www.elsevier.com/locate/caor

A novel Hybrid multi-objective immune algorithm with adaptive differential evolution

Qiuzhen Lin, Qingling Zhu, Peizhi Huang, Jianyong Chen, Zhong Ming, Jianping Yu

PII: S0305-0548(15)00083-0
DOI: http://dx.doi.org/10.1016/j.cor.2015.04.003
Reference: CAOR3759

To appear in: *Computers & Operations Research*

Cite this article as: Qiuzhen Lin, Qingling Zhu, Peizhi Huang, Jianyong Chen, Zhong Ming, Jianping Yu, A novel Hybrid multi-objective immune algorithm with adaptive differential evolution, *Computers & Operations Research,* http://dx.doi.org/10.1016/j.cor.2015.04.003

A Novel Hybrid Multi-objective Immune Algorithm with Adaptive Differential Evolution

Qiuzhen Lin, Qingling Zhu, Peizhi Huang, Jianyong Chen*, Zhong Ming, Jianping Yu
College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, P.R. China,

Abstract:

In this paper, we propose a novel hybrid multi-objective immune algorithm with adaptive differential evolution, named ADE-MOIA, in which the introduction of differential evolution (DE) into multi-objective immune algorithm (MOIA) combines their respective advantages and thus enhances the robustness to solve various kinds of MOPs. In ADE-MOIA, in order to effectively cooperate DE with MOIA, we present a novel adaptive DE operator, which includes a suitable parent selection strategy and a novel adaptive parameter control approach. When performing DE operation, two parents are respectively picked from the current evolved and dominated population in order to provide a correct evolutionary direction. Moreover, based on the evolutionary progress and the success rate of offspring, the crossover rate and scaling factor in DE operator are adaptively varied for each individual. The proposed adaptive DE operator is able to improve both of the convergence speed and population diversity, which are validated by the experimental studies. When comparing ADE-MOIA with several natural-inspired heuristic algorithms, such as NSGA-II, SPEA2, AbYSS, MOEA/D-DE, MIMO and D^2MOPSO, simulations show that ADE-MOIA performs better on most of 21 well-known benchmark problems.

Keywords: Multi-objective optimization; Immune algorithm; Differential evolution; Adaptive parameter control

1. Introduction

Optimization problems widely exist in many domains of scientific research and engineering application [1-4]. Based on the number of objectives needed to be optimized, they are generally classified into two categories, such as single-objective optimization problems (SOPs) and multi-objective optimization problems (MOPs). Generally, MOPs bring more challenges as they are aimed at optimizing several conflicting objectives simultaneously, while SOPs only locate a global optimal value. Due to the complex landscape in decision and objective spaces of MOPs, it is practically impossible for traditional deterministic approaches to travel the entire solution space and find a satisfactory result within a limited time. As a result, evolutionary algorithms (EAs) are presented for solving MOPs, which demonstrate the excellent global search capability in finding optimal solution set [5, 6]. The ability to handle complex MOPs that are characterized with discontinuities, multimodality, disjoint feasible spaces and noisy function evaluations, reinforces the potential effectiveness of multi-objective EAs (MOEAs) [7, 8].

* Corresponding author
Email address: qiuzhlin@szu.edu.cn(Q.Z. Lin), jychen@szu.edu.cn (J.Y. Chen)
Phone: +86-75526001223, Fax: +86-75526534078 (J.Y. Chen)

The first reported literature of MOEAs may be the vector evaluated genetic algorithm (VEGA) in mid-1980s [9]. After that, MOEAs attract more and more interests of researchers and numbers of various MOEAs are presented. The first generation of MOEAs published around 1990s mostly adopted the Pareto-rank based selection and fitness sharing, the representatives of which include multi-objective genetic algorithm (MOGA) [10], niched Pareto genetic algorithm (NPGA) [11] and non-dominated sorting genetic algorithm (NSGA) [12]. In 2000s, the second generation of MOEAs was designed based on the elitist selection strategy, such as strength Pareto evolutionary algorithm (SPEA) [13] and its improved version (SPEA2) [14], Pareto envelop-based selection algorithm (PESA) [15], and a fast non-dominated sorting genetic algorithm (NSGA-II) [16]. Recently, as more and more heuristic algorithms including scatter search [17], simulated annealing [18], particle swarm optimization [19], ant colony optimization [20], differential evolution [21] and immune algorithm [22], are presented, it is found that multiple heuristic algorithms can be hybridized to achieve stronger search capabilities [23-25]. This is realized by combining the advantages of various heuristic algorithms to overcome the natural weakness of each algorithm. For example, an archive-based hybrid scatter search algorithm (AbYSS) is proposed [23], which embeds the mutation and crossover operators of EAs into the framework of scatter search. The experimental studies show that this hybrid approach obviously outperforms the state-of-the-art algorithms, such as SPEA2 and NSGA-II. A novel hybrid multi-objective evolutionary algorithm [24] is designed for real-valued MOPs by combining the concepts of personal best and global best in particle swarm optimization into MOEAs. Multiple crossover operators are also adopted here to enhance its global search capability. In [25], a multi-objective particle swarm optimizer based on decomposition and dominance (D^2MOPSO) is presented, which incorporates the dominance relationship with the decomposition approach. The improved version of D^2MOPSO is also proposed by the same authors with the introduction of a new mechanism for leaders' selection and a new archiving technique [26]. These new features facilitate the attaining of better diversity and coverage in both objective and solution spaces.

