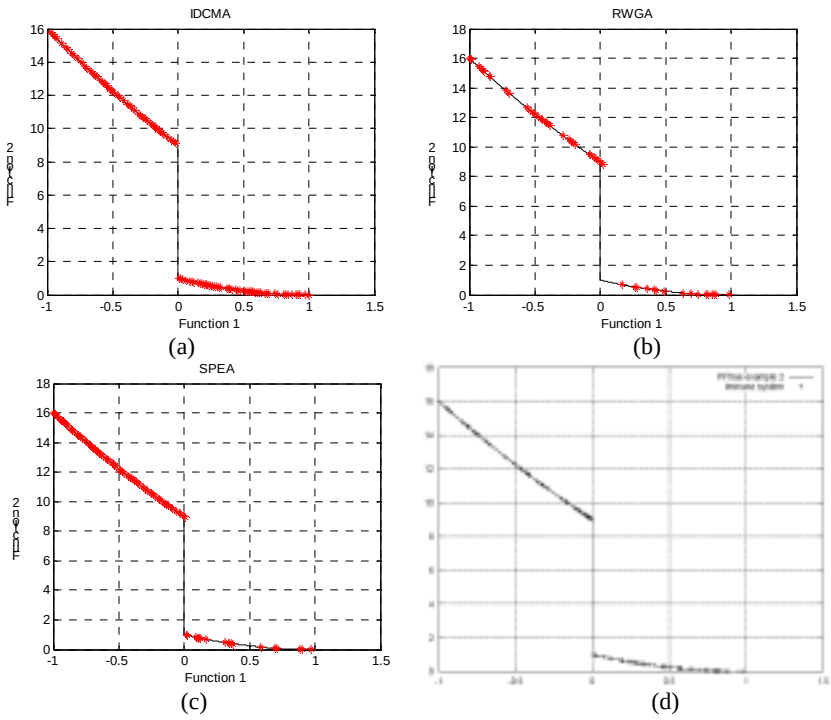

Fig. 2. The Pareto-optimal solution distributions corresponding to test 1. (a) The solution distribution solved by IDCMA; (b) The solution distribution solved by RWGA; (c) The solution distribution solved by SPEA; (d) The solution distribution solved by MISA.

The Pareto-optimal front is divided into two parts because of the discontinuity in f_1. Figure 2 shows the Pareto-optimal solution distributions solved by IDCMA, RWGA, SPEA and MISA, in which the real lines denote the Pareto-optimal fronts. The statistical results for data of the two metrics are shown in table 1 and table 2.

For this problem, IDCMA can obtain 100 Pareto-optimal solutions per time, while SPEA can get 132 Pareto-optimal solutions and RWGA can get 50 Pareto-optimal solutions per time on average. It can be seen from figure 2 that all of the four algorithms can be good convergent to the Pareto fronts, but for the distributions, IDCMA is better than the other three algorithms. Table 1 shows that the quality of the optimal solution set gained by IDCMA is better than those gained by RWGA and SPEA. From table 2 we can see that as far as the metric Spacing is concerned, IDCMA is little worse than SPEA, but it is far better than MISA and RWGA.

Test 2

The second problem is a more complicated multiobjective problem having a Pareto-optimal front that is discontinuous and concave[11].

$$\min F(x,y) = (f_1(x,y),\ f_2(x,y)), \tag{13}$$

$$f_1(x,y) = x, \quad f_2(x,y) = (1+10y)\times[1-(\frac{x}{1+10y})^2 - \frac{x}{1+10y}\sin(8\pi x)],$$

$$S.T. \quad 0 \le x, y \le 1$$

The Pareto-optimal front is divided into four parts because of the periodicity in f_2. Figure 3 shows the Pareto-optimal solution distributions solved by IDCMA, RWGA, SPEA and MISA.

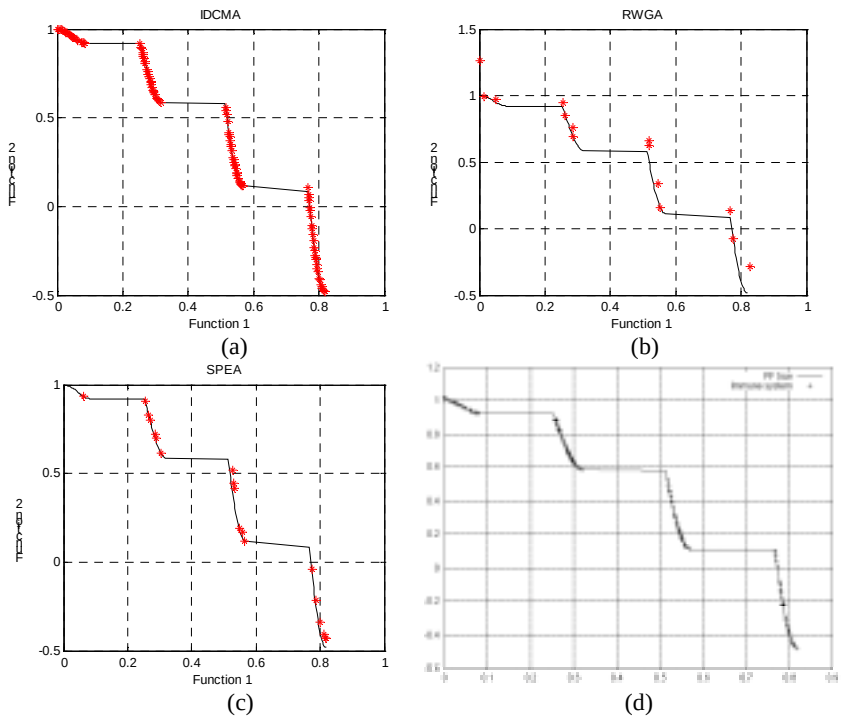

Fig. 3. The Pareto-optimal solution distributions corresponding to test 2.

For this problem, IDCMA can obtain 100 Pareto-optimal solutions per time, while SPEA can get 20 Pareto-optimal solutions and RWGA can get 14 Pareto-optimal

solutions per time on average. It can be seen from table 1 that $\varsigma(\boldsymbol{X}^{\mathrm{I}},\boldsymbol{X}^{\mathrm{S}})$ and $\varsigma(\boldsymbol{X}^{\mathrm{I}},\boldsymbol{X}^{\mathrm{R}})$ are obviously greater than $\varsigma(\boldsymbol{X}^{\mathrm{S}},\boldsymbol{X}^{\mathrm{I}})$ and $\varsigma(\boldsymbol{X}^{\mathrm{R}},\boldsymbol{X}^{\mathrm{I}})$, which shows that the Pareto-optimal solutions gained by IDCMA dominate those gained by the other two algorithms. When compared with MISA, IDCMA is more predominant in the metric Spacing. It can be seen from Figure 3 that the distribution of the solutions gained by IDCMA is better than those gained by RWGA, SPEA and MISA.

Test 3

Next, we consider another multiobjective problem[11].

$$\min F(x,y)=\left(f_1(x,y),\, f_2(x,y)\right), \tag{14}$$

$$f_1(x,y)=\sqrt[8]{x^2+y^2},\quad f_2(x,y)=\sqrt[4]{(x-0.5)^2+(y-0.5)^2},$$

$$S.T.-5\le x,y\le 10$$

This problem is a typical reflection of many to one. Figure 4(a) shows the global Pareto-optimal front in the f_1-f_2 space. Figures 4(b) (c) and (d) show the Pareto-optimal solution distributions solved by IDCMA, RWGA and SPEA separately.

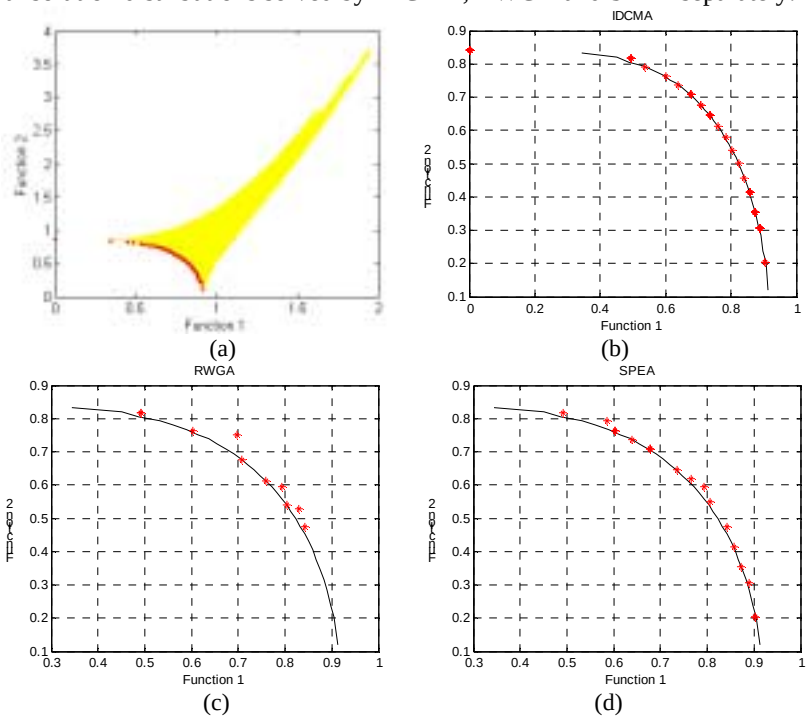

Fig. 4. The Pareto-optimal solution distributions corresponding to test 3. (a) The global Pareto-optimal front; (b) The solution distribution solved by IDCMA; (c) The solution distribution solved by RWGA; (d) The solution distribution solved by SPEA.

For this problem, IDCMA can obtain 100 Pareto-optimal solutions per time, while SPEA can get 26 Pareto-optimal solutions and RWGA can get 10 Pareto-optimal

solutions per time on average. Because this is a reflection of many to one, one dot in the objective space correspond to many dots in the domain, so the number of the dots for 100 optimal solutions gained by algorithm IDCMA displayed in the objective space is much less than 100. Figure 4 (a) shows that its Pareto-optimal front is very complicated, especially for the solution of $f_1 = 0$, SPEA and RWGA are difficult to find it, while IDCMA always find it in 30 times running. Figure 4 shows that the Pareto-optimal solution distribution solved by IDCMA is much better than those solved by SPEA and RWGA. The values of Spacing also show that quantitatively. The values in table 1 show that IDCMA produced solutions that clearly dominated or equals those generated by SPEA and RWGA.

Test 4

Let us consider a three-objective optimization problem having two variables[11]:

$$\min F(x,y) = \left(f_1(x,y), f_2(x,y), f_3(x,y) \right) \tag{15}$$

$$f_1(x,y) = \frac{(x-2)^2}{2} + \frac{(y+1)^2}{13} + 3, \quad f_2(x,y) = \frac{(x+y-3)^2}{36} + \frac{(-x+y+2)^2}{8} - 17,$$

