

A Hybrid Evolutionary Immune Algorithm for Multiobjective Optimization Problems

Qiuzhen Lin, Jianyong Chen, *Member, IEEE*, Zhi-Hui Zhan, *Member, IEEE*, Wei-neng Chen, *Member, IEEE*, Carlos A. Coello Coello, *Fellow, IEEE*, Yilong Yin, Chih-Min Lin, *Fellow, IEEE* and Jun Zhang, *Senior Member, IEEE*

Abstract— In recent years, multi-objective immune algorithms (MOIAs) have shown promising performance in solving multi-objective optimization problems (MOPs). However, basic MOIAs only use a single hypermutation operation to evolve individuals, which may induce some difficulties in tackling complicated MOPs. In this paper, we propose a novel hybrid evolutionary framework for MOIAs, where the cloned individuals are divided into several subpopulations and then evolved using different evolutionary strategies. An example of this hybrid framework is implemented, where simulated binary crossover and differential evolution with polynomial mutation are adopted. A fine-grained selection mechanism and a novel elitism sharing strategy are also adopted for performance enhancement. Various comparative experiments are conducted on 28 test MOPs and our empirical results validate the effectiveness and competitiveness of our proposed algorithm in solving MOPs of different types.

Index Terms—Artificial immune system, multiobjective optimization problems, hybrid evolution, elitism strategy.

I. INTRODUCTION

ANY scientific and engineering applications give rise to problems that require the simultaneous optimization of several (often conflicting) objectives. They are called multiobjective optimization problems (MOPs), and their solu-

tion consists not of a single solution, but of a set of them, representing the trade-offs among the objectives. Such solutions conform the so-called Pareto-optimal set (*PS*). The corresponding objective vectors of *PS* are called the Pareto-optimal front (*PF*) [1].

During the last two decades, evolutionary algorithms (EAs) have been widely used to solve MOPs, because of their generality (they require little specific domain information) and because of their population-based search nature, which allows them to produce multiple Pareto-optimal solutions in a single run [2]. The best-known state-of-the-art multiobjective EAs (MOEAs) include NSGA-II [2], SPEA2 [3], and MOEA/D [4]. NSGA-II adopts a fast nondominated sorting approach combined with a crowded-comparison operator and an elitist strategy. This nondominated sorting approach is recently improved by using more efficient approaches [5]-[6]. SPEA2 uses a fine-grained fitness assignment strategy and a density estimation technique based on clustering. MOEA/D decomposes MOPs into a set of single-objective optimization subproblems and then optimizes them simultaneously. In recent years, many new approaches have been proposed to speed up convergence and to improve diversity of MOEAs [7]-[17]. Regarding convergence speed up, Tan *et al.* [7]-[8] proposed an adaptive mutation operator and an enhanced exploration strategy, Adra *et al.* [9] developed a convergence acceleration operator, and Yu *et al.* [10] presented a trade-off approach to switch between coarse local search and fine local search. Regarding the enhancement of population diversity, Ishibuchi *et al.* [11] proposed a non-geometric binary crossover, Zhan *et al.* [12] presented a new optimization framework for MOPs based on the use of multiple populations, Li *et al.* [13] reported a stable matching model for the individuals and subproblems in MOEA/D, and Gee *et al.* [14] designed an online diversity metric to measure the diversity loss caused by any individual in the population. Additionally, other nature-inspired heuristic algorithms have also been modified to solve MOPs, including coevolutionary algorithms [18], [19], scatter search [20], particle swarm optimization (PSO) [21], [22] and artificial immune systems [23]-[25].

Compared with MOEAs, multiobjective immune algorithms (MOIAs) have shown some advantages related to an improved convergence speed and maintaining a better population diversity [23]. Specifically, MOIAs are characterized by the clonal selection principle, in which only a small proportion of indi-

Manuscript received XX. XX. XXXX; revised XX. XX. XXXX; accepted XX. XX. 2015. This work was supported by the National Nature Science Foundation of China under Grant 61402291, 61402545 and 61379061, the National High-Techonology Research and Development Program (“863” Program) of China under Grant 2013AA01A212, NSFC Joint Fund with Guangdong under Key Project No. U1201258 and the CONACyT Grant with No. 221551.

Q. Z. Lin and J. Y. Chen are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail of Jianyong Chen: jyachen@szu.edu.cn).

Z. H. Zhan, W. N. Chen, and J. Zhang are with the Key Lab. of Machine Intelligence and Advanced Computing, Ministry of Education, Guangzhou 510006, China, and also with the Key Lab. of Software Technology, Education Department of Guangdong Province, Guangzhou 510006, China.

C. A. Coello Coello is with the Department of Computer Science, CINVESTAV-IPN (Evolutionary Computation Group), México, D.F. 07300, MÉXICO.

Y. L. Yin is with the School of Computer Science and Technology, Shandong University, Jinan 250100, China.

C. M. Lin is with the College of Electrical and Communication Engineering, Yuan Ze University, Chung-Li, Taiwan.

Copyright (c) 2012 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org

viduals with better convergence and diversity are proliferated to produce multiple clones [23]-[25]. Then, each clone is evolved by hypermutation to search for the better individuals. In this way, superior individuals have more opportunities for evolving, which helps to speed up convergence. However, most MOIAs only adopt a single hypermutation operator to evolve each clone [23]-[25], which may induce some difficulties in solving complicated MOPs (e.g., complicated unconstrained MOPs such as the Unconstrained Functions (UF) in [26]). This may be due to the fact that a single hypermutation operator will have difficulties to achieve a proper balance between proximity and diversity. This conforms to the no-free-lunch theorem which states that a single search strategy cannot have the best performance in terms of both proximity and diversity [27]. This motivated us to study whether multiple search strategies can be combined in MOIAs so that their search patterns can complement each other. Therefore, in this study, we propose a novel Hybrid Evolutionary framework for MOIAs (named HEIA), in which the cooperation of multiple evolutionary strategies allows us to combine their advantages and overcome the inherent limitations of adopting a single strategy. Our proposed approach is more capable of maintaining the trade-off between proximity and diversity, and, consequently, has a better performance than other multiobjective optimization algorithms when dealing with different types of MOPs. To validate the effectiveness of our hybrid framework, an implementation example is proposed, which considers simulated binary crossover (SBX) [28] and differential evolution (DE) [29] followed by polynomial-based mutation as two evolutionary strategies. The main features of our proposed HEIA are the following:

(1) The cloned population is randomly divided into several subpopulations, which will be subject to different evolutionary strategies, separately. This hybrid framework with multiple evolutionary strategies is different from the traditional MOIAs that only apply one hypermutation operator on the cloned population. The cooperation of multiple evolutionary strategies in our scheme can overcome the inherent limitations of using a single strategy, and enhances the exploratory capability and robustness of our proposed approach, thus allowing it to handle a wide variety of MOPs.

(2) Two evolutionary strategies are included in HEIA. The first one adopts SBX followed by polynomial-based mutation. This strategy is found to perform well on simple MOPs with independent decision variables. The second strategy applies DE crossover and polynomial-based mutation, and is especially effective for complicated problems with variable linkages (i.e., parameter dependencies) [29]. These two evolutionary strategies can complement each other by exploiting their advantages and it is experimentally shown that they perform better than the separate use of any of them.

(3) A fine-grained selection mechanism and a novel elitist sharing strategy are adopted in HEIA. After the hybridized evolution from item (2) shown above is undertaken, the subpopulations are combined with an elitist archive and a fine-grained selection mechanism proposed in our previous work [25], with the aim of preserving the nondominated individuals found in the elitist archive. An elitist sharing strategy is then

operated by selecting some less-crowded individuals (measured by the crowding distance metric [2]) from the elitist archive. Such individuals are cloned and divided into different subpopulations for independent evolution in the next generation. In this way, the search experience of one subpopulation can be shared by another subpopulation.

The improvements of the proposed hybrid framework are also validated with an experimental study. To have a comprehensive evaluation of the performance of our proposed HEIA, four different test suites were adopted. Such problems are the Zitzler-Deb-Thiele (ZDT) test suite [30], the Walking Fish Group (WFG) test suite [31], the Deb-Thiele-Laumanns-Zitzler (DTLZ) test suite [32] and the UF test suite adopted at the competition held at the *2009 IEEE Congress on Evolutionary Computation (CEC'2009)* [26]. When compared with various nature-inspired multiobjective algorithms, i.e., two state-of-the-art MOEAs (NSGA-II [2] and SPEA2 [3]), an Archive-based hYbrid Scatter Search algorithm (AbYSS) [20], MOEA/D [29], a novel PSO-based multiobjective algorithm (SMPSO) [33] and three MOIAs (NICA [23], NNIA [24], and MIMO [25]), HEIA has been found to be advantageous in terms of convergence speed and population diversity. The effectiveness of the proposed hybrid evolutionary strategy and the influence of the parameters settings on its performance are also studied experimentally, which further confirm the improvements yielded by HEIA.

The remainder of this paper is organized as follows. In Section II, the related work of MOPs and MOIAs is provided. In Section III, the proposed hybrid framework and its implementation example are described in detail. Section IV presents the experimental results of HEIA, when compared to other multiobjective metaheuristics. Our conclusions and some possible paths for future work are described in Section V.

II. RELATED WORK

A. Multiobjective Optimization Problems

Multiobjective optimization problems naturally arise in many practical applications, which are aimed at optimizing multiple, possibly conflicting objectives, simultaneously. Without loss of generality, we formulate the general multiobjective problem as follows.

$$\text{Min}_{\mathbf{x} \in \Omega} F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x}))^T \quad (1)$$

where $\mathbf{x} = (x_1, x_2, \dots, x_n)$ is an n -dimensional decision vector bounded in the *decision space* Ω , the mapping function $F: \Omega \rightarrow R^m$ defines m objective functions and R^m is called the *objective space*. However, since the objectives may be in conflict with each other, the optimization of one objective can result in the deterioration of another objective. Normally, no single solution can optimize all the objectives simultaneously. The best trade-offs among the objectives can be properly attained by using the definition of Pareto optimality [27].

Definition 1 (Pareto-dominance): A decision vector \mathbf{x} is said to dominate another decision vector \mathbf{y} (noted as $\mathbf{x} > \mathbf{y}$) if and only if

$$\begin{aligned}
& (\forall i \in \{1, 2, \dots, m\} : f_i(\mathbf{x}) \leq f_i(\mathbf{y})) \\
& \wedge (\exists j \in \{1, 2, \dots, m\} : f_j(\mathbf{x}) < f_j(\mathbf{y})). \quad (2)
\end{aligned}$$

Definition 2 (Pareto-optimal): A solution \mathbf{x} is said to be Pareto-optimal if and only if

$$\neg \exists \mathbf{y} \in \Omega : \mathbf{y} > \mathbf{x}. \quad (3)$$

Definition 3 (Pareto-optimal set): the set \mathbf{PS} includes all Pareto-optimal solutions, defined by

$$\mathbf{PS} = \{\mathbf{x} \mid \neg \exists \mathbf{y} \in \Omega : \mathbf{y} > \mathbf{x}\}. \quad (4)$$

Definition 4 (Pareto-optimal front): The set \mathbf{PF} includes the values of all the objective functions corresponding to the Pareto-optimal solutions in \mathbf{PS} .

$$\mathbf{PF} = \{F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x}))^T \mid \mathbf{x} \in \mathbf{PS}\}. \quad (5)$$

In this paper, the best solutions (possibly sub-optimal) produced by an algorithm can also be treated as a \mathbf{PF} , when the true \mathbf{PF} defined in (5) is unavailable in practical cases. To distinguish these two types of \mathbf{PF} , \mathbf{PF}_{true} is used to refer to the true (or optimal) \mathbf{PF} as defined in (5), while \mathbf{PF}_{known} is employed to represent the best solutions produced by an algorithm.

B. Immunology Terms in MOIAs

As MOIAs are designed based on the principles and processes of biological immune systems [34]-[35], some immunological terms used in MOIAs are introduced next in order to better understand them.

Definition 5 (Antigen): An antigen refers to the problem and constraints to be solved, *e.g.*, the minimization problem $F(\mathbf{x})$ in (1).

Definition 6 (Antibody): An antibody refers to a candidate solution of the target problem, *e.g.*, the decision variable \mathbf{x} in (1).

Definition 7 (Affinity): Affinity usually represents the objective function values or the fitness measurement of the problem related to a candidate solution.

To mimic the clonal selection principle, MOIAs usually apply the cloning operator, which generates copies of the antibodies that are selected due to their better affinities. After that, hypermutation (which refers to applying a mutation operator at a high rate), is applied on each clone to alter the decision variables. This process is aimed at searching the antibodies with better affinities and achieving affinity maturation (*i.e.*, individuals that represent better solutions). The set of antibodies is called an antibody population. Using the definition of Pareto-dominance, an antibody is called a nondominated antibody when it is not dominated by any other antibodies in the population.

C. Related Work on MOIAs

The first MOIA was reported in [36], and incorporated the concept of antibody-antigen affinity into a standard genetic

algorithm to modify its fitness assignment mechanism. Afterwards, many other MOIAs were designed, most of which have a superior performance. Based on the special features inspired by the immune system, MOIAs can be classified into three categories: clonal selection approaches, immune network approaches and hybrid approaches (*i.e.*, combinations of an immune system with another heuristic).

A multiobjective immune system algorithm (MISA) based on clonal selection was proposed in [37]. In this approach, only the antibodies with high affinities are proliferated to generate multiple clones, and an adaptive grid is used to maintain diversity. The performance of MISA was further improved in [38]-[40]. An immune dominance clonal multiobjective algorithm (IDCMA) was introduced in [41]. This approach adopts the antibody-antibody affinity to reflect the similarity among antibodies. This guides the cloning operator to select an effective search region (*i.e.*, the least-crowded region). This approach was further extended to solve dynamic MOPs [42] and it was improved in [24]. A novel MOIA using a multiple-affinity model was presented in [43]. This approach adopts six measures for affinity assignment, where cloning, hypermutation and immune suppression were applied according to such affinity measurements. In this case, immune suppression refers to removing similar antibodies in both variable and objective space.

On the immune network approach, a vector artificial immune system (VAIS) [44], [45] was extended from the artificial immune network algorithm (opt-aiNet) [46] to solve MOPs. In this case, two evolutionary loops are performed. The inner loop is aimed at exploiting the search space while the outer one is aimed to avoid the redundancy produced by similar antibodies (suppression is used to avoid such redundancy). A novel weight-based MOIA was presented in [47]. This approach adopts a random-weighted sum method as its fitness assignment scheme, combined with a new truncation algorithm that eliminates similar individuals. Its authors claimed that this approach has a low computational complexity and is able to obtain a well-distributed \mathbf{PF}_{known} .

Regarding hybrid approaches, one called immune-inspired Pareto archived evolution strategy was introduced in [48]. In this approach, two hypermutation operators are combined to solve a MOP extracted from *ab initio* protein structure prediction problems. An evolutionary artificial immune algorithm designed for solving MOPs was reported in [49]. This approach combines the global search capabilities of EAs with the learning capabilities of artificial immune systems. A novel immunity-based hybrid evolutionary algorithm was proposed in [50] to tackle both unconstrained and constrained MOPs. This approach uses a sorting scheme featuring uniform crossover, multi-point mutation, and crowding distance sorting, to efficiently approximate \mathbf{PF}_{true} .

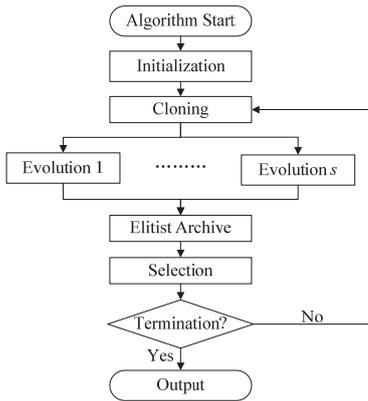


Fig. 1 The proposed framework of HEIA

In recent years, several new MOIAs with competitive performance have been proposed. For example, a dynamic MOIA was introduced in [51] for constrained nonlinear MOPs and it was extended for its use in greenhouse control [52]. A hybrid immune multiobjective algorithm (HIMO) was proposed in [53]. This approach uses a hybrid operator combining Gaussian and polynomial-based mutation. It was further improved by using an adaptive mutation operator [25] and a novel adaptive DE operator [54] with a fine-grained selection mechanism. A novel immune clonal algorithm (NICA) [23] was reported to solve complex MOPs. This approach adopts a full cloning scheme and a novel antibody population updating operation after clonal selection.

However, in the above-mentioned MOIAs, most of them only adopt a simple hypermutation operator to evolve the antibodies, *e.g.*, [23], [47], [51], [52] only use a hypermutation operator while [24], [25], [43], [53] utilize a combination of one crossover operator with a mutation operator. The use of simple evolutionary operators in MOIAs may lead to a monotonous search pattern, which makes the existing MOIAs incapable of tackling complicated MOPs (*e.g.*, the UF test problems [26]). Actually, hybrid mutation approaches have been investigated in immune algorithms [48], [53], [55], [56], with promising results. Working in this same direction, this paper presents a general hybrid evolutionary framework for MOIAs, which adopts different search strategies to complement each other, thus providing superior performance. The cooperation of hybrid evolutionary strategies can overcome the inherent limitations of using a single strategy, and is expected to be better and more robust when solving different types of MOPs. To the best of our knowledge, this is the first attempt to construct a hybrid evolutionary framework for MOIAs. For MOEAs, some hybrid frameworks have been proposed, *e.g.*, Sindhya *et al.* [57] introduced a hybrid framework for MOEAs, which uses a local search module to speed up convergence. Tang and Wang [58] proposed a novel hybrid MOEA, which incorporates the concepts of personal best and global best in particle swarm optimization and multiple crossover operators. The differences between HEIA and existing hybrid MOEAs are the following. One natural difference is that cloning is performed in HEIA, which generates multiple copies of the high-affinity antibodies to be evolved, while existing hybrid MOEAs generally evolve

Algorithm 1: Initialization

```

1 set  $g = 0$ ; //generation number
2 for  $i = 1$  to  $N$ 
3   for  $j = 1$  to  $n$ 
4      $x_{ij} = l_i + rand() \times (u_i - l_i)$ ; //initialize each variable of  $x_i$ 
5   end for
6   evaluate the objective functions;
7 end for
8 add the nondominated antibodies to the elitist archive  $E$ ;
9 calculate the crowding distance for each antibody in  $E$ ;

```

the entire population. The other one is that the proposed hybrid operators in HEIA are modified for MOIAs, *e.g.*, SBX and DE crossover are applied by selecting parent antibodies only from the chosen high-affinity antibodies, while the evolutionary operators in existing hybrid MOEAs usually select parents from the entire population. To describe our proposed hybrid framework in a better way, an implementation example is also included in this study and some experiments are carried out to assess its performance.

III. THE PROPOSED FRAMEWORK AND IMPLEMENTATION

The framework of HEIA is shown in Fig. 1, where s is the number of subpopulations. HEIA starts by initializing the population and by setting some relevant parameters. After that, cloning is performed to get multiple copies of the selected individuals (those with high affinity values). The clones are then randomly divided into the equal-size subpopulations, which are independently evolved using different evolutionary strategies in order to increase their affinities. The multiple evolutionary strategies can search various directions in solution space and avoid the inherent limitation of using a single strategy. Finally, an elitist archive is used to collect all the non-dominated antibodies found by the subpopulations, which enables each subpopulation to share their search results in the next iteration. Any effective evolutionary operators designed for MOPs can be used in this hybrid framework.

According to the framework of HEIA shown in Fig. 1, the antibodies will undergo four important procedures, *i.e.*, cloning, evolution, elitist archive and selection, to approximate \mathbf{PF}_{true} . In order to validate the effectiveness of the proposed framework, an implementation example is given here by using two groups of well-known evolutionary operators. One is SBX followed by polynomial-based mutation, which is adopted in some of MOEAs and MOIAs using the real-value variables [2], [3], [24], [25], [43]. The other one is DE crossover plus polynomial-based mutation, which is especially effective for solving some complicated MOPs with variable linkages [29], [59], [60]. At first, the pseudo-code of initialization is given in **Algorithm 1**, where N is the population size, n is the number of decision variables in each solution, u_i and l_i are respectively the lower and upper bounds of the i -th decision variable, $rand()$ returns a uniformly distributed random number in $[0, 1]$. All the nondominated antibodies in the initial population are added to the elitist archive E and their crowding distances [2] are computed. The implementation details of the other main procedures are described below.

A. Cloning

In biological immune systems, cloning refers to an asexual propagation mechanism, which generates a group of identical cells from a single common ancestor [24]. In MOIAs, cloning is simulated by cloning the high-affinity antibodies to perform hypermutation. In this study, it is noted that the affinity value of an antibody is assigned as the crowding-distance value [2], so that high-affinity antibodies are those located in less-crowded regions of the search space. Analogously to the cloning operators reported in [24], [25], [53], in this paper, only a small proportion of high-affinity antibodies from the elitist archive are cloned. This aims to promote the exploration of less-crowded regions of the search space in order to attain a better distribution of solutions along the current $\mathbf{PF}_{\text{known}}$. Let us assume that the antibodies with higher affinities that have been selected for cloning are denoted by $A = [a_1, a_2, \dots, a_{NA}]$, where NA is the number of selected individuals. The cloning operator T_C can be formulated as follows.

$$T_C(A) = [T_C(a_1), T_C(a_2), \dots, T_C(a_{NA})] \quad (6)$$

where $T_C(a_i) = q_i \times a_i$ ($i = 1, 2, \dots, NA$) is to reproduce q_i copies of a_i . Here, the value of q_i is set as follows.

$$q_i = \left\lceil N \times \frac{\text{fit}(a_i)}{\sum_{j=1}^{NA} \text{fit}(a_j)} \right\rceil \quad (7)$$

where $\text{fit}(a_i)$ is the affinity value of antibody a_i and N is the population size. The affinity value of antibody a_i can be obtained as follows.

$$\text{fit}(a_i) = \sum_{j=1}^m \frac{\text{fit}_j(a_i)}{f_j \max - f_j \min} \quad (8)$$

where $f_j \max$ and $f_j \min$ are, respectively, the maximum and minimum values of the j -th objective obtained by the current population, and

$$\text{fit}_j(a_i) = \begin{cases} \infty, & \text{if } (f_j(a_i) = f_j \min) \text{ or } (f_j(a_i) = f_j \max); \\ f'_j(a'_{I(i)+1}) - f'_j(a'_{I(i)-1}), & \text{otherwise;} \end{cases} \quad (9)$$

where $f'_j(a'_{I(i)})$ is to sort the antibodies according to the j -th objective in descending order and $I(i)$ is the new index of a_i after sorting. As the affinity values of boundary antibodies are set to ∞ in (9), it is not applicable to get the clone number in (7). In this case, it is set as the double of the maximum affinity value except for the boundary antibodies [24], [25], [53].

B. Evolutionary Strategies

In our proposed framework, the clones are randomly divided into multiple subpopulations following a uniformly random distribution. Then, multiple subpopulations are evolved separately by using multiple evolutionary strategies. The use of multiple evolutionary strategies in the hybrid framework mitigates the risk of using a single strategy, which may be inappropriate or ineffective for a particular problem. This phenomenon is validated in the experimental studies described in Section IV.F. This hybrid strategy enhances the global search capability of HEIA as well as its robustness when solving different types of complicated MOPs. Here, two groups of

evolutionary operators are used to illustrate the behavior of our hybrid framework.

1. Evolution 1

In this phase, simulated binary crossover (SBX) followed by polynomial-based mutation are adopted. SBX is an important recombination operator adopted in several MOEAs and MOIAs using real-valued variables [2], [3], [24], [25], [43]. Let us assume that the antibody population after cloning is denoted by $C = [c_1, c_2, \dots, c_N]$, where N is the population size. For each individual $\mathbf{x} = (x_1, x_2, \dots, x_n)$ in C , the other parent $\mathbf{y} = (y_1, y_2, \dots, y_n)$ is randomly selected from A (these are the antibodies selected from the elitist archive for cloning). Let w_i and v_i ($i = 1, 2, \dots, n$) be $\max(x_i, y_i)$ and $\min(x_i, y_i)$, respectively. The SBX operator was originally defined in [28]. Here, a few modifications were made for improving its performance [61]. It works as follows.

$$z_i^0 = 0.5[(w_i + v_i) - \beta_0 \times (w_i - v_i)] \quad (10)$$

$$z_i^1 = 0.5[(w_i + v_i) + \beta_1 \times (w_i - v_i)] \quad (11)$$

where z_i^0 and z_i^1 are two decision variables of the generated child antibodies, and β_j ($j = 0, 1$) are obtained as follows.

$$\beta_j = \begin{cases} [r_j \times a_j]^{1/(\eta+1)}, & \text{if } r_j \leq \frac{1}{a_j} \\ \left[\frac{1}{2 - r_j \times a_j} \right]^{1/(\eta+1)}, & \text{otherwise} \end{cases} \quad (12)$$

where r_j ($j = 0, 1$) are uniformly distributed random numbers in $[0, 1]$; η is a crossover distribution index, a larger value of which will generate child solutions near to the parent solutions with a higher probability; a_j ($j = 0, 1$) are defined as follows (assuming that $w_i \neq v_i$).

$$a_j = \begin{cases} 2 - \left(1 + \left(2 \frac{v_i - l_i}{w_i - v_i} \right) \right)^{-(\eta+1)} & j = 0; \\ 2 - \left(1 + \left(2 \frac{u_i - w_i}{w_i - v_i} \right) \right)^{-(\eta+1)} & j = 1; \end{cases} \quad (13)$$

where l_i and u_i are the lower and upper bounds of the i -th decision variables, respectively. After applying SBX, a new antibody $\mathbf{x}' = (x'_1, x'_2, \dots, x'_n)$ is reorganized by randomly assigning x'_i with z_i^0 or z_i^1 ($i = 1, 2, \dots, n$).

Then, the new antibody \mathbf{x}' is further permuted by using polynomial-based mutation, defined as

$$x''_i = x'_i + \sigma \times (u_i - l_i), \quad i = 1, 2, \dots, n \quad (14)$$

where x''_i and x'_i are the i -th decision variables after and before mutation, respectively; σ is a small variation, generated by

$$\sigma = \begin{cases} [2r + (1 - 2r) \times \sigma_1^{\eta+1}]^{\frac{1}{\eta+1}} - 1, & \text{if } r < 0.5 \\ 1 - [2(1 - r) + (2r - 1) \times \sigma_2^{\eta+1}]^{\frac{1}{\eta+1}}, & \text{otherwise} \end{cases} \quad (15)$$

where r is a uniformly distributed random number in $[0, 1]$ and η is a mutation distribution index. A larger value of η generates smaller variances on average. The values of σ_1 and σ_2 are defined as

Algorithm 2: Selection(P_1, P_2, E)

```

1 collect  $P_1, P_2$  into  $E$ ;
2 for  $i=1$  to  $|E|$ 
3   for  $j=i+1$  to  $|E|$ 
4     if ( $E_j > E_i$ ) //Pareto domination in (2)
5       mark  $E_i$  as a dominated individual;
6     else if ( $E_i > E_j$ )
7       mark  $E_j$  as a dominated individual;
8     end if
9   end for
10 end for
11 delete the dominated solutions from  $E$ ;
12 if ( $|E| > N_E$ )
13   calculate the crowding distance [2] for each individual in  $E$ ;
14   while ( $|E| > N_E$ )
15     delete one individual with smallest crowding distance;
16     recalculate the crowding distance [2] for each individual in  $E$ ;
17   end while
18 end if

```

$$\sigma_1 = \frac{x'_i - l_i}{u_i - l_i} \quad \text{and} \quad \sigma_2 = \frac{u_i - x'_i}{u_i - l_i} \quad (16)$$

At last, x'_i is checked to see if it is still included in the boundary of the i -th decision variable. If not, x'_i is set as the corresponding boundary value, as follows.

$$x'_i = \begin{cases} l_i & \text{if } x'_i < l_i \\ u_i & \text{if } x'_i > u_i \end{cases} \quad (17)$$

2. Evolution 2

Differential evolution is a very powerful recombination operator, which is especially suitable for complicated problems with linkages in the decision variables [29], [59], [60]. It has been adopted to solve a wide variety of optimization problems, including multimodal optimization problems and MOPs. Assuming that each individual in C is represented by $\mathbf{x} = (x_1, x_2, \dots, x_n)$, a new antibody \mathbf{x}' is generated as follows.

$$x'_i = \begin{cases} x_i + F \times (x_i^{r_1} - x_i^{r_2}) & \text{if } r < CR \\ x_i & \text{otherwise} \end{cases} \quad (18)$$

where F and CR are two control parameters; r is a uniformly distributed random number in $[0, 1]$; r_1 and r_2 are two uniformly distributed random integers for selecting two parents in the specified population P . In this study, two strategies for the selection of P are presented. The first one is to select two different parents from A , which encourages performing global search as the antibodies in A are less-crowded in the elitist archive. The other one chooses two distinct parents from the neighbors of \mathbf{x} , which is beneficial for searching the local area. Let us assume that the number of neighbors is T . In our scheme, the definition of neighbors consists in finding the T closest antibodies according to the value of one randomly chosen objective function. This set of neighbors is denoted by $N(\mathbf{x})$. The selection of the parent population is controlled by a probability parameter δ , defined by

$$P = \begin{cases} N(\mathbf{x}) & \text{if } r < \delta \\ A & \text{otherwise} \end{cases} \quad (19)$$

Algorithm 3: The complete algorithm HEIA

```

1 Initialization (Algorithm 1);
2 while  $g < max\_g$ 
3   sort the antibodies in  $E$  descendingly according to the affinities;
4   select  $NA$  antibodies with high affinities;
5   for  $i=1$  to  $NA$ 
6     compute the clone number  $q_i$  for  $a_i$  using (7)-(9);
7     for  $j=1$  to  $q_i$  // subpopulation division
8       if  $rand() < 0.5$ 
9         add  $a_i$  to  $P_1$ ;
10      else
11        add  $a_i$  to  $P_2$ ;
12      end if
13    end for
14  end for
15  evolve subpopulation  $P_1$  using evolution 1 (Section III.B.1);
16  evolve subpopulation  $P_2$  using evolution 2 (Section III.B.2);
17  Selection( $P_1, P_2, E$ ) (Algorithm 2);
18   $g = g+1$ ;
19 end while
20 output  $E$ ;

```

After applying DE crossover, polynomial-based mutation defined in (14) is also adopted to permute the new antibody \mathbf{x}' .