Differential evolution (DE) algorithm is a simple and efficient random search technology that is mainly used for continuous global optimization problems [27, 28]. Because of its excellent global search ability and easy implementation, DE is recently investigated to mix with MOEAs for compensating the defects of lacking diversity in some MOEAs [29-34]. In [29], a differential evolution algorithm for multi-objective optimization with rough sets (DEMORS) theory is proposed, which uses the concept of ε-dominance to keep the population diversity. Two stages are sequentially performed, in which the first stage generates an initial population close to the true Pareto front by using the multi-objective version of DE and the second stage exploits the rough sets theory to further improve the convergence and the diversity of population founded in the first stage. DEMORS is justified to outperform some state-of-the-art MOEAs and extended to solve the complex constrained MOPs in [30]. A novel multi-objective evolutionary algorithm based on decomposition (MOEA/D) is designed by Zhang et al. [31], which decomposes MOPs into multiple SOPs using the weighted aggregation of each objective. A basic DE operator is adopted to replace the simulated binary crossover operator and the experimental studies confirm the advantages of DE when handling some complex MOPs with variable linkages in decision space [32]. Recently, an adaptive differential evolution for MOEA/D (ADEMO/D)

is reported in [33], which introduces the adaptive control strategy of SaDE [28] into the framework of MOEA/D. The improved version of ADEMO/D [34] replaces the self-adaptive DE strategy with a novel adaptive selection strategy (AdapSS) [27].

On the other hand, artificial immune system (AIS) is a new developing heuristic algorithm imitating the information processing mechanism of biological immune system [35], which has found numbers of applications in the fields of computer security [36], optimization [37], and anomaly detection [38]. Especially, immune algorithm has been successfully applied for MOPs and shown pretty promising performance in accelerating the convergence speed and maintaining the population diversity [37]. However, when dealing with some complex MOPs, such as DTLZ and WFG test problems [39, 40], it is quite difficult to fast approach the true Pareto front in limited generations. As the previous studies have shown that the hybridization of MOEAs with DE is especially effective for solving some complex MOPs, it is reasonable to believe that the embedment of DE into multi-objective immune algorithms (MOIAs) is promising. Especially, MOIAs may suffer from the lack of population diversity due to the elitist clonal selection principle. The global search capability of DE operator can repair that defect and enhance the robustness of MOIAs to handle various kinds of MOPs. However, to the best of our knowledge, this integration of MOIAs with DE is rarely investigated. Therefore, in this paper, we propose a novel multi-objective immune algorithm with adaptive DE (ADE-MOIA), where the adaptive DE (ADE) operator substantially improves both of the convergence speed and population diversity. The ADE operation is designed by a suitable parent selection strategy and a novel adaptive parameter control method. By dividing the population into a dominated subpopulation and a non-dominated subpopulation, three parents to run DE operator are respectively chosen from the corresponding subpopulations. The difference between the dominated and non-dominated parents may provide a correct evolutionary direction in DE. Besides that, as the choice of systematic parameters in DE has great impact on the optimization performance, an adaptive control approach is presented to tune the crossover rate (CR) and scaling factor (F), which is aimed at decreasing the influence of parameter settings and enhancing its robustness. In our ADE operation, CR is gradually changed with the evolutionary process while F is adaptively modified for each individual based on the success rate of offspring. The advantage of the proposed ADE operator is verified by the experimental studies. To investigate the performance of ADE-MOIA, 21 well-known benchmark problems such as ZDT problems [41], WFG problems [40] and DTLZ problems [39], are used. When compared with various natural-inspired heuristic algorithms, such as NSGA-II [16], SPEA2 [14], AbYSS [23], MOEA/D-DE [32], MIMO [37] and D^{2}MOPSO [26], ADE-MOIA performs best on most of benchmark problems.