$$f_3(x,y) = \frac{(x+2y-1)^2}{175} + \frac{(2y-x)^2}{17} - 13,$$

$$S.T. -4 \le x, y \le 4$$

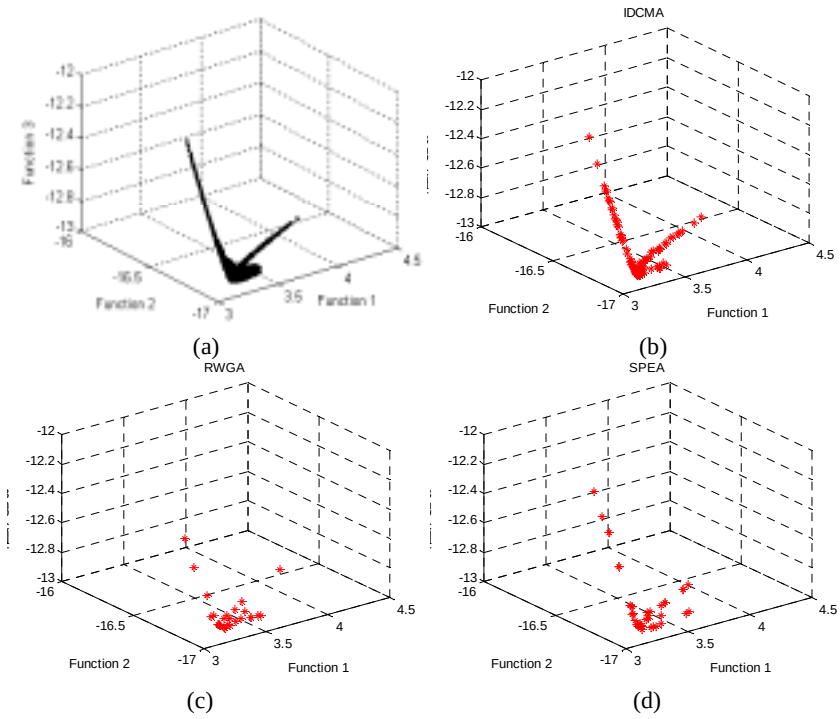

Fig. 5. The Pareto-optimal solution distributions corresponding to test 4.

Figure 5(a) shows the global Pareto-optimal front in function space. Figures 5(b) (c) and (d) show the Pareto-optimal solution distributions solved by IDCMA, RWGA and SPEA separately.

For this three-objective problem, IDCMA can also obtain 100 Pareto-optimal solutions per time, while SPEA can only get 40 Pareto-optimal solutions and RWGA can get 27 Pareto-optimal solutions per time on average. For this problem, IDCMA still has the obvious dominance. Figure 5(b) is most close to Figure 5(a), which reflects directly that the distribution of the solutions gained by IDCMA is the most ideal. It can be seen from table 1 that $\varsigma(\boldsymbol{X}^{\mathrm{I}},\boldsymbol{X}^{\mathrm{S}})$ and $\varsigma(\boldsymbol{X}^{\mathrm{I}},\boldsymbol{X}^{\mathrm{R}})$ are much greater than $\varsigma(\boldsymbol{X}^{\mathrm{S}},\boldsymbol{X}^{\mathrm{I}})$ and $\varsigma(\boldsymbol{X}^{\mathrm{R}},\boldsymbol{X}^{\mathrm{I}})$, which reflects objectively that the optimal solutions gained by IDCMA dominate those gained by the other two algorithms. It can be seen from metric Spacing that the solutions gained by IDCMA is the most uniform.

Test 5

At last, we consider a two-objective problem having three variables [11]:

$$\min F(\vec{x}) = \left(f_1(\vec{x}), f_2(\vec{x}) \right), \tag{16}$$

$$f_1(\vec{x}) = \sum_{i=1}^{n-1}\left(-10e^{(-0.2)\sqrt{x_i^2+x_{i+1}^2}}\right), \quad f_2(\vec{x}) = \sum_{i=1}^{n}\left(\left|x_i\right|^{0.8} + 5\sin\left(x_i\right)^3\right)$$

$$S.T. -5 \le x_i \le 5, i = 1,2,3, n = 3$$

Figure 6(a) shows the global Pareto-optimal front in function space. Figures 6(b) (c) and (d) show the Pareto-optimal solution distributions solved by IDCMA, RWGA and SPEA separately.

For this problem, IDCMA can obtain 100 Pareto-optimal solutions per time, while SPEA can get 24 Pareto-optimal solutions and RWGA can get 12 Pareto-optimal solutions per time on average. It can be seen from table 1 that $\varsigma(\boldsymbol{X}^{\mathrm{I}},\boldsymbol{X}^{\mathrm{S}})$ and $\varsigma(\boldsymbol{X}^{\mathrm{I}},\boldsymbol{X}^{\mathrm{R}})$ are much greater than $\varsigma(\boldsymbol{X}^{\mathrm{S}},\boldsymbol{X}^{\mathrm{I}})$ and $\varsigma(\boldsymbol{X}^{\mathrm{R}},\boldsymbol{X}^{\mathrm{I}})$. The values of metric Spacing show that the solutions gained by IDCMA are the most uniform. In addition, especially for the isolated optimal point $f_1 = 0$ in the objective space, SPEA and RWGA can not find it in 30 times runs, but IDCMA can find this point very well, which can show adequately that IDCMA have a stronger ability for the global search.

The statistical results for data of the two metrics of these nondominated solutions are shown in table 1 and table 2. In which $\boldsymbol{X}^{\mathrm{I}}$ denotes the solutions solved by IDCMA, $\boldsymbol{X}^{\mathrm{S}}$ denotes the solutions solved by SPEA, and $\boldsymbol{X}^{\mathrm{R}}$ denotes the solutions solved by RWGA. '/' means no correlative data.

Table 1. Average results of the metric Coverage of Two Sets

No.	$\varsigma(\boldsymbol{X}^{\mathrm{I}},\boldsymbol{X}^{\mathrm{S}})$	$\varsigma(\boldsymbol{X}^{\mathrm{S}},\boldsymbol{X}^{\mathrm{I}})$	$\varsigma(\boldsymbol{X}^{\mathrm{I}},\boldsymbol{X}^{\mathrm{R}})$	$\varsigma(\boldsymbol{X}^{\mathrm{R}},\boldsymbol{X}^{\mathrm{I}})$	$\varsigma(\boldsymbol{X}^{\mathrm{S}},\boldsymbol{X}^{\mathrm{R}})$	$\varsigma(\boldsymbol{X}^{\mathrm{R}},\boldsymbol{X}^{\mathrm{S}})$
Test 1	0.727525	0.466000	0.722667	0.298667	0.516000	0.342929
Test 2	0.833333	0.014333	0.792857	0.014667	0.238095	0.206667
Test 3	0.994872	0.548667	0.996667	0.277333	0.626667	0.319231
Test 4	0.428333	0.037000	0.388889	0.038000	0.161728	0.163333
Test 5	0.788889	0.016000	0.905556	0.004333	0.441667	0.101389

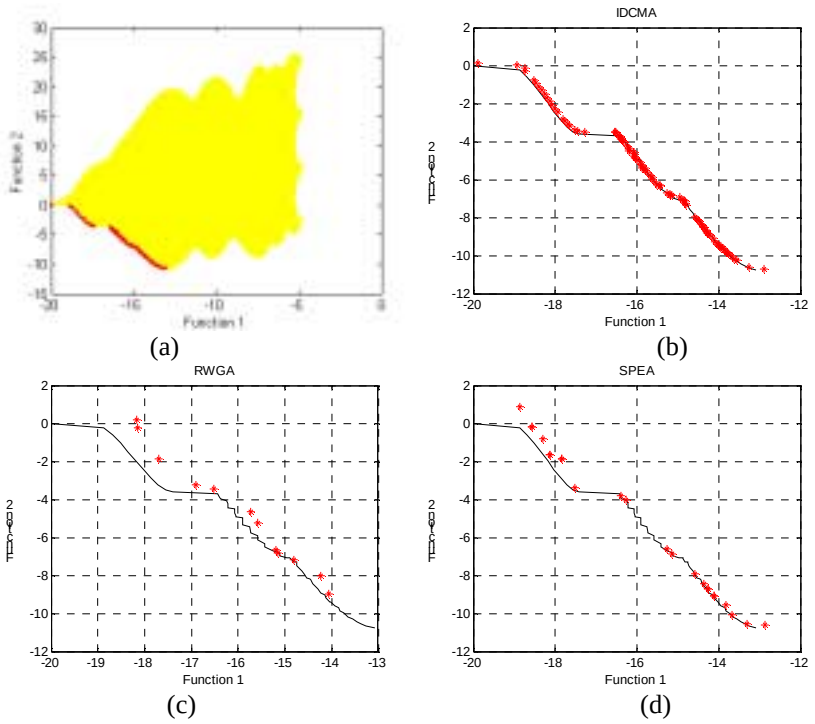

Fig. 6. The Pareto-optimal solution distributions corresponding to test 5. (a) The global Pareto-optimal front; (b) The solution distribution solved by IDCMA; (c) The solution distribution solved by RWGA; (d) The solution distribution solved by SPEA.

