An adaptive immune-inspired multi-objective algorithm with multiple differential evolution strategies

Qiuzhen Lin, Yueping Ma, Jianyong Chen, Qingling Zhu, Carlos A. Coello Coello, Ka-Chun Wong, Fei Chen

AR T I C L E   I N F O

Article history:
Received 3 February 2017
Revised 1 November 2017
Accepted 16 November 2017
Available online 20 November 2017

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

A B S T R A C T

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, 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.

© 2017 Published by Elsevier Inc.

1. Introduction

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 [36], job shop scheduling [18], 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.

* Corresponding author.
E-mail address: fchen@szu.edu.cn (F. Chen).

https://doi.org/10.1016/j.ins.2017.11.030
0020-0255/© 2017 Published by Elsevier Inc.
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 [11], SPEA2 [53], and MOEA/D [49]. Regarding NSGA-II [11], 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 [53], 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 [49], a MOP is decomposed into a set of SOPs and then these SOPs are solved on 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 were used in [32] to speed up convergence towards the global PF, while a reference point based approach was introduced in [12] 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 [26] 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 [50] to dynamically assign the computational resources based on the difficulties of sub-problems, and an economic stable matching model (STM) was designed in [25] to guarantee the balanced match of sub-problems and solutions by mutual preferences. A detailed review of MOEAs can be found in [33]. Especially, some of MOEAs were enhanced based on the use of differential evolution (DE), since DE shows excellent search abilities. The experiments conducted by [4] and [41] 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 [23] and CMODE [43], a specific DE operator was used to substitute the original evolutionary operators of MOEA/D [49] and CPSO [48], giving rise to a better optimization performance; Moreover, in ADEMO/D [42], MOEA/D-FRRMAB [24], and MOEA/D-CDE [31], 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 [3,6,19,37], as inspired from the biologic immune system. The nondominated neighbor-based immune algorithm (NNIA) [19] 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 [6,22,29]. 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 [11]. 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 [11] and most of other MOEAs [12,33,43], 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 [22,38], 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 [27,28,30,34] 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 [14], as the DE operators normally show a better search capability than SBX [28]. Therefore, embedding them can help to enhance the population diversity of MOIAs. However, all the MOIAs proposed in [27,28,30,34] 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 [10]. 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 [24,31,42], 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 [19]) 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 [20] and DTLZ [13]) with various features, AIMA shows evident advantages over five state-of-the-art multi-objective optimization algorithms (i.e., NSGA-II [11], SPEA2 [53], MOEA/D [49], SMS-EMOA [2], and CPMPO [48]) and their DE-based variants [23,43], and four competitive MOIAs (i.e., NNIA [19], IMADE [34], DMMO [27], and HEIA [28]).

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:

\[
\min_{x \in \Omega} F(x) = (f_1(x), f_2(x), ..., f_m(x))^T
\]

where \( x = (x_1, x_2, ..., x_n) \in \Omega \) is a decision vector with \( n \) dimensions, \( \Omega \) is the feasible region of decision space, the mapping function \( F: \Omega \rightarrow \mathbb{R}^m \) defines \( m \) objective functions and \( \mathbb{R}^m \) 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 \( x \succ y \)) if and only if

\[
\forall i \in \{1, 2, ..., m\} : f_i(x) \leq f_i(y) \land \exists j \in \{1, 2, ..., m\} : f_j(x) < f_j(y) \]

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

\[
\neg \exists y \in \Omega : y \succ x
\]

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

\[
PS = \{ x | \neg \exists y \in \Omega : y \succ x \}
\]

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

\[
PF = \{ F(x) = (f_1(x), f_2(x), ..., f_m(x))^T | x \in PS \}
\]

2.2. A brief introduction to MOIAs

The first MOIA was designed in [47] 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) [21], 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 [19]. 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) [29].

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) [16], 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 [17], 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 [9], 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 [40], the global search capabilities of EAs were combined with the learning capabilities of artificial immune systems. In the immunity-based hybrid evolutionary algorithm [46], 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 [22] 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 [34], 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) [14]. Therefore, it provides two different search patterns during the search process. A double-module immune algorithm for MOPs (DMMO) [27] 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 [30], 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) [28] 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 [19]) is introduced. This approach has inspired the design of many other MOIAs [6,22,29,30,34]. Its main contribution is to propose a non-dominated 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.