The remainder of this paper is organized as follows. Section 2 describes the related background, such as MOPs, AIS and related work of immune algorithm. The realization of ADE-MOIA is introduced in Section 3, where the cloning, ADE, perturbance and archive update operators are respectively described in detail. The experimental studies are conducted in Section 4, which gives a comparative study among ADE-MOIA and various nature-inspired heuristic algorithms. Moreover, the advantage of ADE operator is analyzed and its effectiveness is confirmed by the experimental results. At last, the conclusions are summarized in Section 5.

2. Related Background

2.1 Multi-objective Optimization Problems

There exist many MOPs in various practical applications, which may need to handle constraints and optimize multiple conflicting objectives simultaneously. Without loss of generality, the mathematical description of MOPs for minimization can be expressed as follows.

$$\begin{aligned} &Minimize \ \ f(x) = \{f_1(x), f_2(x), ..., f_m(x)\} \\ &s.t.: \quad g_i(x) \le 0, \quad i = 1, 2, ..., q \\ &\qquad h_j(x) = 0, \quad j = 1, 2, ..., p \end{aligned} \tag{1}$$

where $x = (x_1, x_2, ..., x_n) \in \Omega$ is a decision vector with n dimensions, Ω is the decision space, m is the number of objectives, $g_i(x)$ $(i = 1, 2, ..., q)$ are q inequality constraints and $h_j(x)$ $(j = 1, 2, ..., p)$ are p equality constraints.

The goal of MOPs is to minimize all the objective functions in Eq. (1) and the concepts of Pareto optimum theory [42] are important for MOPs, which are described as follows.

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

$$(\forall i \in \{1, 2, ..., m\} : f_i(x) \le f_i(\mathrm{y})) \wedge (\exists j \in \{1, 2, ..., m\} : f_j(x) < f_j(\mathrm{y})) \tag{2}$$

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

$$\neg \exists y \in \Omega : y \succ x \tag{3}$$

Definition 3 (Pareto-optimal set): The set PS includes all the Pareto-optimal solutions, as defined by

$$PS = \{x \mid \neg \exists y \in \Omega : y \succ x\} \tag{4}$$

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

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

2.2 Artificial Immune System

The concept of artificial immune system (AIS) put forward for the first time may trace back to 1996 [43]. Since then, AIS steps into a period of rapid development and becomes a research hotspot in the field of artificial intelligence [44-54]. It is a novel intelligence approach imitating the information processing principle of natural immune system. When external antigen is detected by the biological immune system, its B-cell will be immediately adapted to eliminate invaders by the reactive procedures, such as clonal selection and affinity maturation by hyper-mutation. The antibodies that can identify antigen better will be selected to reproduce by cloning and this process is called clonal selection, while hyper-mutation realizes the affinity maturation process by varying the clones in high frequency. After that, some antibodies with high affinity will retain as memory cells to prevent the re-invasion. This information processing principle can be simulated in AIS to improve the convergence speed and maintain the population diversity when solving MOPs.

In the variation process of biological immune system, antibodies are always mutated in order to better identify the specific antigen. This is similar with the optimizing process when handling MOPs. In MOIAs, the multi-objective problems and the corresponding constraints are considered as antigen. Its

potential solution is regarded as an antibody while the affinity is seen as the fitness of a solution to the target problems. That is to say, taking the MOPs defined in Eq. (1) as example, a solution vector $x=(x_1,x_2,...,x_n)\in\Omega$ is considered to be an antibody and $f(x)=\{f_1(x),f_2(x),...,f_m(x)\}$ is regarded as an antigen. An antibody population is composed by a number of antibodies. When an antibody is said to be non-dominated, it indicates that there is no other antibodies in the antibody population that can dominate it.

2.3 Related Work on Immune Algorithm

Immune algorithm is a new developing intelligent algorithm, which has attracted great interests of researchers in recent years. Most immune algorithms are performed by simulating the two important immune principles, such as clonal selection and affinity maturation by hyper-mutation. The first immune optimization algorithm [22] is designed to solve SOPs by embedding the concept of clonal selection into genetic algorithm. Then, an immune network algorithm called opt-aiNet [44] is presented to solve the multimodel problems, in which the antibodies going to be cloned, inhibited or retained are determined by the immune network. