Available online at www.sciencedirect.com
ScienceDirect
Progress in Natural Science 19 (2009) 237–253
Progress in Natural Science
www.elsevier.com/locate/pnsc

Immune secondary response and clonal selection inspired optimizers

Maoguo Gong [a,*], Licheng Jiao [a], Lining Zhang [a], Haifeng Du [b]

[a] *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*
[b] *School of Public Policy and Administration, Xi'an Jiaotong University, Xi'an 710049, China*

Received 7 March 2008; received in revised form 12 May 2008; accepted 19 May 2008

Abstract

The immune system's ability to adapt its B cells to new types of antigen is powered by processes known as clonal selection and affinity maturation. When the body is exposed to the same antigen, immune system usually calls for a more rapid and larger response to the antigen, where B cells have the function of negative adjustment. Based on the clonal selection theory and the dynamic process of immune response, two novel artificial immune system algorithms, secondary response clonal programming algorithm (SRCPA) and secondary response clonal multi-objective algorithm (SRCMOA), are presented for solving single and multi-objective optimization problems, respectively. Clonal selection operator (CSO) and secondary response operator (SRO) are the main operators of SRCPA and SRCMOA. Inspired by the clonal selection theory, CSO reproduces individuals and selects their improved maturated progenies after the affinity maturation process. SRO copies certain antibodies to a secondary pool, whose members do not participate in CSO, but these antibodies could be activated by some external stimulations. The update of the secondary pool pays more attention to maintain the population diversity. On the one hand, decimal-string representation makes SRCPA more suitable for solving high-dimensional function optimization problems. Special mutation and recombination methods are adopted in SRCPA to simulate the somatic mutation and receptor editing process. Compared with some existing evolutionary algorithms, such as OGA/Q, IEA, IMCPA, BGA and AEA, SRCPA is shown to be able to solve complex optimization problems, such as high-dimensional function optimizations, with better performance. On the other hand, SRCMOA combines the Pareto-strength based fitness assignment strategy, CSO and SRO to solve multi-objective optimization problems. The performance comparison between SRCMOA, NSGA-II, SPEA, and PAES based on eight well-known test problems shows that SRCMOA has better performance in converging to approximate Pareto-optimal fronts with wide distributions.

Keywords: Clonal selection; Secondary response; Artificial immune systems; Evolutionary algorithms

1. Introduction

Artificial immune systems (AIS) make use of the mechanism of vertebrate immune system in terms of the model of information processing, and they construct new intelligent methods for solving problems. These methods provide the evolutionary learning mechanism like robustness, unsupervised learning, self-organization, and memory. Therefore, AIS have received a significant amount of interest from researchers and industrial sponsors in the recent years. Some of the first work in applying immune system paradigms was undertaken in the area of fault diagnosis [1]. Later work applied immune system paradigms to the field of computer security [2], which seemed to act as a catalyst for further investigation of the immune system as a metaphor in many areas, such as anomaly detection [3,4], pattern recognition [5–8], job shop scheduling [9,10], optimization [11–13] and engineering design optimization [14,15]. As de Castro and Timmis said, the field of AIS is showing great promise of being powerful computing paradigms [16].

* Corresponding author. Tel.: +86 29 88202661; fax: +86 29 88201023.
E-mail address: gong@ieee.org (M. Gong).

1002-0071/$ - see front matter
doi:10.1016/j.pnsc.2008.05.026

The immune system's ability to adapt its B cells to the new types of antigen is powered by processes known as clonal selection and affinity maturation [17]. Most immune system-inspired optimization algorithms are based on the applications of the clonal selection principle [18]. Clonal selection algorithms have been popularized mainly by de Castro and Von Zuben's CLONALG [6]. CLONALG selects part fittest antibodies to clone proportionally to their antigenic affinities. The hypermutation operator generates the matured clone population by performing an affinity maturation process inversely proportional to the fitness values. After computing the antigenic affinity of the matured clone population, CLONALG creates randomly part new antibodies to replace the lowest fitness antibodies in current population and retain best antibodies to recycle. Subsequently, de Castro and Timmis proposed an artificial immune network called opt-aiNet [19] 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. Kelsey and Timmis proposed the B cell Algorithm (BCA) in Ref. [20]. Through a process of evaluation, cloning, mutation and selection, BCA evolves a population of individuals (B cells) towards a global optimum. Each member of the B cell population can be considered as an independent entity. Garrett has presented an attempt to remove all the parameters from the clonal selection algorithm [21]. This method, which is called ACS for short, attempts to self-evolve various parameters during a single run. Cutello, Nicosia and Pavone proposed an immune algorithm for optimization called opt-IA [22,23]. Opt-IA uses three immune operators, i.e. cloning, hypermutation and aging. In 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, $(\mu+\lambda)$-selection operator. As far as multi-objective optimization is concerned, MISA [24,25] may be the first attempt to solve general multi-objective 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 [26] proposed a vector artificial immune system (VAIS) for solving multi-objective 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 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. Cutello, Narzisi and Nicosia [27] modified the (1+1)-PAES using 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 them in Table 1 where SOPs are short for single-objective optimization problems and MOPs are short for multi-objective optimization problems.

In this paper, based on the clonal selection theory and the dynamic process of immune response, two novel artificial immune system algorithms, called secondary response clonal programming algorithm (SRCPA) and secondary response clonal multi-objective optimization algorithm (SRCMOA), are presented. Clonal selection operator (CSO) and secondary response operator (SRO) are the main operators of SRCPA and SRCMOA. Inspired by the clonal selection theory, CSO performs a greedy search which reproduces individuals and selects their improved maturated progenies after the affinity maturation process, and so single individuals will be optimized locally and the newcomers yield a broader exploration of the search space. SRO copies certain antibodies to a secondary pool, whose members do not participate in CSO, but they could be activated by some external stimulation. The update of the secondary pool pays more attention to maintain the population diversity. SRCPA adopts decimal representation rather than binary representation and uses special mutation and recombination methods designed for large parameter optimization problems. The experimental results indicate that SRCPA converges faster than some existing evolutionary algorithms, such as BGA [28], AEA [29], OGA/Q [30], IEA [31] and IMCPA [32], for solving high-dimensional function optimization problems. SRCMOA tries to get more various non-dominated solutions by combining CSO, SRO and previous MOEAs' techniques such as Pareto-strength based fitness assignment strategy [33]. The performance comparison between SRCMOA, NSGA-II [34], SPEA [35], and PAES [36] based on the eight well-known test problems shows that SRCMOA has a good performance in converging to approximate Pareto-optimal fronts with wide distributions.

Table 1
Main optimization algorithms inspired by immune system.

Algorithm	Key operators	Applications
CLONALG	Elitist selection, cloning, mutation, death	SOPs
Opt-aiNet	Elitist selection, cloning, mutation, network suppression, death	SOPs
BCA	Cloning, metadynamics, contiguous mutation, selection	SOPs
ACS	Elitist selection, cloning, mutation, death	SOPs
opt-IA	$(\mu+\lambda)$-selection, cloning, hypermutation, hypermacromutation, aging	SOPs
MISA	Elitist selection according to fitness and diversity, cloning, uniform mutation and non-uniform mutation, archive update	MOPs
VAIS	Cloning, mutation, clonal selection, suppression, death, archive update	MOPs
I-PAES	Cloning, mutation, clonal selection, (1+1)-selection, archive update	MOPs

2. Immunology background and terms

2.1. Clonal selection and immune response

The ability of the immune system to respond to an antigen exists before it ever encounters that antigen. The immune system relies on the prior formation of an incredibly diverse population of B cells and T cells. The specificity of both the B-cell receptors (BCRs) and the T-cell receptors (TCRs), i.e. the epitope to which a given receptor can bind, is created by a remarkable genetic mechanism. Each receptor is created even though the epitope it recognizes may never have been present in the body. If an antigen with that epitope should enter the body, those few lymphocytes able to bind to it will do so. If they also receive the second costimulatory signal, they may begin repeated rounds of mitosis. In this way, clones of antigen-specific lymphocytes (B and T) provide the basis of the immune response. This phenomenon is called clonal selection [18], which is a dynamic process of the immune system self-adaptive against antigen stimulation. According to Burnet, biological clonal selection occurs according to the degree that a B-cell matches an antigen. A strong match causes a B-cell to be cloned many times, and a weak match results in little cloning.

As shown in Fig. 1, after recovering from an infection, the concentration of antibodies against the infectious agent gradually declines over the ensuing weeks, months, or even years. A time may come till antibodies against that agent can no longer be detected. Nevertheless, the individual often is still protected against the second case of the disease, i.e. the person is still immune. In fact, the second exposure to the agent usually calls for a more rapid and larger response to the antigen. This is called the secondary response. The secondary response reflects a larger number of antigen-specific cells than those existed before the primary response. During the initial expansion of clones, some of the progeny cells neither went on dividing nor developed into plasma cells. Instead, they reverted to small lymphocytes bearing the same BCR on their surface that their ancestors had. This lays the foundation for a more rapid and massive response when the next time the antigen enters the body.

In immunology, the association among immune systems is usually described as links of communication, and the factors determining the communication degree include cognizing or mistake, memory or forgetting, and knowledge or ignorance. The recent studies primarily clarify that the ability of organism in selecting and recognizing adventitious antigens comes from the selection and deletion of lymphocytes during the growth and differentiation process of immune system. The change of antibodies' quality and quantity is closely influenced by antigens, B and T cells, and other accessorial cells. B cells have the function of negative adjustment, maybe clonal deletion or clonal anergy [37]. If the difference between B cells and antibodies is ignored, the action of an antibody may be proliferation, anergy or deletion during the immune response.

2.2. Terms in this paper

In order to describe the algorithms well, we define the notations in our algorithms as follows, where boldface italic capital letters indicate matrices and boldface italic small letters indicate vectors.

(i) Antigen: the objective function(s) to be optimized (maximized or minimized).

(ii) Antibody: the representation of solution candidates. The limited-length character string $\boldsymbol{a}$ is the coding of variable vector $\boldsymbol{x}$, denoted by $\boldsymbol{a} = encode(\boldsymbol{x})$; **and** $\boldsymbol{x}$ is called the decoding of antibody $\boldsymbol{a}$, expressed as $\boldsymbol{x} = decode(\boldsymbol{a})$.

Set $\boldsymbol{I}$ is called antibody space, namely $\boldsymbol{a} \in \boldsymbol{I}$. The antibody population space is

$$\boldsymbol{I}^n = \{\boldsymbol{A} : \boldsymbol{A} = (\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n), \quad \boldsymbol{a}_k \in \boldsymbol{I}, \quad 1 \leqslant k \leqslant n\}, \quad (1)$$

where the positive integer n is the size of antibody population, and the antibody population $\boldsymbol{A} = \{\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n\}$ is an n-dimensional group of antibody $\boldsymbol{a}$, and it is a spot in the antibody space $\boldsymbol{I}$.

