

An Adaptive Immune-inspired Multi-objective Algorithm with Multiple Differential Evolution Strategies

Qiuzhen Lin¹, Yueping Ma¹, Jianyong Chen^{1*}, Qingling Zhu¹, Carlos A. Coello Coello², Ka-Chun Wong³, Fei Chen^{1*}

¹*College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, PR. China,*

²*CINVESTAV-IPN, Department of Computer Science, Mexico, D.F., 07360, Mexico*

³*Department of Computer Science, City University of Hong Kong, Hong Kong, PR. China*

Abstract:

Most multi-objective immune algorithms (MOIAs) adopt clonal selection to speed up convergence, as this operator only clones the best individuals during the search process. However, this approach somehow deteriorates the population diversity, which may cause a MOIA to be trapped in a local optimum and could also lead to premature convergence when tackling some complicated multi-objective optimization problems (MOPs). In order to overcome this problem, an adaptive immune-inspired multi-objective algorithm (AIMA) is presented in this paper, in which multiple differential evolution (DE) strategies having distinct advantages are embedded into a conventional MOIA. Our proposed approach strengthens the exploration capabilities of a MOIA while also improving its population diversity. At each generation, and based on the current search stage, an adaptive selection method is designed to choose an appropriate DE strategy for evolution. The core idea is to effectively combine the advantages of three DE strategies when solving different MOPs. A number of comparative experiments are conducted on the well-known and frequently-used WFG and DTLZ test problems. Our experimental results validate the superiority of our proposed AIMA, as it performs better than some state-of-the-art multi-objective optimization algorithms and some state-of-the-art MOIAs.

Keywords: Multi-objective optimization; Immune algorithm; Differential evolution; Adaptive strategy selection

1. Introduction

* Corresponding author
Email address: jyichen@szu.edu.cn (J.Y. Chen), fchen@szu.edu.cn (F. Chen)

Optimization problems widely exist in scientific research and engineering applications. Based on the number of objectives to be optimized, they are generally classified into single-objective optimization problems (SOPs) and multi-objective optimization problems (MOPs). This paper focuses on tackling MOPs, which give rise to several challenges due to the aim of optimizing several (often conflicting) objectives simultaneously. There is a wide variety of real-world MOPs, such as route planning [39], job shop scheduling [20], and data classification [1]. Due to the conflict among the objectives, the optimization of a MOP generates a set of solutions representing the best possible trade-off among all the objectives, which compose the so-called Pareto-optimal set (*PS*). The corresponding mapping of *PS* in objective space is termed Pareto-optimal front (*PF*). Without any further preference information, the goal of multi-objective optimization is to produce a set of solutions, which approximate the true *PF* as close as possible and are distributed along the true *PF* as uniformly as possible.

Due to the population-based nature of evolutionary algorithms (EAs), they are very suitable for tackling MOPs since they can process a set of solutions in a single run. During the recent decades, a number of multi-objective EAs (MOEAs) have been designed, showing a very promising performance on tackling different MOPs. There are three well-known representatives of state-of-the-art MOEAs, i.e., NSGA-II [12], SPEA2 [56], and MOEA/D [52]. Regarding NSGA-II [12], it incorporates a fast nondominated sorting approach to direct the search, while a crowding-distance metric is used to maintain the population's diversity. On SPEA2 [56], a nearest neighbor density estimation technique is proposed to maintain the population's diversity, combined with a fine-grained fitness assignment strategy that is used to guide the search. For MOEA/D [52], a MOP is decomposed into a set of SOPs and then these SOPs are solved in a cooperative manner using evolutionary search. These state-of-the-art MOEAs have inspired many enhanced variants. For example, regarding NSGA-II, a novel parent inheritance operator was embedded and several jumping gene adaptations have been used in [35] to speed up convergence towards the global PF, while a reference point based approach was introduced in [13] to maintain the population's diversity when tackling many-objective optimization problems (*i.e.*, MOPs having more than three objectives); for SPEA2, a shift-based density estimation (SDE) strategy [29] was presented to enhance its performance on tackling many-objective optimization problems; with respect to MOEA/D, a dynamic resource allocation (DRA) strategy was introduced in [53] to dynamically assign the computational resources based on the difficulties of sub-problems, and an economic stable matching model (STM) was designed in [28] to guarantee the balanced match of sub-problems and solutions by mutual preferences. A detailed review of MOEAs can be found in [36]. Especially, some of MOEAs were enhanced based on the use of differential evolution (DE), since DE shows excellent search capabilities. The experiments conducted by [4] and [44] showed that DE can significantly enhance the performance of MOEAs, as the

DE-based variants of three state-of-the-art MOEAs (i.e., NSGA-II, SPEA2 and IBEA) significantly outperformed the original ones. In MOEA/D-DE [26] and CMODE [46], a specific DE operator was used to substitute the original evolutionary operators of MOEA/D [52] and CMPSO [51], giving rise to a better optimization performance; Moreover, in ADEMO/D [45], MOEA/D-FRRMAB [27], and MOEA/D-CDE [34], multiple DE operators were further combined to enhance their performance. These promising results have evidenced the advantages of incorporating single or multiple DE operators into a MOEA.

On the other hand, multi-objective immune algorithms (MOIAs) are designed to mimic the process of clonal selection [3][6][21][40], as inspired from the biologic immune system. The nondominated neighbor-based immune algorithm (NNIA) [21] may be the first real-coded MOIA using the clonal selection approach. Since the report of NNIA, a number of other MOIAs have been designed and enhanced under its framework [6][24][25][32]. Among these MOIAs, clonal selection is employed to pick out a few of less-crowded nondominated solutions, which are then proportionally cloned according to their crowding-distance values [12]. Then, the clones undergo the heuristic search operations, such as recombination and mutation. By this way, the less-crowded search area will be assigned with more clones for exploration. Note that the boundary area is considered as the sparsest area and, therefore, it will be explored by more clones. Compared to the selection operator in NSGA-II [12] and most of other MOEAs [13][36][46], clonal selection enables MOIAs to allocate more search efforts to the boundary and less-crowded areas, which helps to improve the convergence speed and tries to extend the population's diversity. However, when dealing with some complicated MOPs, MOIAs may easily fall into local optimum and suffer from premature convergence or stagnation due to the lack of population diversity [25][41], as only a few of nondominated solutions are selected for cloning, especially at the early stages of the search. To overcome this limitation, some MOIAs [30][31][33][37] have been recently proposed to embed the DE operators. These embedded DE operators have been often used to replace or cooperate with the simulated binary crossover (SBX) operator [16], as the DE operators normally show a better search capability than SBX [31]. Therefore, embedding them can help to enhance the population diversity of MOIAs. However, all the MOIAs proposed in [30][31][33][37] only adopt one single DE strategy, which may not provide optimal performance when tackling different kinds of MOPs, as different DE strategies are suitable for solving certain kinds of problems with different features [11]. Moreover, a single DE strategy with fixed parameters settings only presents a monotonous search pattern, which may limit the search capability of these MOIAs. In some MOEAs [27][34][45], it was experimentally validated that multiple DE strategies seemed more advantageous when solving different kinds of MOPs. Thus, it is reasonable to expect that, the introduction of multiple DE strategies into MOIAs may also be very promising. Therefore, in this paper, an Adaptive Immune-inspired Multi-objective Algorithm, called

AIMA, is proposed in this paper. Three DE strategies with different parameters settings are embedded into a state-of-the-art MOIA (*i.e.*, NNIA [21]) and an adaptive DE strategy selection approach is designed to automatically run an appropriate DE strategy at each generation based on the current evolutionary stage. These three DE strategies can provide different search properties. Thus, they can significantly enhance the search capability and population diversity of NNIA when appropriately selected using the adaptive DE strategy. When solving the well-known and frequently-used test MOPs (WFG [23] and DTLZ [14]) with various features, AIMA shows evident advantages over five state-of-the-art multi-objective optimization algorithms (*i.e.*, NSGA-II [12], SPEA2 [56], MOEA/D [52], SMS-EMOA [2], and CMPSO [51]) and their DE-based variants [26][46], and four competitive MOIAs (*i.e.*, NNIA [21], IMADE [37], DMMO [30], and HEIA [31]).

The remainder of this paper is organized as follows. Section 2 provides some basic background on multi-objective optimization, as well as a brief introduction to MOIAs, and a description of the clonal selection operator in NNIA. The details of AIMA are given in Section 3, including the complete pseudo-code of AIMA, the three adopted DE strategies, and the adaptive DE strategy selection approach. In Section 4, our proposed AIMA is compared with respect to several state-of-the-art MOEAs and MOIAs. The advantages of our proposed mechanisms and the sensitivity of our proposed approach to its parameters are also analyzed in this section. Finally, our conclusions and some possible paths of future research are given in Section 5.

2. Background

2.1. Multi-objective Optimization Problems

Multi-objective optimization problems aim to optimize multiple (often conflicting) objectives simultaneously. Without loss of generality, a MOP can be formulated as follows:

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

where $x = (x_1, x_2, \dots, x_n) \in \Omega$ is a decision vector with n dimensions, Ω is the feasible region of *decision space*, the mapping function $F: \Omega \rightarrow R^m$ defines m objective functions and R^m is the *objective space*. As the objectives may conflict with each other, it is usually impossible to find a solution that can optimize all the objectives at the same time. Thus, the definition of Pareto optimality [5] is adopted to find out the best trade-offs among all the objectives.

Definition 1. (Pareto-dominance): A decision vector x is said to dominate another decision vector y (denoted as $x \prec y$) if and only if

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

Definition 2. (Pareto optimality): A solution x is said to be Pareto-optimal if and only if

$$\neg \exists y \in \Omega : y \prec x \quad (3)$$

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

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

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} .

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

2.2. A Brief Introduction to MOIAs

The first MOIA was designed in [50] to solve a multi-criterion design problem, in which the concept of antibody-antigen affinity is used to modify the fitness assignment mechanism of a standard genetic algorithm. After that, a variety of MOIAs were designed with the aim of enhancing performance. Based on the special features inspired by the immune system, most MOIAs can be classified into three main categories.

The first kind of MOIAs is designed based on the clonal selection approach [3], which applies the cloning operator to produce copies of antibodies having the highest affinity values. In the multi-objective immune system algorithm (MISA) [8], the antibodies with high affinities were cloned to have multiple copies and an adaptive grid was used to maintain the population's diversity. In the immune dominance clonal multi-objective algorithm (IDCMA) [24], the antibody-antibody affinity was used to reflect the similarity among antibodies, which helps to guide the application of the cloning operator on an effective search region (*i.e.*, the least-crowded region). This approach was modified to implement a real-coded MOIA in NNIA [21]. In the hybrid immune multi-objective algorithm (HIMO) [6], a hybrid operator combining Gaussian mutation and polynomial-based mutation was designed, which was further enhanced by using an adaptive mutation operator in the micro-population immune multi-objective algorithm (MIMO) [32].

There is a second class of MOIAs which adopts immune network theory to evolve the population and to maintain the population's diversity. In the vector artificial immune system (VAIS) [18], two evolutionary loops were performed, in which the inner loop is used to exploit the search space, while the outer one using the suppression mechanism is adopted to lower the redundancy among similar antibodies. In the weight-based MOIA [19], a random weighted sum method was used as a fitness assignment scheme and a new truncation approach was presented to eliminate similar individuals.

The last kind of MOIAs combines an immune system with another heuristic, by embedding operators of another heuristic into MOIAs. In the immune-inspired Pareto archived evolution strategy [10], two hypermutation operators were integrated to solve a MOP extracted from the problems of *ab initio* protein structure prediction. In the evolutionary artificial immune algorithm [43], the global search capabilities of EAs were combined with the learning capabilities of artificial immune systems. In the immunity-based

hybrid evolutionary algorithm [49], a sorting scheme featuring uniform crossover, multi-point mutation, and crowding distance sorting were hybridized to solve both unconstrained and constrained MOPs.

In recent years, some new MOIAs with competitive performance have been designed. For example, a novel MOIA was introduced in [25] with two interesting methods. One is to design a resource allocation model to speed up convergence, while the other is to use a double-sphere crowding distance measure to improve the population's diversity. In [37], the immune multi-objective optimization algorithm with differential evolution inspired recombination (IMADE) was reported. This approach presents a novel recombination operator, combining a newly designed DE operator and simulated binary crossover (SBX) [16]. Therefore, it provides two different search patterns during the search process. A double-module immune algorithm for MOPs (DMMO) [30] was presented with two evolutionary modules, with the aim of simultaneously enhancing the convergence speed and the population's diversity. The first module is used to optimize each objective independently using DE in each sub-population, while the second module follows the traditional procedures of MOIAs to optimize multiple conflicting objectives. In [33], a novel hybrid multi-objective immune algorithm with adaptive differential evolution (ADE-MOIA) was designed. This approach introduces an adaptive DE operator to further enhance the robustness of MOIAs in solving various kinds of MOPs. More recently, the hybrid evolutionary immune algorithm for MOPs (HEIA) [31] was reported, providing a novel hybrid evolutionary framework for MOIAs. This approach divides the cloned individuals into several sub-populations, which are then separately evolved by using different evolutionary strategies. An implementation of this framework with two evolutionary strategies (SBX and DE) is shown in HEIA.

2.3. Clonal Selection Operator in NNIA

To illustrate the way in which the clonal selection operator works, here a well-known state-of-the-art MOIA (*i.e.*, NNIA [21]) is introduced. This approach was inspired on the design of many other MOIAs [6][24][25][32][33][37]. Its main contribution is to propose a nondominated neighbor-based selection and a crowding-distance-based proportional cloning mechanism. These two approaches actually constitute the clonal selection operator in NNIA. To show the way in which this operator works, in **Fig. 1**, we show the population evolution in a single generation at time t for NNIA.

to the method in [12], *i.e.*, 2.0, 1.0, 0.5, 1.0, and 2.0, respectively for the five individuals. Assuming that $N=65$, we get $q_1 = 20$, $q_2 = 10$, $q_3 = 5$, $q_4 = 10$ and $q_5 = 20$ according to Eq. (7).

According to the above process of clonal selection, it can be easily observed that the clonal selection operator in NNIA puts more clones to search the less-crowded and boundary areas, thus it shows very promising convergence speed, as validated in [21][31]. However, according to the No-Free-Lunch theorem [48], as we produce more clones of the best solutions, the higher will be the loss of population diversity [41]. As experimentally validated in [41], when the nondominated solutions are few (this case often happens at the early stage of the search), the population diversity in NNIA is extremely poor and it is very easy to get trapped into a local optimum when solving some complicated MOPs with many local *PFs* (*e.g.*, DTLZ1, DTLZ3 and WFG1) or with disconnected *PFs* (*e.g.*, DTLZ7 and WFG2). To overcome this problem, some MOIAs [30][31][33][37] have tried to embed the search capabilities of DE. Although these MOIAs were shown to obtain promising results, they only adopted one single DE strategy, which provides limited search capabilities and may not perform well in tackling various types of MOPs with different features. Therefore, this paper proposes to embed three DE strategies with different parameter settings into a MOIA and then adaptively select an appropriate DE strategy for running at each generation, aiming to enhance the search capabilities of MOIAs during the different search stages, as introduced in Section 3.

3. Our Proposed Algorithm AIMA

The main contribution of AIMA is to present the use of multiple DE strategies during the evolutionary process. The other components of AIMA are designed following the framework of NNIA, as shown in **Fig. 1**. Two main enhancements in AIMA are clarified with respect to NNIA. NNIA uses the SBX operator and polynomial-based mutation to produce offspring in each generation, while AIMA adopts an adaptively selected DE strategy and polynomial-based mutation. Moreover, in the archive update process, if the number of nondominated solutions in D_t (in **Fig. 1**) is larger than N , NNIA selects N of the least-crowded nondominated solutions based on their crowding-distance values [12] to construct the new archive E_{t+1} , while AIMA adopts the archive truncation method from SPEA2 [56], keeping the N least-crowded nondominated solutions based on their Euclidean distances to the nearest neighbor. This way, the weakness of NNIA regarding the loss of population diversity can be somehow alleviated. To show the details of AIMA, its pseudo-code is provided in **Algorithm 1**, where t and MAX_G are respectively the current generation and the maximum number of generations, the parameter *selected* records the selected index for DE, and the parameter *hold_gens* records the number of generations that the stagnant state holds.

Algorithm 1: Pseudo-code of AIMA

```
1 Initialize  $t=0$ ,  $selected=1$ ,  $hold\_gens=0$ ,  $E_0 = \emptyset$  ;
2 Randomly generate an initial population  $P_0$ ;
3 while  $t < MAX\_G$ 
4     // Evolution
5     Apply  $DE_{selected}$  strategy to  $P_t$  to produce the offspring population  $O_t$ ;
6     Apply polynomial-based mutation to  $O_t$ ;
7     // Identification of nondominated solutions
8     Identify the nondominated solutions in  $E_t \cup O_t$  to form  $D_t$ ;
9     // Selection of an appropriate DE strategy for next generation
10    [ $selected$ ,  $hold\_gens$ ] = Adaptive DE Strategy Selection( $hold\_gens$ ,  $E_t$ ,  $O_t$ ) (Algorithm 2);
11    // Archive update
12     $E_{t+1} = D_t$ ;
13    if  $|E_{t+1}| > N$ 
14        Keep  $N$  less-crowded solutions in  $E_{t+1}$  by the archive truncation method in [56];
15    end if
16     $t = t + 1$ ;
17    // Clonal Selection
18    Calculate the crowding-distance values of solutions in  $E_t$ ;
19    Select  $n_A$  solutions with the larger crowding-distance values to form  $A_t$ ;
20    Apply proportional cloning (Eq. 6) on  $A_t$  to generate a new population  $P_t$ ;
21 end while
22 return  $E_t$ ;
```

At the beginning of AIMA, some related parameters (i.e., t , $selected$, and $hold_gens$) as well as the archive E_0 are initialized in line 1, and then the evolution population is randomly generated in line 2. Then, AIMA goes into its main loop in lines 3-21, until t reaches the maximum number of generations, i.e., MAX_G . Regarding each generation at time t , the offspring population O_t is generated in lines 5-6 by running an adaptively selected DE strategy ($DE_{selected}$) and by applying polynomial-based mutation to the population P_t . Then, in line 8, the offspring population O_t and the archive E_t are combined, and their nondominated solutions are selected to form the temporary population D_t . After that, in line 10, our proposed adaptive DE strategy selection is run to select a preferred DE strategy to be used in the next generation, by setting the value of $selected$. Next, the archive update procedure is carried out and all the solutions in D_t are collected into the archive E_{t+1} in line 12. If the size of E_{t+1} exceeds the predefined maximum archive size N , the archive truncation process as introduced in [56] is executed on E_{t+1} to keep the N least-crowded individuals. At last, the generation time t is increased by 1, and the evolution population P_t is produced by the clonal selection approach in lines 18-20, as introduced in Section 2.3. At the end of AIMA, the solutions in the archive E_t are reported as the approximation set. From the pseudo-code of AIMA, the main contributions of this paper focus on the adopted DE strategies in line 5 and the corresponding adaptive DE strategy selection in line 10. To further clarify the way in which AIMA works, the three adopted DE strategies and the adaptive DE strategy selection mechanism are respectively introduced in the following subsections.

3.1. The Three Adopted DE Strategies

Differential evolution is a simple yet powerful stochastic search method, which has been shown to be an efficient and effective optimizer for both SOPs and MOPs [26]. Different DE strategies with their control parameter settings show distinct search characteristics and, therefore, they can provide different advantages and cooperate with each other to solve different kinds of MOPs. In this paper, three well-known DE strategies (*i.e.*, rand/1/bin, rand/2/bin and rand/1/bin~, respectively named DE₁, DE₂, and DE₃) with fixed parameter settings are used, as respectively listed in Eqs. (8)-(10).

$$DE_1 : U_{i,j} = \begin{cases} X_{r1,j} + F \cdot (X_{r2,j} - X_{r3,j}) + F \cdot (X_{r4,j} - X_{r5,j}) & \text{if } rand < Cr \text{ or } j = j_{rand}, \text{ with } Cr = 0.9 \text{ and } F = 0.7 \\ X_{i,j} & \text{otherwise} \end{cases} \quad (8)$$

$$DE_2 : U_{i,j} = \begin{cases} X_{r1,j} + F \cdot (X_{r2,j} - X_{r3,j}) & \text{if } rand < Cr \text{ or } j = j_{rand}, \text{ with } Cr = 0.5 \text{ and } F = 0.5 \\ X_{i,j} & \text{otherwise} \end{cases} \quad (9)$$

$$DE_3 : U_{i,j} = \begin{cases} X_{i,j} + F \cdot (X_{r1,j} - X_{r2,j}) & \text{if } rand < Cr \text{ or } j = j_{rand}, \text{ with } Cr = 0.1 \text{ and } F = 0.5 \\ X_{i,j} & \text{otherwise} \end{cases} \quad (10)$$

In the above equations, $U_{i,j}$ is called the trial vector and $X_{i,j}$ is the target vector in the current population. Cr is the crossover rate and F is the scaling factor. $r1$, $r2$, $r3$, $r4$, and $r5$ are the distinct integers randomly selected from $[1, N]$. The subscript j represents the j th variable of the decision vector ($j=1, 2, \dots, n$). $rand$ is a real number randomly generated in $[0, 1]$ and j_{rand} is an integer randomly selected from $[1, 2, \dots, n]$. At each generation, one DE strategy will be selected to produce trial vectors for all the target vectors in the current population. The details of the proposed adaptive DE strategy selection will be introduced in the following subsection.

Among these three DE strategies, rand/1/bin and rand/2/bin are the two most commonly used when tackling MOPs. All the used vectors in them are randomly selected from the population, so they can provide different search directions and bear strong exploration capabilities [47]. Thus, these two strategies are good for solving multimodal problems. Compared to rand/1/bin, rand/2/bin has a better perturbation and can generate more different trial vectors since it has two difference vectors [38][42][55]. Thus, rand/2/bin shows better exploration capabilities than rand/1/bin, which may make it more suitable for solving MOPs with complex **PF**, *e.g.*, disconnected **PF**. For rand/1/bin~, the difference vector is added to the current solution rather than a randomly selected solution, which leads to local search around the target vector. On the other hand, the settings of control parameters in DE, *i.e.*, Cr and F , also have great impact on the performance of DE. The value of Cr determines how much information the offspring (the trial vector) can inherit from its parent (the target vector). A large Cr value makes the trial vector very different from the target vector, since more information in the trial vector is inherited from the mutant vector created by DE rather than from the target vector in the current population [11][47]. This results in exploration around the target vector and helps to enhance the population diversity. On the contrary, a

small value of Cr lets the trial vector be different from the target vector on only few dimensions, thus encouraging exploitation around the target vector. Considering the scaling factor, a large value of F can provide a large search step to extend the search area and to diversify the population, while a small value of F makes the search around the target vector and thus, it can speed up convergence [47].

Therefore, the three DE strategies with specific parameters settings in Eqs. (8)-(10) exhibit distinct advantages. For DE_1 , it employs rand/2/bin with large values of Cr and F , thus it has a very strong global search capability (i.e., exploration) and can greatly enhance the population diversity. On the contrary, DE_3 applies rand/1/bin~ and couples with a small value of Cr , which encourages to conduct local search (i.e., exploitation). Whereas, in DE_2 , the rand/1/bin strategy with moderate values of Cr and F are used. This lets DE_2 be able to balance exploration and exploitation during the search. It can be regarded as a trade-off strategy between DE_1 and DE_3 .

3.2. Adaptive DE Strategy Selection

The three DE strategies introduced in this paper are expected to complement one another for tackling MOPs with different characteristics. To reach this goal, an adaptive DE strategy selection is introduced here. As pointed out in [26] and [33], when dealing with some complicated MOPs, a global search capability is usually required to explore the entire search space and to maintain a diversified population at the early stages of the search. In contrast, at later stages of the search process, local search is preferred to speed up convergence towards the true PF. Thus, in this paper, AIMA divides the search process into three stages, *i.e.*, early stage, middle stage and later stage.

At the early stages of the search, the information about the search space is limited; so, a DE strategy with a strong global search capability is needed to effectively explore the whole search space, which helps to maintain the population diversity and to prevent the algorithm from falling into a local optimum. The middle stage of the search will come after exploring for a certain number of generations, in which some attention should be paid to speeding up convergence. A DE strategy that can balance exploration and exploitation is preferred at this stage to speed up convergence without lowering too much the diversity. At the later stages of the search, the algorithm often cannot find many good solutions, even after a long search period, since the solutions found so far may be close to the true **PF** in this stage. Therefore, a DE strategy encouraging local search is preferred to find better solutions around the current solutions. Based on the above analysis, DE_1 should be assigned with a large execution probability at the early stage to encourage exploration, while DE_3 should be run with a large probability at the later stage to encourage exploitation. Regarding the middle stage, DE_2 should be executed more times to balance exploration and exploitation.

In AIMA, the execution probabilities of DE_1 and DE_3 at each generation are respectively denoted by

p_1 and p_3 , which are tuned according to the stages of the search. For DE_2 , the execution probability at each generation is calculated by $1-p_1-p_3$.