First, an offspring population \( O_t \) is produced by applying the evolutionary operators (i.e., SBX and polynomial-based mutation [19]) on the current population \( P_t \) (with size \( N \)). Then, \( O_t \) is merged with the archive population \( E_t \), and the non-dominated solutions from this union are selected to form a temporary population \( D_t \). If the size of \( D_t \) is smaller than \( N \), all the solutions in \( D_t \) are chosen for the new archive population \( E_{t+1} \). Otherwise, \( N \) solutions in \( D_t \) with the largest values on crowding distance [11] are selected. Thus, the size of the archive is not fixed, and such size may be very small especially at the early stages of the search. At last, the iteration counter \( t \) is increased by 1 and clonal selection is performed to produce a new population \( P_t \). This approach first selects \( n_A \) solutions with the largest crowding-distance values [11], to form the population with cloned parents \( A_t \) (\( n_A \) is always set to be far smaller than \( N \)). Then, the parents \( q_i (i=1,2, \ldots, n_A) \) in \( A_t \) are all cloned proportionally based on their crowding-distance values. The mathematical model of proportional cloning is defined by

\[
P_{t+1} = \bigcup_{i=1}^{n_A} \{ q_i \otimes a_i \}, a_i \in A_t
\]

where the operator \( \otimes \) represents the cloning operator and the parameter \( q_i \) denotes the number of clones corresponding to each solution \( a_i \) in \( A_t \). The value of \( q_i \) is calculated by

\[
q_i = \left[ \frac{N \times cd(a_i)}{\sum_{j=1}^{n_A} cd(a_j)} \right]
\]

where \( cd(a_i) \) is the crowding-distance value [11] for \( a_i \). In this way, the solution with a larger crowding-distance value will have more clones, as a larger value of \( q_i \) will be assigned in Eq. (7). Since the crowding-distance values for boundary solutions are positive infinity, their crowding-distance values are reset to be the double of the maximum value of solutions in \( A_t \) except for the boundary solutions [19]. An example is given here to clarify this procedure. Suppose that there are five solutions in population \( A_t \) and the corresponding values of the objective functions are \((0, 1.0), (0.3, 0.6), (0.5, 0.5), (0.6, 0.4)\) and \((1.0, 0)\) for a bi-objective problem. Therefore, their crowding-distance values can be computed according to the method in [11], 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 [19,28]. However, according to the No-Free-Lunch theorem [45], as we produce more clones of the best solutions,
the higher will be the loss of population diversity [38]. As experimentally validated in [38], 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 [27,28,30,34] 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 [11] to construct the new archive $E_{t+1}$, while AIMA adopts the archive truncation method from SPEA2 [53], 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.

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$D$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 [53] 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.

Algorithm 1 Pseudo-code of AIMA.

```python
1 Initialize $t = 0$, $selected = 1$, $hold gens = 0$, $E_0 = \phi$;
2 Randomly generate an initial population $P_0$;
3 while $t < MAX_G$
4     // Evolution
5     Apply DE$selected$ 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 $E_t$; $O_t$ 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 [53];
15     end if
16     $t = t + 1$;
17     // Clonal Selection
18     Calculate the crowding-distance values of solutions in $E_t$;
19     Select $n_0$ 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 $E_t$;
```
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 [23]. 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).

\[
\begin{align*}
\text{DE1: } U_{i,j} &= \begin{cases} 
X_{1,j} + F \cdot (X_{2,j} - X_{3,j}) + F \cdot (X_{r4,j} - X_{r5,j}) & \text{if } \text{rand} < Cr \text{ or } j = j_{\text{rand}}, \text{ with } Cr = 0.9 \text{ and } F = 0.7 \\
X_{i,j} & \text{otherwise}
\end{cases} \\
\text{DE2: } U_{i,j} &= \begin{cases} 
X_{1,j} + F \cdot (X_{2,j} - X_{3,j}) & \text{if } \text{rand} < Cr \text{ or } j = j_{\text{rand}}, \text{ with } Cr = 0.5 \text{ and } F = 0.5 \\
X_{i,j} & \text{otherwise}
\end{cases} \\
\text{DE3: } U_{i,j} &= \begin{cases} 
X_{i,j} + F \cdot (X_{1,j} - X_{2,j}) & \text{if } \text{rand} < Cr \text{ or } j = j_{\text{rand}}, \text{ with } Cr = 0.1 \text{ and } F = 0.5 \\
X_{i,j} & \text{otherwise}
\end{cases}
\]

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 \((j = 1, 2, ..., n)\). \(\text{rand}\) is a real number randomly generated in \([0, 1]\) and \(j_{\text{rand}}\) is an integer randomly selected from \([1, 2, ..., n]\). 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 [44]. 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 [35, 39, 52]. 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 offsprings (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 [10, 44]. 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 searching 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 [44].

Therefore, the three DE strategies with specific parameters settings in Eqs. (8)–(10) exhibit distinct advantages. For DE1, 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, DE3 applies rand/1/bin~ and couples with a small value of \(Cr\), which encourages to conduct local search (i.e., exploitation). Whereas, in DE2, the rand/1/bin strategy with moderate values of \(Cr\) and \(F\) is used. This lets DE2 be able to balance exploration and exploitation during the search. It can be regarded as a trade-off strategy between DE2 and DE3.

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 [23] and [30], 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 toward 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, DE1 should be
assigned with a large execution probability at the early stage to encourage exploration, while DE3 should be run with a large probability at the later stage to encourage exploitation. Regarding the middle stage, DE2 should be executed more times to balance exploration and exploitation.

In AIMA, the execution probabilities of DE1 and DE3 at each generation are respectively denoted by \( p_1 \) and \( p_3 \), which are tuned according to the stages of the search. For DE2, the execution probability at each generation is calculated by \( 1 - p_1 - p_3 \).

As it is difficult to define the boundary between the early stage and the middle stage, there is not 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:

\[
p_1 = 0.9 \times \frac{1.0}{1 + \exp(20 \times (\frac{t}{\text{MAX}_G} - 0.3))}
\]

where \( t \) is the current generation number, and \( \text{MAX}_G \) 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 \( \text{MAX}_G \). Therefore, AIMA has a large probability to use DE1 and may spread the population diversity at the early stage of the search. After running for about half of \( \text{MAX}_G \), DE1 is applied with a very small probability and more computational resources are assigned to run DE2 and DE3.

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 \( \text{avgRatio} \)) 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 \( \text{avgRatio} \) 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 stagnate state holds, as denoted by \( \text{holdgens} \) and then uses \( \text{holdgens} \) to tune the value of \( p_3 \). It is noted that, once the stagnant state is broken, i.e., \( \text{avgRatio} \) is larger than \( LR\% \), the value of \( \text{holdgens} \) will be reset to zero. Once AIMA is unable to find many good solutions during a long period of search, the value of \( \text{holdgens} \) 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 \( \text{PF} \) 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 \( \text{holdgens} \) is getting larger. Thus, the value of \( p_3 \) is assigned to dynamically change with \( \text{holdgens} \), so as to adaptively control the execution probability of DE3, as expressed by

\[
p_3 = 0.9 \times \frac{1.0}{1 + \exp(-20 \times (\frac{\text{holdgens}}{\text{MAX}_G} - 0.25))}
\]

(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 [11] 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.

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 [20]) and seven three-objective problems (DTLZ1-DTLZ7 [13]), 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 PFs. Thus, the WFG and DTLZ test suites are widely used in many experimental studies for performance comparisons among different multi-objective optimization algorithms [2,19,48]. 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) [23], 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.

$$IGD(P^*, P) = \frac{\sum_{x \in P^*} d(x, P)}{|P^*|}$$  \hspace{1cm} (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