(iii) Affinity: the fitness measurement for an antibody. For single-objective optimization problems, the affinity $F(\cdot) \geqslant 0$ is a linear or nonlinear transformation of the value of the objective function $f(\cdot)$ for a given antibody. For multi-objective optimization problems, the affinity $F(\cdot)$ is computed based on the Pareto-strength based fitness assignment strategy in Ref. [33] but with a modification on diversity preservation, which will be described in Section 4.2, and $F(\cdot)$ is to be maximized both in SRCPA and in SRCMOA.

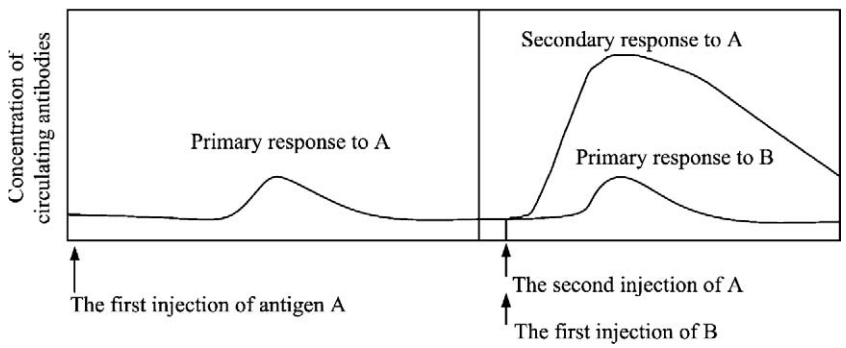

Fig. 1. Primary and secondary immune responses.

3. Clonal selection operator and secondary response operator

The mechanism of immune response and its characteristics, such as self–non-self recognition, clonal selection, memory learning, antibody diversity, immune tolerance and adaptive regulation, are all important paradigms of the research on AIS. Clonal selection operator (CSO) and secondary response operator (SRO) are two operators simulating the dynamic process of immune response. CSO, including clonal proliferation operation, affinity maturation operation, and clonal selection operation, reproduces antibodies and selects their improved progenies after the affinity maturation process, respectively, and so single individuals will be optimized locally and the newcomers yield a broader exploration of the search space. SRO copies certain antibodies to the secondary pool and the antibodies in the secondary pool do not participate in CSO, but they could be activated by some external stimulation, which is denoted as the activation operation of the secondary pool. A diversity maintenance strategy in the update of the secondary pool can maintain the population diversity efficiently.

3.1. Clonal selection operator

Inspired by the clonal selection theory, the CSO implements the clonal proliferation operation ($T_{\mathrm{P}}^{\mathrm{C}}$), affinity maturation operation ($T_{\mathrm{M}}^{\mathrm{A}}$) and clonal selection operation ($T_{\mathrm{S}}^{\mathrm{C}}$) on the antibody population $\boldsymbol{A}(k)$, where the antibody population at time k is represented by the time-dependent variable matrix $\boldsymbol{A}(k) = \{\boldsymbol{a}_1(k), \boldsymbol{a}_2(k), \ldots, \boldsymbol{a}_n(k)\}$. The evolution process can be described as

$$\boldsymbol{A}(k) \xrightarrow{T_{\mathrm{P}}^{\mathrm{C}}} \boldsymbol{Y}(k) \xrightarrow{T_{\mathrm{M}}^{\mathrm{A}}} \boldsymbol{Z}(k) \cup \boldsymbol{A}(k) \xrightarrow{T_{\mathrm{S}}^{\mathrm{C}}} \boldsymbol{A}(k+1) \tag{2}$$

The main operation of CSO is shown in Fig. 2.

3.1.1. Clonal proliferation operation

Define

$$\boldsymbol{Y}(k) = T_{\mathrm{P}}^{C}(\boldsymbol{A}(k)) = [T_{\mathrm{P}}^{C}(\boldsymbol{a}_1(k))T_{\mathrm{P}}^{C}(\boldsymbol{a}_2(k)), \ldots, T_{\mathrm{P}}^{C}(\boldsymbol{a}_n(k))]^{\mathrm{T}} \tag{3}$$

where $\boldsymbol{Y}_i(k) = T_{\mathrm{P}}^{C}(\boldsymbol{a}_i(k)) = \boldsymbol{e}_i \times \boldsymbol{a}_i(k), \quad i = 1, 2, \ldots, n$, and $\boldsymbol{e}_i$ is a q_{i}-dimensional identity column vector.

There are various methods for calculating q_i. In this study, we compute it as follows:

$$q_i(k) = \left\lceil N_{\mathrm{c}} \times \frac{F(\boldsymbol{a}_i(k))}{\sum_{j=1}^{n} F(\boldsymbol{a}_j(k))} \right\rceil \quad i = 1, 2, \ldots, n \tag{4}$$

where $N_{\mathrm{c}} > n$ is a given integer called clonal size. The value of $q_i(k)$ is proportional to the value of $F(\boldsymbol{a}_i(k))$.

After clonal proliferation, the population becomes

$$\boldsymbol{Y}(k) = \{\boldsymbol{Y}_1(k), \boldsymbol{Y}_2(k), \ldots, \boldsymbol{Y}_n(k)\} \tag{5}$$

where

$$\boldsymbol{Y}_i(k) = \{\boldsymbol{y}_{ij}(k)\} = \{\boldsymbol{y}_{i1}(k), \boldsymbol{y}_{i2}(k), \ldots, \boldsymbol{y}_{iq_i}(k)\} \text{ and}$$
$$\boldsymbol{y}_{ij}(k) = \boldsymbol{a}_i(k), \quad j = 1, 2, \ldots, q_i \quad i = 1, 2, \ldots, n \tag{6}$$

3.1.2. Affinity maturation operation

Inspired by immune response process, the affinity maturation operation $T_{\mathrm{M}}^{\mathrm{A}}$ is diversified basically by two mechanisms: hypermutation and receptor editing [12,38,39].

Random changes are introduced into the genes, i.e. mutation. Such changes may lead to an increase in the affinity of the clonal antibody occasionally. Antibodies had deleted their low-affinity receptors (genes) and developed entirely new ones through recombination, i.e. receptor editing [12]. Receptor editing offers the ability to escape from local optima on an affinity landscape.

After affinity maturation operation, the population becomes

$$\boldsymbol{Z}(k) = \{\boldsymbol{Z}_1(k), \boldsymbol{Z}_2(k), \ldots, \boldsymbol{Z}_n(k)\} \tag{7}$$

where

$$\boldsymbol{Z}_i(k) = \{\boldsymbol{z}_{ij}(k)\} = \{\boldsymbol{z}_{i1}(k), \boldsymbol{z}_{i2}(k), \ldots, \boldsymbol{z}_{iq_i}(k)\} \text{ and}$$
$$\boldsymbol{z}_{ij}(k) = T_{\mathrm{M}}^{\mathrm{A}}(\boldsymbol{y}_{ij}(k)), \quad j = 1, 2, \ldots, q_i \quad i = 1, 2, \ldots, n \tag{8}$$

3.1.3. Clonal selection operation

Define $\forall i = 1, 2, \ldots, n, \quad \boldsymbol{b}_i(k) \in \boldsymbol{Z}_i(k)$ is the best antibody (the antibody with the highest affinity) in $\boldsymbol{Z}_i(k)$, then

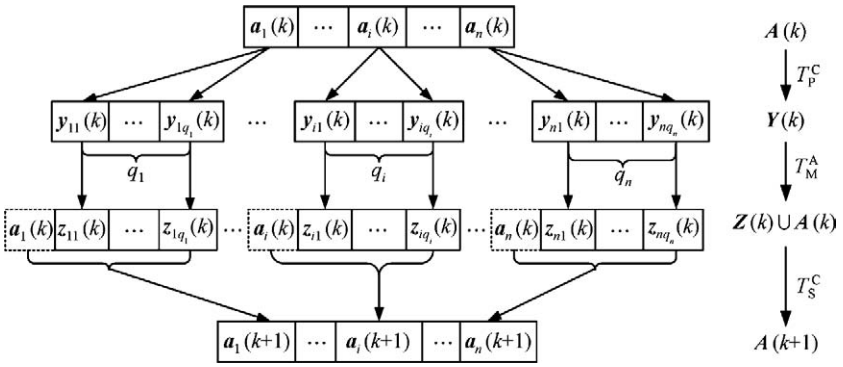

Fig. 2. The main operational process of CSO.

$$\begin{aligned}\boldsymbol{a}_i(k+1) &= T_{\mathrm{S}}^{\mathrm{C}}(\boldsymbol{Z}_i(k)\cup\boldsymbol{a}_i(k))\\ &= \begin{cases}\boldsymbol{b}_i(k) & \text{if } F(\boldsymbol{a}_i(k)) < F(\boldsymbol{b}_i(k))\\ \boldsymbol{a}_i(k) & \text{if } F(\boldsymbol{a}_i(k)) \geqslant F(\boldsymbol{b}_i(k))\end{cases}\end{aligned} \tag{9}$$

The newcome population is

$$\begin{aligned}\boldsymbol{A}(k+1) &= T_{\mathrm{S}}^{\mathrm{C}}(\boldsymbol{Z}(k)\cup\boldsymbol{A}(k))\\ &= [T_{\mathrm{S}}^{\mathrm{C}}(\boldsymbol{Z}_1(k)\cup\boldsymbol{a}_1(k)), T_{\mathrm{S}}^{\mathrm{C}}(\boldsymbol{Z}_2(k)\cup\boldsymbol{a}_2(k)),\ldots,\\ &\quad T_{\mathrm{S}}^{\mathrm{C}}(\boldsymbol{Z}_n(k)\cup\boldsymbol{a}_n(k))]^{\mathrm{T}}\\ &= \{\boldsymbol{a}_1(k+1),\boldsymbol{a}_2(k+1),\ldots,\boldsymbol{a}_n(k+1)\}\end{aligned} \tag{10}$$

where $\boldsymbol{a}_i(k+1) = T_{\mathrm{S}}^{\mathrm{C}}(\boldsymbol{Z}_i(k)\cup\boldsymbol{a}_i(k)) \quad i=1,2,\ldots,n$.

Similar to the $(\mu+\lambda)-$evolution strategies, i.e. $(\mu+\lambda)-$ ES, CSO also selects new populations among the offspring and parents together and employs self-adapting genetic variation mechanisms. But here it is necessary to point out that CSO has its own characteristics.