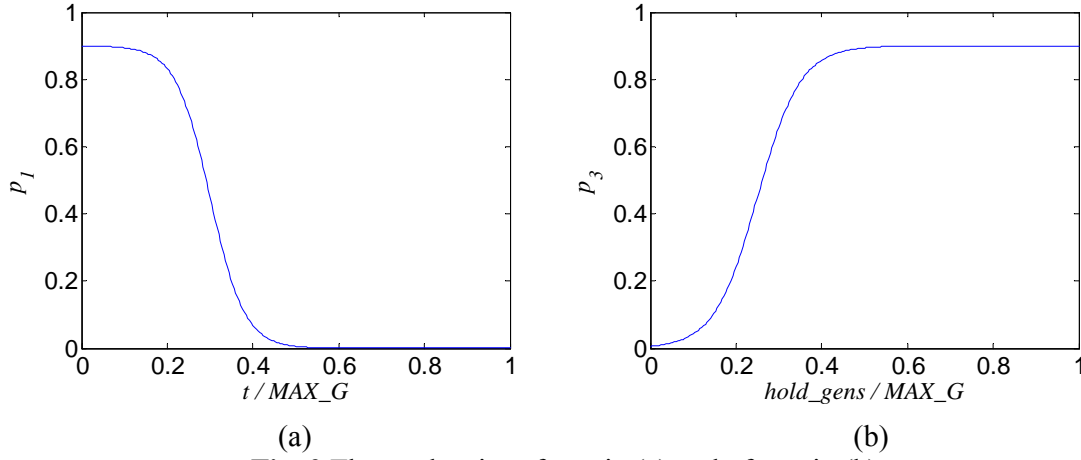


Fig. 2 The tendencies of p_1 in (a) and of p_3 in (b)

As it is difficult to define the boundary between the early stage and the middle stage, there isn't a numerical indicator that can exactly decide the current stage of the search. Here, we simply let p_1 be dynamically reduced as the search goes on, and it is expressed as follows:

$$p_1 = 0.9 \times \frac{1.0}{1 + \exp(20 \times (\frac{t}{MAX_G} - 0.3))} \quad (11)$$

where t is the current generation number, and MAX_G is the pre-defined maximum number of generations. To clearly show the dynamic change of p_1 , its tendency is illustrated in **Fig. 2(a)**, where the value of p_1 is reduced from 0.9 to 0. As observed from **Fig. 2(a)**, p_1 is set to be large at the start of the search and it decreases rapidly after about one quarter of MAX_G . Therefore, AIMA has a large probability to use DE_1 and may spread the population diversity at the early stage of the search. After running for about half of MAX_G , DE_1 is applied with a very small probability and more computational resources are assigned to run DE_2 and DE_3 .

In order to tune the value of p_3 , an estimation method is designed in this paper to reflect the possibility that the search process has entered the later stage. When the estimated possibility is larger, a larger value will also be assigned to p_3 . Here, to clearly describe the estimation method, two relevant terms are introduced:

- (1) **Elimination**: if a solution X in archive E_t is dominated by one solution in the offspring population O_t , it will be eliminated from E_t . The number of eliminated solutions in E_t is recorded by $eNum$, and then the elimination ratio in E_t is recorded by $eRatio = eNum / |E_t|$ ($|E_t|$ returns the size of E_t). If AIMA can generate many good solutions at the t th generation, most of the

solutions in E_t will be eliminated, which results in a large value of $eRatio$. Otherwise, a small value of $eRatio$ is obtained when AIMA only obtains a few good solutions.

(2) **Stagnant state:** In AIMA, we monitor the average $eRatio$ value (denoted as $avgRatio$) during the last previous LP generations. Then, we set a pre-defined low ratio $LR\%$ to judge whether the search of AIMA is stagnant. If $avgRatio$ is lower than $LR\%$, it indicates that a few better solutions are produced by AIMA during the previous LP generations and the algorithm may stay in a stagnant state. The values of LP and LR were respectively set to 5 and 10 after running several experiments.

AIMA records the number of generations that the stagnant state holds, as denoted by $hold_gens$ and then uses $hold_gens$ to tune the value of p_3 . It is noted that, once the stagnant state is broken, *i.e.*, $avgRatio$ is larger than $LR\%$, the value of $hold_gens$ will be reset to zero. Once AIMA is unable to find many good solutions during a long period of search, the value of $hold_gens$ will be increased, which indicates that the stagnant state has been going on for lots of generations. In such case, we consider that the solutions found so far may be close to the true PF and there is a larger possibility that the evolution has entered the later stage. Then, a large value should be assigned to p_3 for encouraging a local search, when the value of $hold_gens$ is getting larger. Thus, the value of p_3 is assigned to dynamically change with $hold_gens$, so as to adaptively control the execution probability of DE_3 , as expressed by

$$p_3 = 0.9 \times \frac{1.0}{1 + \exp(-20 \times (\frac{hold_gens}{MAX_G} - 0.25))} \quad (12)$$

where MAX_G is the pre-defined maximum number of generations. To clearly show the dynamic change of p_3 , its tendency is also demonstrated in **Fig. 2(b)**, where the value of p_3 is increased from 0 to 0.9. When the value of $hold_gens$ is small, a small p_3 will be used, as the search may get stuck temporarily and it is not so sure that the search has stepped into the later stage. Conversely, if the value of $hold_gens$ is large, it is more likely that the search has turned into the later stage and, therefore, a large value of p_3 is adopted.

Finally, after the values of p_1 and p_3 are computed, the roulette wheel selection method [12] is employed to select a preferred DE strategy to be used in the next generation according to p_1 and p_3 . It is noted that p_1 and p_3 will be normalized when their summation is larger than 1. To clearly describe our adaptive DE strategy selection, its pseudo-code is shown in **Algorithm 2**.

Algorithm 2: Adaptive DE Strategy Selection ($hold_gens, E_t, O_t$)

- 1 Get p_1 by Eq.(11);
- 2 Calculate the average elimination ratio ($avgRatio$) of last LP generations;
- 3 **if** $avgRatio \leq LR\%$
- 4 $hold_gens = hold_gens + 1$;
- 5 **Else**

```

6      hold_gens = 0;
7  end if
8  Get  $p_3$  by Eq. (12);
9  /* the roulette wheel method is used here */
10 Randomly generate the value of  $rand$  in  $[0,1]$ ;
11 if  $rand < p_1$ 
12     Set  $selected = 1$ ;           // select  $DE_1$ 
13 else if  $rand < p_1 + p_3$ 
14     Set  $selected = 3$ ;           // select  $DE_3$ 
15 Else
16     Set  $selected = 2$ ;           // select  $DE_2$ 
17 end if
18 return  $selected$  and  $hold\_gens$ ;

```

4. Experimental Studies

In this section, to assess the overall performance of AIMA, several experimental studies are launched. First, some related information about the experiments is provided, including the benchmark problems, the performance measures, and the parameters settings of all the compared algorithms. Then, we provide experimental results comparing AIMA with different kinds of MOEAs and their DE-based variants, as well as with four competitive MOIAs. Moreover, the advantages of the proposed mechanisms in AIMA and their sensitivity to their corresponding parameters are also investigated in this section.

4.1. Benchmark Problems

In this study, sixteen frequently-used test MOPs without any constraints, including nine two-objective problems (WFG1-WFG9 [23]) and seven three-objective problems (DTLZ1-DTLZ7 [14]), were used to assess the performance of AIMA. These test MOPs are characterized for having different complex features, such as convexity, concavity, discontinuity, non-uniformity, and many local **PF**s. Thus, the WFG and DTLZ test suites are widely used in many experimental studies for performance comparisons among different multi-objective optimization algorithms [2][21][51]. It is noted that, the number of decision variables in all these MOPs is set to 10. Especially for the WFG test problems, 10 decision variables are composed by 8 position parameters and 2 distance parameters.

4.2. Performance Measures

It is well-known that the main aim of solving MOPs includes two aspects, *i.e.*, convergence and diversity. That is to say, a set of solutions is preferred to approximate the true **PF** as close as possible and to cover the true **PF** as widely as possible. Therefore, in this paper, the widely used performance measure, inverted generational distance (IGD) [26], is adopted to assess the performance of all the compared algorithms in terms of both convergence and diversity.

Let P^* denote a subset of Pareto-optimal solutions uniformly distributed along the true **PF** and P be an approximation set obtained by multi-objective optimization algorithms. The IGD metric measures the

average distance from each point in P^* to its nearest point in P regarding the objective space, and is calculated as follows.

$$IGD(P^*, P) = \frac{\sum_{x \in P^*} d(x, P)}{|P^*|} \quad (13)$$

where $d(x, P)$ represents the Euclidean distance from x to its nearest neighbor in P . Generally, a smaller value of this measure indicates a better performance on convergence and diversity. To clearly show the calculation of IGD in this paper, the numbers of used points in P^* for all the test problems adopted are listed in **Table 1**. All these true **PFs** were downloaded from <http://jmetal.sourceforge.net/problems.html> and we have pre-processed these **PFs** by removing some duplicated points in them.

Table 1 Number of points in the subset of true **PFs**

Problem	Points	Problem	Points	Problem	Points	Problem	Points
WFG1	605	WFG5	2601	WFG9	2601	DTLZ4	4000
WFG2	111	WFG6	2601	DTLZ1	10000	DTLZ5	333
WFG3	301	WFG7	2601	DTLZ2	10000	DTLZ6	140
WFG4	1181	WFG8	10201	DTLZ3	4000	DTLZ7	676

4.3. Experimental Settings

In this study, AIMA is compared to five state-of-the-art multi-objective optimization algorithms, *i.e.*, NSGA-II [12], SPEA2 [56], MOEA/D [52], SMS-EMOA [2], and CMPSO [51]. Moreover, to have a more fair comparison, AIMA is also compared to five DE-based variants of the above algorithms, *i.e.*, NSGA-II-DE, SPEA2-DE, MOEA/D-DE [26], SMS-EMOA-DE, and CMODE [46], as AIMA is also a DE-based variant for MOIAs. The original SBX is replaced by the “DE/rand/1/bin” strategy in NSGA-II-DE, SPEA2-DE, and SMS-EMOA-DE in this paper. These DE-based variants have been shown to outperform the original ones when solving most test MOPs [44][4]. MOEA/D-DE and CMODE are respectively the improved versions of MOEA/D and CMPSO by embedding DE. Besides that, to further validate the superiority of AIMA over other MOIAs, AIMA is compared to four competitive MOIAs, including NNIA [21], IMADE [37], DMMO [30], and HEIA [31]. In order to make a fair comparison, all the compared MOIAs also use the same archive truncation approach of SPEA2 in the experiments.

It is noted that the implementations of NSGA-II, SPEA2, MOEA/D-DE, and SMS-EMOA can be found in jMetal [17], while the source codes of CMPSO, CMODE, NNIA, IMADE, DMMO, and HEIA are realized by us based on the framework of jMetal. In order to allow a fair comparison and provide a better performance for all the compared algorithms, their relevant parameters are set as suggested in the corresponding references. For NSGA-II-DE, SPEA2-DE, and SMS-EMOA-DE, the values of Cr and F are respectively set to 0.3 and 0.5 as used in [44], which show significant improvements with respect to

the original algorithms. The rest of the parameters in these three variants are set the same as those in NSGA-II, SPEA2, and SMS-EMOA. For AIMA, the number of individuals picked out from the archive for cloning, n_A , is set to $N/5$ as other MOIAs do. The mutation probability p_m and distribution index η_m in polynomial-based mutation (PM) are set the same with the compared algorithms, *i.e.*, $1/n$ and 20. The values of LP and LR are respectively set to 5 and 10. When solving all the test problems, the maximum number of function evaluations (FEs) is set to 25000 for all the compared algorithms. The population size N and the archive size are all set to 100, except that CMPSO and CMODE which still adopt a population size of 20 for each swarm as suggested by their authors.

All the compared algorithms are launched 30 times on each problem using a personal computer with 3.6 GHZ CPU and 8 GB RAM. The mean value (Mean) and standard deviation (Std) from 30 runs regarding IGD are used for comparison purposes. The best result and second best result in the comparison tables are respectively marked with dark gray background and light gray background. Moreover, in order to have a statistically sound conclusion, Wilcoxon's rank sum test with a significance level $\alpha=0.05$ is further run, which shows the statistical significance of the difference between the results obtained by AIMA and those obtained by the other algorithms.