C. Elitist Archive and Selection

After applying the two above evolutionary strategies, two subpopulations (P_1 and P_2) are collected into the elitist archive (E) and then Pareto-dominance [2] is applied to find the non-dominated antibodies. With the evolution of multiple generations, the number of nondominated antibodies may be very large. Therefore, a proper selection mechanism is needed to limit the size of the elitist archive and for helping to guide the search direction towards \mathbf{PF}_{true} . In most of the current selection strategies, Pareto dominance is first employed to determine nondominated individuals and then the density estimation information is further adopted to maintain the population diversity [2], [3], [23]-[25]. In this study, a selection mechanism presented in our previous work [25] is adopted, which performs a fine-grained selection procedure for the nondominated antibodies. Once the size of the elitist archive is larger than the predefined size N_E , the most crowded individual is deleted and then the crowding-distance values of its neighbors are recalculated. The pseudo-code of this fine-grained selection mechanism is shown in **Algorithm 2**, where the inputs are the two resultant subpopulations P_1 and P_2 that have been respectively permuted by the two above evolutionary strategies, as well as the elitist archive E . This selection procedure will finally maintain N_E nondominated solutions in the elitist archive E .

D. The Full Algorithm HEIA

The above subsections have introduced the main components of HEIA, *i.e.*, cloning, evolutionary strategies, elitist archive and selection. Other implementation details are described in the pseudo-code of HEIA, as illustrated in **Algorithm 3**, where g and max_g respectively denote the current generation and the maximum number of generations, and $rand()$ returns a uniformly distributed random number in $[0, 1]$.

After the initialization described in **Algorithm 1**, HEIA enters the loop of the evolutionary process until the maximum

TABLE I
PARAMETERS SETTINGS OF ALL THE ALGORITHMS COMPARED

Algorithms	Parameter settings
NSGA-II	$N=100, p_c=0.9, p_m=1/n, \eta_c=20, \eta_m=20$
SPEA2	$N=100, p_c=0.9, p_m=1/n, \eta_c=20, \eta_m=20$
AbYSS	$N=100, N_{\text{RefSet1}}=10, N_{\text{RefSet2}}=10, p_c=0.9, p_m=1/n, \eta_c=20, \eta_m=20$
MOEA/D	$N=100, CR=1.0, F=0.5, p_m=1/n, \eta_m=20, T=20, \delta=0.9, n_r=2$
SMPSO	$C_1 \in [1.5, 2.5], C_2 \in [1.5, 2.5], p_m=1/n, \eta_m=20$
NICA	$N=100, R=4, p_m=1/n$
NNIA	$N=100, NA=20, p_c=1.0, p_m=1/n, \eta_c=20, \eta_m=20$
MIMO	$N=100, NA=20, p_c=1.0, p_m=1/n, \eta_c=20, A=0.1, B=3.0$
HEIA	$N=100, NA=20, p_c=1.0, p_m=1/n, \eta_c=20, \eta_m=20, CR=1.0, F=0.5, T=20, \delta=0.9$

number of generations, max_g , is achieved. At first, NA antibodies with high affinities are picked out from the elitist archive E , by sorting the antibodies descendingly according to their affinities. Then, each selected antibody a_i ($i=1,2,\dots,NA$) is reproduced by cloning q_i copies, and each clone is randomly assigned to two subpopulations P_1 and P_2 in lines 7-13. After that, two subpopulations P_1 and P_1 are respectively permuted using **evolution 1** and **evolution 2** as described in Sections III.B.1 and III.B.2. At last, the mutant subpopulations P_1, P_2 and the original archive E are used as the input to run **Algorithm 2**, which will keep N_E nondominated solutions in E . The above evolutionary phase will be repeated until the (pre-defined) maximum number of generations, max_g , is reached. At the end of the algorithm, the nondominated solutions in E are reported as our final PF_{known} .

IV. EXPERIMENTAL RESULTS

A. Test Problems

Several types of test problems are adopted to evaluate the performance of HEIA. First, the popular ZDT problems are used [30]. Due to their lack of features such as variable linkages and objective function multimodality, this test suite is not particularly challenging. Therefore, the bi-objective WFG and UF problems are also adopted as they are characterized by presenting convexity, concavity, discontinuity, non-uniformity and the existence of many local Pareto-optimal fronts [26], [31]. Moreover, the three-objective DTLZ test problems are used to further examine the performance of HEIA in handling MOPs with more than two objectives [32]. Thus, we used a total of 28 test problems (ZDT1-ZDT4, ZDT6, WFG1-WFG9, UF1-UF7 and DTLZ1- DTLZ7) in our experimental studies. This large set is comprehensive and sufficient to assess the performance of multiobjective algorithms. It is noted that for ZDT1-ZDT3 and all UF problems, the number of decision variables is 30, while the number of decision variables in ZDT4, ZDT6, all the WFG and the DTLZ problems is 10. In the WFG problems, their 10 decision variables consist of 8 position parameters and 2 distance parameters. The details of the ZDT, WFG, UF and DTLZ test problems are available in [30], [31], [26] and [32], respectively.

B. Performance Measure

The goal of MOPs is to find a uniformly distributed set that is as close to the PF_{true} as possible. As the inverted generational distance (IGD) metric [29] can examine convergence and

diversity simultaneously, it is used to assess the performance of all the compared algorithms in our experimental studies.

Let S be a set of solutions that are uniformly distributed along PF_{true} and let S' be the set of best solutions (*i.e.*, PF_{known}) that are found by an algorithm. The IGD value of S to S' , *i.e.*, $IGD(S, S')$ is defined as

$$IGD(S, S') = \frac{\sum_{i=1}^{|S|} d(S_i, S')}{|S|} \quad (20)$$

where $|S|$ returns the size of the set S and $d(S_i, S')$ denotes the minimum Euclidean distance in objective space between S_i and the individuals in S' . IGD requires to know PF_{true} in advance. The subsets of PF_{true} adopted in our experiments can be found in <http://jmetal.sourceforge.net/problems.html>. In general, a lower value of $IGD(S, S')$ is preferred as it indicates that S' obtains a more even coverage of PF_{true} and is closer to PF_{true} .

C. Experimental Settings

In our experiments, in order to assess the performance of HEIA, we compared it with respect to several types of nature-inspired heuristic algorithms for solving MOPs, including NSGA-II [2], SPEA2 [3], AbYSS [20], MOEA/D [29], and SMPSO [33]. Moreover, we also compared HEIA with respect to three recently proposed MOIAs, *i.e.*, NICA [23], NNIA [24] and MIMO [25]. All the algorithms have shown a competitive performance when solving MOPs. Thus, a comparison of our results with those produced by such algorithms should provide a comprehensive performance assessment for the proposed HEIA algorithm.

The parameters settings of the compared algorithms were established as recommended in their original references [2], [3], [20], [23]-[25], [29], [33], as summarized in Table I. It is worth noting that the parameters of the compared algorithms were properly tuned to solve most of the MOPs adopted in our experimental studies. To allow a fair comparison, the parameters of HEIA were set according to those of the compared algorithms. In Table I, N is the population size; p_c is the crossover probability and p_m is the mutation probability; η_c and η_m are the distribution indexes of SBX and polynomial mutation, respectively. For AbYSS, N_{RefSet1} and N_{RefSet2} are the sizes of RefSet1 and RefSet2, respectively. In MOEA/D, T defines the size of the neighborhood in the weight coefficients, δ controls the probability that parent solutions are chosen from T neighbors and n_r is the maximum number of parent

TABLE II
COMPARISON OF RESULTS ON THE ZDT TEST PROBLEMS

Problems	Algorithms	NSGA-II	SPEA2	AbYSS	MOEA/D	SMP SO	HEIA
ZDT1	Mean	4.95E-03	4.26E-03	3.90E-03	1.75E-02	3.68E-03	3.90E-03
	Std	1.89E-04	1.09E-04	1.16E-04	5.51E-03	2.61E-05	6.57E-05
	Rank	5-	4-	3≈	6-	1+	2
ZDT2	Mean	5.06E-03	4.64E-03	4.24E-03	1.33E-02	3.81E-03	3.96E-03
	Std	2.17E-04	1.86E-03	2.15E-03	5.41E-03	3.24E-05	5.23E-05
	Rank	5-	4-	3-	6-	1+	2
ZDT3	Mean	5.68E-03	5.93E-03	1.91E-02	6.04E-02	4.48E-03	4.43E-03
	Std	2.96E-03	5.11E-03	2.38E-02	2.27E-02	2.53E-04	5.41E-05
	Rank	3-	4-	5-	6-	2≈	1
ZDT4	Mean	7.28E-03	1.98E-02	1.05E-02	3.12E-01	3.77E-03	3.87E-03
	Std	2.14E-03	2.51E-02	1.73E-02	2.34E-01	4.44E-05	2.00E-04
	Rank	3-	5-	4-	6-	1+	2
ZDT6	Mean	8.29E-03	1.55E-02	3.04E-03	2.45E-03	3.03E-03	3.02E-03
	Std	8.47E-04	2.32E-03	1.13E-04	7.09E-04	1.93E-04	1.29E-04
	Rank	5-	6-	4≈	1+	3≈	2
Rank Sum		21	23	19	25	8	9
Final Rank		4	5	3	6	1	2
<i>better/worse/similar</i>		0/5/0	0/5/0	0/3/2	1/4/0	3/0/2	7

“+”, “-”, and “≈” indicate that the results obtained by the algorithm are significantly better than, worse than, and similar to the ones obtained by HEIA using Wilcoxon’s rank sum test with a significant level $\alpha=0.05$, respectively.

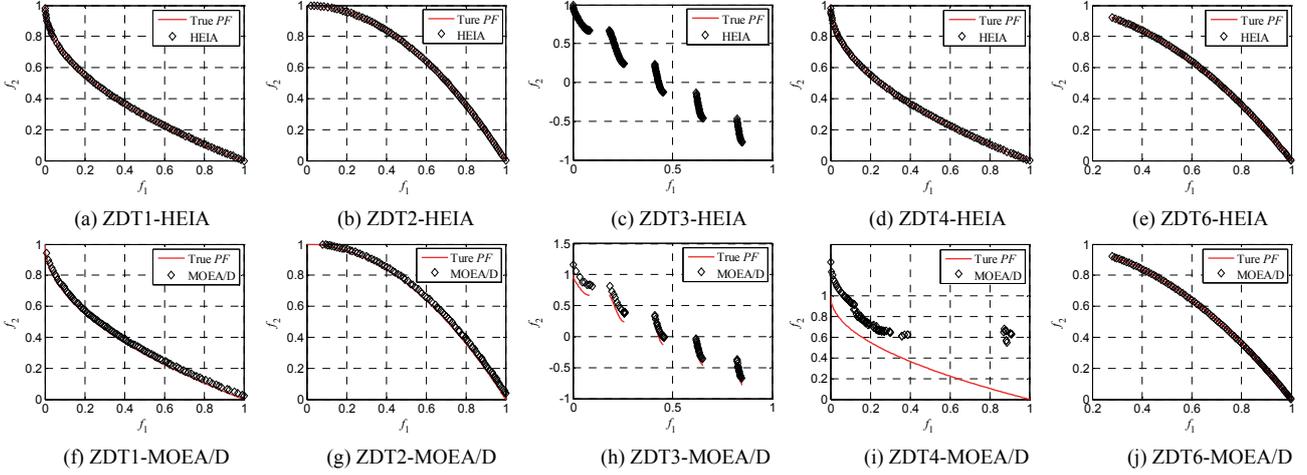


Fig. 2 The nondominated solution sets found by HEIA and MOEA/D on the ZDT problems

solutions that are replaced by each child solution. C_1 and C_2 are two control parameters randomly picked within the range [1.5, 2.5] in SMP SO. For NNIA, MIMO and HEIA, NA is the size of selected antibodies for cloning proliferation. A and B are the two control parameters used in the adaptive mutation operator of MIMO and R is the clonal rate in the entire cloning of NICA. The external archive size N_E is usually set to the same value as N .

It is noted that the settings of N and NA listed in Table I are only applied for the ZDT problems and that the maximum number of function evaluations was set to 25000. When handling other MOPs, the population size and the maximum number of function evaluations were adjusted based on the difficulty and complexity of the MOP to be solved. For solving the more difficult WFG and three-objective DTLZ test problems, the population sizes were set to 200 and 500, respectively. In this case, the maximum number of function evaluations was set to 10^5 . As the UF problems are extremely complicated, all the algorithms adopted a population size of 300 and the maximum number of function evaluations was set to 3×10^5 . The

settings of N_{RefSet1} , N_{RefSet2} and NA on these test problems were proportionally adjusted with the population size N . The rest of the parameters settings were the same as listed in Table I. All the experiments were run 100 times (using different random seeds), the mean IGD values and the corresponding standard deviations (std) of which were collected for comparison. The best results are identified in **boldface** in the comparison tables. Moreover, in order to have a statistically sound conclusion, the Wilcoxon’s rank sum test was further conducted to assess the statistical significance of the difference between the results obtained by HEIA and those obtained by the other algorithms with a significance level $\alpha=0.05$.

D. Comparisons of HEIA with NSGA-II, SPEA2, AbYSS, MOEA/D and SMP SO

1) Comparisons on the ZDT Test Problems

Table II provides the experimental results of all the algorithms on the ZDT problems. The results show that NSGA-II, SMP SO and HEIA were able to find good approximations of PF_{true} for all the ZDT problems as the corresponding mean

TABLE III
COMPARISON OF RESULTS ON THE WFG TEST PROBLEMS

Problems	Algorithms	NSGA-II	SPEA2	AbYSS	MOEA/D	SMPSO	HEIA
		WFG1	Mean Std Rank	6.29E-01 2.18E-01 4-	1.22E+00 2.07E-01 5-	1.29E+00 2.41E-01 6-	1.60E-02 7.81E-03 2-
WFG2	Mean Std Rank	1.04E-01 7.35E-02 5-	9.33E-02 6.53E-02 4-	1.77E-01 7.91E-02 6-	4.44E-02 2.61E-02 2≈	4.81E-03 2.73E-04 1+	7.82E-02 7.54E-02 3
WFG3	Mean Std Rank	7.40E-03 3.39E-04 6-	6.07E-03 4.01E-04 3-	6.08E-03 9.31E-04 4≈	6.73E-03 6.41E-06 5-	5.52E-03 6.52E-04 1+	5.79E-03 4.17E-04 2
WFG4	Mean Std Rank	6.90E-03 3.91E-04 5-	6.26E-03 1.76E-04 3-	5.11E-03 1.33E-04 1+	8.07E-03 1.33E-03 6-	6.74E-03 3.68E-04 4-	5.46E-03 1.64E-04 2
WFG5	Mean Std Rank	6.54E-02 3.07E-03 6-	6.49E-02 3.19E-03 2+	6.47E-02 3.26E-03 1+	6.54E-02 1.68E-04 5-	6.53E-02 8.85E-04 4-	6.50E-02 3.19E-03 3
WFG6	Mean Std Rank	8.64E-03 9.77E-04 4-	1.27E-02 6.86E-03 5-	1.35E-02 8.99E-03 6-	7.59E-03 4.01E-03 3-	7.30E-03 1.28E-03 2≈	7.07E-03 1.17E-03 1
WFG7	Mean Std Rank	8.72E-03 4.79E-04 5-	7.18E-03 5.52E-04 3-	7.71E-03 2.65E-03 4-	8.82E-03 6.33E-04 6-	5.78E-03 8.73E-05 1+	6.20E-03 1.18E-04 2
WFG8	Mean Std Rank	2.26E-02 3.28E-03 3-	5.13E-02 7.98E-03 5-	6.79E-02 7.25E-02 6-	2.04E-02 2.29E-03 2-	2.70E-02 3.65E-03 4-	7.89E-03 4.25E-04 1
WFG9	Mean Std Rank	7.96E-03 3.89E-04 6-	6.45E-03 2.91E-04 1≈	7.15E-03 2.31E-03 3≈	7.48E-03 3.73E-04 4-	7.86E-03 7.83E-04 5-	6.47E-03 1.68E-04 2
Rank Sum		44	31	37	35	25	17
Final Rank		6	3	5	4	2	1
<i>better/worse/similar</i>		0/9/0	1/7/1	2/5/2	0/8/1	3/5/1	/

“+”, “-”, and “≈” indicate that the results obtained by the algorithm are significantly better than, worse than, and similar to the ones obtained by HEIA using Wilcoxon’s rank sum test with a significant level $\alpha=0.05$, respectively.

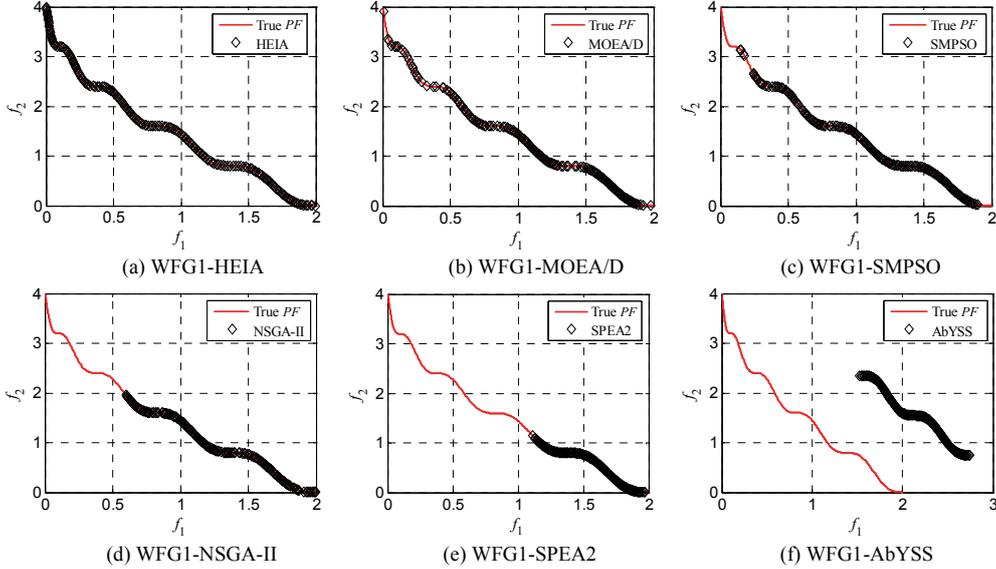


Fig. 3 The nondominated solution sets found by all the algorithms on WFG1

values of IGD are under an accuracy level of 10^{-3} . SPEA2 also obtained good approximations of \mathbf{PF}_{true} on ZDT1, ZDT2 and ZDT3, while AbYSS performed well on ZDT1, ZDT2 and ZDT6. Although MOEA/D obtained the best performance on ZDT6, it gave the worst results on ZDT1-ZDT4. As ZDT3 has a \mathbf{PF}_{true} with multiple disconnections, AbYSS and MOEA/D failed to approach all the disconnected parts of \mathbf{PF}_{true} in some runs. ZDT4 has many local Pareto-optimal fronts, which increases the difficulty in searching for \mathbf{PF}_{true} . SPEA2, AbYSS and MOEA/D could not effectively approach the \mathbf{PF}_{true} of

ZDT4. SPEA2 performed worse on ZDT6 as it has a non-uniform search space. Moreover, the Wilcoxon’s rank sum test indicates that HEIA performed similarly to AbYSS on ZDT1 and ZDT6, and to SMPSO on ZDT3 and ZDT6, respectively. The last third row, labeled “Rank Sum” summarizes the ranks obtained by all the algorithms in solving all the ZDT problems, and the last second row labeled “Final Rank” shows the final ranks of all the algorithms according to “Rank Sum”. As observed from the “Final Rank” row, SMPSO and HEIA obtained the first and second ranks, respectively, while

TABLE IV
COMPARISON OF RESULTS ON THE UF TEST PROBLEMS

Problems \ Algorithms		NSGA-II	SPEA2	AbYSS	MOEA/D	SMPSO	HEIA
UF1	Mean	7.30E-02	9.02E-02	9.37E-02	1.79E-03	6.39E-02	2.70E-03
	Std	1.14E-02	1.45E-02	2.86E-02	1.93E-04	9.38E-03	1.42E-04
	Rank	4-	5-	6-	1+	3-	2
UF2	Mean	2.34E-02	2.55E-02	3.69E-02	6.53E-03	2.23E-02	5.81E-03
	Std	6.66E-03	6.15E-03	1.89E-02	1.74E-03	2.26E-03	4.98E-04
	Rank	4-	5-	6-	2-	3-	1
UF3	Mean	1.17E-01	1.61E-01	2.31E-01	1.13E-02	1.18E-01	1.28E-02
	Std	2.93E-02	4.29E-02	7.98E-02	1.23E-02	3.81E-02	6.10E-03
	Rank	3-	5-	6-	1+	4-	2
UF4	Mean	4.09E-02	4.12E-02	4.19E-02	6.09E-02	4.86E-02	3.77E-02
	Std	5.92E-04	6.70E-04	1.15E-03	4.96E-03	2.73E-03	7.49E-04
	Rank	2-	3-	4-	6-	5-	1
UF5	Mean	2.23E-01	2.32E-01	2.80E-01	2.95E-01	1.17E+00	2.05E-01
	Std	4.22E-02	5.43E-02	5.69E-02	1.15E-01	4.91E-01	1.17E-01
	Rank	2-	3-	4-	5-	6-	1
UF6	Mean	1.17E-01	1.37E-01	1.94E-01	1.95E-01	4.47E-01	1.52E-01
	Std	3.52E-02	6.37E-02	1.04E-01	1.73E-01	6.38E-02	8.47E-02
	Rank	1 ≈	2+	4-	5≈	6-	3
UF7	Mean	6.51E-02	9.51E-02	2.28E-01	2.95E-03	2.15E-02	3.09E-03
	Std	8.31E-02	1.18E-01	1.49E-01	2.57E-03	2.54E-03	5.56E-04
	Rank	4-	5-	6-	1+	3-	2
Rank Sum		20	28	36	21	30	12
Final Rank		2	4	6	3	5	1
<i>better/worse/similar</i>		0/6/1	1/6/0	0/7/0	3/3/1	0/7/0	/

“+”, “-”, and “≈” indicate that the results obtained by the algorithm are significantly better than, worse than, and similar to the ones obtained by HEIA using Wilcoxon’s rank sum test with a significant level $\alpha=0.05$, respectively.

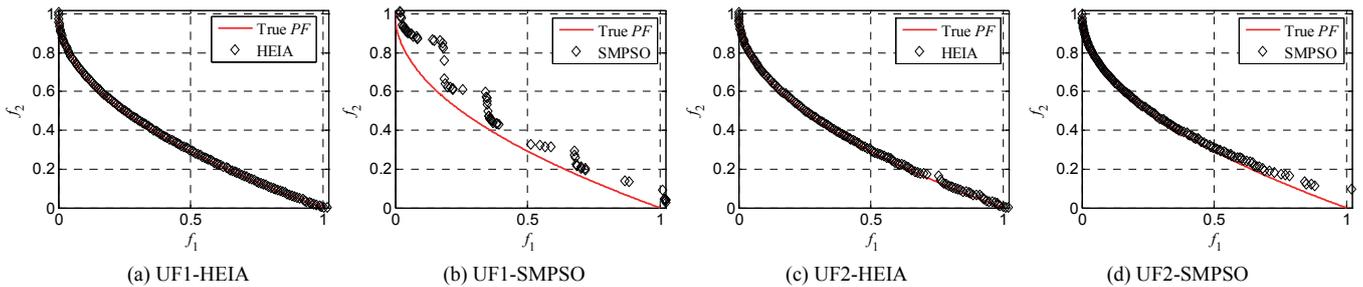


Fig. 4 The nondominated solution sets found by HEIA and SMPSO on UF1 and UF2

MOEA/D obtained the last rank. The last row “*better/worse/similar*” indicates the number of test problems in which the performance of the compared algorithm was better than, worse than or similar to that of HEIA. This row also indicates that HEIA performed worse than SMPSO, but it was better than any of the other algorithms.

It is pointed out that when the *IGD* values are under an accuracy level of 10^{-3} , the found solution sets are uniform enough to closely approximate \mathbf{PF}_{true} . In Fig. 2, the \mathbf{PF}_{known} obtained by HEIA on all the ZDT problems are illustrated. As other compared algorithms also obtained mean *IGD* results under an accuracy level of 10^{-3} for some ZDT problems, their performance was similar and, therefore, is relatively indistinguishable when graphed; however we plotted the results of MOEA/D in Fig. 2, which delivered the worst overall results, to give a sense of the range of solution qualities obtained. It is noted that only one final set that has the *IGD* value closest to the mean *IGD* value in 100 runs was plotted in Fig. 2. Except for SMPSO, which performed better than HEIA, all the other algorithms had a performance ranking located between HEIA and MOEA/D. As observed in Fig. 2, where the \mathbf{PF}_{true} are identified with red color, the \mathbf{PF}_{known} found by HEIA have a uniform distribution

along the \mathbf{PF}_{true} on all the ZDT problems. For MOEA/D, it is found that the obtained sets of \mathbf{PF}_{known} are also distributed uniformly, but not so close to \mathbf{PF}_{true} , especially on ZDT4, which has many local Pareto-optimal fronts. These plots confirm that HEIA can consistently find an evenly distributed \mathbf{PF}_{known} to closely approach \mathbf{PF}_{true} of all the ZDT problems, while MOEA/D fails to get a good approximation set for some ZDT problems.

2) Comparisons on the WFG Test Problems

In Table III, the comparisons on the WFG problems are illustrated. As observed in Table III, HEIA performed best on WFG1, WFG6 and WFG8, while SMPSO obtained the best results on WFG2, WFG3 and WFG7. AbYSS provided the best performance on WFG4 and WFG5, while SPEA2 performed best on WFG9. The Wilcoxon’s rank sum test shows that HEIA achieved statistically similar results to SPEA2 on WFG9, to AbYSS on WFG3 and WFG9, to MOEA/D on WFG2, and to SMPSO on WFG6. As revealed by the “Final rank” row, HEIA performed best when considering all the WFG problems. SMPSO and SPEA2 obtained the second and third ranks, respectively. Moreover, MOEA/D, AbYSS and NSGA-II obtained the 4th, 5th, and 6th ranks, respectively. As the WFG

TABLE V
COMPARISON OF RESULTS ON THE DTLZ TEST PROBLEMS

Problems \ Algorithms		NSGA-II	SPEA2	AbYSS	MOEA/D	SMPSO	HEIA
DTLZ1	Mean	2.18E-02	3.32E-02	2.53E-02	4.23E-02	1.28E-02	1.16E-02
	Std	1.98E-02	3.48E-02	5.12E-02	1.17E-01	3.82E-04	3.98E-04
	Rank	3-	5-	4-	6≈	2-	1
DTLZ2	Mean	3.07E-02	2.38E-02	3.02E-02	2.82E-02	3.12E-02	3.08E-02
	Std	7.43E-04	2.76E-04	7.26E-04	1.76E-04	7.21E-04	7.57E-04
	Rank	4≈	1+	3+	2+	6-	5
DTLZ3	Mean	5.04E-02	3.83E-01	3.67E-02	1.45E-01	3.10E-02	3.04E-02
	Std	1.95E-02	2.08E-01	1.63E-02	5.13E-01	5.73E-04	6.47E-04
	Rank	4-	6-	3-	5≈	2-	1
DTLZ4	Mean	3.08E-02	2.09E-02	2.81E-02	1.87E-02	2.60E-02	3.24E-02
	Std	2.39E-03	1.52E-03	2.07E-03	6.04E-04	6.46E-03	2.33E-03
	Rank	5+	2+	4+	1+	3+	6
DTLZ5	Mean	1.07E-03	8.73E-04	7.96E-04	1.85E-03	8.16E-04	8.44E-04
	Std	4.20E-05	3.41E-05	3.11E-05	2.06E-05	2.97E-05	3.29E-05
	Rank	5-	4-	1+	6-	2+	3
DTLZ6	Mean	1.41E-01	2.75E-01	2.04E-02	1.72E-03	7.87E-04	7.96E-04
	Std	3.91E-02	2.09E-02	1.92E-02	7.27E-06	4.16E-05	4.17E-05
	Rank	5-	6-	4-	3-	1≈	2
DTLZ7	Mean	3.05E-02	2.39E-02	1.20E-01	7.99E-02	3.35E-02	3.21E-02
	Std	8.03E-04	3.98E-04	1.37E-01	8.61E-02	1.53E-03	1.12E-03
	Rank	2+	1+	6≈	5-	4-	3
Rank Sum		28	25	25	28	20	21
Final Rank		4	3	3	4	1	2
<i>better/worse/similar</i>		2/4/1	3/4/0	3/3/1	2/3/2	2/4/1	7

“+”, “-”, and “≈” indicate that the results obtained by the algorithm are significantly better than, worse than, and similar to the ones obtained by HEIA using Wilcoxon’s rank sum test with a significant level $\alpha=0.05$, respectively.

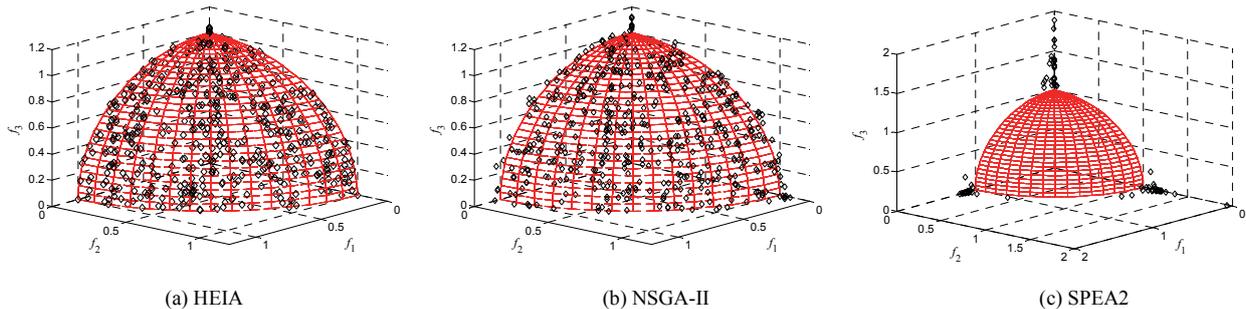


Fig. 5 The nondominated solution sets found by HEIA, NSGA-II and SPEA2 on DTLZ3

problems are more difficult than the ZDT problems, the superior performance of HEIA on the WFG problems further confirms its advantages.

In Fig. 3, we plotted the final nondominated sets obtained by all the algorithms on WFG1, as WFG1 seems to be the most difficult problem to solve, since some algorithms failed to reach a close \mathbf{PF}_{known} for it. One final set that has the IGD value closest to the mean IGD value obtained from 100 independent runs was plotted in Fig. 3. It is observed that the \mathbf{PF}_{known} of HEIA is distributed evenly along \mathbf{PF}_{true} . Although MOEA/D and SMPSO also produced the final sets of \mathbf{PF}_{known} close to \mathbf{PF}_{true} , such sets did not have a uniform distribution. NSGA-II and SPEA2 only approached half of \mathbf{PF}_{true} , while AbYSS failed to approximate \mathbf{PF}_{true} in this case.

3) Comparisons on the UF Series Problems

Table IV lists our experimental results on the UF problems, which have very complicated Pareto-optimal sets and thus present more challenges to multiobjective algorithms. As observed in Table IV, HEIA performed best in three test problems, *i.e.*, UF2, UF4, and UF5. MOEA/D obtained the best results on UF1, UF3 and UF7, while NSGA-II produced the

best approximation on UF6. Moreover, the Wilcoxon’s rank sum test shows that HEIA obtained statistically similar results to NSGA-II and MOEA/D on UF6. Considering all the UF problems, HEIA performed best as it obtained the first rank in the “Final rank” row. Moreover, the last row in Table IV also indicates that HEIA performed better than NSGA-II, SPEA2, AbYSS, and SMPSO on most of the UF problems. Only MOEA/D obtained a similar performance to that of HEIA as it performed better in 3 out of 7 UF problems and had a similar performance in another one. The promising results on the UF problems also confirm the advantages of HEIA.

In Fig. 4, we plotted the final nondominated sets obtained by some of the compared algorithms on UF1 and UF2. As the IGD values of NSGA-II, SPEA2, AbYSS and SMPSO on UF1 and UF2 are all under an accuracy level of 10^{-2} , their final sets of \mathbf{PF}_{known} will look very similar in these plots. Thus, the \mathbf{PF}_{known} obtained by SMPSO is plotted as the representation of NSGA-II, SPEA2, AbYSS and SMPSO. Also, the \mathbf{PF}_{known} obtained by HEIA is plotted as the representation of HEIA and MOEA/D. The plots in Fig. 4 show that HEIA can closely approach the \mathbf{PF}_{true} of UF1, while SMPSO only finds some disconnected parts that are not so near to \mathbf{PF}_{true} . For UF2, the

TABLE VI
FINAL RANK OF ALL THE ALGORITHMS ON THE ZDT, WFG, UF AND DTLZ PROBLEMS

Problems	Algorithms	NSGA-II	SPEA2	AbYSS	MOEAD	SMPSO	HEIA
Rank Sum on ZDTs		21	23	19	25	8	9
Rank Sum on WFGs		44	31	37	35	25	17
Rank Sum on UFs		20	28	36	21	30	12
Rank Sum on DTLZs		28	25	25	28	20	21
Total Rank Sum on all the Problems		113	107	117	109	83	59
Final Rank on all the Problems		5	3	6	4	2	1

TABLE VII
FINAL COMPARISONS OF ALL THE ALGORITHMS ON THE ZDT, WFG, UF AND DTLZ PROBLEMS

Problems	Algorithms	NSGA-II	SPEA2	AbYSS	MOEAD	SMPSO
ZDTs		0/5/0	0/5/0	0/3/2	1/4/0	3/0/2
WFGs		0/9/0	1/7/1	2/5/2	0/8/1	3/5/1
UFs		0/6/1	1/6/0	0/7/0	3/3/1	0/7/0
DTLZs		2/4/1	3/4/0	3/3/1	2/3/2	2/4/1
Total <i>better/worse/similar</i>		2/24/2	5/22/1	5/18/5	6/18/4	8/16/4
Final result (HEIA vs other algorithms)		<i>better</i>	<i>better</i>	<i>better</i>	<i>better</i>	<i>better</i>

\mathbf{PF}_{known} of HEIA achieves a more even coverage of \mathbf{PF}_{true} and is closer to \mathbf{PF}_{true} than the approximation obtained by SMPSO.

4) Comparisons on the DTLZ Test Problems

In the above experiments, HEIA had a very promising performance when solving all of the ZDT, WFG and UF test problems. However, these problems have only two objectives. In order to further assess the performance of HEIA on MOPs with more than two objectives, the DTLZ problems [32] were adopted. Table V shows the comparison of results when solving all the DTLZ test problems. As observed in Table V, HEIA provided the best results on DTLZ1 and DTLZ3. SPEA2 performed best on DTLZ2 and DTLZ7. MOEA/D, AbYSS, and SMPSO achieved the best results on DTLZ4, DTLZ5, and DTLZ6, respectively. Moreover, the Wilcoxon's rank sum test indicates that HEIA had a similar performance as MOEA/D on DTLZ1 and DTLZ3. However, the mean and std values of MOEA/D are larger than those of HEIA, which indicate that MOEA/D cannot consistently approach the \mathbf{PF}_{true} of DTLZ1 and DTLZ3. Also, HEIA obtained statistically similar results to NSGA-II on DTLZ2, to AbYSS on DTLZ7, and to SMPSO on DTLZ6. The "Final rank" row indicates that SMPSO and HEIA obtained the first rank and the second rank, respectively, while AbYSS and MOEA/D had a comparable performance with the 3rd place in the ranking. SPEA2 and NSGA-II also performed similarly with 4th place in the ranking. Moreover, the last row in Table V also indicates that HEIA performed better than NSGA-II, SPEA2, and SMPSO as HEIA performed better in 4 out of 7 DTLZ test problems. Also, HEIA outperformed MOEA/D as HEIA performed better, worse, and similarly on 3, 2 and 2 out of 7 test problems, respectively. For AbYSS, HEIA had a comparable performance as they both performed better on 3 DTLZ problems.

In Fig. 5, some plots of the final sets on DTLZ3 are given as DTLZ3 has many local Pareto-optimal fronts and is difficult to solve. The plots of AbYSS, MOEA/D and SMPSO are not evidently different from those of HEIA. Therefore, the plot of HEIA is used as a representation of all of them. As observed from Fig. 5, both the final sets of HEIA and NSGA-II corre-

spond to good approximation sets, but the one of HEIA provides a more even coverage of \mathbf{PF}_{true} and is closer to \mathbf{PF}_{true} . SPEA2 cannot approach \mathbf{PF}_{true} and only finds the solutions near the boundaries of \mathbf{PF}_{true} .

In Table VI, we collected the rank sums of HEIA, NSGA-II, SPEA2, AbYSS, MOEA/D and SMPSO on all the ZDT, WFG, UF and DTLZ test problems. When considering all of the test problems, the final rank indicates that HEIA performed better than NSGA-II, SPEA2, AbYSS, MOEA/D and SMPSO. Moreover, we also gathered the comparison results of HEIA with other algorithms in Table VII. The last second row illustrates the total comparison results of HEIA on all the test problems, and the last row provides the summary of the comparison performance of HEIA with respect to the other algorithms. The final ranking shown in Table VII also confirms the advantages of HEIA as it performs better than the compared algorithms on most of the test problems adopted. Based on the above experimental results, it is reasonable to conclude that HEIA is able to tackle various kinds of test problems and its advantages are more evident when solving complicated test problems, such as those included in the WFG and UF test suites.

E. Comparisons with Other MOIAs

The above simulations have shown the advantages of HEIA when compared with other nature-inspired multiobjective algorithms. In this subsection, we further compare HEIA with three MOIAs (*i.e.*, NICA [23], NNIA [24] and MIMO [25]), all of which were designed based on the clonal selection principle. As pointed out above, the use of a simple evolutionary operator in basic MOIAs is not so suitable for solving different types of MOPs due to its monotonic search patterns. This will cause some difficulties when handling complicated MOPs. Therefore, we provide a comparison of results with respect to NNIA, MIMO and HEIA on the ZDT problems and the UF test problems in Tables VIII and IX, respectively.

For the ZDT problems in Table VIII, it is observed that all the *IGD* values are under an accuracy level of 10^{-3} , which indicates that all of NNIA, MIMO and HEIA can closely ap-

TABLE VIII
COMPARISON OF RESULTS WITH RESPECT TO NNIA, MIMO AND HEIA
ON THE ZDT TEST PROBLEMS

Problems	Algorithms	NNIA	MIMO	HEIA
		Mean	4.84E-03	3.98E-03
ZDT1	Std	2.25E-04	8.47E-05	6.57E-05
	Rank	3-	2-	1
	Mean	4.86E-03	4.06E-03	3.96E-03
ZDT2	Std	2.10E-04	7.21E-05	5.23E-05
	Rank	3-	2-	1
	Mean	7.22E-03	5.98E-03	4.43E-03
ZDT3	Std	7.62E-03	6.54E-03	5.41E-05
	Rank	3-	2-	1
	Mean	6.43E-03	6.44E-03	3.87E-03
ZDT4	Std	1.46E-03	1.94E-03	2.00E-04
	Rank	2-	3-	1
	Mean	3.88E-03	3.16E-03	3.02E-03
ZDT6	Std	4.03E-04	1.35E-04	1.29E-04
	Rank	3-	2-	1
	Rank Sum	14	11	5
Final Rank	3	2	1	
<i>better/worse/similar</i>	0/5/0	0/5/0	/	

“+”, “-”, and “≈” indicate the results obtained by the algorithm are significantly better than, worse than, and similar to the ones obtained by HEIA using Wilcoxon’s rank sum test with a significant level $\alpha=0.05$.

proach PF_{true} . The Wilcoxon’s rank sum test shows that HEIA is significantly better than MIMO and NNIA on all the ZDT problems. Furthermore, when solving the UF test problems, it is found that HEIA performs better than NNIA and MIMO on all of them as indicated by the results in Table IX. The Wilcoxon’s rank sum test indicates that the IGD results of HEIA are all significantly better than those obtained by NNIA and MIMO. These experimental results validate that the simple evolutionary operator in NNIA and MIMO only performs well in solving simple test problems (*i.e.*, the ZDT problems), but is unable to reach the PF_{true} of complicated MOPs (*i.e.*, the UF test problems). However, HEIA can consistently achieve promising results on both of the ZDT and the UF test problems, which validates the effectiveness of our proposed hybrid framework for MOIAs. Moreover, referring to Tables II, IV, VIII and IX, one interesting phenomenon is observed: early reported MOIAs generally outperform classical MOEAs on most of the simple ZDT problems (*i.e.*, NNIA performs better than NSGA-II, and MIMO performs better than NSGA-II and SPEA2). This is mainly due to the clonal selection principle adopted in NNIA and MIMO, which helps to accelerate the convergence rate on the simple ZDT problems. However, when tackling the complicated UF problems, it is found that NNIA and MIMO are outperformed by NSGA-II and SPEA2 as the evolutionary search based on simulated binary crossover and polynomial-based mutation is ineffective for searching the complicated PS of the UF problems and the cloning of a small proportion of the population will lower the population diversity of NNIA and MIMO. Different from NNIA and MIMO, a hybrid evolutionary framework, where different subpopulations undergo different evolutionary strategies separately, is adopted in HEIA, which makes it capable of solving both of the ZDT and the UF test problems.

Furthermore, a comparison of results between NICA and HEIA on ZDT1-ZDT4 is provided in Table X. These results were obtained by performing 50000 function evaluations. Two

TABLE IX
COMPARISON OF RESULTS WITH RESPECT TO NNIA, MIMO AND HEIA
ON THE UF TEST PROBLEMS

Problems	Algorithms	NNIA	MIMO	HEIA
		Mean	8.38E-02	9.70E-02
UF1	Std	2.61E-02	4.48E-02	1.42E-04
	Rank	2-	3-	1
	Mean	3.52E-02	3.47E-02	5.81E-03
UF2	Std	1.32E-02	1.55E-02	4.98E-04
	Rank	3-	2-	1
	Mean	2.14E-01	2.07E-01	1.28E-02
UF3	Std	5.12E-02	5.66E-02	6.10E-03
	Rank	3-	2-	1
	Mean	4.14E-02	4.08E-02	3.77E-02
UF4	Std	5.73E-04	7.15E-04	7.49E-04
	Rank	3-	2-	1
	Mean	2.95E-01	2.94E-01	2.05E-01
UF5	Std	9.32E-02	9.47E-02	1.17E-01
	Rank	3-	2-	1
	Mean	2.04E-01	2.34E-01	1.52E-01
UF6	Std	1.18E-01	1.45E-01	8.47E-02
	Rank	2-	3-	1
	Mean	1.79E-01	1.61E-01	3.09E-03
UF7	Std	1.61E-01	1.90E-01	5.56E-04
	Rank	3-	2-	1
	Rank Sum	19	16	7
Final Rank	3	2	1	
<i>better/worse/similar</i>	0/7/0	0/7/0	/	

“+”, “-”, and “≈” indicate the results obtained by the algorithm are significantly better than, worse than, and similar to the ones obtained by HEIA using Wilcoxon’s rank sum test with a significant level $\alpha=0.05$.

TABLE X
COMPARISONS OF RESULTS WITH RESPECT TO NICA AND HEIA ON
ZDT1-ZDT4

Algorithms	Problems	ZDT1	ZDT2	ZDT3	ZDT4
		Mean	1.29e-4	1.25e-4	5.89e-4
NICA(GD)	Std	2.14e-6	6.35e-5	1.14e-5	5.89e-4
	Mean	9.66e-5	9.30e-5	1.12e-4	9.08e-5
HEIA(GD)	Std	1.02e-5	7.93e-6	4.66e-6	5.41e-6
	Mean	0.2595	0.2352	0.3525	0.2776
NICA(Diversity)	Std	0.0001	0.0013	0.0000	0.0000
	Mean	0.0934	0.0879	0.4178	0.0791
HEIA(Diversity)	Std	0.0098	0.0089	0.0057	0.0089

additional performance measures, *i.e.*, generational distance (GD) and spacing [2], were adopted to assess convergence and diversity, respectively. It is noted that 500 uniformly distributed points of PF_{true} are used to obtain the GD values. Regarding the GD values, it is obvious that HEIA always found solution sets with better convergence than NICA. When considering the diversity metric, HEIA was better than NICA on ZDT1, ZDT2 and ZDT4. Only for ZDT3, which has a disconnected PF_{true} , HEIA performed worse. However, as observed from Fig. 2(c), the population diversity of HEIA on ZDT3 is quite uniform. Therefore, we claim that HEIA is better than NICA in terms of both convergence and diversity when considering all the ZDT problems in Table X.

F. Advantages of the Hybrid Evolutionary Strategies

In this study, two evolutionary strategies were adopted to form HEIA as an implementation example of our proposed hybrid framework. In order to study the advantages of such a framework and to show how the two evolutionary strategies cooperate with each other, we also provide experimental results of HEIA using only one evolutionary strategy at a time, *i.e.*,

TABLE XI

COMPARISON OF RESULTS WITH RESPECT TO HEIA-1, HEIA-2 AND HEIA ON THE ZDT TEST PROBLEMS

Problems	Algorithms	HEIA-I	HEIA-II	HEIA
ZDT1	Mean	3.99E-03	1.29E-02	3.90E-03
	Std	1.15E-04	4.79E-03	6.57E-05
	Rank	2-	3-	1
ZDT2	Mean	4.05E-03	8.02E-03	3.96E-03
	Std	8.77E-05	2.96E-03	5.23E-05
	Rank	2-	3-	1
ZDT3	Mean	6.24E-03	8.77E-03	4.43E-03
	Std	7.13E-03	2.77E-03	5.41E-05
	Rank	2-	3-	1
ZDT4	Mean	6.57E-03	7.40E+00	3.87E-03
	Std	2.03E-03	1.33E+01	2.00E-04
	Rank	2-	3-	1
ZDT6	Mean	3.10E-03	3.39E-03	3.02E-03
	Std	1.27E-04	2.63E-04	1.29E-04
	Rank	2 \approx	3-	1
Rank Sum		10	15	5
Final Rank		2	3	1
<i>better/worse/similar</i>		0/4/1	0/5/0	/

“+”, “-”, and “ \approx ” indicate the results obtained by the algorithm are significantly better than, worse than, and similar to the ones obtained by HEIA using Wilcoxon’s rank sum test with a significant level $\alpha=0.05$.

HEIA-I utilizes the first evolutionary strategy (SBX and polynomial-based mutation) and HEIA-II adopts the second one (DE crossover and polynomial-based mutation). In Tables XI and XII, a comparison of results among HEIA-I, HEIA-II and HEIA on the ZDT and the UF test problems is provided. It can be seen from Table XI that HEIA-I is better than HEIA-II in solving all the ZDT problems. Particularly, on the ZDT4 problem, HEIA-I can obtain a uniform approximation set close to PF_{true} ; however, HEIA-II is unable to reach PF_{true} as its IGD value on ZDT4 is larger than 7 (the ideal IGD value is zero). These experiments indicate that the first evolutionary strategy is more capable for tackling the ZDT problems. Moreover, the cooperation of the two evolutionary strategies consistently performs better on all the ZDT problems, which validates our hypothesis that each evolutionary strategy can enhance the search capabilities of the other one. In Table XII, HEIA performed best on 6 out of 7 of the UF test problems. Only for UF3, HEIA performed worse than HEIA-II. For the rest of the UF test problems, it is shown that the cooperation of the two evolutionary strategies in HEIA can significantly enhance the performance of HEIA-I and HEIA-II. For example, for the UF1, UF2 and UF7 test problems, HEIA obtained a significantly better performance and its corresponding IGD values were all under an accuracy level of 10^{-3} , which indicates that HEIA was able to find good approximation sets for these problems. This further confirms the effectiveness of our hybrid framework and that the cooperation of the hybrid evolutionary strategies is beneficial for enhancing their search capabilities.

G. Analysis of the Parameters Settings

The parameters settings adopted in HEIA are listed in Table I. In order to study their effect on the performance of HEIA, we conducted an experimental study of one-at-a-time sensitivity analysis. As the parameters settings in the SBX and the polynomial-based mutation operators have been substantially investigated in [28], their analysis is not repeated here. The effect of parameter N (population size) is quite evident: under the

TABLE XII

COMPARISON OF RESULTS WITH RESPECT TO HEIA-I, HEIA-II AND HEIA ON THE UF TEST PROBLEMS

Problems	Algorithms	HEIA-I	HEIA-II	HEIA
UF1	Mean	8.31E-02	4.15E-02	2.70E-03
	Std	2.81E-02	1.50E-02	1.42E-04
	Rank	3-	2-	1
UF2	Mean	3.44E-02	1.74E-02	5.81E-03
	Std	1.17E-02	5.82E-03	4.98E-04
	Rank	3-	2-	1
UF3	Mean	2.15E-01	8.41E-03	1.28E-02
	Std	4.99E-02	5.34E-03	6.10E-03
	Rank	3-	1+	2
UF4	Mean	4.12E-02	6.13E-02	3.77E-02
	Std	4.47E-04	5.41E-03	7.49E-04
	Rank	2-	3-	1
UF5	Mean	2.79E-01	6.27E-01	2.05E-01
	Std	7.64E-02	1.39E-01	1.17E-01
	Rank	2-	3-	1
UF6	Mean	2.31E-01	3.21E-01	1.52E-01
	Std	1.45E-01	8.29E-02	8.47E-02
	Rank	2-	3-	1
UF7	Mean	1.44E-01	2.78E-02	3.09E-03
	Std	1.56E-01	1.01E-02	5.56E-04
	Rank	3-	2-	1
Rank Sum		18	16	8
Final Rank		3	2	1
<i>better/worse/similar</i>		0/7/0	1/6/0	/

“+”, “-”, and “ \approx ” indicate the results obtained by the algorithm are significantly better than, worse than, and similar to the ones obtained by HEIA using Wilcoxon’s rank sum test with a significant level $\alpha=0.05$.

same number of generations, a larger value of N will produce a better performance. The neighbor size T is chosen depending on the value of N and the number of the objectives. For the rest of the parameters, such as NA , δ , CR and F , they are respectively studied in the following subsections to investigate their influence on HEIA. Five representative values for NA (*i.e.*, 20, 40, 60, 80 and 100), δ (*i.e.*, 0.1, 0.3, 0.5, 0.7 and 0.9), CR (*i.e.*, 0.2, 0.4, 0.6, 0.8 and 1.0) and F (*i.e.*, 0.1, 0.3, 0.5, 0.7 and 0.9) were adopted to solve three test problems with different types of difficulties, *i.e.*, ZDT1, WFG1 and UF1. Thus, there is a total of 625 combinations of NA , δ , CR and F values for each test problem and other parameters settings were the same as those listed in Table I except that the population sizes are all set to 100 for ZDT1, WFG1 and UF1. All the simulations have been repeated 100 times for each combination and their corresponding mean IGD results are illustrated using box plots [62], in which the central red line indicates the median value, the edges of the box are the 25th and 75th percentiles, and the red symbol “+” denotes outliers. Finally, the Kolmogorov-Smirnov test with a 5% significance level was further used to detect the statistical differences between two samples in each plot. It is noted that a black star * above the box indicates that the corresponding sample has statistical differences with other samples.

1) NA : The number of selected antibodies for cloning

When performing the cloning operator in (6), only NA antibodies with high affinities are selected from the external archive. Fig. 6 shows the distribution of IGD results on ZDT1, WFG1 and UF1, which were obtained by HEIA with different NA values from 625 combinations. For the ZDT1 problem, it is observed that the IGD results are significantly increased with respect to the NA values, so that a smaller value of NA is better

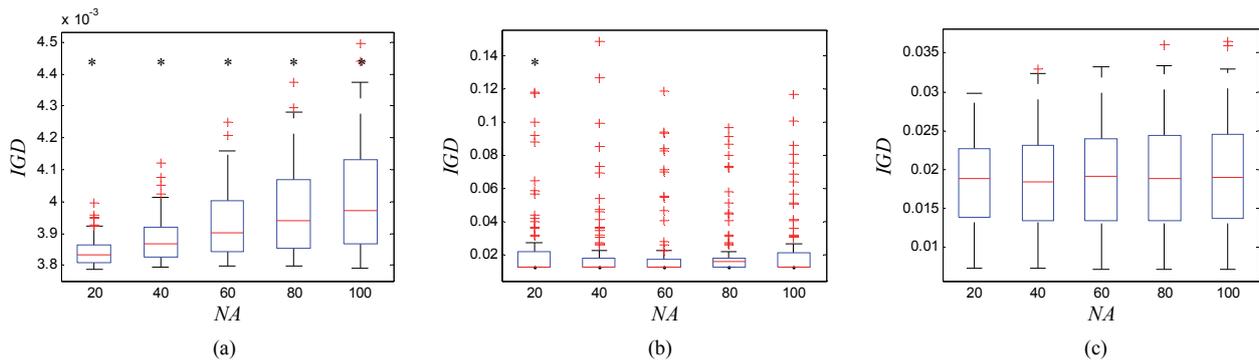


Fig. 6 Box plots of the IGD results obtained by HEIA with different NA values on (a) ZDT1, (b) WFG1 and (c) UF1

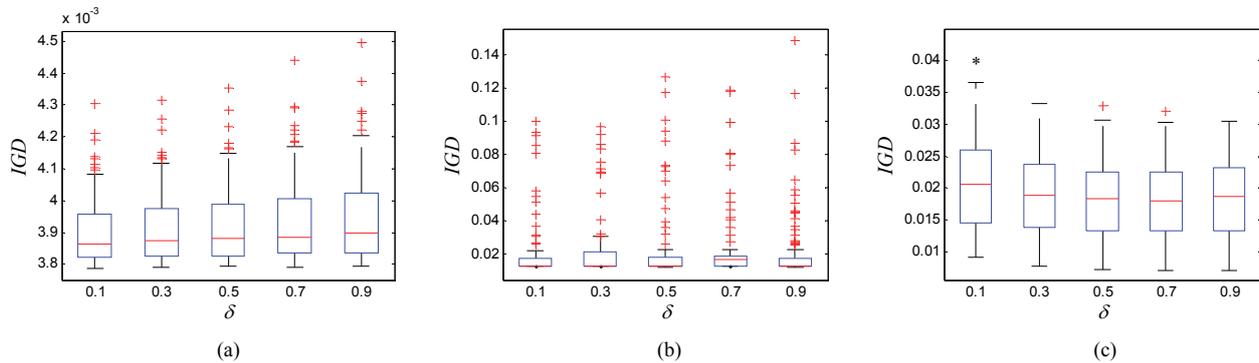


Fig. 7 Box plots of the IGD results obtained by HEIA with different δ values on (a) ZDT1, (b) WFG1 and (c) UF1

for ZDT1. This is mainly because more cones can be assigned to a small proportion of high-affinity antibodies under a fixed population size, which helps to accelerate the convergence speed. When solving more difficult problems such as WFG1 and UF1, most of the IGD results are statistically similar, so that the impact of NA is less significant on the WFG1 and UF1 problems. Therefore, if we consider all the test problems, a small value of NA (e.g., between 20 and 40) is suggested.

2) δ : A control parameter for selecting the parent individuals in the DE operator

As introduced in Section III.B, two parents are selected from population P in order to perform DE crossover. P can be specified as A (the selected antibody set for cloning) or the neighbors of \mathbf{x} , which is controlled by the probability parameter δ . When P is set as A , DE crossover can perform global search as the antibodies in A are the least-crowded ones, which cover most of the currently-found approximated front. Otherwise, P is assigned by the neighbors of \mathbf{x} , which can search in the local area around \mathbf{x} .

Fig. 7 presents the box plots of the IGD results obtained by HEIA with different δ values on ZDT1, WFG1 and UF1. The Kolmogorov-Smirnov test shows that all the IGD results plotted in Figs. 7(a)-(c) do not have a significant difference in solving ZDT1, WFG1 and UF1. These experimental results indicate that the performance of HEIA is not so sensitive to δ on ZDT1, WFG1 and UF1.

3) CR and F : two control parameters in the DE operator

The crossover rate CR and the scaling factor F are two important control parameters when generating new antibodies in (18), where CR controls how many variables are inherited from the mutant vectors and F adjusts the mutation scale.

The box plots of the IGD results obtained by HEIA with different values of CR and F are illustrated in Figs. 8-9. From Figs. 8(a)-(c) we can observe that a smaller value of CR can give significantly better performance for solving ZDT1, while a larger value of CR performs significantly better on WFG1 and UF1. This is reasonable as ZDT1 is a simple test problem, in which a smaller CR value can make offspring to inherit more information from the parents, thus speeding up convergence. However, the use of more inheritance from the parents will lower population's diversity, which makes HEIA unsuitable for difficult problems such as WFG1 and UF1. Thus, the performance of HEIA is sensitive to the selection of CR when tackling different kinds of MOPs. On the other hand, the IGD results shown in Figs. 9(a)-(c) indicate that a larger value of F is preferred for ZDT1 and WFG1, while a smaller value of F is more suitable for UF1. This is also supported by the Kolmogorov-Smirnov test, which indicates that HEIA performs significantly worse on ZDT1 and WFG1, but significantly better on UF1 when the F value is set to 0.1. Therefore, the performance of HEIA is also sensitive to the F values when solving various kinds of MOPs. That is to say, no fixed parameter settings of F and CR are guaranteed to always perform well. To have a good overall performance, intermediate values

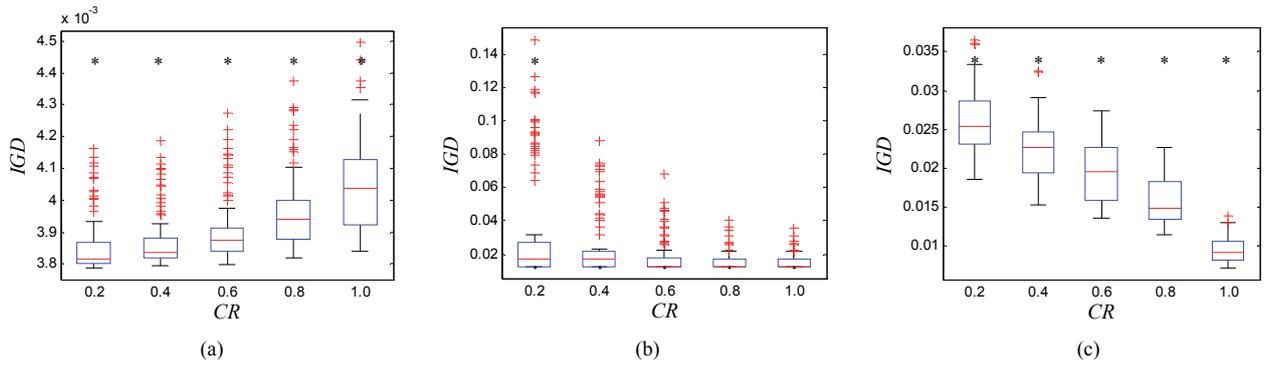


Fig. 8 Box plots of the IGD results obtained by HEIA with different CR values on (a) ZDT1, (b) WFG1 and (c) UF1

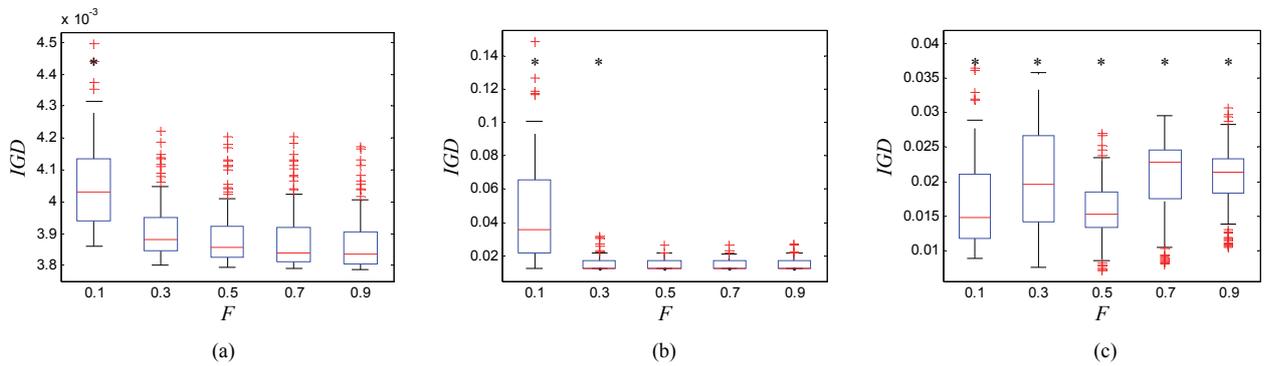


Fig. 9 Box plots of the IGD results obtained by HEIA with different F values on (a) ZDT1, (b) WFG1 and (c) UF1

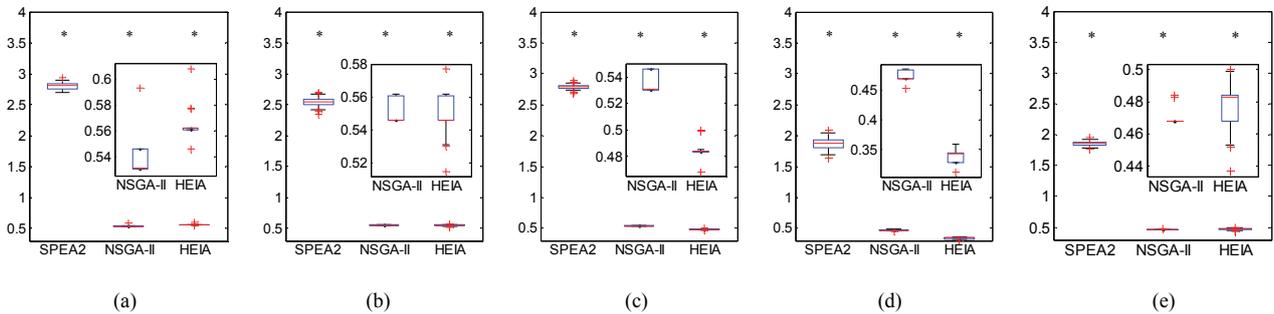


Fig. 10 Mean computational times (s) obtained by NSGA-II, SPEA2 and HEIA on (a) ZDT1, (b) ZDT2, (c) ZDT3, (d) ZDT4, and (e) ZDT6

of CR and F are suggested when considering all the test problems adopted in this paper.

H. Time Complexity Analysis

In this subsection, the time complexity analysis and the practical implementation efficiency of HEIA are investigated. The aim is to show that the hybrid framework proposed in this paper will not have a significant negative effect on execution efficiency. Based on the pseudo-code of HEIA (**Algorithm 3**), the time complexity of HEIA is mainly determined by the evolutionary loop in lines 2-18. When calculating the time complexity, the impact of the decision variables and the objectives are ignored as they are much smaller than the population size N . In lines 3-4, the time complexity is $O(N \log N + NA)$ for the sorting procedure and for picking out NA high-affinity antibodies; in lines 5-14, the selected high-affinity antibodies are cloned and randomly divided into two subpopulations,

which take the time complexity $O(N)$; in lines 15-16, two evolutionary strategies are respectively performed on the two subpopulations and the corresponding time complexity is $O(2N)$; at last, the fine-grained selection mechanism is operated with time complexity $O(N^2)$. Therefore, the total time complexity of HEIA is $O(N^2 + N \log N + NA + 3N) \sim O(N^2)$. Note that the time complexity of NSGA-II [2], SPEA2 [3], NNIA [24], NICA [23] and MIMO [25] are respectively $O(N^2)$, $O(N^3)$, $O(N^2)$, $O(N^2)$ and $O(N^2)$. Therefore, HEIA has a time complexity which is comparable with that of two state-of-the-art MOEAs (NSGA-II and SPEA2) and three MOIAs (NNIA, NICA and MIMO).

We further investigated the practical implementation efficiency of HEIA with NSGA-II and SPEA2 as they all use Pareto-dominance-based comparisons and diversity maintenance strategies on the combined population formed by the offspring and the elitist archive. Fig. 10 shows the box plots of

the computational times obtained by NSGA-II, SPEA2, and HEIA on all the ZDT problems with 25000 function evaluations. All the algorithms were implemented on the Java environment [61] and were run 100 times on the same computer with dual-core 3.2 GHz CPU, 2Gbytes of RAM and Windows 7 operation system. The parameters settings were those listed in Table I. It is noted that the Kolmogorov-Smirnov test with a 5% significance level was also used to detect the statistical differences between two samples in each plot and the sample owning the statistical differences with other samples was identified by a black star * above the box.

From Fig. 10, we can observe that HEIA performed faster than NSGA-II on ZDT2, ZDT3 and ZDT4 with statistical differences. This is mainly because NSGA-II needs to find multiple ranks of nondominated solutions while HEIA only finds out the first rank of nondominated solutions using a fine-grained selection mechanism. Besides that, they all use the crowding-distance metric [2] to maintain the population diversity. SPEA2 has a slower running speed than that of HEIA and NSGA-II, as it has to calculate the distance of the nearest neighbor to keep the population diversity. It is very time-consuming to find the nearest neighbor for each solution using Euclidean distances. For all the ZDT problems, HEIA only needs less than 0.6s to execute 25000 function evaluations. These experiments indicate that HEIA has a promising running speed. Moreover, as HEIA enables different subpopulations to evolve separately, it can be implemented in a distributed parallel computing architecture. Therefore, the running speed of HEIA can be enhanced greatly, which makes it more useful for practical applications.

V. CONCLUSIONS

In this paper, a hybrid evolutionary framework for MOIAs (HEIA) is introduced. In the proposed framework, multiple evolutionary strategies are applied after cloning. The cooperation of multiple evolutionary strategies is able to enhance the capabilities and the robustness of the proposed approach, allowing it to handle different types of MOPs. An implementation example of this hybrid framework is provided using two different evolutionary strategies, which adopt either SBX or DE crossover followed by polynomial-based mutation, respectively. Simulation results showed that HEIA is capable of successfully handling different types of MOPs, *i.e.*, the ZDT, WFG, UF and DTLZ test problems. When compared with three MOIAs (*i.e.*, NICA, NNIA, and MIMO) and other nature-inspired multi-objective algorithms (*i.e.*, NSGA-II, SPEA2, AbYSS, MOEA/D, and SMP SO), HEIA was found to present advantages in terms of finding a solution set with better convergence and diversity to approach PF_{true} . Moreover, the effectiveness of our hybrid framework has justified that two different search strategies (SBX and DE in our case) can complement each other, thus performing better than the isolated use of any of them. The reason for this behavior may be that the search patterns of SBX and DE are indeed complementary, which would allow their search outputs to be successfully shared through the use of cloning. Finally, the influence that its parameters have on the performance of our proposed HEIA was

also studied (experimentally) in the paper.

Although promising results were obtained using HEIA, there are still several issues worth studying for further improvements. Our future research work will continue on the following directions: 1) study the possibility to integrate other evolutionary operators into our hybrid framework, such as simplex crossover, parent centric crossover, Gaussian mutation and Cauchy mutation; 2) design an adaptive approach to dynamically assign different computational resources to different subpopulations, based on their historical performance; 3) extend HEIA to solve MOPs with more than 3 objectives [63]-[64] or in a noisy environments [65], and apply it to real-world applications [66].

REFERENCES

- [1] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, Inc., New York, USA, 2001.
- [2] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.
- [3] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: improving the strength Pareto evolutionary algorithm," Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, Technical report 103, 2001.
- [4] Q.F. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, 2007.
- [5] M. Drozdik, Y. Akimoto, H. Aguirre, and K. Tanaka, "Computational Cost Reduction of Nondominated Sorting Using the M-Front," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 659–678, 2015.
- [6] X.Y. Zhang, Y. Tian, R. Cheng, and Y.C. Jin, "An Efficient Approach to Nondominated Sorting for Evolutionary Multiobjective Optimization Evolutionary Computation," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 201–213, 2015.
- [7] K.C. Tan, C.K. Goh, A.A. Mamun, and E.Z. Ei, "An evolutionary artificial immune system for multi-objective optimization," *Eur. J. Oper. Res.*, vol. 187, no. 2, pp. 371–392, 2008.
- [8] K.C. Tan, S.C. Chiam, A.A. Mamun, and C.K. Goh, "Balancing exploration and exploitation with adaptive variation for evolutionary multi-objective optimization," *Eur. J. Oper. Res.*, vol. 197, no. 2, pp. 701–713, 2009.
- [9] S.F. Adra, T.J. Dodd, I.A. Griffin, and P.J. Fleming, "Convergence acceleration operator for multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 4, pp. 825–847, 2009.
- [10] Z.W. Yu, H.S. Wong, D.W. Wang, and M. Wei, "Neighborhood knowledge-based evolutionary algorithm for multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 15, no. 6, pp. 812–831, 2011.
- [11] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Diversity improvement by non-Geometric binary crossover in evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 14, no. 6, pp. 985–998, 2010.
- [12] Z.H. Zhan, J.J. Li, J.N. Cao, J. Zhang, H.S.H. Chung, and Y.H. Shi, "Multiple populations for multiple objective a coevolutionary technique for solving multiobjective optimization problems," *IEEE Trans. Cybernetics*, vol. 43, no. 2, pp. 445–463, 2013.
- [13] K. Li, A. Fialho, S. Kwong, and Q.F. Zhang, "Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 114–140, 2014.
- [14] S.B. Gee, K.C. Tan, V.A. Shim, and N.R. Pal, "Online Diversity Assessment in Evolutionary Multiobjective Optimization: A Geometrical Perspective," *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 542–559, 2015.
- [15] L. Bezerra, M. Lopez-Ibanez, and T. Stuetzle, "Automatic Component-Wise Design of Multi-Objective Evolutionary Algorithms," *IEEE Trans. Evol. Comput.*, DOI: 10.1109/TEVC.2015.2474158, 2015.
- [16] J. Branke, S. Greco, R. Slowinski, and P. Zielniewicz, "Learning Value Functions in Interactive Evolutionary Multiobjective Optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 1, pp. 88–102, 2015.