Firstly, the clonal selection operation $T_{\mathrm{S}}^{\mathrm{C}}$ in CSO is different from the truncation selection in $(\mu+\lambda)-$ ES [40]. $T_{\mathrm{S}}^{\mathrm{C}}$ selects the best single individual $\boldsymbol{a}_i(k+1)$ from the sub-population $\boldsymbol{Z}_i(k)\cup\boldsymbol{a}_i(k)$, respectively, and all the best individuals $\boldsymbol{a}_i(k+1), \quad i=1,2,\ldots,n$ constitute the new population. Therefore, single individuals will be optimized locally (exploitation of the surrounding space) by the affinity maturation process. However, truncation selection with deterministic choice of the best μ individuals from a set of λ offspring and μ parents emphasizes on the global selection on the whole population.

Secondly, $(\mu+\lambda)-$ ES often uses Gaussian or Cauchy mutations and deterministic rules to control the step size. For CSO, the mutation is implemented based on some somatic mutation-inspired methods.

In fact, the mechanism of CSO is based on the clonal selection principle rather than on the natural evolution.

3.2. Secondary response operator

In this study, we construct a secondary pool, which is called SP for short. The SP is a special antibody population whose members are not involved in CSO. The initial SP $\boldsymbol{M}(0) = \{\boldsymbol{m}_1(0),\boldsymbol{m}_2(0),\ldots,\boldsymbol{m}_s(0)\}$, where $\boldsymbol{m}_i(0), \quad i=1,2,\ldots,s$, is generated randomly. The SP at time k is represented by the time-dependent variable matrix $\boldsymbol{M}(k) = \{\boldsymbol{m}_1(k),\boldsymbol{m}_2(k),\ldots,\boldsymbol{m}_s(k)\}$.

SRO includes update operation of SP, and activation operation of SP. Since the problem features of single-objective and multi-objective optimization are different, we design different update operations of SP for SRCPA and SRCMOA, respectively.

3.2.1. Update operation of SP

For single-objective optimization problems, define $\boldsymbol{a}^*(k+1)\in\boldsymbol{A}(k+1)$ as the best antibody in antibody population $\boldsymbol{A}(k+1)$, then the update operation of SP can be described as follows.

Calculating the distance between $\boldsymbol{a}^*(k+1)$ and each member of $\boldsymbol{M}(k)$. If the distance between $\boldsymbol{m}_j(k)$ and $\boldsymbol{a}^*(k+1)$ is the minimum one, and its value is δ_{j0}, then there are two cases to be considered. In the first case, $\delta_{j0}\leqslant\delta_0$, where δ_0 is the minimum distance between every two individuals of $\boldsymbol{M}(k)$, then compare the affinities of $\boldsymbol{a}^*(k+1)$ and $\boldsymbol{m}_j(k)$, and let

$$\boldsymbol{m}_j(k+1) = \begin{cases}\boldsymbol{a}^*(k+1) & \text{if } F(\boldsymbol{a}^*(k+1)) > F(\boldsymbol{m}_j(k))\\ \boldsymbol{m}_j(k) & \text{else}\end{cases} \tag{11}$$

In the second case, $\delta_{j0}>\delta_0$, then add the antibody $\boldsymbol{a}^*(k+1)$ to SP, and delete the worst individual of SP.

For multi-objective optimization problems, SP preserves the current non-dominated individuals of the antibody population. So the update operation of SP is performed by copying all the non-dominated antibodies from the combined population of $\boldsymbol{A}(k)$ and $\boldsymbol{M}(k)$ to $\boldsymbol{M}(k+1)$. The individuals of SP can survive for several generations, unless they are removed when dominated by other new members or the current number of non-dominated individuals s' exceeds its upper limited s. If s' is larger than s, then excess individuals should be discarded using the environmental selection strategy in Ref. [33]. Therefore, SP serves as an elite set which is useful in maintaining the diversity and in improving the performance of SRCMOA.

After the update operation, the SP becomes $\boldsymbol{M}(k+1) = \{\boldsymbol{m}_1(k+1),\boldsymbol{m}_2(k+1),\ldots,\boldsymbol{m}_s(k+1)\}$.

3.2.2. Activation operation of SP

The individuals of SP can retroact at antibody population. The activation operation of $\boldsymbol{M}(k) = \{\boldsymbol{m}_1(k), \boldsymbol{m}_2(k),\ldots,\boldsymbol{m}_s(k)\}$ on the antibody population $\boldsymbol{A}(k) = \{\boldsymbol{a}_1(k),\boldsymbol{a}_2(k),\ldots,\boldsymbol{a}_n(k)\}$ is described as follows: randomly select $t = \lfloor T\%\times s\rfloor$ antibodies of $\boldsymbol{M}(k+1)$ to replace the worst t antibodies of $\boldsymbol{A}(k)$, where $T\%$ is a constant called activation ratio and s is the population size of SP. It can be implemented as the following Algorithm 1:

Algorithm 1. Activation operation of SP

Step 1: Sort the antibodies in population $\boldsymbol{A}(k)$ as their affinities and set $\boldsymbol{A}'(k) = \{\boldsymbol{a}'_1(k),\boldsymbol{a}'_2(k),\ldots,\boldsymbol{a}'_n(k)\}$ to the resulting population, where $F(\boldsymbol{a}'_i(k)) \geqslant F(\boldsymbol{a}'_{i+1}(k)) \quad i=1,2,\ldots,n-1$.

Step 2: Calculate the number of antibodies that should be activated: $t = \lfloor T\%\times s\rfloor$.

Step 3: Antibody population updating: $\mathbf{A}(k) = \{\mathbf{a}'_1(k), \mathbf{a}'_2(k),\cdots,\mathbf{a}'_{n-t}(k),\mathbf{m}_1(k),\mathbf{m}_2(k),\cdots,\mathbf{m}_t(k)\}$.

4. SRCPA and SRCMOA

Based on CSO and SRO, two algorithms are proposed in this section. SRCPA is designed for solving single-objective optimization problems and SRCMOA is designed for

solving multi-objective function optimization problems. The main power of SRCPA and SRCMOA arises mainly from CSO and SRO. In addition, decimal representation makes SRCPA more suitable for solving high-dimensional function optimization problems than other clonal selection algorithms with binary representation. Special mutation and recombination methods are adopted in SRCPA to simulate the hypermutation and receptor editing process. SRCMOA adopts the Pareto strength based fitness assignment strategy.

4.1. Secondary response clonal programming algorithm

SRCPA is designed for being an efficient general-purpose optimization algorithm for solving large-scale optimization problems. We consider the following optimization problem

$$\text{maximize } f(\boldsymbol{x}), \boldsymbol{x} = (x_1, x_2, \ldots, x_D) \in \boldsymbol{\Omega} \tag{12}$$

where $f(\boldsymbol{x})$ is the objective function, $\boldsymbol{x}$ is a variable vector in h^D, $\boldsymbol{\Omega} \subseteq h^D$ defines the feasible solution space which is a D-dimensional space bounded by the parametric constraints $\underline{x}_i \leqslant x_i \leqslant \bar{x}_i, \quad i = 1, 2, \ldots, D$. Thus, the feasible solution space $\boldsymbol{\Omega} = [\underline{\boldsymbol{x}}, \bar{\boldsymbol{x}}]$, where $\underline{\boldsymbol{x}} = (\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_D)$ and $\bar{\boldsymbol{x}} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_D)$.

4.1.1. Antibody representation and affinity measure

In this study, an antibody represents a search point in feasible solution space. For the single-objective optimization problem described in Eq. (12), an antibody $\boldsymbol{a} = (a_1, a_2, \ldots, a_l) = encode(\boldsymbol{x})$ is a normalized real-valued string, i.e.

$$\boldsymbol{a} = encode(\boldsymbol{x}) = (\boldsymbol{x} - \underline{\boldsymbol{x}})./(\bar{\boldsymbol{x}} - \underline{\boldsymbol{x}}) \tag{13}$$

and so $l = D$, and $0 \leqslant a_i \leqslant 1, \quad i = 1, 2, \ldots, D$.

We adopt real-valued representation rather than binary representation since the test problems that we are dealing with have continuous spaces, and real encoding should be preferred to avoid problems related to hamming cliffs.

For the single-objective optimization problem described in Eq. (12), the affinity measure $F(\boldsymbol{a}) > 0$ for antibody $\boldsymbol{a}$ is a mapping of the value of the objective function $f(\boldsymbol{x})$ where $\boldsymbol{a} = encode(x), \ x \in \boldsymbol{\Omega}$. In this study, we adopt the following mapping function:

$$F(\boldsymbol{a}) = \mathrm{e}^{f(decode(\boldsymbol{a}))} \tag{14}$$

The exponential function is an increasing function and it holds that $F(\boldsymbol{a}) > 0$ for all antibodies. Furthermore, the rate of grade of the exponential function also increases with the increasing variant value. It will redound to accelerate the convergence during final iterations.

4.1.2. Main loop of SRCPA

The simple SRCPA can be described as follows:

Algorithm 2. Secondary response clonal programming algorithm (SRCPA)

Step 1: Initialization: Randomly generate the initial antibody population
$\boldsymbol{A}(0) = \{\boldsymbol{a}_1(0), \boldsymbol{a}_2(0), \ldots, \boldsymbol{a}_n(0)\} \quad \in \boldsymbol{I}^n$
Randomly generate the SP
$\boldsymbol{M}(0) = \{\boldsymbol{m}_1(0), \boldsymbol{m}_2(0), \ldots, \boldsymbol{m}_s(0)\} \quad \in \boldsymbol{I}^s$
Calculate the affinity of all antibodies in $\boldsymbol{A}(0)$ and $\boldsymbol{M}(0)$, k:=0.

Step 2: CSO:

- **Step 2.1: Clonal proliferation operation:** $\boldsymbol{Y}(k)$ Get by applying $T_{\mathrm{P}}^{\mathrm{C}}$ to $\mathbf{A}(k)$.
- **Step 2.2: Affinity maturation operation:** Get $\boldsymbol{Z}(k)$ by applying $T_{\mathrm{M}}^{\mathrm{A}}$ to $\boldsymbol{Y}(k)$.
- **Step 2.3: Evaluation:** Calculate the affinity of all antibodies of $\boldsymbol{Z}(k)$.
- **Step 2.4: Clonal selection operation:** Get $\boldsymbol{A}(k+1)$ by applying $T_{\mathrm{S}}^{\mathrm{C}}$ to $\boldsymbol{Z}(k) \cup \boldsymbol{A}(k)$.

Step 3: SRO:

- **Step 3.1: Update operation of SP:** Get the new SP $\boldsymbol{M}(k+1)$ by update operation of $\boldsymbol{M}(k)$.
- **Step 3.2: Activation operation of SP:** Update $\boldsymbol{A}(k+1)$ by activation operation of $\boldsymbol{M}(k+1)$.

Step 4: Termination test: If a stopping condition is satisfied, stop the algorithm; otherwise, k:=k+1, go to **Step 2**.

In SRCPA, we design the hypermutation and receptor editing of CSO as follows.

For the antibody $\boldsymbol{y}_{ij}(k), \quad i = 1, 2, \ldots, n; \quad j = 1, 2, \ldots q_i$ in the antibody population $\boldsymbol{Y}(k)$, replace its certain numbers by a random integer between 0 and 9. For example, if a three-dimensional optimization problem is to be solved, the antibody $\boldsymbol{y}_{ij}(k) = (1.233567, 12.33\mathbf{4}567, 0.123356)$, and the second variable's fifth number is selected to mutate. Then we can randomly generate an integer between 0 and 9 to take the place of the number '4' in bold.

The receptor editing is designed similar to uniform recombination. Let $\boldsymbol{a} = (a_1, a_2, \ldots, a_D)$ be the antibody to be edited. Then randomly select a member $\boldsymbol{m} = (m_1, m_2, \ldots, m_D)$ from the current SP $\boldsymbol{M}(k)$. Randomly generate a Boolean vector $\boldsymbol{r} = (r_1, r_2, \ldots, r_D)$, where $r_i \in \{0, 1\}, i = 1, 2, \ldots, D$. Then the new antibody $\boldsymbol{a}'$ is expressed as $\boldsymbol{a}' = \{a'_1, a'_2, \ldots, a'_D\}$, where

$$a'_i = \begin{cases} a_i & \text{if } r_i = 0 \\ m_i & \text{if } r_i = 1 \end{cases}, \quad i = 1, 2, \ldots, D \tag{15}$$

Furthermore, the hypermutation and receptor editing in CSO is applied to each antibody of the antibody population $\boldsymbol{Y}(k)$ with probability 1.

4.2. Secondary response clonal multi-objective optimization algorithm

SRCMOA incorporates the Pareto strength-based fitness assignment strategy, CSO and SRO, for solving multi-objective optimization problems. The simple SRCMOA can be written as follows:

Algorithm 3. Secondary response clonal multi-objective optimization algorithm (SRCMOA).

Step 1: Initialization: Randomly generate the initial antibody population
$\boldsymbol{A}(0) = \{\boldsymbol{a}_1(0), \boldsymbol{a}_2(0), \ldots, \boldsymbol{a}_n(0)\} \in \boldsymbol{I}^n$
Randomly generate the SP
$\boldsymbol{M}(0) = \{\boldsymbol{m}_1(0), \boldsymbol{m}_1(0), \ldots, \boldsymbol{m}_s0)\} \in \boldsymbol{I}^s$
Calculate the affinity of all antibodies in $\boldsymbol{A}(0)$ and $\boldsymbol{M}(0)k := 0$.

Step 2: CSO:

Step 2.1: Clonal proliferation operation: Get $\boldsymbol{Y}(k)$ by applying T_{P}^{C} to $\boldsymbol{A}(k)$.

Step 2.2: Affinity maturation operation: Get $\boldsymbol{Z}(k)$ by applying $T_{\mathrm{M}}^{\mathrm{A}}$ to $\boldsymbol{Y}(k)$.

Step 2.3: Evaluation: Calculate the affinity of all antibodies of $\boldsymbol{Z}(k)$, $\boldsymbol{A}(k)$ and $\boldsymbol{M}(k)$.

Step 2.4: Clonal selection operation: Get $\boldsymbol{A}(k+1)$ by applying $T_{\mathrm{S}}^{\mathrm{C}}$ to $\boldsymbol{Z}(k) \cup \boldsymbol{A}(k)$.

Step 3: SRO:

Step 3.1: Update operation of SP: Get the new SP $\boldsymbol{M}(k+1)$ by update operation of $\boldsymbol{M}(k)$.

Step 3.2: Activation operation of SP: Update $\boldsymbol{A}(k+1)$ by activation operation of $\boldsymbol{M}(k+1)$.

Step 4: Termination test: If a stopping condition is satisfied, stop the algorithm; otherwise, $k{:=}k+1$, go to **Step 2**.

The main difference between the flows of SRCMOA and SRCPA is the evaluation in **Step 1** and **Step 2.3**.

In SRCMOA, each antibody $\boldsymbol{b}_{\mathrm{i}} \in \boldsymbol{B}$ (in Step 1, $\boldsymbol{B} = A(0)$; in Step 2.3, $\boldsymbol{B} = \boldsymbol{Z}(k) \cup \boldsymbol{A}(k) \cup \boldsymbol{M}(k)$) is assigned an affinity $F(\boldsymbol{b}_i)$ as follows.

Each individual $\boldsymbol{b}_{\mathrm{i}} \in \boldsymbol{B}$ is assigned a strength value $S(\boldsymbol{b}_i)$, representing the number of individuals it dominates

$$S(\boldsymbol{b}_i) = |\{\boldsymbol{b}_j | j \in \boldsymbol{B} \wedge \boldsymbol{b}_i \succ \boldsymbol{b}_j\}| \tag{16}$$

where the symbol $\succ$ corresponds to the Pareto-dominance relation. Then let

$$R(\boldsymbol{b}_i) = \sum_j (S(\boldsymbol{b}_j) | \boldsymbol{b}_j \in \boldsymbol{B} \wedge \boldsymbol{b}_j \succ \boldsymbol{b}_i). \tag{17}$$

$R(\boldsymbol{b}_i) = 0$ indicates that the individual $\boldsymbol{b}_i$ is not dominated by any other antibodies, i.e. a non-dominated individual, while a high $R(\boldsymbol{b}_i)$ value means that $\boldsymbol{b}_i$ is dominated by many individuals.

For each individual $\boldsymbol{b}_i$ the distances (in objective space) to all individuals $\boldsymbol{b}_{\mathrm{j}} \in \boldsymbol{B}$ are calculated and stored in a list. After sorting the list in increasing order, the sum of two smallest elements gives the distance sought, denoted by $d(p_{\mathrm{i}})$. The density $D(\boldsymbol{b}_i)$ is defined as $D(\boldsymbol{b}_i)\frac{1}{d(p_i)+1}$. Afterwards, the affinity of $\boldsymbol{b}_i$ is defined as

$$F(\boldsymbol{b}_i) = N - R(\boldsymbol{b}_i) - D(\boldsymbol{b}_i). \tag{18}$$

Here, N is added in the affinity to make $F(\boldsymbol{b}_i)$ positive and is maximized in the evaluation. Those individuals with affinity between $[N-1, N]$ are considered to be non-dominated individuals. We adopt binary representation and a constant clonal size in SRCMOA. The hypermutation and receptor editing in CSO are designed as the conventional mutation operation, e.g. bit-inverse mutation.

5. Evaluation of SRCPA's effectiveness

Experiments were carried out to evaluate the performance of SRCPA by comparing with some existing EAs using nine benchmark functions gleaned from the literatures. The test function, parameter domain and global optimum are listed in Table 2. In this study, when the variable dimension is between 10 and 1000, the functions are called high-dimensional functions. When the variable dimension is larger than 1000, the functions are called superhigh-dimensional functions.

Before comparing SRCPA with BGA [28], AEA [29], OGA/Q [30], IEA [31], and IMCPA [32] in the following experiments, we first give a brief description of the five algorithms.

(1) BGA: It is based on an artificial selection similar to that used by human breeders, and is a recombination of evolution strategies and genetic algorithms. BGA uses truncation selection as performed by breeders. This selection scheme is similar to the (μ, λ) strategy in evolution strategies. The search process of BGA is mainly driven by recombination, making BGA a genetic algorithm.

Table 2
Benchmark functions for SRCPA.

Test functions	Parameter domain	Optimum
$f_1(\boldsymbol{x}) = \sum_{i=1}^{D} x_i^2$	$[-100, 100]$	0 (min)
$f_2(\boldsymbol{x}) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	$[-10, 10]$	0 (min)
$f_3(\boldsymbol{x}) = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_j)^2$	$[-100, 100]$	0 (min)
$f_4(x) = \sum_{i=1}^{D} i x_i^4 + \mathrm{random}[0, 1)$	$[-1.28, 1.28]$	0 (min)
$f_5(x) = \frac{1}{D}\sum_{i=1}^{D} (x_i^4 - 16x_i^2 + 5x_i)$	$[-5, 5]$	-78.33236 (min)
$f_6(\boldsymbol{x}) = 10D + \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i))$	$[-5.12, 5.12]$	0 (min)
$f_7(\boldsymbol{x}) = -\sum_{i=1}^{D} x_i \sin(\sqrt{\lvert x_i \rvert})$	$[-500, 500]$	$-412.9829D$ (min)
$f_8(\boldsymbol{x}) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600, 600]$	0 (min)
$f_9(\boldsymbol{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_i^D x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + e$	$[-30, 30]$	0 (min)

(2) AEA: This is a modified version of BGA. Besides the new recombination operator and the mutation operator, each individual of AEA is coded as a vector with components all in the unit interval, and inversion is applied with some probability to the parents before recombination is performed.
(3) OGA/Q: This is a modified version of the classical genetic algorithm (CGA). It is the same as CGA, except that it uses the orthogonal design to generate the initial population and the offspring of the crossover operator.
(4) IEA: This is also an evolutionary algorithm based on the orthogonal design. IEA uses a novel intelligent gene collector (IGC) in recombination. Based on the orthogonal experimental design, IGC uses a divide-and-conquer approach, which includes adaptively dividing two individuals of parents into N pairs of gene segments, economically identifying the potentially better one of two gene segments of each pair, and systematically obtaining a potentially good approximation to the best one of all combinations using at most $2N$ fitness evaluations.
(5) IMCPA: IMCPA is a modified clonal selection algorithm for solving the global single-objective optimization problem. IMCPA runs at two populations, namely antibody population and SP, side-by-side. The main operations on two populations are similar, but the practical handles have some differences, and they have different functions: the antibody population is the basic population acting on antigens and emphasizes the global search, while SP emphasizes the self-adaptive local search in the defining space.

5.1. Performance comparisons on high-dimensional functions

Table 3 shows the statistical results of SRCPA, IMCPA, IEA and OGA/Q, where the reported results of IMCPA, IEA and OGA/Q are obtained from Refs. [3–32]. The population size of SRCPA is 10, the clonal size is 15, the size of SP is 5, and the activation ratio is 50%. The termination criterion of SRCPA is that the given optimal value is reached or the best solution cannot be further improved in successive 30 iterations, which is the same as that in Ref. [31]. Each result of SRCPA is obtained from 50 independent runs. All the test functions can be categorized into three classes by carefully examining the simulation results in Table 3 as follows:

(1) Functions f_1, f_2, f_3, f_6, and f_8: The globally optimal solution exists in the orthogonal array-based initial populations generated using the methods of OGA/Q and IEA, respectively. Therefore, OGA/Q and IEA can obtain the optimal solution with a zero standard deviation. Owing to the termination criterion, the solution quality of SRCPA is worse than that of OGA/Q and IEA, but SRCPA can reduce the number of function evaluations sufficiently. SRCPA performs better than IMCPA in terms of both solution quality and computational cost.
(2) Functions f_4 and f_9: The globally optimal solution exists in the orthogonal array-based initial populations generated using the method of IEA, but not in that of OGA/Q. Therefore, IEA can obtain the optimal solution with a zero standard deviation. The solution quality of SRCPA is worse than that of IEA, but SRCPA performs better than IMCPA and OGA/Q in terms of both solution quality and computational cost.
(3) Functions f_5 and f_7: The globally optimal solution does not exist in either the orthogonal array-based initial populations generated using the method of IEA or that of OGA/Q. SRCPA performs much better than OGA/Q, IEA, and IMCPA, not only in terms of the solution quality but also in terms of computational cost.

Table 3
Performance comparison of SRCPA, IMCPA, IEA and OGA/Q.

f	D	Mean number of function evaluations				Mean function value (standard deviation)			
		SRCPA	IMCPA	IEA	OGA/Q	SRCPA	IMCPA	IEA	OGA/Q
f_1	30	**1,238**	2,173	16,840	112,559	1.592031×10^{-7} (2.674927×10^{-7})	2.038235×10^{-7} (3.170165×10^{-7})	**0** **(0)**	**0** **(0)**
f_2	30	**1,304**	2,729	8,420	112,612	2.224406×10^{-9} (3.472134×10^{-9})	2.443007×10^{-9} (3.669766×10^{-9})	**0** **(0)**	**0** **(0)**
f_3	30	**2,636**	4,196	16,840	112,576	2.206835×10^{-10} (2.862637×10^{-10})	2.575336×10^{-10} (3.175653×10^{-10})	**0** **(0)**	**0** **(0)**
f_4	30	**8,244**	22,402	8,420	112,652	2.576724×10^{-4} (1.740283×10^{-4})	4.381959×10^{-4} (2.897946×10^{-4})	**0** **(0)**	6.301×10^{-3} (4.069×10^{-4})
f_5	100	**2,752**	5,806	184,711	245,930	**−78.33234** **(2.339041×10^{-8})**	−78.33119 (2.902509×10^{-4})	−78.33232 (6.353×10^{-6})	−78.3000296 (6.288×10^{-3})
f_6	30	**1,295**	2,087	8,420	224,710	7.763717×10^{-12} (1.678557×10^{-11})	1.158469×10^{-11} (2.334151×10^{-11})	**0** **(0)**	**0** **(0)**
f_7	30	**2,083**	4,973	54,706	302,166	**−12569.49** **(1.034761×10^{-6})**	−12569.49 (5.946202×10^{-6})	−12569.49 (6.079×10^{-3})	−12569.4537 (6.447×10^{-4})
f_8	30	**1,624**	3,516	16,840	134,000	1.6565×10^{-15} (2.743451×10^{-15})	1.931788×10^{-15} (2.773929×10^{-15})	**0** **(0)**	**0** **(0)**
f_9	30	**1,993**	4,295	8,420	112,421	1.8829×10^{-16} (2.490513×10^{-17})	3.019807×10^{-15} (3.432247×10^{-15})	**0** **(0)**	4.440×10^{-16} (3.989×10^{-17})

OGA/Q and IEA apply the orthogonal design to generate an initial population of points that are scattered uniformly over the feasible solution space, so that the algorithm can evenly scan the search space once to locate good points for further exploration in subsequent iterations. They also use the recombination methods based on the orthogonal design. Therefore, OGA/Q and IEA can obtain an accurate solution, but a large number of function evaluations are needed.

SRCPA reproduces individuals and selects their improved maturated progenies after the affinity maturation process, thus single individuals will be optimized locally and the newcomers yield a broader exploration of the search space. The update operation of SP pays more attention to maintain the population diversity, and the activation operation of SP accelerates the convergence speed. Therefore, SRCPA can obtain a satisfying solution, even though it is not the optimal solution with a much smaller number of function evaluations.

Because the size of the search space and the number of local minima increase with the problem dimension, the higher the dimension is, the more difficult the problem is. Therefore, the following experiment studies the performance of SRCPA on functions with 20–1000 dimensions.

The experiment results of SRCPA, IMCPA, AEA and BGA optimizing the functions f_6, f_7, f_8, and f_9 with various dimensions are shown in Table 4. The termination criterion of SRCPA is that one of the objectives, $|f_{best} - f_{min}| < \varepsilon \cdot |f_{min}|$ or $|f_{best}| < \varepsilon$ if $f_{min} = 0$, is reached. We use $\varepsilon = 10^{-1}$ for f_6, $\varepsilon = 10^{-4}$ for f_7 and $\varepsilon = 10^{-3}$ for f_8 and f_9, which is the same as that in BGA, AEA and IMCPA. Each result of SRCPA is obtained from 30 independent runs. The reported results of IMCPA, AEA, and BGA are obtained from their references.

Table 4
Performance comparisons of SRCPA, IMCPA, AEA and BGA.[a]

Functions	D	Mean number of function evaluations			
		SRCPA	IMCPA	AEA	BGA
f_6	20	**693**	1469	1247	3608
	100	**2453**	4988	4798	25040
	200	**3877**	5747	10370	52948
	400	**6455**	12563	23588	112634
	1000	**11592**	24408	46024	337570
f_7	20	**1197**	3939	1603	16100
	100	**3789**	11896	5106	92000
	200	**6990**	16085	8158	248000
	400	**11722**	26072	13822	699803
	1000	**22375**	60720	23687	/
f_8	20	**1035**	2421	3581	40023
	100	**3181**	6713	17228	307625
	200	**5037**	8460	36760	707855
	400	**8564**	15365	61975	1600920
	1000	**15138**	30906	97600	/
f_9	20	**803**	1776	7040	197420
	100	**2943**	5784	22710	53860
	200	**4859**	9728	43527	107800
	400	**6092**	13915	78216	220820
	1000	**13362**	26787	160940	548306

[a] '/' denotes that relevant data were not found.

As can be seen from Table 4, for all the four functions, the number of function evaluations of SRCPA is much smaller than those of IMCPA, AEA and BGA at all dimensions. Therefore, SRCPA obtains satisfying solutions at a lower computational cost than BGA, AEA and IMCPA, and it displays a good performance in solving large parameter optimization problems.

5.2. Performance comparisons on superhigh-dimensional functions

In order to further test the scalability of SRCPA along the problem dimension further, we use SRCPA and other algorithms to optimize f_6, f_7, f_8 and f_9 with higher dimensions, respectively.

With the same parameters as used in Section 5.1, the problem dimension is increased from 2000 to 30,000. The termination criterion of SRCPA is that one of the objectives, $|f_{best} - f_{min}| < \varepsilon \cdot |f_{min}|$ or $|f_{best}| < \varepsilon$ if $f_{min} = 0$, is reached. We use $\varepsilon = 10^{-1}$ for f_6, $\varepsilon = 10^{-4}$ for f_7 and $\varepsilon = 10^{-3}$ for f_8 and f_9. Table 5 shows the mean number of function evaluations of SRCPA and IMCPA obtained from 30 independent runs. AEA and BGA cannot obtain satisfying solution in 12 h, so we do not list them in the table. When $N > 10{,}000$, IMCPA also cannot attain satisfying solution in 12 h which denoted by "/" in the table. The simulation results show that SRCPA also has a fast convergence speed for superhigh-dimensional functions, which is impossible for some other algorithms.

Further analysis indicates that, under the same condition, the numbers of function evaluations are approximately linear with the problem dimension. The numbers of function evaluations for f_6, f_7, f_8, f_9 versus the problem dimension are shown in Fig. 3. It can be seen that when the variable dimension added one, the mean number of function evaluations increases no more than 13.

5.3. Sensitivity in relation to parameters

In order to study the effects of the main parameters on algorithm performance when running SRCPA, the following experiments consider the problems of f_6, f_7, f_8, f_9 to be solved by SRCPA with various antibody population size, clonal size, secondary pool size, and activation ratio.

5.3.1. Sensitivity in relation to antibody population size and clonal size

The experimental results of SRCPA on optimizing the functions f_6, f_7, f_8, f_9 with the antibody population size n increasing from 5 to 70 and the ratio between clonal size and population size, i.e. N_c/n, called clonal scale, increasing from 1 to 20 are shown in Fig. 4. The values of the other parameters are as follows: the SP size is 5, the activa-

Table 5
Performance comparison of SRCPA and IMCPA on superhigh-dimensional functions.

D	f_6		f_7		f_8		f_9	
	SRCPA	IMCPA	SRCPA	IMCPA	SRCPA	IMCPA	SRCPA	IMCPA
2000	**21346**	37879	**42794**	90001	**23153**	43003	**19480**	41880
5000	**32590**	87245	**64352**	136869	**36088**	125847	**28858**	83125
10000	**55730**	143700	**123600**	235840	**58698**	147037	**80716**	138487
20000	**103546**	/	**196578**	/	**143654**	/	**118078**	/
30000	**151990**	/	**269402**	/	**213096**	/	**145719**	/

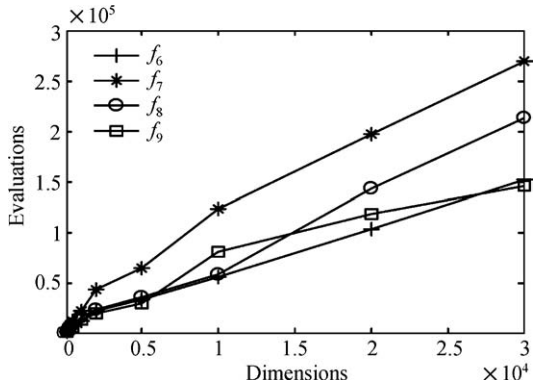

Fig. 3. The mean number of function evaluations versus problem dimensions.

tion ratio is 50%, and the problem dimension is 20. The termination criterion is the same as that in Section 5.2. The data are the statistical results obtained from 50 independent runs.

The results in Fig. 4 show that, antibody population size and clonal size have large effects on the performance of SRCPA. The number of function evaluations increases almost linearly with the increasing antibody population size or clonal scale. After approximating the number of function evaluations by $O(\mu \times n \times N_c)$, we find that when the antibody population size increases by one the average numbers of function evaluations on optimizing functions f_6, f_7, f_8, f_9 increase by about $\mu_{f_6} \approx 43.7067$, $\mu_{f_7} \approx 73.4533$, $\mu_{f_8} \approx 57.96673$, and $\mu_{f_9} \approx 57.96673$, respectively. When the clonal size increases by one, the average numbers of function evaluations on optimizing functions f_6, f_7, f_8, f_9 increase about $\mu_{f_6} \approx 16.3847$, $\mu_{f_7} \approx 21.5700$, $\mu_{f_8} \approx 20.6946$, and $\mu_{f_9} \approx 18.9660$, respectively. Thus, increasing the clonal size will result in a less increment in the number of function evaluations than increasing the antibody population size. In addition, the bigger clonal size also helps to extend the searching scope. However, we also found in experiment that the larger antibody size improves the diversity of population.

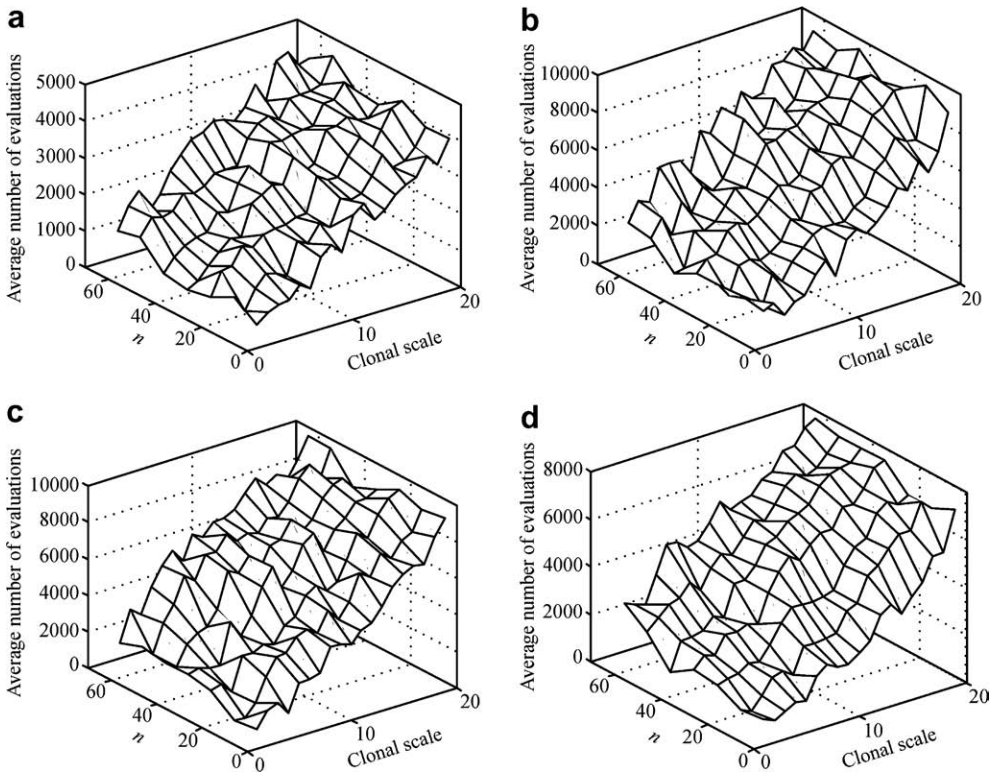

Fig. 4. SRCPA sensitivity in relation to antibody population size and clonal size. (a) f_6; (b) f_7; (c) f_8; (d) f_9.

5.3.2. Sensitivity in relation to SP size and activation ratio

The experimental results of SRCPA on optimizing functions f_6, f_7, f_8, f_9 with the secondary pool size s increasing from 1 to 10 and activation ratio T% increasing from 0 to 1 are shown in Fig. 5. The values of the other parameters are as follows: the antibody population size is 10, the clonal size is 15, and the problem dimension is 20. The termination criterion is the same as that in Section 5.2. The data are the statistical results obtained from 50 independent runs.

From the above results, we conclude that the effect of SP size on the number of function evaluations is weaker than those of antibody population size and clonal size. The larger the SP size is, the fewer the number of function evaluations is. The activation ratio has a rather weak influence on the performance, especially when the variable dimension is low. Generally, when T% is set around 50%, the number of function evaluations is rather few.

6. Evaluation of SRCMOA's effectiveness

In this section, we compare the performance of SRCMOA with NSGA-II, SPEA and PAES through eight well-known multi-objective test functions. In order to demonstrate the workings of these methods, we give both the statistical results of two performance metrics, the convergence metric and the diversity metric. In the last part of this section, we also describe the additional experiments to show the behavior of SRCMOA with different parameter settings.

6.1. Test functions and performance metrics

At first, we describe the eight test functions in this study. Veldhuizen have cited most of these test functions in Ref. [41], and here we choose three of them, i.e. Fonseca's second problem, Poloni's problem and Schaffer's first problem, and call them FON, POL and SCH, respectively. We also choose five test problems which followed the guidelines suggested by Deb [42] and then designed by Zilter, Deb and Thiele in 2000 [43]. We call them ZDT1, ZDT2, ZDT3, ZDT4, ZDT6 here. Table 6 summarizes the numbers of variables, the variable domain, the description of objective functions of these test problems and the comments about their Pareto-optimal fronts.

The goal of the multi-objective optimization algorithms is to find a Pareto-optimal set or approximate it. On one hand, the smaller the distance to the Pareto-optimal set, the better it would be; on the other hand, the more diverse the achieved non-dominated solution, the better it would be. Zitzler et al. [44] 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. Several widely used metrics have been proposed. Since there has not been such a performance metric that can deal with both tasks adequately, we adopt two performance metrics suggested by Deb [34], the convergence metric γ and the diversity metric Δ, which are more direct in evaluating the above two goals in a solution set and make a complement in the whole performance comparison.

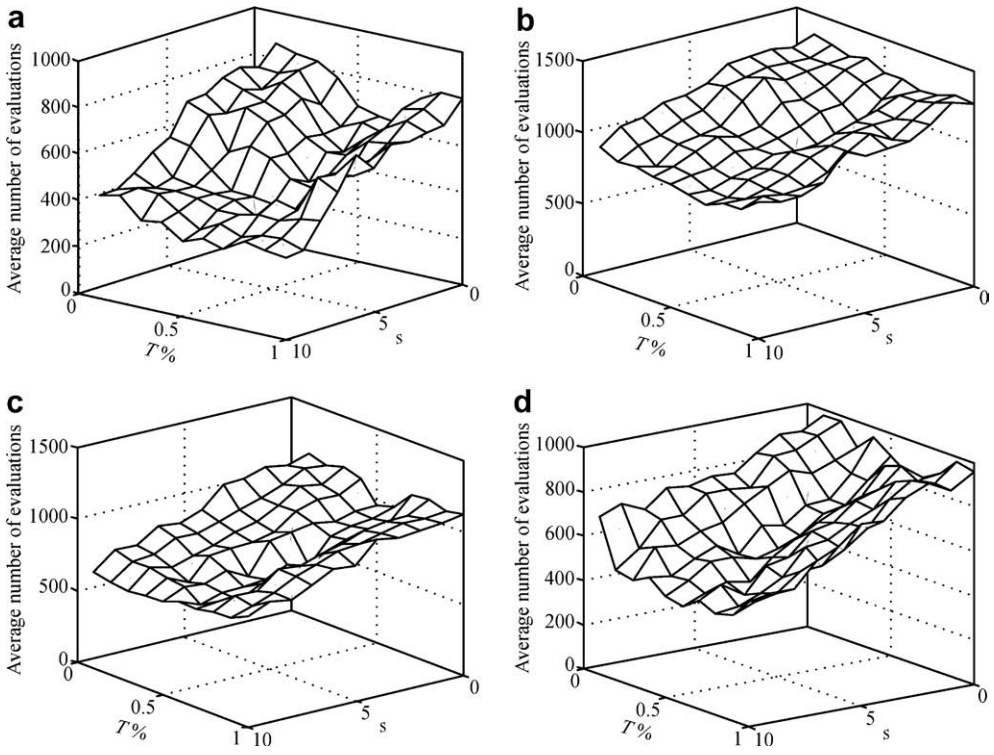

Fig. 5. SRCPA sensitivity in relation to SP size and activation ratio. (a) f_6; (b) f_7; (c) f_8; (d) f_9.

Table 6
Test problems for SRCMOA.

Problem	n	Variable domain	Objective function	Comments
FON	3	$[-4,4]$	$f_1(x) = 1 - \exp\left(-\sum_{i=1}^{n}(x_i - \frac{1}{\sqrt{n}})^2\right)$ $f_2(x) = 1 - \exp\left(-\sum_{i=1}^{n}(x_i + \frac{1}{\sqrt{n}})^2\right)$	Non-convex
POL	2	$[-\pi,\pi]$	$f_1(x) = [1 + (A_1 - B_1)^2 + (A_2 - B_2)^2]$ $f_2(x) = [(x+3)^2 + (y+1)^2]$ $A_1 = 0.5\sin 1 - 2\cos 1 + \sin 2 - 1.5\cos 2,$ $A_2 = 1.5\sin 1 - \cos 1 + 2\sin 2 - 0.5\cos 2,$ $B_1 = 0.5\sin x - 2\cos x + \sin y - 1.5\cos y,$ $B_2 = 1.5\sin x - \cos x + 2\sin y - 0.5\cos y$	Non-convex disconnected
SCH	1	$[-10^3, 10^3]$	$f_1(x) = x^2$ $f_2(x) = (x-2)^2$	Convex
ZDT1	30	$[0,1]$	$f_1(x) = x_1$ $f_2(x) = g(x)\left[1 - \sqrt{x_1/g(x)}\right]$ $g(x) = 1 + 9\left(\sum_{i=2}^{n} x_i\right)\Big/(n-1)$	Convex
ZDT2	30	$[0,1]$	$f_1(x) = x_1$ $f_2(x) = g(x)\left[1 - (x_1/g(x))^2\right]$ $g(x) = 1 + 9\left(\sum_{i=2}^{n} x_i\right)\Big/(n-1)$	Non-convex
ZDT3	30	$[0,1]$	$f_1(x) = x_1$ $f_2(x) = g(x)\left[1 - \sqrt{x_1/g(x)} - \frac{x_1}{g(x)}\sin(10\pi x_i)\right]$ $g(x) = 1 + 9\left(\sum_{i=2}^{n} x_i\right)\Big/(n-1)$	Convex disconnected
ZDT4	10	$[0,1]$	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{x_1/g(x)}]$ $g(x) = 1 + 10(n-1) + \sum_{i=2}^{n}[x_i^2 - 10\cos(4\pi x_i)]$	Non-convex
ZDT6	10	$[0,1]$	$f_1(x) = 1 - \exp(-4x_1)\sin^6(4\pi x_1)$ $f_2(x) = g(x)[1 - (f_1(x)/g(x))^2]$ $g(x) = 1 + 9[(\sum_{i=2}^{n} x_i)/(n-1)]^{0.25}$	Non-convex non-uniformly spaced

6.2. Performance comparisons with NSGA-II, SPEA and PAES

In the following experiments, we performed 10 independent runs for each algorithm on each test function. The termination criterion is to run 250 generations. For NSGA-II, SPEA and PAES, we directly use the results in Ref. [34]. Both binary coded and real-coded NSGA-II are able to converge to the Pareto-optimal front on these eight test problems, but real-coded NSGA-II is able to find a better spread of solutions than binary-coded NGSA-II on most problems. Therefore, we use real-coded NSGA-II as the compared algorithm here. For SRCMOA, we have some common parameter settings the same as the other compared algorithms, including the antibody population size of 100. Each decision variable is also binary-coded with 30 bits. The special parameters concerned with the CSO and SRO in IFCOMA are summarized as follows: the size of SP size is 100, the clonal scale is 3, and the activation ratio is 10%.

The final non-dominated solutions obtained by IFCOMA at the end of 250 generation are reported. Tables 7 and 8 show the direct comparison of SRCMOA with NSGA-II, SPEA and PAES based on the two performance metrics. The mean and the variance of the results are presented for each test problem.

As can be seen from Tables 7 and 8 SRCMOA can converge to the true Pareto-optimal fronts with a good distribution for most test problems except ZDT4, for which SRCMOA does not find a well-converged non-dominated solution set. The mean values of diversity metric obtained by SRCMOA are the best for all the test problems, although the mean values of convergence metric obtained by SRCMOA are not always the best. SRCMOA can find a better convergence in ZDT3, while NSGA-II does better in FON, POL and ZDT4, SPEA in ZDT1 and ZDT2, and PAES in SCH and ZDT6. However, the difference in the mean values of γ between SRCMOA and the algorithms which can find better convergence is small, as well as the variance of γ in the 10 runs. Therefore, Tables 7 and 8 indicate the effective convergence ability in the objective space and the reasonable distribution obtained by SRCMOA for solving most test problems.

6.3. Sensitivity in relation to parameters

In this section, we have some additional experiments based on various secondary pool size, antibody population

Table 7
Statistical values of the convergence metric γ.

Algorithm	FON	POL	SCH	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
SRCMOA								
Mean	0.021221	0.026850	0.001618	0.022175	0.026488	**0.014208**	5.739007	0.263232
Variance	0.000004	0.000002	0	0.000024	0.000048	**0.000047**	5.635990	0.001702
NSGA-II								
Mean	**0.001391**	**0.015553**	0.003391	0.033482	0.072931	0.114500	**0.513053**	0.296564
Variance	**0**	**0.000001**	0	0.004750	0.031689	0.007940	**0.118460**	0.013135
SPEA								
Mean	0.125692	0.037812	0.003403	**0.001799**	**0.001339**	0.047517	7.340299	0.221138
Variance	0.000038	0.000088	0	**0.000010**	**0**	0.000047	6.572516	0.000449
PAES								
Mean	0.151263	0.030864	**0.001313**	0.082085	0.126276	0.023872	0.854816	**0.085469**
Variance	0.000905	0.000431	**0.000003**	0.008679	0.036877	0.000010	0.527238	**0.006664**

Table 8
Statistical values of the diversity metric Δ.

Algorithm	FON	POL	SCH	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
SRCMOA								
Mean	**0.137296**	**0.012475**	**0.146967**	**0.189493**	**0.252527**	**0.126205**	**0.310665**	**0.311588**
Variance	**0.001519**	**0.000001**	**0.000715**	**0.001803**	**0.002000**	**0.000576**	**0.018005**	**0.007565**
NSGA-II								
Mean	0.378065	0.452150	0.477899	0.390307	0.430776	0.738540	0.702612	0.668025
Variance	0.000639	0.002868	0.003471	0.001876	0.004721	0.019706	0.064648	0.009923
SPEA								
Mean	0.792352	0.972783	1.021110	0.784525	0.755148	0.672938	0.798463	0.849389
Variance	0.005546	0.008475	0.004372	0.004440	0.004521	0.003587	0.014616	0.002713
PAES								
Mean	1.162528	1.020007	1.063288	1.229794	1.165942	0.789920	0.870458	1.153052
Variance	0.008945	0	0.002868	0.004839	0.007682	0.001653	0.101399	0.003916

size, clonal scale, and activation ratio. We made these efforts to find the effects of the main parameters on algorithm performance when running SRCMOA, in order to choose a group of more reasonable parameters for SRCMOA.

6.3.1. Sensitivity in relation to SP size and antibody population size

First, we keep other parameters the same as before, but increase the secondary pool size to 150. Table 9 presents the statistical results of 10 independent runs based on the convergence metric and diversity metric for the eight problems. We can see that after increasing the secondary pool size, SRCMOA could convergence to the true Pareto-optimal front better and obtain a better distribution on most test problems.

In Table 9, SRCMOA converges better on the test problems FON, ZDT1, ZDT2 and ZDT3. For POL and SCH, the mean values of γ are similar to the values obtained with the secondary pool size 100. But for ZDT4 and ZDT6, it seems difficult for SRCMOA to have some improvement on the convergence by simply increasing the secondary pool size. Although SRCMOA behaves differently on the above test problems based on the convergence metric γ, the distribution of the non-dominated solutions is improved except ZDT4. So the ability to converge to the true Pareto-optimal front and find a diverse set is not a difficulty for SRCMOA in solving the above test problems except ZDT4 and ZDT6.

Therefore, we do additional experiments on ZDT4 and ZDT6 with a larger antibody population size from 150 to 300 and the secondary pool size of 200. For them, 10 independent runs are performed with the other parameters, the same as those in Section 6.2. Fig. 6 shows the distribution of the convergence metric with different parameter settings. We can see that when the antibody population size and the secondary pool size are increased, the performance of SRCMOA on ZDT4 and ZDT6 is greatly improved. A larger population size may provide more opportunities for the algorithm to find global optimal solutions and a larger secondary pool size may maintain a better diversity, which also helps the algorithm to jump out the local optimal solutions.

6.3.2. Sensitivity in relation to activation ratio and clonal scale

Experiments are designed in this section to find a responsible activation ratio $T\%$ for SRCMOA to have a better performance on most test problems. The activation ratio $T\%$ is assigned from 10% to 50% while keeping other parameters the same as those in Section 6.2. Fig. 7 shows the distribution of the convergence metric on POL and ZDT1.

Fig. 7 shows that we can get the better evaluation values when $T\%$ is set to be 10%. When $T\%$ becomes higher, the values of the convergence metric γ rise too, which indicates that the quality of non-dominated solutions obtained by SRCMOA declines. Since the secondary pool stores the current non-dominated solutions, a too high activation ratio in the activation operation of SP will lead to a low diversity in the antibody population and make SRCMOA easy to drop in the local optimal solutions.

Table 9
Statistical values of the convergence metric γ and diversity metric Δ.

Metric	FON	POL	SCH	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
γ								
Mean	**0.020419**	0.027437	0.001657	0.017363	0.018886	0.008582	6.398918	0.318327
Variance	**0.000004**	0.000003	0	0.000003	0.000017	0.000002	3.003854	0.002116
Δ								
Mean	**0.113090**	0.012107	0.133890	0.191964	0.215481	0.103501	0.484570	0.287365
Variance	**0.000927**	0	0.000777	0.000601	0.001137	0.000104	0.034426	0.009445

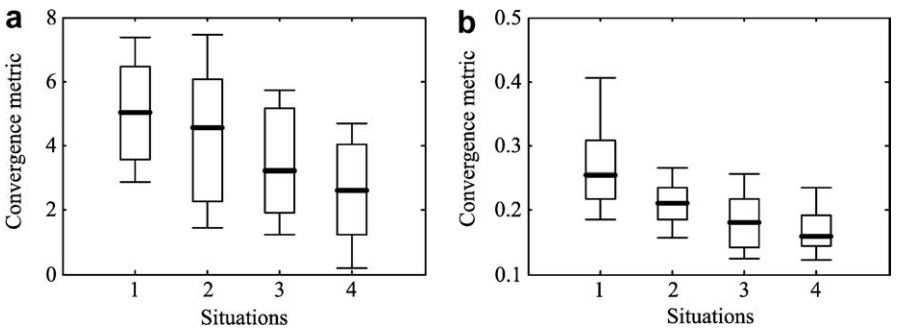

Fig. 6. The distributions of the convergence metric for ZDT4 and ZDT6. (a) ZDT4; (b) ZDT6. The values of horizontal coordinate correspond to four situations, from left to right are (150, 200), (200, 200), (250, 200), and (300, 200), where the first number is the antibody population size and the second one is the secondary pool size.

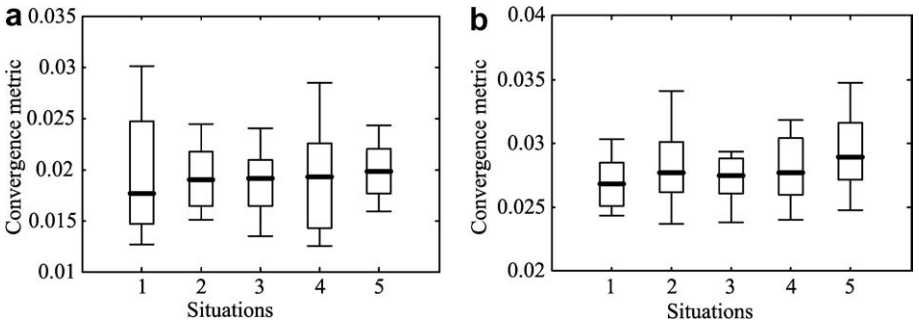

Fig. 7. The distributions of the convergence metric for POL and ZDT1. (a) POL; (b) ZDT1. The values of horizontal coordinate corresponding to different values of activation ratio, from left to right are: 10%, 20%, 30%, 40%, and 50%.

Fig. 8 shows the distribution of the convergence metric of SRCMOA with different clonal scales obtained from 10 independent runs on ZDT1, ZDT2, ZDT3 and ZDT6. Clonal selection operation reproduces antibodies and selects their improved progenies after the affinity maturation process. The higher clonal scale allows a more diverse population around each individual, both the convergence and diversity would be improved obviously. However, the numbers of function evaluation increase with the increasing clonal scale. For a roughly equivalent number of function evaluation with the other algorithms, the clonal scale is set to be 3 in the normal tests.

7. Conclusions

In this study, we have proposed two novel artificial immune system algorithms SRCPA and SRCMOA by using two immune operators CSO and SRO to solve single and multi-objective optimization problems. CSO performs a greedy search, which reproduces individuals and selects their improved maturated progenies after the affinity maturation process, and so single individuals will be optimized locally and the newcomers yield a broader exploration of the search space. SRO copies certain antibodies to the secondary pool whose members do not participate in CSO. Furthermore, the update of the secondary pool pays more attention to maintain the population diversity. SRCPA adopts decimal representation rather than binary representation and uses special mutation and recombination methods designed for large parameter optimization problems. The experimental results indicate that SRCPA converges faster than some existing evolutionary algorithms such as BGA, AEA, OGA/Q, IEA and IMCPA, in terms of solving complex problems such as high-dimensional function optimizations. SRCMOA combines the CSO, SRO and previous MOEAs' techniques to get more various non-dominated solutions. The numerical experiments on the well-known test problems show that SRCMOA has a good

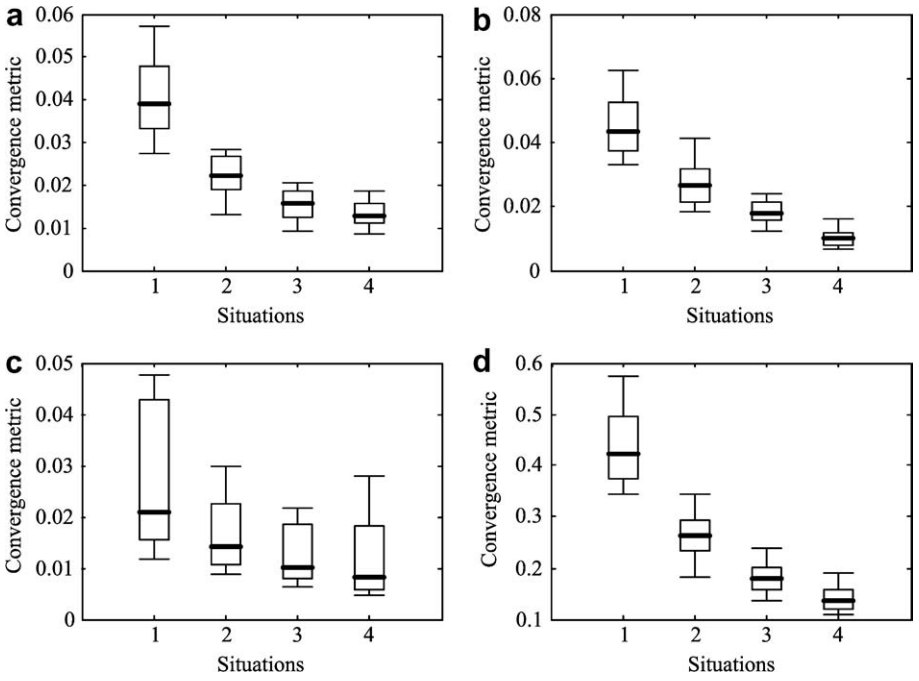

Fig. 8. The distributions of the convergence metric for ZDT1, ZDT2, ZDT3 and ZDT6. (a) ZDT1; (b) ZDT2; (c) ZDT3; (d) ZDT6. The values of horizontal coordinate corresponding to different values of clonal scale, from left to right are 2, 3, 4, and 5.

performance in converging to approximate Pareto-optimal fronts with wide distributions on most of the problems.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant Nos. 60703107 and 60703108), the National High Technology Research and Development Program (863 Program) of China (Grant Nos. 2006AA01Z107 and 2008AA12Z2475853), the National Basic Research Program (973 Program) of China (Grant No. 2006CB705700), and the Program for Cheung Kong Scholars and Innovative Research Team in University (Grant No. IRT0645).

References

[1] Ishida Y. Fully distributed diagnosis by PDP learning algorithm: towards immune network PDP model. In: Proceedings of international joint conference on neural networks, San Diego, 1990:777–82.
[2] Forrest S, Perelson AS, Allen L, et al. Self-nonself discrimination in a computer. In: Proceedings of the IEEE symposium on research in security and privacy. Los Alamitos, CA: IEEE Computer Society Press, 1994:202–12.
[3] Dasgupta D. Artificial immune systems and their applications. Berlin, Heidelberg, New York: Springer-Verlag; 1999.
[4] Gonzalez F, Dasgupta D, Kozma R. Combining negative selection and classification techniques for anomaly detection. In: Proceedings of the special sessions on artificial immune systems in congress on evolutionary computation, 2002 IEEE world congress on computational intelligence. Honolulu, Hawaii, May 2002.
[5] Carter JH. The immune system as a model for pattern recognition and classification. J Am Med Inform Assoc 2000;7(3):28–41.
[6] de Castro LN, Von Zuben FJ. The clonal selection algorithm with engineering applications. In: Proceedings of GECCO'00, workshop on artificial immune systems and their applications, 2000:36–7.
[7] Tarakanov A, Skormin V. Pattern recognition by immunocomputing. In: Proceedings of the special sessions on artificial immune systems. In: Congress on evolutionary computation, 2002 IEEE world congress on computational intelligence. Honolulu, Hawaii, May 2002.
[8] García-Pedrajas N, Fyfe C. Immune network based ensembles. Neurocomputing 2007;70(7–9):1155–66.
[9] Hart E, Ross P. The evolution and analysis of a potential antibody library for use in job-shop scheduling. New ideas in optimization. Columbus: McGraw-Hill; 1999, p. 185–202.
[10] Coello Coello CA, Rivera DC, Cortes NC. Use of an artificial immune system for job shop scheduling. In: Proceedings of the second international conference on artificial immune systems, September 1–3, 2003. Napier University, Edinburgh, UK, 2003.
[11] Jiao LC, Wang L. A novel genetic algorithm based on immunity. IEEE Trans Syst Man Cybernet – Part A 2000;30(5):552–61.
[12] de Castro LN, Von Zuben FJ. Learning and optimization using the clonal selection principle. IEEE Trans Evol Comput 2002;6(3):239–51.
[13] Gong MG, Jiao LC, Du HF, et al. Multi-objective immune algorithm with pareto-optimal neighbor-based selection. Evol Comput 2008;16(2):225–55.
[14] Hajela P, Yoo J, Lee J. GA based simulation of immune networks applications in structural optimization. Eng Optimization 1997;29(1–4):131–49.
[15] Gong MG, Du HF, Jiao LC. Optimal approximation of linear systems by artificial immune response. Sci Chin: Ser F Inform Sci 2006;49(1):63–79.
[16] de Castro LN, Timmis JI. Artificial immune systems: A new computational intelligence approach. London: Springer-Verlag; 2002.
[17] Garrett SM. How do we evaluate artificial immune systems. Evol Comput 2005;13(2):145–78.
[18] Burnet FM. The clonal selection theory of acquired immunity. Cambridge: Cambridge University Press; 1959.
[19] de Castro LN, Timmis J. An artificial immune network for multimodal function optimization. In: Proceedings of the 2002 IEEE congress on evolutionary computation, 2002:699–704.
[20] Kelsey J, Timmis J. Immune inspired somatic contiguous hypermutation for function optimisation. In: Proceedings of the genetic and evolutionary computation conference, 2003:207–18.
[21] Garrett SM. Parameter-free, adaptive clonal selection. In: Proceedings of the 2004 IEEE congress on evolutionary computing. Portland, OR, 2004:1052–8.
[22] Cutello V, Nicosia G, Pavone M. Exploring the capability of immune algorithms: a characterization of hypermutation operators. In: Proceedings of the third international conference on artificial immune systems. Catania, Italy, September 13–16, 2004, Lecture Notes in Computer Science 2004;3239:263–76.
[23] Cutello V, Narzisi G, Nicosia G, et al. Clonal selection algorithms: a comparative case study using effective mutation potentials. In: Proceedings of the Fourth International Conference on Artificial Immune Systems, Banff, Canada, August 14–17, 2005, Lecture Notes in Computer Science 2005;3627:13–28.
[24] Coello Coello CA, Cruz Cortss N. An approach to solve multiobjective optimization problems based on an artificial immune system. In: Proceedings of the First International Conference on Artificial Immune Systems, University of Kent at Canterbury, UK, September 9–11, 2002, 212–21.
[25] Coello Coello CA, Cruz Cortss N. Solving multi-objective optimization problems using an artificial immune system. Genet Prog Evolvable Mach 2005;6:163–90.
[26] Freschi F, Repetto M. Multi-objective optimization by a modified artificial immune system algorithm. In: Proceedings of the fourth international conference on artificial immune systems, Banff, Canada. August 14–17, 2005, Lecture Notes in Computer Science 2005;3627:248–61.
[27] Cutello V, Narzisi G, Nicosia G. A class of pareto archived evolution strategy algorithms using immune inspired operators for Ab-Initio protein structure prediction. In: Proceedings of the third european workshop on evolutionary computation and bioinformatics, Evo-Workshops 2005, Lausanne, Switzerland, 2005. Lecture Notes in Computer Science 2005;3449:54–63.
[28] Mühlenbein H, Dirk S. Predictive models for the breeder genetic algorithm. Evol Comput 1993;1(1):25–49.
[29] Pan ZJ, Kang LS. An adaptive evolutionary algorithms for numerical optimization. In: Simulated evolution and earning. Berlin: Springer-Verlag, 1997:27–34.
[30] Leung YW, Wang YP. An orthogonal genetic algorithm with quantization for global numerical optimization. IEEE Trans Evol Comput 2001;5(1):41–53.
[31] Ho SY, Shu LS, Chen JH. Intelligent evolutionary algorithms for large parameter optimization problems. IEEE Trans Evol Comput 2004;8(6):522–40.
[32] Du HF, Gong MG, Jiao LC, et al. A novel artificial immune system algorithm for high-dimensional function numerical optimization. Prog Nat Sci 2005;15(5):463–71.
[33] Zitzler E, Laumanns M, Thiele L. SPEA2: improving the strength pareto evolutionary algorithm. In: Proceedings of the evolutionary methods for design, optimization and control with applications to industrial problems. Athens, Greece, 2002:95–100.
[34] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 2002;6(2):182–97.
[35] Zitzler E, Thiele L. Multi-objective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Transactions on Evolutionary Computation 1999;3(4):257–71.

[36] Knowles J, Corne D. The pareto archived evolution strategy: a new baseline algorithm for multi-objective optimization. In: Processing of the 1999 congress on evolutionary computation. Piscataway, NJ: IEEE Service Center 1999:397–414.
[37] Akdis CA, Blaser K, Akdis M. Genes of tolerance. Eur J Allergy Clin Immunol 2004;59(9):897–913.
[38] Berek C, Ziegner M. The maturation of the immune response. Immunol Today 1993;14(8):400–2.
[39] George AJT, Gray D. Receptor editing during affinity maturation. Immunol Today 1999;20(4):196.
[40] Beyer HG, Schwefel HP. Evolution strategies: a comprehensive introduction. Nat Comput 2002;1(1):3–52.
[41] Van Veldhuizen DA. Multi-objective evolutionary algorithms: classification, analyses, and new innovations. PhD thesis, Air University, 1999.
[42] Deb K. Multi-objective genetic algorithms: problem difficulties and construction of test problems. Evol Comput 1999;7(3):205–30.
[43] Zitzler E, Deb K, Thiele L. Comparison of multi-objective evolutionary algorithms: empirical results. Evol Comput 2000;8(2):173–95.
[44] Zitzler E, Thiele L, Laumanns M, et al. Performance assessment of multi-objective optimizers: an analysis and review. IEEE Trans Evol Comput 2003;7(2):117–32.