4.4. Experimental Results

4.4.1. Performance Comparisons of AIMA with Various Multi-objective Optimization Algorithms

A) Performance comparisons with five state-of-the-art multi-objective optimization algorithms

In **Table 2**, the comparison results of AIMA with NSGA-II, SPEA2, MOEA/D-DE, SMS-EMOA, and CMPSO are provided when solving the DTLZ and WFG test suites. All the comparison results of AIMA with each algorithm are summarized in the last row of **Table 2**, where “-”, “+”, and “ \approx ” indicate that the results of the algorithm are respectively worse than, better than, and similar to those of AIMA using Wilcoxon's rank sum test with $\alpha = 0.05$. Based on these results, it is clear that AIMA respectively outperforms NSGA-II, SPEA2, MOEA/D, SMS-EMOA, and CMPSO on 16, 13, 12, 13, and 16 out of 16 test problems, while AIMA is only worse than SPEA2 on DTLZ2 and WFG4, worse than MOEA/D on DTLZ2, and worse than SMS-EMOA on WFG4 and WFG9. Particularly, AIMA performs significantly better on WFG2 when compared to its five competitors. For WFG2, AIMA can obtain a better IGD result with an accuracy level of 10^{-2} , while other competitors only obtain IGD results under an accuracy level of 10^{-1} . As WFG2 has five disconnected segments on its true PF , the competitors failed to find all the segments, while AIMA was able to cover the entire PF due to its strong search capability.

To visually display the superiority of AIMA, the nondominated solution sets found for DTLZ6 and WFG8 are plotted in **Fig. 3** and **Fig. 4**, respectively, where the true **PFs** are identified with red color. It is noted that, only the solution set with the median IGD value in 30 runs is illustrated for comparison. Regarding DTLZ6, the solution sets obtained by AIMA and CMPSO seem well-converged and have a uniform distribution along the true **PF**. Their performances are not visually distinguished from the plots. For MOEA/D and SMS-EMOA, although they are able to closely approximate the true **PF**, they fail to get an even distribution for the solution sets. The other two MOEAs, *i.e.*, NSGA-II and SPEA2, are unable to approach the true **PF** closely. On WFG8, all the algorithms except for CMPSO can find the solutions that converge well to the true **PF**. However, as shown in **Fig. 4**, the solutions obtained by AIMA seem more even than those of its competitors.

Table 2 IGD comparison results of AIMA and five state-of-the-art multi-objective optimization algorithms

Problems		NSGA-II	SPEA2	MOEA/D	SMS-EMOA	CMPSO	AIMA
DTLZ1	Mean	8.163E-01 -	4.958E-01 -	6.551E-02 ≈	2.021E-01 ≈	2.041E+00 -	1.934E-01
	Std	6.04E-01	3.32E-01	1.31E-01	2.03E-01	6.20E-01	2.09E-01
DTLZ2	Mean	6.829E-02 -	5.495E-02 +	5.155E-02 +	7.396E-02 -	6.342E-02 -	5.568E-02
	Std	2.81E-03	1.01E-03	3.49E-04	1.29E-03	2.03E-03	1.13E-03
DTLZ3	Mean	2.772E+00 -	2.124E+00 -	2.376E-01 ≈	7.223E-01 -	4.657E+00 -	3.992E-01
	Std	2.06E+00	1.47E+00	3.56E-01	6.73E-01	1.33E+00	6.85E-01
DTLZ4	Mean	6.390E-02 -	1.166E-01 -	1.048E-01 ≈	1.374E-01 -	1.701E-01 -	4.106E-02
	Std	7.28E-03	1.15E-01	1.11E-01	1.25E-01	1.19E-01	2.53E-03
DTLZ5	Mean	5.528E-03 -	4.406E-03 ≈	1.905E-02 -	4.919E-03 -	6.019E-03 -	4.449E-03
	Std	2.78E-04	1.68E-04	1.56E-05	1.50E-04	3.49E-04	1.46E-04
DTLZ6	Mean	5.332E-01 -	5.170E-01 -	3.565E-02 -	3.557E-02 -	4.502E-03 -	4.018E-03
	Std	5.18E-02	4.25E-02	1.58E-02	2.47E-02	2.06E-04	1.10E-04
DTLZ7	Mean	7.602E-02 -	1.219E-01 -	1.338E-01 -	2.022E-01 -	7.356E-02 -	5.789E-02
	Std	3.94E-03	1.59E-01	4.30E-02	9.46E-02	4.34E-03	1.30E-03
WFG1	Mean	1.561E+00 -	1.787E+00 -	1.766E+00 -	1.855E+00 -	7.287E-01 -	9.581E-02
	Std	1.47E-01	7.16E-02	1.09E-01	1.24E-01	2.35E-01	1.90E-01
WFG2	Mean	1.400E-01 -	1.255E-01 -	2.178E-01 -	1.751E-01 -	1.910E-01 -	2.460E-02
	Std	8.46E-02	8.05E-02	6.65E-02	8.20E-02	7.44E-02	2.48E-02
WFG3	Mean	1.559E-02 -	1.575E-02 -	4.146E-02 -	2.020E-02 -	1.476E-02 -	1.170E-02
	Std	9.72E-04	2.87E-03	2.19E-02	7.03E-03	6.77E-04	1.09E-04
WFG4	Mean	1.385E-02 -	1.239E-02 +	1.611E-02 -	1.017E-02 +	1.404E-02 -	1.268E-02
	Std	6.40E-04	3.91E-04	1.82E-03	3.53E-04	1.02E-03	3.63E-04
WFG5	Mean	6.794E-02 -	6.693E-02 -	6.787E-02 -	6.709E-02 -	6.565E-02 -	6.536E-02
	Std	4.52E-04	2.79E-04	7.87E-04	4.35E-04	5.66E-03	5.71E-03
WFG6	Mean	2.180E-02 -	2.476E-02 -	3.368E-02 -	2.684E-02 -	2.889E-02 -	1.409E-02
	Std	6.81E-03	1.27E-02	1.33E-02	1.01E-02	7.98E-03	4.43E-03
WFG7	Mean	1.968E-02 -	2.254E-02 -	1.763E-02 -	3.928E-02 -	1.570E-02 -	1.249E-02
	Std	5.60E-03	8.80E-03	2.62E-03	1.75E-02	9.95E-04	3.49E-04
WFG8	Mean	6.489E-02 -	1.045E-01 -	1.247E-01 -	1.066E-01 -	1.397E-01 -	3.825E-02
	Std	9.14E-03	4.26E-02	4.38E-02	3.23E-02	1.10E-01	4.92E-03
WFG9	Mean	1.595E-02 -	1.295E-02 -	1.493E-02 -	1.221E-02 +	1.636E-02 -	1.246E-02
	Std	1.60E-03	6.06E-04	5.42E-04	7.49E-04	1.23E-03	3.37E-04
-/+/>≈		16/0/0	13/2/1	12/1/3	13/2/1	16/0/0	

“-”, “+”, and “ \approx ” indicate that the results of the algorithm are worse than, better than, and similar to that of AIMA using Wilcoxon’s rank sum test with $\alpha = 0.05$

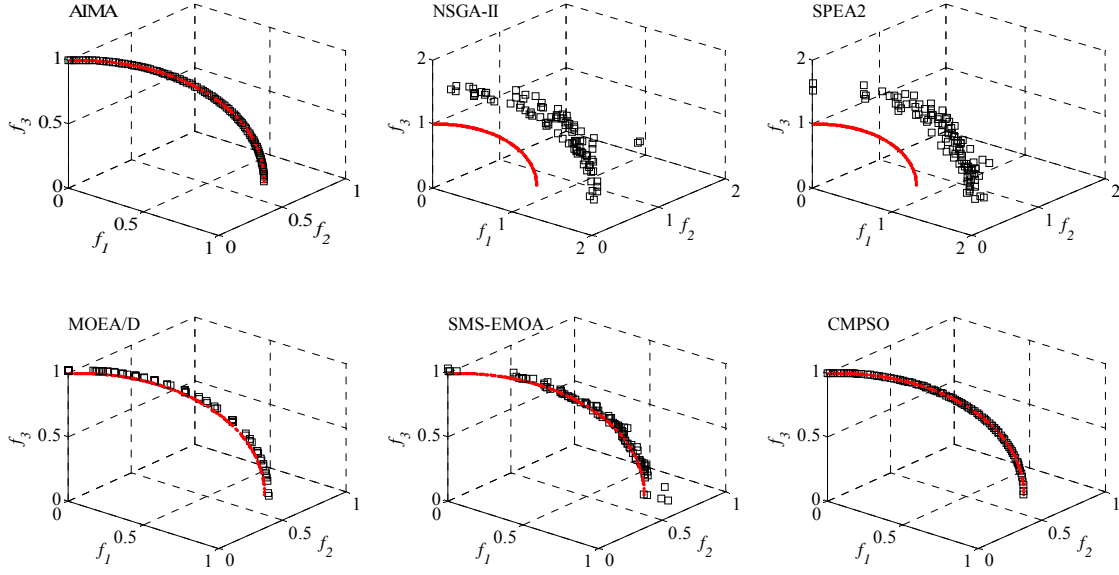


Fig. 3 The nondominated solution set of each algorithm on DTLZ6

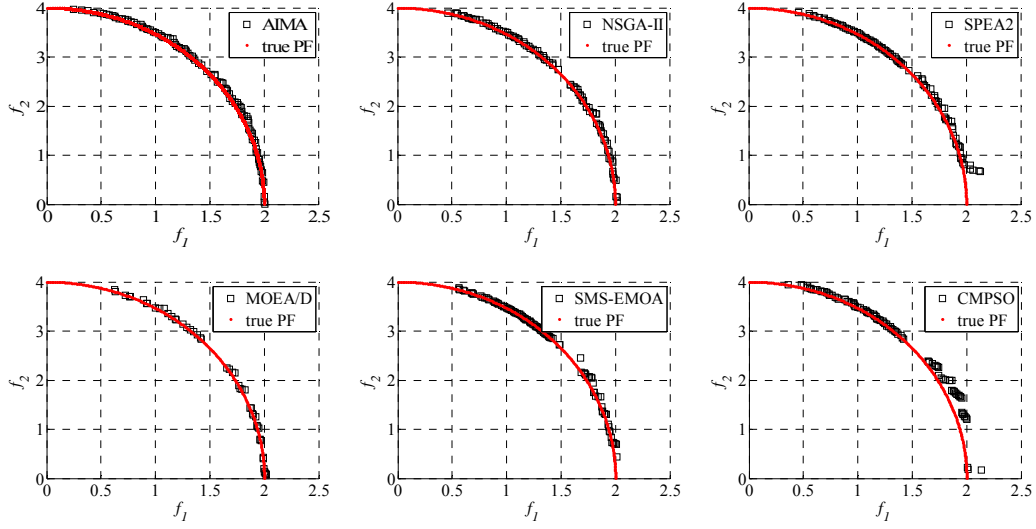


Fig. 4 The nondominated solution set of each algorithm on WFG8

Table 3 The IGD comparison results of AIMA and DE-based variants of five state-of-the-art multi-objective optimization algorithms

Problems		NSGA-II-DE	SPEA2-DE	MOEA/D-DE	SMS-EMOA-DE	CMODE	AIMA
DTLZ1	Mean	5.314E-02 \approx	4.474E-02 \approx	9.944E-01 -	4.950E-02 +	6.476E+00 -	1.934E-01
	Std	8.63E-02	5.50E-02	7.44E-01	8.38E-02	3.28E+00	2.09E-01
DTLZ2	Mean	6.386E-02 -	5.616E-02 -	5.232E-02 +	7.430E-02 -	5.941E-02 -	5.568E-02
	Std	2.17E-03	1.32E-03	4.59E-04	9.53E-04	1.63E-03	1.13E-03
DTLZ3	Mean	6.518E-02 \approx	1.127E-01 \approx	1.954E+00 -	1.396E-01 \approx	1.553E+01 -	3.992E-01
	Std	3.50E-03	1.80E-01	1.52E+00	2.40E-01	7.57E+00	6.85E-01
DTLZ4	Mean	6.359E-02 -	4.916E-02 -	3.007E-02 +	4.871E-02 -	5.799E-02 -	4.106E-02
	Std	6.01E-03	2.08E-02	5.78E-04	2.08E-02	3.40E-02	2.53E-03
DTLZ5	Mean	5.302E-03 -	5.334E-03 -	1.901E-02 -	4.959E-03 -	7.036E-03 -	4.449E-03