---

**Algorithm 2 Adaptive DE Strategy Selection ($hold\_gens$, $E_i$, $O_i$).**

1. Get $p_1$ by Eq. (11);
2. Calculate the average elimination ratio (avgRatio) of last LP generations;
3. If avgRatio <= LR%
4. $hold\_gens = hold\_gens + 1$;
5. Else
6. $hold\_gens = 0$;
7. End if
8. Get $p_3$ by Eq. (12);
9. If 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$_i$
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 $hold\_gens$;
Table 1
Number of points in the subset of true PFs.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Points</th>
<th>Problem</th>
<th>Points</th>
<th>Problem</th>
<th>Points</th>
<th>Problem</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>WFG1</td>
<td>605</td>
<td>WFG5</td>
<td>2601</td>
<td>WFG9</td>
<td>2601</td>
<td>DTLZ4</td>
<td>4000</td>
</tr>
<tr>
<td>WFG2</td>
<td>111</td>
<td>WFG6</td>
<td>2601</td>
<td>DTLZ1</td>
<td>10,000</td>
<td>DTLZ5</td>
<td>333</td>
</tr>
<tr>
<td>WFG3</td>
<td>301</td>
<td>WFG7</td>
<td>2601</td>
<td>DTLZ2</td>
<td>10,000</td>
<td>DTLZ6</td>
<td>140</td>
</tr>
<tr>
<td>WFG4</td>
<td>1181</td>
<td>WFG8</td>
<td>10,201</td>
<td>DTLZ3</td>
<td>4000</td>
<td>DTLZ7</td>
<td>676</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Problems</th>
<th>NSGA-II</th>
<th>SPEA2</th>
<th>MOEA/D</th>
<th>SMS-EMOA</th>
<th>CMPSO</th>
<th>AIMA</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTLZ1</td>
<td>Mean</td>
<td>Std</td>
<td>6.551E-02</td>
<td>2.021E-01</td>
<td>2.041E+00</td>
<td>1.934E-01</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>Std</td>
<td>5.155E-02</td>
<td>7.396E-02</td>
<td>6.342E-02</td>
<td>5.568E-02</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>Std</td>
<td>2.376E-01</td>
<td>7.223E+00</td>
<td>6.731E+00</td>
<td>5.992E-01</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>Std</td>
<td>1.048E-01</td>
<td>1.374E-01</td>
<td>1.701E-01</td>
<td>4.106E-02</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>Std</td>
<td>1.11E-01</td>
<td>1.25E-01</td>
<td>1.19E-01</td>
<td>2.53E-03</td>
</tr>
<tr>
<td>DTLZ2</td>
<td>Mean</td>
<td>Std</td>
<td>4.406E-03</td>
<td>4.919E-03</td>
<td>6.019E-03</td>
<td>4.449E-03</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>Std</td>
<td>1.565E-04</td>
<td>1.605E-04</td>
<td>1.50F-04</td>
<td>1.466E-04</td>
</tr>
<tr>
<td>DTLZ3</td>
<td>Mean</td>
<td>Std</td>
<td>3.565E-02</td>
<td>3.557E-02</td>
<td>4.502E-03</td>
<td>4.018E-03</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>Std</td>
<td>1.338E-01</td>
<td>2.022E-01</td>
<td>7.356E-02</td>
<td>5.789E-02</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>Std</td>
<td>1.766E+00</td>
<td>1.855E+00</td>
<td>7.287E+00</td>
<td>9.581E-02</td>
</tr>
<tr>
<td>WFG1</td>
<td>Mean</td>
<td>Std</td>
<td>1.09E-01</td>
<td>1.24E-01</td>
<td>2.35E-01</td>
<td>1.90E-01</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>Std</td>
<td>2.178E-01</td>
<td>1.751E-01</td>
<td>1.910E-01</td>
<td>2.460E-02</td>
</tr>
<tr>
<td>WFG2</td>
<td>Mean</td>
<td>Std</td>
<td>2.19E-02</td>
<td>2.020E-02</td>
<td>4.176E-02</td>
<td>1.701E-02</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>Std</td>
<td>2.87E-03</td>
<td>7.03E-03</td>
<td>6.71E-04</td>
<td>1.096E-03</td>
</tr>
<tr>
<td>WFG3</td>
<td>Mean</td>
<td>Std</td>
<td>1.239E+02</td>
<td>1.017E+02</td>
<td>1.404E-02</td>
<td>1.268E-02</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>Std</td>
<td>6.693E-02</td>
<td>6.709E-02</td>
<td>6.565E-02</td>
<td>6.536E-02</td>
</tr>
<tr>
<td>WFG4</td>
<td>Mean</td>
<td>Std</td>
<td>2.476E-02</td>
<td>2.684E-02</td>
<td>2.889E-02</td>
<td>1.490E-02</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>Std</td>
<td>1.27E-02</td>
<td>1.33E-02</td>
<td>1.01E-02</td>
<td>7.98E-03</td>
</tr>
<tr>
<td>WFG5</td>
<td>Mean</td>
<td>Std</td>
<td>2.254E-02</td>
<td>3.928E-02</td>
<td>1.570E-02</td>
<td>1.249E-02</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>Std</td>
<td>2.62E-03</td>
<td>9.05E-03</td>
<td>9.05E-04</td>
<td>3.49E-04</td>
</tr>
<tr>
<td>WFG6</td>
<td>Mean</td>
<td>Std</td>
<td>1.247E-01</td>
<td>1.066E-01</td>
<td>1.397E-01</td>
<td>3.825E-02</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>Std</td>
<td>1.212E+02</td>
<td>1.636E-02</td>
<td>1.246E-02</td>
<td>3.57E-04</td>
</tr>
</tbody>
</table>

α = 0.05

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.

4.3. Experimental settings

In this study, AIMA is compared to five state-of-the-art multi-objective optimization algorithms, i.e., NSGA-II [11], SPEA2 [53], MOEA/D [49], SMS-EMOA [2], and CMPSO [48]. 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 [23], SMS-EMOA-DE, and CMODE [43], 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 [41,44]. MOEA/D-DE and CMODE are respectively the improved versions of MOEA/D and CMPSO by embedding DE. Besides that, further validate the superiority of AIMA over other MOIAs, AIMA is compared to four competitive MOIAs, including NNIA [19], IMAD [34], DMMO [27], and HEIA [28]. 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 [15], while the source codes of CMPSO, CMODE, NNIA, IMAD, 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 $C_r$ and $F$ are respectively set to 0.3 and 0.5 as used in [41], 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 $\eta_m$ 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 25,000 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 “$\sim$” 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

![Fig. 3. The nondominated solution set of each algorithm on DTLZ6.](image-url)
**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.