Table 2. Average results of the metrics Spacing

Algorithm		IDCMA	MISA	SPEA	RWGA
Spacing *S*	Test 1	0.057842	0.107427	0.051856	0.127075
	Test 2	0.047654	0.114692	0.077124	0.438289
	Test 3	0.033705	/	0.045433	0.059720
	Test 4	0.032992	/	0.076490	0.127397
	Test 5	0.134716	/	0.479929	0.738698

We adopted two popular numerical metrics, Convergence of Two Sets and Spacing, selected five typical multiobjective problems, and compared with the other three advanced multiobjective algorithms. The simulation results show that Immune Dominance Clone Multiobjective algorithm proposed in this paper can solve the low-dimensional multiobjective problems very well. Especially for the discontinuous Pareto-optimal fronts or the isolated optimal solutions, IDCMA can also construct and find them while the other algorithms seem incapable sometimes. In addition, Figure 2 to Figure 6 testifies the rationality of the conclusion above intuitively.

5 Conclusion and Prospective

In this paper, the basic concepts of artificial immune system are presented and a novel algorithm, Immune Dominance Clonal Multiobjective Algorithm, inspired by the concept of immunodominance and the clonal selection theory, is proposed. When compared with RWGA, SPEA and MISA, IDCMA is more effective for low-dimensional multiobjective optimization problems in the two popular metrics, Spacing and Coverage of Two Sets.

Although IDCMA can solve some low-dimensional multiobjective problems preferably, it adopts binary coding, so it can not solve high-dimensional problems with low computational complexity. To design a suitable antibody coding mode and computing method of antibody-antibody affinity is our next work.

References

1. Schaffer, J.D.: Multiple objective optimization with vector ecaluated genetic algorithms. PhD thesis, Vanderbilt University (1984)
2. Abido, M.A.: Environmental economic power dispatch using multiobjective evolutionary algorithms. IEEE Trans. Power Systems, Vol.18, No. 4, November (2003)
3. Zitzler, E., Thiele, L.: Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. IEEE Trans. Evolutionary Computation. Vol. 3, No. 4, November (1999)
4. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In: Giannakoglou, K., Tsahalis, D., Periaux, J., Papailou, P., Fogarty, T. (eds.): EUROGEN 2001, Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems. Athens, Greece (2002)95–100
5. Coello Coello, C.A., Nareli, C.C.: An Approach to Solve Multiobjective Optimization Problems Based on an Artificial Immune System. In: Jonathan, T., Peter, J.B. (eds.): Proceedings of the First International Conference on Artificial Immune Systems. Canterbury, UK (2002)212–221
6. Du, H.F., Jiao, L.C., Gong, M.G., Liu, R.C.: Adaptive Dynamic Clone Selection Algorithms. In: Zdzislaw, P., Lotfi, Z. (eds.): Proceedings of the Fourth International Conference on Rough Sets and Current Trends in Computing. Uppsala, Sweden (2004)
7. Ishibuchi, H., Murata, T.: A multiobjective genetic local search algorithm and its application to flowshop scheduling. IEEE Trans. System, Man and Cybernetics. Vol. 28, No. 3, (1998) 392–403
8. Abbas, A.K., Lichtman, A.H., Pober, J.S.: Cellular and Molecular Immunology. 3rd edn. W. B. Saunders Company, New York (1998)
9. Zitzler, E.: Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. A dissertation submitted to the Swiss Federal Institute of Technology Zurich for the degree of Doctor of Technical Sciences. Diss. Eth No. 13398 (1999)
10. Schott, J.R.: Fault Tolerant Design Using Single and Multictiteria Gentetic Algorithm Optimization. Master's thesis, Massachusetts Institute of Technology,Cambridge, Massachusetts, May (1995)
11. David, A.V.: Multiobjective Evolutionary Algorithms: Classification, Analyses, and New Innovations. PhD thesis. Presented to the Faculty of the Graduate School of Engineering of he Air Force Institute of Technology. Air University. USA. AFIT/DS/ENG (1999)