	Std	2.13E-04	3.04E-04	4.70E-05	1.46E-04	6.65E-04	1.46E-04
DTLZ6	Mean	5.865E-03 -	4.057E-03 \approx	1.714E-02 -	4.817E-03 -	4.953E-02 -	4.018E-03
	Std	3.68E-04	1.24E-04	4.97E-05	1.53E-04	1.06E-01	1.10E-04
DTLZ7	Mean	7.399E-02 -	5.964E-02 -	1.407E-01 -	1.647E-01 -	9.277E-02 -	5.789E-02
	Std	3.89E-03	1.57E-03	5.89E-02	5.67E-02	8.78E-02	1.30E-03
WFG1	Mean	1.104E-01 -	1.045E-01 \approx	5.705E-02 +	1.281E-01 -	1.317E-02 +	9.581E-02
	Std	1.95E-01	2.26E-01	1.18E-01	2.71E-01	1.69E-03	1.90E-01
WFG2	Mean	4.995E-02 -	6.130E-02 -	1.976E-01 -	5.974E-02 -	3.947E-02 -	2.460E-02
	Std	4.36E-02	1.44E-02	7.50E-02	1.52E-02	2.52E-02	2.48E-02
WFG3	Mean	1.399E-02 -	1.218E-02 -	1.325E-02 -	1.188E-02 -	1.250E-02 -	1.170E-02
	Std	5.15E-04	2.04E-04	1.74E-04	1.38E-03	2.12E-04	1.09E-04
WFG4	Mean	1.277E-02 \approx	1.317E-02 -	1.582E-02 -	1.021E-02 +	1.283E-02 \approx	1.268E-02
	Std	8.00E-04	6.25E-04	7.38E-04	5.62E-04	3.97E-04	3.63E-04
WFG5	Mean	6.800E-02 -	6.614E-02 -	6.716E-02 -	6.677E-02 -	6.369E-02 +	6.536E-02
	Std	4.86E-04	5.56E-03	2.79E-04	5.95E-03	8.89E-03	5.71E-03
WFG6	Mean	1.834E-02 -	1.400E-02 \approx	2.799E-02 -	1.787E-02 -	1.632E-02 -	1.409E-02
	Std	8.44E-03	5.02E-03	1.24E-02	8.95E-03	6.27E-03	4.43E-03
WFG7	Mean	1.447E-02 -	1.277E-02 -	1.779E-02 -	1.101E-02 +	1.287E-02 -	1.249E-02
	Std	8.54E-04	3.47E-04	6.78E-03	2.43E-04	3.36E-04	3.49E-04
WFG8	Mean	7.031E-02 -	1.633E-01 -	3.854E-02 \approx	1.517E-01 -	8.737E-02 -	3.825E-02
	Std	8.77E-03	3.38E-02	4.60E-03	2.56E-02	5.06E-02	4.92E-03
WFG9	Mean	1.522E-02 -	1.314E-02 -	1.469E-02 -	1.258E-02 \approx	1.397E-02 -	1.246E-02
	Std	8.79E-04	4.64E-04	2.06E-04	9.00E-04	7.53E-04	3.37E-04
-/+/ \approx		13/0/3	11/0/5	12/3/1	11/3/2	13/2/1	

“-”, “+”, and “ \approx ” indicate that the results of the algorithm are worse than, better than, and similar to those of AIMA using Wilcoxon’s rank sum test with $\alpha = 0.05$

B) Performance comparisons with the DE-based variants of five state-of-the-art multi-objective optimization algorithms

As AIMA is designed based on the DE strategies, to further investigate the performance of AIMA, the “DE/rand/1/bin” strategy is also embedded into the above compared algorithms and AIMA is further compared to five DE-based variants, *i.e.*, NSGA-II-DE, SPEA2-DE, MOEA/D-DE, SMS-EMOA-DE, and CMODE. **Table 3** lists the IGD comparison results of AIMA with these DE-based variants. As observed from **Table 3**, all these variants have better overall performance when compared to the original ones. Nevertheless, AIMA still shows superior performance over these algorithms, as AIMA performs significantly better than NSGA-II-DE, SPEA2-DE, MOEA/D-DE, SMS-EMOA-DE, and CMODE on 13, 11, 12, 11, and 13 out of 16 test problems, respectively; whereas, AIMA is respectively outperformed by MOEA/D-DE, SMS-EMOA-DE, and CMODE on 3, 3, and 2 out of 16 test problems.

To visually show the advantages of AIMA, **Fig. 5** provides the nondominated solution sets found by all the compared algorithms on WFG2. For AIMA, the obtained solutions are evenly distributed along all the disconnected segments of WFG2, while other competitors only approximate some parts of the segments of the true **PF**. In order to show the evolutionary progress of some compared algorithms (NSGA-II, SPEA2, MOEA/D-DE, SMS-EMOA, CMODE, and AIMA), **Fig. 6** plots the evolutionary curves of the average IGD values from 30 runs versus the number of function evaluations on each WFG problem. It is easily observed that AIMA achieves the best overall performance when solving most of the

WFG test problems.

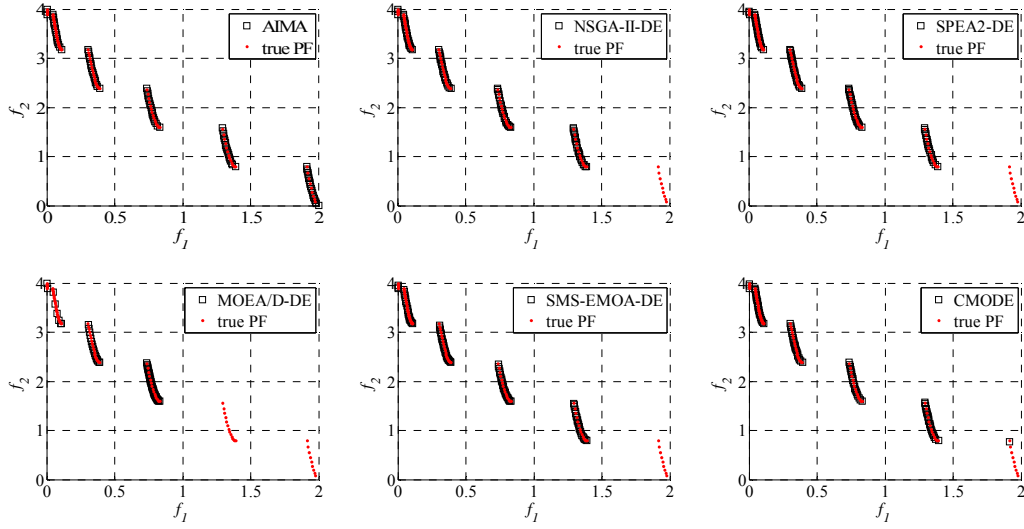
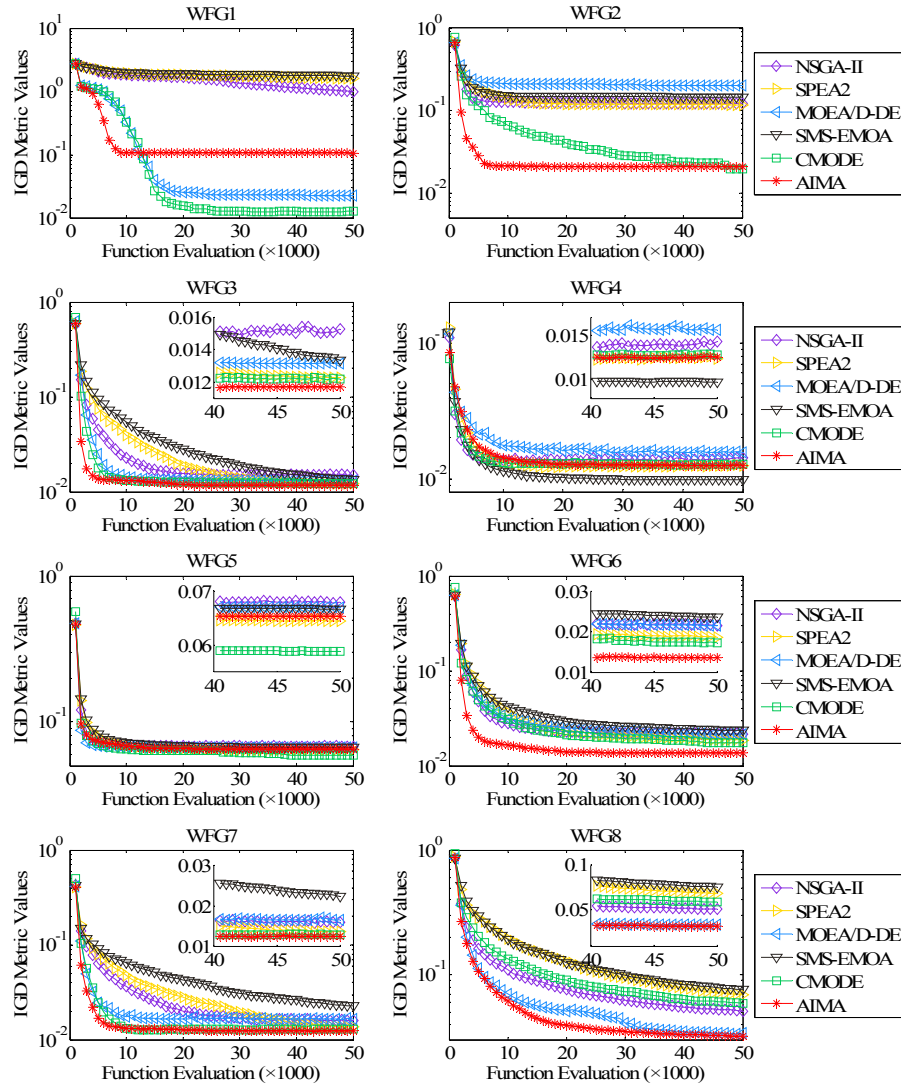


Fig. 5 The nondominated solution set obtained by each algorithm on WFG2



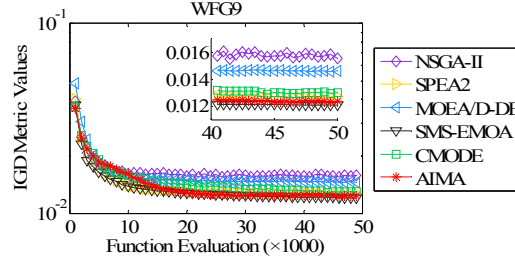


Fig. 6 Evolutionary curves of the average IGD values from 30 runs versus the number of function evaluations on the WFG test problems

From the above discussion on the performance comparisons of AIMA with respect to five state-of-the-art multi-objective optimization algorithms and their DE-based variants regarding the IGD values, it is reasonable to conclude that AIMA performs better than its five competitors and their DE-based variants on tackling most of the DTLZ and WFG test problems.

Table 4 IGD comparison results of all the compared MOIAs

	NNIA Mean(Std)	IMADE Mean(Std)	DMMO Mean(Std)	HEIA Mean(Std)	AIMA Mean(Std)
DTLZ1	2.660E+00(1.92E+00) -	8.127E-01(1.19E+00) -	3.122E-01(2.83E-01) -	5.237E-01(7.62E-01) -	1.934E-01(2.09E-01)
DTLZ2	5.593E-02(1.40E-03) ≈	6.067E-02(1.78E-03) -	5.657E-02(1.16E-03) -	5.809E-02(1.03E-03) -	5.568E-02(1.13E-03)
DTLZ3	3.544E+00(3.60E+00) -	1.361E+00(2.96E+00) -	5.785E-01(6.92E-01) -	7.946E-01(8.25E-01) -	3.992E-01(6.85E-01)
DTLZ4	1.081E-01(1.09E-01) -	4.686E-02(5.46E-03) -	8.044E-02(8.35E-02) -	5.047E-02(7.09E-03) -	4.106E-02(2.53E-03)
DTLZ5	4.720E-03(2.81E-04) -	5.119E-03(2.82E-04) -	4.803E-03(3.54E-04) -	5.217E-03(5.31E-04) -	4.449E-03(1.46E-04)
DTLZ6	5.175E-01(1.02E-01) -	4.051E-03(1.41E-04) ≈	4.033E-03(1.32E-04) ≈	4.059E-03(1.13E-04) ≈	4.018E-03(1.10E-04)
DTLZ7	1.157E-01(1.53E-01) -	6.047E-02(1.47E-03) -	2.298E-01(2.03E-01) -	6.950E-02(5.07E-02) -	5.789E-02(1.30E-03)
WFG1	1.364E+00(1.80E-01) -	1.251E-02(3.94E-04) ≈	1.244E-02(3.37E-04) ≈	1.249E-02(3.88E-04) ≈	9.581E-02(1.90E-01)
WFG2	1.781E-01(7.99E-02) -	3.765E-02(4.56E-02) ≈	2.104E-01(5.79E-02) -	8.387E-02(7.92E-02) -	2.460E-02(2.48E-02)
WFG3	1.272E-02(1.24E-03) -	1.214E-02(1.97E-04) -	1.200E-02(2.75E-04) -	1.187E-02(1.82E-04) -	1.170E-02(1.09E-04)
WFG4	1.247E-02(5.03E-04) +	1.296E-02(4.16E-04) -	1.261E-02(3.95E-04) ≈	1.253E-02(4.24E-04) ≈	1.268E-02(3.63E-04)
WFG5	6.658E-02(3.84E-04) -	6.549E-02(5.63E-03) ≈	6.643E-02(2.03E-04) ≈	6.663E-02(2.56E-04) -	6.536E-02(5.71E-03)
WFG6	2.181E-02(6.04E-03) -	1.638E-02(7.87E-03) -	2.200E-02(7.38E-03) -	2.032E-02(8.78E-03) -	1.409E-02(4.43E-03)
WFG7	1.735E-02(5.92E-03) -	1.286E-02(3.52E-04) -	1.291E-02(5.68E-04) -	1.263E-02(4.05E-04) ≈	1.249E-02(3.49E-04)
WFG8	7.671E-02(4.77E-02) -	2.676E-02(3.80E-03) +	4.575E-02(4.44E-02) -	3.714E-02(4.47E-02) +	3.825E-02(4.92E-03)
WFG9	1.360E-02(1.51E-03) -	1.302E-02(5.15E-04) -	1.254E-02(4.61E-04) ≈	1.286E-02(5.99E-04) -	1.246E-02(3.37E-04)
-/+/~	14/1/1	11/1/4	11/0/5	11/1/4	