<table>
<thead>
<tr>
<th>Problems</th>
<th>NSGA-II-DE</th>
<th>SPEA2-DE</th>
<th>MOEA/D-DE</th>
<th>SMS-EMOA-DE</th>
<th>CMODE</th>
<th>AIMA</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTLZ1</td>
<td>Mean 5.314E-02 ≈ 4.474E-02 ≈ 9.944E-01 ≈ 4.950E-02 + 6.476E+00 - 1.934E-01</td>
<td>8.63E-02</td>
<td>9.45E-01 - 7.44E-01 - 8.38E-02 + 3.28E+00 - 2.09E-01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DTLZ2</td>
<td>Mean 6.386E-02 - 5.616E-02 - 5.232E-02 - 7.430E-02 - 5.941E-02 - 5.568E-02</td>
<td>2.17E-03</td>
<td>1.32E-03 + 8.59E-04 + 9.53E-04 + 1.63E-03 + 1.13E-03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DTLZ3</td>
<td>Mean 6.518E-02 ≈ 1.127E-01 ≈ 1.954E+00 ≈ 1.396E-01 ≈ 1.535E-01 ≈ 3.992E-01</td>
<td>3.50E-03</td>
<td>1.80E-01 - 1.52E-01 - 2.40E-01 - 7.57E+00 - 6.85E-01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DTLZ4</td>
<td>Mean 6.359E-02 - 4.916E-02 - 3.007E-02 - 4.871E-02 - 5.799E-02 - 4.106E-02</td>
<td>6.01E-03</td>
<td>2.08E-02 + 5.78E-04 + 2.08E-04 + 3.40E-02 + 2.53E-02</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DTLZ5</td>
<td>Mean 5.302E-03 - 5.334E-03 - 1.901E-02 - 4.959E-03 - 7.036E-03 - 4.449E-03</td>
<td>2.13E-04</td>
<td>3.04E-04 - 4.70E-05 - 1.46E-04 - 6.55E-04 - 1.46E-04</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DTLZ6</td>
<td>Mean 5.865E-03 - 4.057E-03 ≈ 1.714E-02 - 4.817E-03 - 4.953E-02 - 4.018E-03</td>
<td>3.68E-04</td>
<td>1.24E-04 - 4.76E-05 - 1.53E-04 - 1.06E-04 - 1.10E-04</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DTLZ7</td>
<td>Mean 7.399E-02 - 5.964E-02 - 1.407E-01 - 1.647E-01 - 9.277E-02 - 5.799E-02</td>
<td>3.89E-03</td>
<td>1.57E-01 - 5.89E-02 - 5.67E-02 - 8.78E-02 - 1.30E-03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WFG1</td>
<td>Mean 1.104E-01 - 1.045E-01 ≈ 5.705E-02 + 1.281E-01 + 1.317E-02 + 9.581E-02</td>
<td>1.95E-01</td>
<td>2.26E-01 - 1.18E-01 - 2.71E-01 - 1.69E-03 - 1.90E-01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WFG2</td>
<td>Mean 4.995E-02 - 6.130E-02 - 1.976E-01 - 5.974E-02 - 3.947E-02 - 2.460E-02</td>
<td>4.36E-02</td>
<td>1.44E-02 - 7.50E-02 - 1.52E-02 - 2.52E-02 - 2.48E-02</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WFG3</td>
<td>Mean 1.399E-02 - 1.218E-02 - 1.325E-02 - 1.188E-02 - 1.250E-02 - 1.170E-02</td>
<td>5.15E-04</td>
<td>2.04E-04 - 1.74E-04 - 2.12E-04 - 1.09E-04 - 1.32E-04</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WFG4</td>
<td>Mean 1.277E-02 ≈ 1.317E-02 ≈ 1.582E-02 ≈ 1.021E-02 + 1.283E-02 ≈ 1.268E-02</td>
<td>8.00E-04</td>
<td>6.25E-04 - 7.38E-04 - 5.62E-04 - 3.97E-04 - 3.63E-04</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WFG5</td>
<td>Mean 6.800E-02 - 6.614E-02 - 6.716E-02 - 6.677E-02 - 6.369E-02 + 6.536E-02</td>
<td>4.86E-04</td>
<td>5.56E-03 - 2.79E-04 - 5.95E-03 - 8.89E-03 - 5.71E-03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WFG6</td>
<td>Mean 1.834E-02 - 1.400E-02 ≈ 2.799E-02 - 1.787E-02 - 1.632E-02 - 1.409E-02</td>
<td>8.44E-03</td>
<td>5.02E-03 - 1.24E-02 - 8.95E-03 - 6.27E-03 - 4.43E-03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WFG7</td>
<td>Mean 1.447E-02 - 1.277E-02 - 1.779E-02 - 1.161E-02 - 1.287E-02 - 1.249E-02</td>
<td>8.54E-04</td>
<td>3.47E-04 - 6.78E-03 - 2.43E-04 - 3.36E-04 - 3.49E-04</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WFG8</td>
<td>Mean 7.031E-02 - 6.633E-01 - 3.854E-02 ≈ 1.517E-01 - 8.737E-02 - 8.25E-02</td>
<td>8.77E-03</td>
<td>3.38E-02 - 4.60E-03 - 2.56E-02 - 5.06E-02 - 4.92E-03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WFG9</td>
<td>Mean 1.522E-02 - 1.314E-02 - 1.469E-02 - 1.258E-02 - 1.397E-02 - 1.246E-02</td>
<td>8.79E-04</td>
<td>4.64E-04 - 2.06E-04 - 9.00E-04 - 7.53E-04 - 3.37E-04</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

"≈", "≈", and "≈" 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 $\alpha = 0.05$.
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.

(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.

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.

(C) Performance comparisons of AIMA with four competitive MOIA

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 MOIA in this subsection, i.e., NNIA [19], IMADE [34], DMMO [27], and HEIA [28], 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 MOIA 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 MOIA 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 MOIA 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. 6. Evolutionary curves of the average IGD values from 30 runs versus the number of function evaluations on the WFG test problems.
Table 4
IGD comparison results of all the compared MOIAs.

<table>
<thead>
<tr>
<th></th>
<th>NNIA Mean(n=10)</th>
<th>IMADE Mean(n=10)</th>
<th>DMMO Mean(n=10)</th>
<th>HEIA Mean(n=10)</th>
<th>AIMA Mean(n=10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTLZ1</td>
<td>2.660E+00(1.929-00)</td>
<td>8.127E-01(1.199-00)</td>
<td>3.122E-01(2.838-01)</td>
<td>5.237E-01(7.633-01)</td>
<td>1.934E-01(2.990-01)</td>
</tr>
<tr>
<td>DTLZ2</td>
<td>5.593E-02(1.400E-03)</td>
<td>6.067E-02(1.783-03)</td>
<td>5.657E-02(1.166-03)</td>
<td>5.809E-02(1.036-03)</td>
<td>5.568E-02(1.136-03)</td>
</tr>
<tr>
<td>DTLZ3</td>
<td>3.544E+00(3.660E-00)</td>
<td>1.361E+00(2.969E-00)</td>
<td>5.785E-01(6.929E-01)</td>
<td>7.946E-01(8.251E-01)</td>
<td>3.992E-01(8.853E-01)</td>
</tr>
<tr>
<td>DTLZ4</td>
<td>1.081E-01(1.090E-01)</td>
<td>4.686E-02(5.460E-02)</td>
<td>8.044E-02(8.352E-02)</td>
<td>5.047E-02(7.086E-03)</td>
<td>4.106E-02(2.513E-03)</td>
</tr>
<tr>
<td>DTLZ5</td>
<td>4.720E-03(0.81E-04)</td>
<td>5.119E-03(3.823E-04)</td>
<td>4.803E-03(3.546E-04)</td>
<td>5.217E-03(5.310E-04)</td>
<td>4.449E-03(4.466E-04)</td>
</tr>
<tr>
<td>DTLZ6</td>
<td>5.175E-01(1.025E-01)</td>
<td>4.051E-03(1.415E-03)</td>
<td>4.033E-03(1.325E-04)</td>
<td>4.059E-03(1.310E-04)</td>
<td>4.018E-03(1.059E-03)</td>
</tr>
<tr>
<td>DTLZ7</td>
<td>1.157E-01(1.535E-01)</td>
<td>6.047E-02(1.476E-02)</td>
<td>2.298E-01(2.039E-01)</td>
<td>6.950E-02(9.797E-02)</td>
<td>5.789E-02(2.916E-03)</td>
</tr>
<tr>
<td>WFG1</td>
<td>1.364E+00(1.856E-01)</td>
<td>1.251E-02(3.942E-04)</td>
<td>1.244E-02(3.376E-04)</td>
<td>1.249E-02(3.883E-04)</td>
<td>9.581E-02(9.080E-01)</td>
</tr>
<tr>
<td>WFG2</td>
<td>1.781E-01(7.995E-02)</td>
<td>3.756E-02(4.582E-02)</td>
<td>2.104E-01(5.799E-02)</td>
<td>8.387E-02(7.922E-02)</td>
<td>2.460E-02(2.485E-03)</td>
</tr>
<tr>
<td>WFG3</td>
<td>1.272E-02(1.240E-03)</td>
<td>1.214E-02(1.975E-04)</td>
<td>1.206E-02(2.759E-04)</td>
<td>1.187E-02(8.828E-04)</td>
<td>1.170E-02(9.089E-04)</td>
</tr>
<tr>
<td>WFG4</td>
<td>1.247E-02(3.034E-03)</td>
<td>1.296E-02(4.888E-04)</td>
<td>1.261E-02(2.955E-04)</td>
<td>1.253E-02(2.428E-04)</td>
<td>1.268E-02(2.634E-03)</td>
</tr>
<tr>
<td>WFG5</td>
<td>6.658E-02(3.845E-04)</td>
<td>6.549E-02(2.633E-03)</td>
<td>6.643E-02(2.035E-04)</td>
<td>6.663E-02(2.543E-04)</td>
<td>6.536E-02(5.711E-03)</td>
</tr>
<tr>
<td>WFG6</td>
<td>2.181E-02(6.043E-04)</td>
<td>1.638E-02(7.878E-03)</td>
<td>2.206E-02(7.348E-03)</td>
<td>2.032E-02(8.789E-03)</td>
<td>1.409E-02(4.317E-03)</td>
</tr>
<tr>
<td>WFG7</td>
<td>1.735E-02(2.923E-03)</td>
<td>1.286E-02(3.322E-04)</td>
<td>1.291E-02(2.565E-04)</td>
<td>1.263E-02(4.058E-04)</td>
<td>1.249E-02(3.497E-03)</td>
</tr>
<tr>
<td>WFG8</td>
<td>7.671E-02(2.377E-02)</td>
<td>2.676E-02(4.303E-03)</td>
<td>4.575E-02(4.446E-02)</td>
<td>3.714E-02(4.472E-02)</td>
<td>3.825E-02(2.919E-03)</td>
</tr>
<tr>
<td>WFG9</td>
<td>1.360E-02(5.195E-03)</td>
<td>1.302E-02(5.159E-04)</td>
<td>1.254E-02(4.619E-04)</td>
<td>1.286E-02(5.995E-04)</td>
<td>1.246E-02(2.379E-04)</td>
</tr>
</tbody>
</table>

\( \sim \), \( \sim\sim\), and \( \sim\sim\sim\) 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 \( \alpha = 0.05 \)

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

Table 5
IGD comparison results of AIMA and four variants.

<table>
<thead>
<tr>
<th></th>
<th>AIMA-DE$_1$ Mean(Sd)</th>
<th>AIMA-DE$_2$ Mean(Sd)</th>
<th>AIMA-DE$_3$ Mean(Sd)</th>
<th>AIMA-EP Mean(Sd)</th>
<th>AIMA Mean(Sd)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTLZ1</td>
<td>1.634E-01 (0.56E+00)</td>
<td>1.074E-01 (2.17E+00)</td>
<td>3.270E-01 (4.32E+00)</td>
<td>1.162E-01 (2.33E+00)</td>
<td>1.934E-01 (2.09E+00)</td>
</tr>
<tr>
<td>DTLZ2</td>
<td>6.420E-02 (1.38E+00)</td>
<td>5.560E-02 (1.17E+00)</td>
<td>5.573E-02 (9.29E+00)</td>
<td>5.716E-02 (1.17E+00)</td>
<td>5.568E-02 (1.13E+00)</td>
</tr>
<tr>
<td>DTLZ3</td>
<td>1.476E-01 (9.99E+00)</td>
<td>5.494E-01 (1.43E+00)</td>
<td>7.622E-01 (1.05E+00)</td>
<td>4.383E-01 (6.64E+00)</td>
<td>3.992E-01 (6.85E+00)</td>
</tr>
<tr>
<td>DTLZ4</td>
<td>4.856E-02 (4.28E+00)</td>
<td>4.250E-02 (4.56E+00)</td>
<td>4.387E-02 (4.41E+00)</td>
<td>5.151E-02 (8.90E+00)</td>
<td>4.106E-02 (2.58E+00)</td>
</tr>
<tr>
<td>DTLZ5</td>
<td>1.006E-02 (1.32E+00)</td>
<td>4.480E-03 (1.72E+00)</td>
<td>4.309E-03 (3.74E+00)</td>
<td>5.013E-03 (4.29E+00)</td>
<td>4.449E-03 (1.48E+00)</td>
</tr>
<tr>
<td>DTLZ6</td>
<td>4.042E+03 (1.32E+00)</td>
<td>4.015E-03 (1.32E+00)</td>
<td>4.015E-03 (1.32E+00)</td>
<td>4.087E-03 (1.19E+00)</td>
<td>4.018E-03 (1.19E+00)</td>
</tr>
<tr>
<td>DTLZ7</td>
<td>6.011E-02 (2.92E+00)</td>
<td>7.813E-02 (7.08E+00)</td>
<td>6.799E-02 (5.20E+00)</td>
<td>5.958E-02 (1.43E+00)</td>
<td>5.789E-02 (1.59E+00)</td>
</tr>
<tr>
<td>WFG1</td>
<td>1.092E-01 (1.98E+01)</td>
<td>3.627E-01 (3.00E+01)</td>
<td>2.198E-01 (3.28E+00)</td>
<td>2.064E-01 (2.43E+00)</td>
<td>9.581E-02 (1.95E+01)</td>
</tr>
<tr>
<td>WFG2</td>
<td>1.160E-02 (1.91E+00)</td>
<td>5.970E-02 (3.94E+00)</td>
<td>1.602E-01 (9.16E+00)</td>
<td>2.385E-02 (6.49E+00)</td>
<td>2.460E-02 (2.84E+00)</td>
</tr>
<tr>
<td>WFG3</td>
<td>1.349E-02 (2.93E+00)</td>
<td>1.180E-02 (2.78E+00)</td>
<td>1.186E-02 (2.16E+00)</td>
<td>1.199E-02 (8.08E+00)</td>
<td>1.170E-02 (1.99E+00)</td>
</tr>
<tr>
<td>WFG4</td>
<td>1.314E-02 (2.25E+00)</td>
<td>1.248E-02 (5.06E+00)</td>
<td>1.255E-02 (3.78E+00)</td>
<td>1.274E-02 (3.07E+00)</td>