- [17] N. Chen, W. Chen, Y. Gong, Z.-H. Zhan, J. Zhang, Y. Li, and Y. S. Tan, "An evolutionary algorithm with double-level archives for multiobjective optimization," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 1851–1863, 2015.
- [18] K.C. Tan, Y.J. Yang, and C.K. Goh, "A distributed cooperative coevolutionary algorithm for multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 527–549, 2006.
- [19] C.K. Goh and K.C. Tan, "A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 1, pp. 103–127, 2009.
- [20] A.J. Nebro, F. Luna, E. Alba, B. Dorronsoro, J.J. Durillo, and A. Beham, "AbYSS: Adapting scatter search to multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 12, no. 4, pp. 439–457, 2008.
- [21] W. Hu and G.G. Yen, "Adaptive Multiobjective Particle Swarm Optimization Based on Parallel Cell Coordinate System," *IEEE Trans. Evol. Comput.*, vol. 19, no. 1, pp. 1–18, 2015.
- [22] K.-B. Lee and J.-H. Kim, "Multiobjective Particle Swarm Optimization With Preference-Based Sort and Its Application to Path Following Footstep Optimization for Humanoid Robots," *IEEE Trans. Evol. Comput.*, vol. 17, no. 6, pp. 755–766, 2013.
- [23] R.H. Shang, L.C. Jiao, F. Liu, and W.P. Ma, "A novel immune clonal algorithm for MO problems," *IEEE Trans. Evol. Comput.*, vol. 16, no. 1, pp. 35–50, 2012.
- [24] M.G. Gong, L.C. Jiao, H.F. Du, and L.F. Bo, "Multi-objective immune algorithm with nondominated neighbor-based selection," *Evol. Comput.*, vol. 16, no. 2, pp. 225–255, 2008.
- [25] Q.Z. Lin and J.Y. Chen, "A novel micro-population immune multiobjective optimization algorithm," *Comput. Oper. Res.*, vol. 40, no. 6, pp. 1590–1601, 2013.
- [26] Q.F. Zhang, A.M. Zhou, S.Z. Zhao, P.N. Suganthan, W.D. Liu, and S. Tiwari, "Multiobjective optimization test instances for the CEC 2009 special session and competition," The School of Computer Science and Electronic Engineering, University of Essex, *Technical Report CES-487*, 2009.
- [27] P.A.N. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 174–188, 2003.
- [28] K. Deb and H.G. Beyer, "Self-adaptive genetic algorithms with simulated binary crossover," *Evol. Comput.*, vol. 9, no. 2, pp. 197–221, 2001.
- [29] H. Li and Q.F. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284–302, 2009.
- [30] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2000.
- [31] S. Huband, L. Barone, R. While, and P. Hingston, "A scalable multi-objective test problem toolkit," in *Proc. 3rd Conf. Evol. Multi-Criterion Optimiz.*, *Lecture Notes in Computer Science*, vol. 3410, 2005, pp. 280–295.
- [32] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multi-objective optimization," in *evolutionary multi-objective optimization, advanced information and knowledge processing series*. Berlin Springer, 2005, pp. 105–145.
- [33] A.J. Nebro, J.J. Durillo, J. Garcia-Nieto, C.A. Coello Coello, F. Luna, and E. Alba, "SMPSO: A new PSO-based metaheuristic for multi-objective optimization," in *Proc. IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making*, 2009, pp. 66–73.
- [34] L.N. De Castro and J. Timmis, *Artificial Immune Systems: A New Computational Intelligence Approach*. London, U. K.: Springer, 2002.
- [35] L.N. De Castro and F.J. Von Zuben, "Learning and optimization using the clonal selection," *IEEE Trans. Evol. Comput.*, vol. 6, no. 3, pp. 239–251, 2002.
- [36] J. Yoo and P. Hajela, "Immune network simulations in multicriterion design," *Struct. Optim.*, vol. 18, no. 2-3, pp. 85–94, 1999.
- [37] C.A. Coello Coello and C.N. Cruz, "An approach to solve multiobjective optimization problems based on an artificial immune system," in *Proc. 1st Conf. on Artif. Immune Syst.*, 2002, pp. 212–221.
- [38] C.N. Cruz and C.A. Coello Coello, "Multiobjective optimization using ideas from the clonal selection principle," in *Proc. 5th Annual Gene. and Evol. Comput. Conf.*, 2003, pp. 158–170.
- [39] C.N. Cruz and C.A. Coello Coello, "Using artificial immune systems to solve optimization problems," in *Proc. Workshop Program of Genetic and Evolutionary Computation Conference*, 2003, pp. 312–315.
- [40] C.A. Coello Coello and C.N. Cruz, "Solving multiobjective optimization problems using an artificial immune system," *Genet. Program. Evol. Mach.*, vol. 6, no. 2, pp. 163–190, 2005.
- [41] L.C. Jiao, M.G. Gong, R.H. Shang, H.F. Du, and B. Lu, "Clonal selection with immune dominance and energy based multiobjective optimization," in *Proc. 3rd Conf. Evol. Multi-Criterion Optimiz.*, *Lecture Notes in Computer Science*, vol. 3410, 2005, pp. 474–489.
- [42] R.H. Shang, L.C. Jiao, M.G. Gong, and B. Lu, "Clonal selection algorithm for dynamic multiobjective optimization," in *Proc. Conf. Comput. Intelligence and Security*, 2005, pp. 846–851.
- [43] Z.H. Hu, "A multiobjective immune algorithm based on a multiple-affinity model," *Eur. J. Oper. Res.*, vol. 202, no. 1, pp. 60–72, 2010.
- [44] F. Freschi and M. Repetto, "Multi-objective optimization by a modified artificial immune system algorithm," in *Proc. 4th Conf. Artif. Immune Syst.*, vol. 3627, 2005, pp. 248–261.
- [45] F. Freschi and M. Repetto, "VIS: an artificial immune network for multi-objective optimization," *Eng. Optimiz.*, vol. 38, no. 8, pp. 975–996, 2006.
- [46] L.N. De Castro and J. Timmis, "An artificial immune network for multimodal function optimization," in *Proc. IEEE Congr. Evol. Comput.*, vol. 1, 2002, pp. 699–704.
- [47] J.Q. Gao and J. Wang, "WBMOAIS: a novel artificial immune system for multiobjective optimization," *Comput. Oper. Res.*, vol. 37, no. 1, pp. 50–61, 2010.
- [48] V. Cutello, G. Narzisi, and G. Nicosia, "A class of pareto archived evolution strategy algorithms using immune inspired operators for ab initio protein structure prediction," in *Lecture Notes in Computer Science*, vol. 3449. Lausanne, Switzerland: Springer, 2005, pp. 54–63.
- [49] K.C. Tan, C.K. Goh, A.A. Mamun, and E.Z. Ei, "An evolutionary artificial immune system for multi-objective optimization," *Eur. J. Oper. Res.*, vol. 187, no. 2, pp. 371–392, 2008.
- [50] E.Y.C. Wong, H.S.C. Yeung, and H.Y.K. Lau, "Immunity-based hybrid evolutionary algorithm for multi-objective optimization in global container repositioning," *Eng. Appl. Artif. Intell.*, vol. 22, no. 6, pp. 842–854, 2009.
- [51] Z.H. Zhang, "Immune optimization algorithm for constrained nonlinear multiobjective optimization problems," *Appl. Soft Comput.*, vol. 7, no. 3, pp. 840–857, 2007.
- [52] Z.H. Zhang, "Multiobjective optimization immune algorithm in dynamic environments and its application to greenhouse control," *Appl. Soft Comput.*, vol. 8, no. 2, pp. 959–971, 2008.
- [53] J.Y. Chen, Q.Z. Lin, and Z. Ji, "A hybrid immune multiobjective optimization algorithm," *Eur. J. Oper. Res.*, vol. 204, no. 2, pp. 294–302, 2010.
- [54] Q.Z. Lin, Q.L. Zhu, P.Z. Huang, J.Y. Chen, Z. Ming, and J.P. Yu, "A novel hybrid multi-objective immune algorithm with adaptive differential evolution," *Comput. Oper. Res.*, vol. 65, pp. 95–111, 2015.
- [55] T. Jansen and C. Zarges, "Reevaluating Immune-Inspired Hypermutations Using the Fixed Budget Perspective," *IEEE Trans. Evol. Comput.*, vol. 18, no. 5, pp. 674–688, 2014.
- [56] Z.P. Liang, R.Z. Song, Q.Z. Lin, Z.H. Du, J.Y. Chen, Z. Ming, and J.P. Yu, "A double-module immune algorithm for multi-objective optimization problems," *Appl. Soft Comput.*, vol. 35, pp. 161–174, 2015.
- [57] K. Sindhya, K. Miettinen, and K. Deb, "A hybrid framework for evolutionary multi-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 4, pp. 495–511, 2013.
- [58] L.X. Tang and X.P. Wang, "A hybrid multiobjective evolutionary algorithm for multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 17, no. 1, pp. 20–46, 2013.
- [59] Q.F. Zhang, W.D. Liu, E. Tsang, and B. Virginas, "Expensive multi-objective optimization by MOEA/D with gaussian process model," *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 456–474, 2010.
- [60] S. Das and P.N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, 2011.

- [61] J.J. Durillo and A.J. Nebro, "jMetal: a Java framework for multi-objective optimization," *Adv. Eng. Softw.*, vol. 42, no. 10, pp. 760–771, 2011.
- [62] R. McGill, J.W. Tukey, and W.A. Larsen, "Variations of Box plots," *Am. Stat.*, vol. 32, no. 1, pp. 12–16, 1978.
- [63] S. Bandyopadhyay and A. Mukherjee, "An Algorithm for Many-Objective Optimization With Reduced Objective Computations: A Study in Differential Evolution," *IEEE Trans. Evol. Comput.*, vol. 19, no. 3, pp. 400–413, 2015.
- [64] K. Deb and H. Jain, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints Evolutionary Computation," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, 2014.
- [65] V.A. Shim, K.C. Tan, J.Y. Chia, and A.A.I. Mamun, "Multi-objective optimization with estimation of distribution algorithm in a noisy environment," *Evol. Comput.*, vol. 21, no. 1, pp.149–177, 2013.
- [66] Z.-H. Zhan, X. F. Liu, Y. J. Gong, J. Zhang, H. S. H. Chung, and Y. Li, "Cloud computing resource scheduling and a survey of its evolutionary approaches," *ACM Comput. Surv.*, vol. 47, no. 4, pp. 1–33, 2015.



Qiuzhen Lin received the B.S. degree from Zhaoqing University and the M.S. degree from Shenzhen University, China, in 2007 and 2010, respectively. He received the Ph.D. degree from Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong, in 2014.

He is currently a Lecturer in College of Computer Science and Software Engineering, Shenzhen University. He has published over 20 research papers since 2008. His current research interests include artificial immune system, multi-objective optimization and dynamic system.



Jianyong Chen is a professor in College of Computer Science and Software Engineering, Shenzhen University. He got his PhD from City University of Hong Kong, Hong Kong, China, in 2003. He is interested in Artificial Intelligence and Information Security.

He worked for ZTE Corporation as senior engineer of network technology from 2003 to 2006. After that, he joined the Shenzhen University. He was vice-chairman of International Telecommunication Union-Telecommunication (ITU-T) SG17 from 2004 to 2012, and editor of three recommendations developed in ITU-T SG17. He has published more than 30 papers and got more than 30 patents in the field of Artificial Intelligence and Information Security.



Zhi-Hui Zhan (M'13) received the Bachelor's degree and the Ph. D degree in 2007 and 2013, respectively, from the Department of Computer Science of Sun Yat-Sen University, Guangzhou, China.

He has published over 50 international journal and conference papers, including about 20 *IEEE Transactions* and *Magazines* papers. His current research interests include evolutionary computation, cloud computing, and evolutionary computation applications in real-world problems.

Dr. Zhan's doctoral dissertation was awarded the China Computer Federation Outstanding Dissertation in 2013. Dr. Zhan received the Natural Science Foundation for Distinguished Young Scientists of Guangdong Province, China in 2014 and was awarded the Pearl River New Star in Science and Technology in 2015. Dr. Zhan is listed as one of the Most Cited Chinese Researchers in Computer Science.



Wei-Neng Chen (S'07-M'12) received the Bachelor's degree and the Ph.D. degree from the Department of Computer Science of Sun Yat-sen University, Guangzhou, China, in 2006 and 2012, respectively.

His doctoral thesis received the IEEE Computational Intelligence Society (CIS) Outstanding Dissertation Award in 2016 and the China Computer Federation (CCF) Outstanding Dissertation in 2012. He also received Natural Science Foundation for Distinguished Young Scientists of Guangdong Province, China, in 2015, the "Guangdong Special Support Program" for Outstanding Young Scientists in 2015, and the Pearl River New Star in Science and Technology in 2014. He has published 50 papers in international journals and conferences. His current research interests include swarm intelligence algorithms and their applications.



Carlos A. Coello Coello received PhD degree in computer science from Tulane University, USA, in 1996. He is currently Professor (CINVESTAV-3F Researcher) at the Computer Science Department of CINVESTAV-IPN, in Mexico City, México. Dr. Coello has authored and co-authored over 450 technical papers and book chapters. He has also co-authored the book *Evolutionary Algorithms for Solving Multi-Objective Problems* (Second Edition, Springer, 2007). His publications currently report over 29,000 citations in Google Scholar (his h-index is 67). Currently, he is associate editor of the *IEEE Transactions on Evolutionary Computation* and serves in the editorial board of 12 other international journals. His major research interests are: evolutionary multi-objective optimization and constraint-handling techniques for evolutionary algorithms. He received the 2007 *National Research Award* from the Mexican Academy of Sciences in the area of *Exact Sciences*, the 2013 *IEEE Kiyo Tomiyasu Award* and the 2012 *National Medal of Science and Arts* in the area of *Physical, Mathematical and Natural Sciences*. He is a Fellow of the IEEE, and a member of the ACM, Sigma Xi, and the Mexican Academy of Science.



Yilong Yin received his PhD in 2000 from Jilin University, China. From 2000 to 2002, he worked as a post-doctoral fellow in the Department of Electronics Science and Engineering, Nanjing University, China. He is currently the Director of MLA Group and a Professor of the School of Computer Science and Technology, Shandong University, China. His research interests include machine learning, data mining, and computational medicine.



Chih-Min Lin (Fellow'10) was born in Taiwan, in 1959. He received the B.S. and M.S. degrees from Department of Control Engineering and the Ph.D. degree from Institute of Electronics Engineering, National Chiao Tung University, Hsinchu, Taiwan, in 1981, 1983 and 1986, respectively.

He is currently a Chair Professor and the Dean of the College of Electrical and Communication Engineering, Yuan Ze University, Chung-Li, Taiwan. His current research interests include fuzzy neural network, cerebellar model articulation controller, intelligent control systems and signal processing. He has published 166 journal papers and 167 conference papers.

Dr. Lin was the recipient of an Honor Research Fellow at the University of Auckland, Auckland, New Zealand, from 1997 to 1998. He also serves as an Associate Editor of *IEEE Transactions on Cybernetics*, *IEEE Transactions on Fuzzy Systems*, *International Journal of Fuzzy Systems* and *International Journal of Machine Learning and Cybernetics*. He is also an IET fellow.



Jun Zhang (M'02–SM'08) received the Ph.D. degree under Prof. Henry Chung's supervision in Electrical Engineering from the City University of Hong Kong, Kowloon, Hong Kong, in 2002. Since 2004, he has been with Sun Yat-Sen University, Guangzhou, China, where he is a Changjiang Scholars Professor. He has authored seven research books and book chapters, and over 100 technical papers in his research areas. His current research interests include computational intelligence, cloud computing, high performance computing, data mining, wireless sensor networks, operations research, and power electronic circuits.

Dr. Zhang was a recipient of the China National Funds for Distinguished Young Scientists from the National Natural Science Foundation of China in 2011 and the First-Grade Award in Natural Science Research from the Ministry of Education, China, in 2009. He is currently an Associate Editor of the *IEEE Transactions on Evolutionary Computation*, the *IEEE Transactions on Industrial Electronics*, and the *IEEE Transactions on Cybernetics*. He is the Founding and Current Chair of the IEEE Guangzhou Subsection and IEEE Beijing (Guangzhou) Section Computational Intelligence Society Chapters. He is the Founding and Current Chair of the ACM Guangzhou Chapter.