“-”, “+”, and “≈” indicate that the results of the algorithm are worse than, better than, and similar to that of AIMA using

Wilcoxon’s rank sum test with $\alpha = 0.05$

C) Performance comparisons of AIMA with four competitive MOIAs

In the above subsections, it is experimentally validated that AIMA is able to outperform some state-of-the-art multi-objective optimization algorithms and their DE-based variants in solving the WFG and DTLZ test problems. To further study the performance of AIMA, AIMA is compared to four

competitive MOIAs in this subsection, *i.e.*, NNIA [21], IMADE [37], DMMO [30], and HEIA [31], in tackling the WFG and DTLZ test problems. In these algorithms, IMADE, DMMO, and HEIA are three improved variants of NNIA with a single DE strategy. It is noted that the compared MOIAs also adopt the same archive truncation approach of SPEA2 in our experiments, which highlights the influences of our proposed approaches and makes the comparisons fairer. All the IGD results of the compared MOIAs are listed in **Table 4**. As observed from the last row of **Table 4**, AIMA yields better results than NNIA, IMADE, DMMO, and HEIA on 14, 11, 11, and 11 out of 16 test problems, respectively. On the contrary, AIMA is respectively outperformed by NNIA, IMADE, and HEIA on only 1 test problem. These IGD results substantially validate that the three DE strategies and the adaptive DE strategy selection in AIMA can significantly enhance the optimization performance of MOIA, since the main difference among these MOIAs relies on the adopted evolutionary operators, such as SBX in NNIA, both of SBX and DE in DMMO, IMADE and HEIA, and three DE strategies in our algorithm.

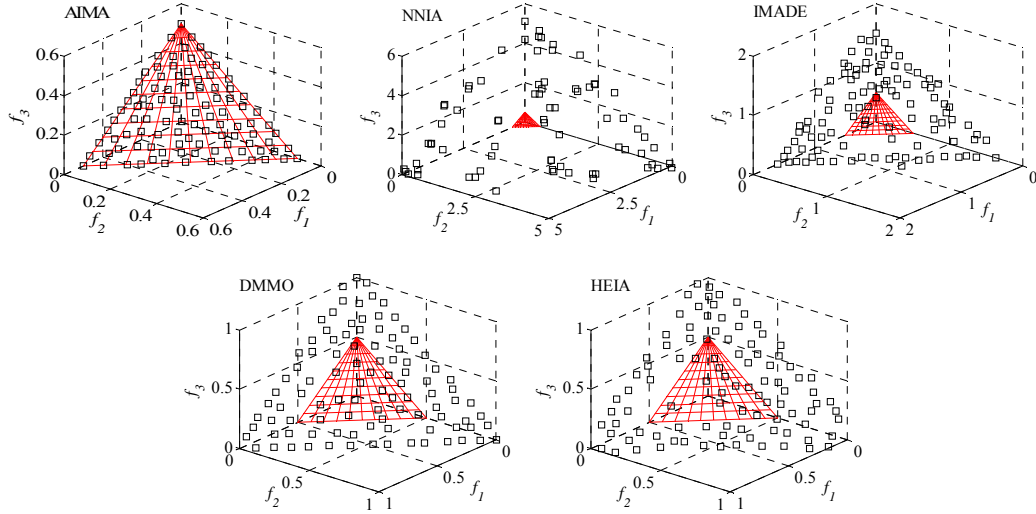
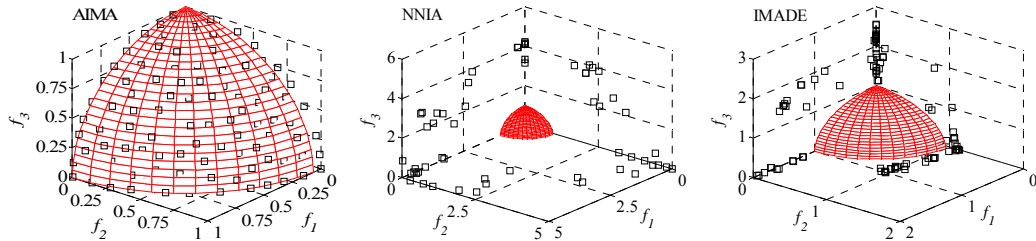


Fig. 7 The nondominated solution set with the median IGD value found by each compared MOIA on DTLZ1



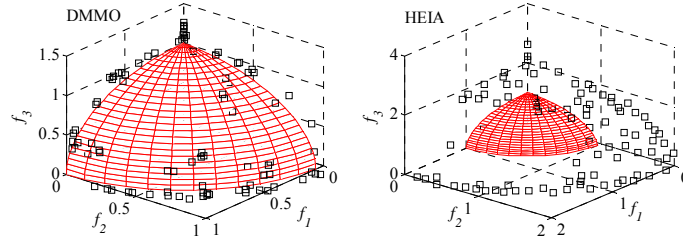
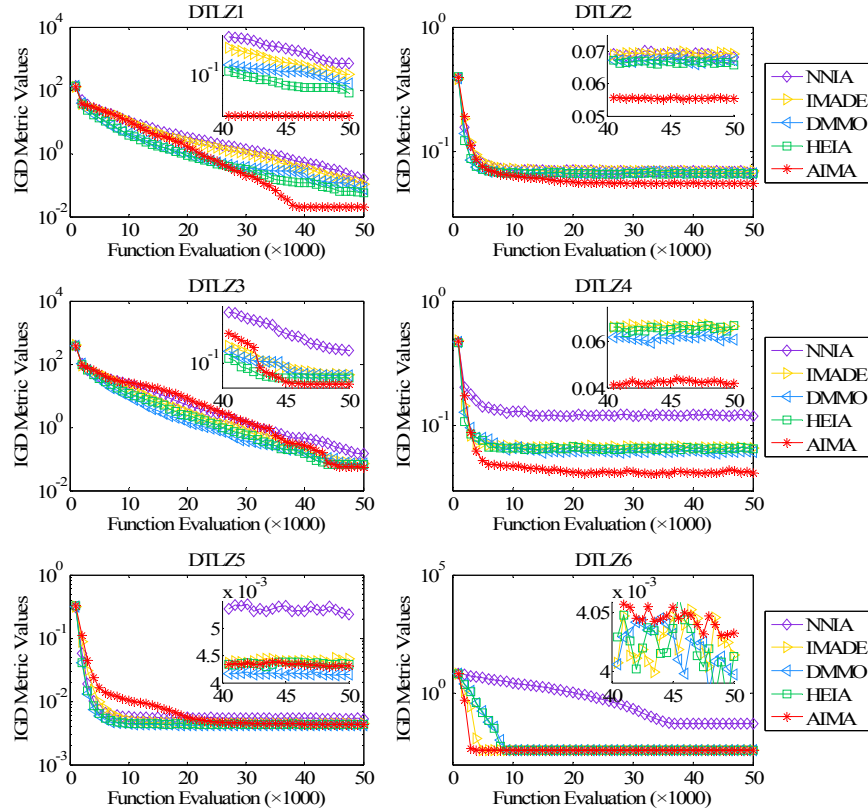


Fig. 8 The nondominated solution set with the median IGD value found by each compared MOIA on DTLZ3

To visually show the advantages of AIMA over these compared MOIAs, the final nondominated solution set with the median IGD value in 30 runs is plotted in **Figs 7-8**, respectively for DTLZ1 and DTLZ3. From the plots in **Fig. 7**, it is observed that the solutions found by AIMA can closely approach and evenly cover the true **PF** of DTLZ1, while other four compared MOIAs only find scattered solutions distributed far away from the true **PF**. Regarding **Fig. 8**, the solutions obtained by AIMA are also with good diversity and coverage for the true **PF** of DTLZ3. However, other compared MOIAs can neither converge to the true **PF** nor keep a good population diversity. Moreover, to show the evolutionary progress of all the compared MOIAs, **Fig. 9** gives the plots of the evolutionary curves of the average IGD values from 30 runs versus the number of function evaluations on each DTLZ test problem. It also illustrates the superiority of AIMA over the compared MOIAs, on tackling most of the DTLZ test problems.



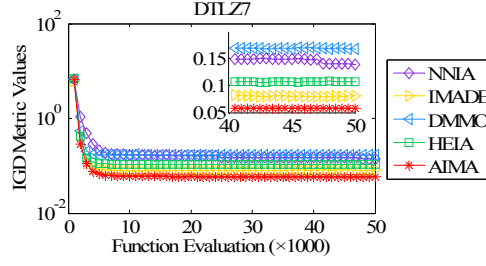


Fig. 9 Evolutionary curves of the average IGD values from 30 runs versus the number of function evaluations on the DTLZ test problems

4.4.2. Further Discussions of AIMA

In the above subsections, it is experimentally shown that AIMA outperforms five state-of-the-art multi-objective optimization algorithms and their DE-based variants, as well as four competitive MOIAs. This superior performance, as we expect, is mainly brought by using the multiple DE strategies and the corresponding adaptive DE strategy selection. Here, the following experiments are further conducted to verify our above opinion. Four variants of AIMA, namely, AIMA-DE₁, AIMA-DE₂, AIMA-DE₃, and AIMA-EP, are adopted for performance comparison. Regarding AIMA-DE_{*i*} (*i* = 1, 2, 3), the single DE_{*i*} strategy defined in Section 3.1 is used, while AIMA-EP chooses one from these three DE strategies with an equal probability, *i.e.*, 1/3, at each generation. To make a fair comparison, all the parameter settings in the AIMA variants are set the same as that in AIMA. Their IGD comparison results on all the WFG and DTLZ test problems are provided in **Table 5**.

Table 5 IGD comparison results of AIMA and four variants

	AIMA-DE ₁ Mean(Std)	AIMA-DE ₂ Mean(Std)	AIMA-DE ₃ Mean(Std)	AIMA-EP Mean(Std)	AIMA Mean(Std)
DTLZ1	1.634E+01(8.66E+00) -	1.074E-01(2.37E-01) +	3.270E-01(4.32E-01) ≈	1.162E-01(2.31E-01) ≈	1.934E-01(2.09E-01)
DTLZ2	6.420E-02(1.98E-03) -	5.560E-02(1.12E-03) ≈	5.573E-02(9.28E-04) ≈	5.716E-02(1.27E-03) -	5.568E-02(1.13E-03)
DTLZ3	1.476E+01(9.99E+00) -	5.494E-01(1.48E+00) -	7.622E-01(1.05E+00) -	4.383E-01(6.64E-01) ≈	3.992E-01(6.85E-01)
DTLZ4	4.856E-02(4.36E-03) -	4.250E-02(4.56E-03) ≈	4.387E-02(4.41E-03) -	5.515E-02(6.89E-02) -	4.106E-02(2.53E-03)
DTLZ5	1.006E-02(1.25E-03) -	4.480E-03(1.72E-04) ≈	4.309E-03(9.74E-05) -	5.013E-03(4.29E-04) -	4.449E-03(1.46E-04)
DTLZ6	4.042E-03(1.26E-04) ≈	4.015E-03(1.15E-04) ≈	4.015E-03(1.25E-04) ≈	4.087E-03(1.19E-04) -	4.018E-03(1.10E-04)
DTLZ7	6.011E-02(1.29E-03) -	7.813E-02(7.06E-02) -	6.709E-02(5.20E-02) -	5.958E-02(1.43E-03) -	5.789E-02(1.30E-03)
WFG1	1.092E-01(1.98E-01) ≈	3.627E-01(3.08E-01) -	2.198E-01(3.28E-01) -	2.064E-01(2.43E-01) -	9.581E-02(1.90E-01)
WFG2	1.160E-02(1.01E-02) +	5.970E-02(3.94E-02) -	1.602E-01(9.16E-02) -	2.835E-02(2.64E-02) ≈	2.460E-02(2.48E-02)
WFG3	1.349E-02(4.83E-04) -	1.180E-02(1.78E-04) -	1.186E-02(1.63E-04) -	1.199E-02(1.80E-04) -	1.170E-02(1.09E-04)
WFG4	1.314E-02(5.26E-04) -	1.248E-02(5.00E-04) +	1.255E-02(3.78E-04) ≈	1.274E-02(3.70E-04) ≈	1.268E-02(3.63E-04)
WFG5	6.980E-02(2.91E-03) -	6.665E-02(3.21E-04) -	6.642E-02(1.67E-04) ≈	6.544E-02(5.67E-03) ≈	6.536E-02(5.71E-03)
WFG6	1.385E-02(4.90E-04) +	2.015E-02(8.98E-03) -	2.148E-02(6.57E-03) -	1.967E-02(1.53E-02) -	1.409E-02(4.43E-03)
WFG7	1.365E-02(5.33E-04) -	1.243E-02(3.43E-04) ≈	1.263E-02(5.33E-04) ≈	1.284E-02(3.10E-04) -	1.249E-02(3.49E-04)
WFG8	4.214E-02(4.07E-03) -	4.833E-02(3.69E-03) -	6.263E-02(7.31E-02) -	3.625E-02(3.18E-03) ≈	3.825E-02(4.92E-03)
WFG9	1.649E-02(9.43E-04) -	1.288E-02(5.68E-04) -	1.315E-02(6.58E-04) -	1.278E-02(3.71E-04)	1.246E-02(3.37E-04)
-/+ / ≈	12/2/2	9/2/5	10/0/6	10/0/6	