<td>1.268E-02 (3.07E+00)</td>
</tr>
<tr>
<td>WFG5</td>
<td>6.980E-02 (2.93E+00)</td>
<td>6.605E-02 (2.81E+00)</td>
<td>6.642E-02 (1.67E+00)</td>
<td>6.344E-02 (6.57E+00)</td>
<td>6.356E-02 (5.71E+00)</td>
</tr>
<tr>
<td>WFG6</td>
<td>1.385E-02 (2.98E+00)</td>
<td>2.015E-02 (6.98E+00)</td>
<td>2.148E-02 (6.57E+00)</td>
<td>1.967E-02 (1.33E+00)</td>
<td>1.409E-02 (2.43E+00)</td>
</tr>
<tr>
<td>WFG7</td>
<td>1.365E-02 (2.33E+00)</td>
<td>1.243E-02 (3.43E+00)</td>
<td>1.263E-02 (3.33E+00)</td>
<td>1.284E-02 (1.18E+00)</td>
<td>1.249E-02 (3.49E+00)</td>
</tr>
<tr>
<td>WFG8</td>
<td>4.214E-02 (4.07E+00)</td>
<td>4.833E-02 (2.34E+00)</td>
<td>6.263E-02 (7.31E+00)</td>
<td>3.625E-02 (1.38E+00)</td>
<td>3.825E-02 (4.92E+00)</td>
</tr>
<tr>
<td>WFG9</td>
<td>1.649E-02 (9.43E+00)</td>
<td>1.288E-02 (6.88E+00)</td>
<td>1.315E-02 (6.85E+00)</td>
<td>1.278E-02 (3.71E+00)</td>
<td>1.246E-02 (5.37E+00)</td>
</tr>
</tbody>
</table>

\[ \pm \] indicates that the results of the algorithm are worse than, better than, and similar to that of AIMA using Wilcoxon's rank sum test with \( \alpha = 0.05 \)

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-DE$_1$, AIMA-DE$_2$, AIMA-DE$_3$, 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.
Fig. 9. Evolutionary curves of the average IGD values from 30 runs versus the number of function evaluations on the DTLZ test problems.

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-DE$_1$ and AIMA-DE$_2$ outperform AIMA on a few problems, e.g., WFG2 and WFG6. In detail, AIMA obtains better IGD results than AIMA-DE$_1$, AIMA-DE$_2$, AIMA-DE$_3$, 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.

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 PFs), DTLZ3 (concave PF and many local PFs), 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 (avgRtiao), 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 LP and LR 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.
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 MOIs (i.e., NNSM, IMM, 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 [7,51]. Furthermore, the extension of AIMA to solve some real-world applications will also be studied in our future work.

Acknowledgments

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 grant 2017A030313338, and CONACyT under grant 221551.

References