“-”, “+”, and “ \approx ” indicate that the results of the algorithm are worse than, better than, and similar to that of AIMA using Wilcoxon’s rank sum test with $\alpha = 0.05$

Compared to the variants with a single DE strategy, AIMA performs better than each of them on most of the test MOPs, even though AIMA-DE₁ and AIMA-DE₂ outperform AIMA on a few problems, *e.g.*, WFG2 and WFG6. In detail, AIMA obtains better IGD results than AIMA-DE₁, AIMA-DE₂, AIMA-DE₃, and AIMA-EP respectively on 12, 9, 10, and 10 out of 16 test problems. This indicates that the three DE strategies in AIMA can cooperate with each other to yield the overall better performance when solving different kinds of MOPs. Although AIMA-EP also adopts three DE strategies, AIMA still achieves better results than AIMA-EP on 10 out of 16 test problems. This confirms that our adaptive DE strategy selection is reasonable and effective to solve various MOPs.

As our adaptive DE strategy selection is designed based on the elimination ratio of nondominated solutions in the external archive, this approach can also be applied to other Pareto-based multi-objective optimization algorithms. To further investigate the performance of our proposed method, it is applied to NSGA-II and SPEA2. These two variants are respectively called NSGA-II-MDE and SPEA2-MDE in this paper. NSGA-II-MDE is compared to the original NSGA-II and its DE-based variant (NSGA-II-DE). Similar comparison experiments are also conducted for SPEA2-MDE. Their IGD comparison results on 16 test problems are provided in **Table 6**.

As observed in the last row of **Table 6**, NSGA-II-MDE performs better than NSGA-II and NSGA-II-DE respectively on 12 and 11 test problems; on the contrary, NSGA-II-MDE is only outperformed by NSGA-II and NSGA-II-DE respectively on 3 and 4 test problems. Similarly, SPEA2-MDE outperforms SPEA2 and SPEA2-DE respectively on 12 and 8 test problems, while SPEA2-MDE is only outperformed by SPEA2 and SPEA2-DE on 3 test problems. These two algorithms (NSGA-II-MDE and SPEA2-MDE) show significant superiority over their original algorithms (NSGA-II and SPEA2), and their single-DE variants (NSGA-II-DE and SPEA2-DE). Based on these observations, it is reasonable to conclude that our proposed adaptive DE strategy selection is able to improve the overall performance of Pareto-based multi-objective optimization algorithms.

Table 6 IGD comparison results of NSGA-II, SPEA2 and their DE-based variants

		NSGA-II	NSGA-II-DE	NSGA-II-MDE	SPEA2	SPEA2-DE	SPEA2-MDE
DTLZ1	Mean	8.163E-01 +	5.314E-02 +	3.305E+00	4.958E-01 -	4.474E-02 +	1.208E-01
	Std	6.04E-01	8.63E-02	1.00E+00	3.32E-01	5.50E-02	2.21E-01
DTLZ2	Mean	6.829E-02 -	6.386E-02 +	6.683E-02	5.495E-02 +	5.616E-02 +	5.897E-02
	Std	2.81E-03	2.17E-03	1.54E-03	1.01E-03	1.32E-03	1.37E-03
DTLZ3	Mean	2.772E+00 +	6.518E-02 +	9.166E+00	2.124E+00 -	1.127E-01 \approx	1.841E-01
	Std	2.06E+00	3.50E-03	2.80E+00	1.47E+00	1.80E-01	3.06E-01
DTLZ4	Mean	6.390E-02 +	6.359E-02 +	6.836E-02	1.166E-01 -	4.916E-02 -	4.850E-02
	Std	7.28E-03	6.01E-03	5.37E-03	1.15E-01	2.08E-02	4.61E-03
DTLZ5	Mean	5.528E-03 -	5.302E-03 -	4.351E-03	4.406E-03 +	5.334E-03 +	8.002E-03
	Std	2.78E-04	2.13E-04	1.10E-04	1.68E-04	3.04E-04	7.17E-04
DTLZ6	Mean	5.332E-01 -	5.865E-03 -	4.039E-03	5.170E-01 -	4.057E-03 \approx	4.067E-03

	Std	5.18E-02	3.68E-04	1.28E-04	4.25E-02	1.24E-04	1.65E-04
DTLZ7	Mean	7.602E-02 \approx	7.399E-02 \approx	7.432E-02	1.219E-01 -	5.964E-02 \approx	5.986E-02
	Std	3.94E-03	3.89E-03	4.12E-03	1.59E-01	1.57E-03	1.40E-03
WFG1	Mean	1.561E+00 -	1.104E-01 -	1.244E-02	1.787E+00 -	1.045E-01 -	1.257E-02
	Std	1.47E-01	1.95E-01	2.50E-04	7.16E-02	2.26E-01	3.74E-04
WFG2	Mean	1.400E-01 -	4.995E-02 -	1.016E-02	1.255E-01 -	6.130E-02 -	1.068E-02
	Std	8.46E-02	4.36E-02	4.28E-04	8.05E-02	1.44E-02	8.26E-04
WFG3	Mean	1.559E-02 -	1.399E-02 -	1.187E-02	1.575E-02 -	1.218E-02 -	1.191E-02
	Std	9.72E-04	5.15E-04	1.27E-04	2.87E-03	2.04E-04	1.01E-04
WFG4	Mean	1.385E-02 -	1.277E-02 -	1.069E-02	1.239E-02 +	1.317E-02 -	1.283E-02
	Std	6.40E-04	8.00E-04	2.89E-04	3.91E-04	6.25E-04	6.35E-04
WFG5	Mean	6.794E-02 -	6.800E-02 -	6.623E-02	6.693E-02 -	6.614E-02 -	6.484E-02
	Std	4.52E-04	4.86E-04	3.05E-03	2.79E-04	5.56E-03	7.50E-03
WFG6	Mean	2.180E-02 -	1.834E-02 -	1.270E-02	2.476E-02 -	1.400E-02 \approx	1.276E-02
	Std	6.81E-03	8.44E-03	2.55E-04	1.27E-02	5.02E-03	2.45E-04
WFG7	Mean	1.968E-02 -	1.447E-02 -	1.247E-02	2.254E-02 -	1.277E-02 \approx	1.265E-02
	Std	5.60E-03	8.54E-04	2.36E-04	8.80E-03	3.47E-04	3.00E-04
WFG8	Mean	6.489E-02 -	7.031E-02 -	5.814E-02	1.045E-01 -	1.633E-01 -	8.031E-02
	Std	9.14E-03	8.77E-03	7.64E-03	4.26E-02	3.38E-02	8.69E-03
WFG9	Mean	1.595E-02 -	1.522E-02 -	1.278E-02	1.295E-02 \approx	1.314E-02 -	1.282E-02
	Std	1.60E-03	8.79E-04	3.27E-04	6.06E-04	4.64E-04	4.20E-04
-/+/ \approx		12/3/1	11/4/1		12/3/1	8/3/5	

“-”, “+”, and “ \approx ” indicate that the results of the algorithm are worse than, better than, and similar to that of NSGA-II-MDE or SPEA2-MDE using Wilcoxon’s rank sum test with $\alpha = 0.05$

4.4.3. Parameter sensitivity analysis in AIMA

When designing AIMA, some parameters, such as LP and LR , are introduced. In order to study their impact on AIMA, different combinations of parameter values are examined on six test problems with different properties, *i.e.*, DTLZ1 (linear PF and many local PF s), DTLZ3 (concave PF and many local PF s), DTLZ5 (degenerate PF), WFG1 (mixed and biased PF), WFG2 (discontinuous PF), and WFG9 (deceptive PF).

In Section 3.2, LP is the number of previous generations used for calculating the average elimination ratio ($avgR_{tiao}$), while LR is the threshold of judging whether stagnation has occurred. To study their impact on AIMA, seven representative values for LP (*i.e.*, 1, 5, 10, 20, 30, 50, and 100) and for LR (*i.e.*, 5, 10, 15, 20, 30, 50, and 100) are used. In **Fig. 10**, the IGD values of DTLZ1, DTLZ3, DTLZ5, WFG1, WFG2, and WFG9 are obtained using AIMA with different combinations of LP and LR . Other parameters settings are the same as listed in Section 4.3. It is noted that also 30 independent runs have been conducted for each combination on each test problem.

As observed from **Fig. 10**, a small LR (5 to 15) is good for DTLZ1 and DTLZ3, while for the remaining problems, the IGD values change a little when LP and LR change. Only for DTLZ5 and WFG9, the performance of AIMA will be significantly worsened when some extremely small values of LR and LP are used. Thus, it is evident that AIMA is less sensitive to the settings of LP and LR except for some

extreme values. These figures also indicate that our settings for LP and LR are reasonable and effective.

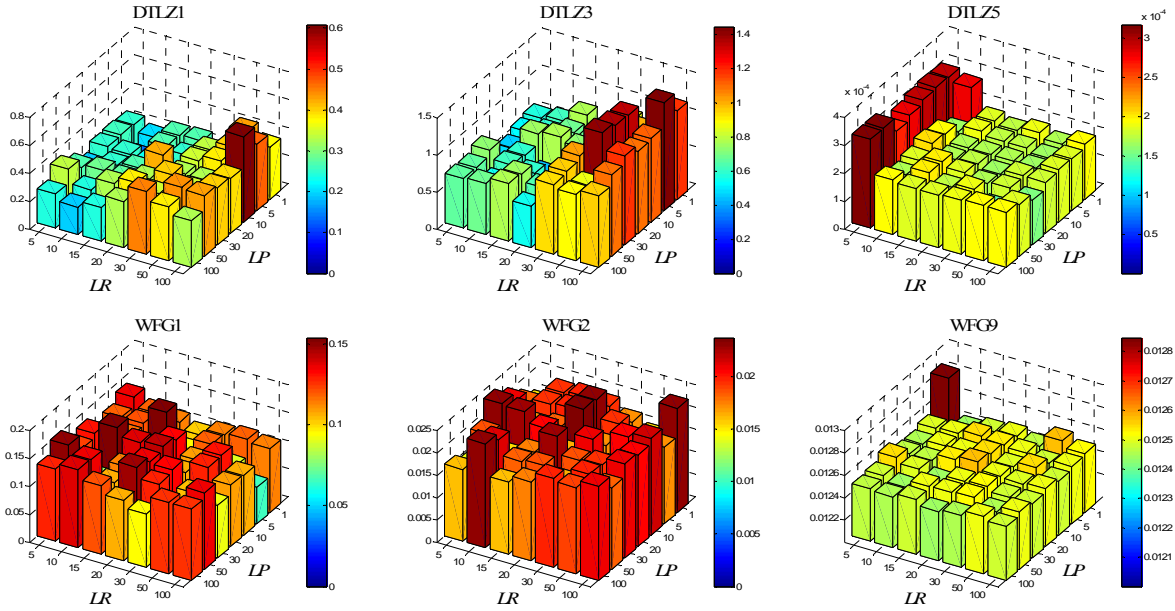


Fig.10 IGD values for DTLZ1, DTLZ3, DTLZ5, WFG1, WFG2, and WFG9 with different combinations of LP and LR

5. Conclusions and Future Work

In this paper, an adaptive immune-inspired multi-objective algorithm with multiple differential evolution strategies (AIMA) was proposed. In this approach, three DE strategies with distinct features are embedded into MOIAs. At each generation, one of them is adaptively selected to be used based on the current search stage. This adaptive DE strategy selection makes three DE strategies effectively cooperate with each other, which significantly improves the search capability and enhances the population diversity. The experimental results have revealed that AIMA is able to successfully solve different types of MOPs, *i.e.*, the DTLZ and WFG test problems. When compared to five state-of-the-art multi-objective optimization algorithms (*i.e.*, NSGA-II, SPEA2, MOEA/D, SMS-EMOA, and CMPSO) and their DE-based variants (*i.e.*, NSGA-II-DE, SPEA2-DE, MOEA/D-DE, SMS-EMOA-DE, and CMODE), and four competitive MOIAs (*i.e.*, NNIA, IMADE, DMMO, and HEIA), AIMA was shown to have a superior overall performance in finding an approximation set, which is close to the true PF and distributed evenly along the true PF . Moreover, the advantages of incorporating multiple DE strategies and the corresponding adaptive DE strategy selection are experimentally confirmed, and the impact of some parameters in AIMA is also investigated.

Our future study will aim to further enhance the performance of AIMA, and extend it to tackle more complicated MOPs, such as large scale problems on both the objective and decision spaces [7][54].

Furthermore, the extension of AIMA to solve some real-world applications will also be studied in our future work.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grants 61402291 and 61672358, the Natural Science Foundation of Guangdong Province under Grants 2017A030313338, and CONACyT under Grant 221551.

Reference

- [1] M. Antonelli, P. Ducange, F. Marcelloni, A fast and efficient multi-objective evolutionary learning scheme for fuzzy rule-based classifiers, *Inform. Sci.* 283 (2014) 36-54.
- [2] N. Beume, B. Naujoks, M. Emmerich, SMS-EMOA: Multiobjective selection based on dominated hypervolume, *Eur. J. Oper. Res.* 181 (2007) 1653–1669.
- [3] S.F.M. Burnet. The clonal selection theory of acquired immunity. Nashville: Vanderbilt University Press. (1959).
- [4] L.C.T. Bezerra, M. López-Ibáñez, T. Stützle. To DE or Not to DE? Multi-objective Differential Evolution Revisited from a Component-Wise Perspective, in: Gaspar-Cunha A., Henggeler Antunes C., Coello C. (eds) *Evolutionary Multi-Criterion Optimization. EMO 2015. Lecture Notes in Computer Science*, vol 9018. Springer, Cham.
- [5] P.A.N. Bosman, and D. Thierens. The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Trans. Evolut. Comput.* 7 (2003) 174-188.
- [6] J.Y. Chen, Q.Z. Lin, Z. Ji, A hybrid immune multiobjective optimization algorithm. *Eur. J. Oper. Res.* 204 (2010), 294-302.
- [7] R. Cheng, Y. Jin, M. Olhofer, B. Sendhoff, Test problems for large-scale multiobjective and many-objective optimization. *IEEE Trans. Cybern.*, in press, doi: 10.1109/TCYB.2016.2600577.
- [8] C.A. Coello Coello, N.C. Cortes, An Approach to Solve Multiobjective Optimization Problems Based on an Artificial Immune System, in: *Proceedings of the first international Conference on artificial immune systems*, 2002, pp. 212-221.
- [9] L.Z Cui, G.H. L, Q.Z. Lin, J.Y. Chen, N. Lu, Adaptive differential evolution algorithm with novel mutation strategies in multiple sub-populations, *Comput. Oper. Res.* 67 (2016) 155–173.
- [10] V. Cutello, G. Narzisi, G. Nicosia, A Class of Pareto Archived Evolution Strategy Algorithms Using Immune Inspired Operators for Ab-Initio Protein Structure Prediction, in: *Workshops on Applications of Evolutionary Computation 2005, LNCS*, vol. 3449, pp. 54-63. Springer Berlin Heidelberg.

- [11]S. Das, P. N. Suganthan, Differential evolution: a survey of the state-of-the-art, *IEEE Trans. Evolut. Comput.* 15 (2011) 4-31.
- [12]K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Trans. Evolut. Comput.* 6 (2002) 182–197.
- [13]K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part I: Solving problems with box constraints, *IEEE Trans. Evolut. Comput.* 18 (2014) 557–601.
- [14]K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable test problems for evolutionary multi-objective optimization,” in: *evolutionary multiobjective optimization, advanced information and knowledge processing series*. Berlin Springer, 2005, pp. 105–145.
- [15]K. Deb, M. Goyal, A combined genetic adaptive search (geneas) for engineering design. *Comput. Sci. Inform.* 26 (1996) 30-45.
- [16]K. Deb, R.B. Agrawal, Simulated binary crossover for continuous search space, *Complex Syst.* 9 (1995) 115–148.
- [17]J.J. Durillo, A.J. Nebro, E. Alba, The jMetal framework for multi-objective optimization: Design and architecture, in: *Proceedings of the 2010 IEEE Congress on Evolutionary Computation (CEC)*, Barcelona, Spain, 2010, pp. 1–8.
- [18]F. Freschi, Repetto, Vis: an artificial immune network for multi-objective optimization, *Eng. Optimiz.* 38 (2006) 975-996.
- [19]J.Q.Gao, J. Wang, WBMOAIS: a novel artificial immune system for multiobjective optimization, *Comput. Oper. Res.* 37(2010), 50-61.
- [20]K.Z. Gao, P.N. Suganthan, Q.K. Pan, T.J. Chua, T.X. Cai, C.S. Chong, Pareto-based grouping discrete harmony search algorithm for multi-objective flexible job shop scheduling, *Inform. Sci.* 289 (2014) 76-90.
- [21]M.G. Gong, L.C. Jiao, H.F. Du, L.F. Bo, Multi-objective immune algorithm with nondominated neighbor-based selection, *Evol. Comput.* 16 (2008), 225-255.
- [22]Z.H. Hu, A multiobjective immune algorithm based on a multiple-affinity model, *Eur. J. Oper. Res.* 202 (2010) 60-72.
- [23]S. Huband, L. Barone, L. While, P. Hingston, A scalable multi-objective test problem toolkit, in: C.A.C. Coello, A.H. Aguirre, E. Zitzler (Eds.), *Proceedings of Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science*, 3410, Guanajuato, Mexico, 2005, pp. 280–295.
- [24]L.C. Jiao, M.G. Gong, R.H. Shang, H.F. Du, B. Lu, Clonal selection with immune dominance and

- energy based multiobjective optimization, in: Coello Coello C.A., Hernández Aguirre A., Zitzler E. (eds) *Evolutionary Multi-Criterion Optimization. EMO 2005. Lecture Notes in Computer Science*, 2005, vol 3410, pp. 474–489. Springer, Berlin, Heidelberg.
- [25] Y. Lei, M.G. Gong, J. Zhang, W. Li, L.C. Jiao, Resource allocation model and double-sphere crowding distance for evolutionary multi-objective optimization, *Eur. J. Oper. Res.* 234 (2014) 197-208.
- [26] H. Li, Q. Zhang, Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II, *IEEE Trans. Evolut. Comput.* 13 (2009) 284–302.
- [27] K. Li, A. Fialho, S. Kwong, Q.F. Zhang, Adaptive operator selection with bandits for multiobjective evolutionary algorithm based decomposition, *IEEE Trans. Evolut. Comput.* 19 (2014) 114–130.
- [28] K. Li, Q. Zhang, S. Kwong, M. Li, R. Wang, Stable matching-based selection in evolutionary multiobjective optimization, *IEEE Trans. Evolut. Comput.* 18 (6) (2014) 909–923.
- [29] M.Q. Li, S.X. Yang, X.H. Liu, Shift-based density estimation for Pareto-based algorithms in many-objective optimization, *IEEE Trans. Evolut. Comput.* 18 (3) (2014) 348–365.
- [30] 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.* 35 (2015) 161–174.
- [31] Q.Z. Lin, J.Y. Chen, Z.H. Zhan, W.N. Chen, C.A. Coello Coello, Y.L. Yin, C.M. Lin, J. Zhang, A hybrid evolutionary immune algorithm for multiobjective optimization problems, *IEEE Trans. Evolut. Comput.* 20 (2016) 711-729.
- [32] Q.Z. Lin, J.Y. Chen, A novel micro-population immune multiobjective optimization algorithm, *Comput. Oper. Res.* 40 (2013) 1590–1601.
- [33] Q.Z. Lin, Q.L. Zhu, P.Z. Huang, J.Y. Chen, Z. Ming, J.P. Yu, A novel hybrid multi-objective immune algorithm with adaptive differential evolution, *Comput. Oper. Res.* 65 (2015) 95–111.
- [34] Q.Z. Lin, Z.W. Liu, Q. Yan, Z.H. Du, C.A. Coello Coello, Z.P. Liang, W.J. Wang, J.Y. Chen, Adaptive composite operator selection, and parameter control for multiobjective evolutionary algorithm, *Inform. Sci.* 339 (2016) 332-352.
- [35] K. Mithilesh, C. Guria, The elitist non-dominated sorting genetic algorithm with inheritance (i-NSGA-II) and its jumping gene adaptations for multi-objective optimization, *Inform. Sci.* 382–383 (2016) 15-37.
- [36] N. Nedjah, L.D.M Mourelle, Evolutionary multi-objective optimisation: a survey, *Int. J. Bio-Inspired Comput.* 7 (2015) 1-25.
- [37] Y.T. Qi, Z.T. Hou, M.L. Yin, H.L. Sun, J.B. Huang, An immune multi-objective optimization

- algorithm with differential evolution inspired recombination, *Appl. Soft Comput.* 29 (2015) 395–410.
- [38] A.K. Qin, V.L. Huang, P.N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, *IEEE Trans. Evolut. Comput.* 13 (2009) 398–417.
- [39] K. Sethanan, W. Neungmatcha, Multi-objective particle swarm optimization for mechanical harvester route planning of sugarcane field operations, *Eur. J. Oper. Res.* 252 (2016) 969–984.
- [40] R.H. Shang, L.C. Jiao, F. Liu, W.P. Ma, A novel immune clonal algorithm for MO problems, *IEEE Trans. Evol. Comput.* 16 (2012) 35–50.
- [41] J. Shi, M. Gong, W. Ma, L. Jiao, A multipopulation coevolutionary strategy for multiobjective immune algorithm, *Sci. World. J.* (2014).
- [42] R. Storn, On the usage of differential evolution for function optimization, in *Proc. Biennial Conf. North Amer. Fuzzy Inf. Process. Soc.*, Berkeley, CA, 1996, pp. 519–523.
- [43] K.C. Tan, C.K. Goh, A.A. Mamun, E.Z. Ei, An evolutionary artificial immune system for multi-objective optimization, *Eur. J. Oper. Res.* 187 (2008) 371–392.
- [44] T. Tušar, B. Filipič, Differential Evolution versus Genetic Algorithms in Multiobjective Optimization, in: *Proceedings of the 4th international conference on Evolutionary multi-criterion optimization*, Matsushima, Japan, 2007, pp. 257–271.
- [45] S.M. Venske, R.A. Gonçalves, M.R. Delgado, ADEMO/D: multiobjective optimization by an adaptive differential evolution algorithm, *Neurocomputing*, 127 (2014) 65–77.
- [46] J.H. Wang, W.W. Zhang, J. Zhang, Cooperative differential evolution with multiple populations for multiobjective optimization, *IEEE Trans. Cybern.* 46 (2015) 2848–2861.
- [47] Y. Wang, Z.X. Cai, Q.F. Zhang, Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Trans. Evolut. Comput.* 15 (2011) 55–66.
- [48] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, *IEEE Trans. Evol. Comput.* 1 (1997) 67–82.
- [49] E.Y.C. Wong, H.S.C. Yeung, H.Y.K. Lau, Immunity-based hybrid evolutionary algorithm for multi-objective optimization in global container repositioning, *Eng. Appl. Artif. Intell.* 22 (2009) 842–854.
- [50] J. Yoo, P. Hajela, Immune network simulations in multicriterion design, *Struct. Optim.* 18 (1999) 85–94.
- [51] 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. Cybern* 43 (2013) 445–463.
- [52] Q.F. Zhang, H. Liu, MOEA/D: A multiobjective evolutionary algorithm based on decomposition,

IEEE Trans. Evolut. Comput. 11 (2007) 712–731.

- [53] Q.F. Zhang, W.D. Liu, H. Li, The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances, in: Proceedings of the Eleventh conference on Congress on Evolutionary Computation, Trondheim, Norway, 2009, pp. 203-208.
- [54] X. Zhang, Y. Tian, R. Cheng, Y. Jin, A decision variable clustering based evolutionary algorithm for large-scale many-objective optimization, IEEE Trans. Evolut. Comput. (2016), in press, doi:10.1109/TEVC.2016.2600642.
- [55] W.J. Zhang, X.F. Xie, DEPSO: Hybrid particle swarm with differential evolution operator, in Proc. IEEE Int. Conf. Syst. Man Cybern., Washington, DC, 2003, pp. 3816–3821.
- [56] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the strength Pareto evolutionary algorithm, Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische Informatik und Kommunikationsnetze (TIK), 2001.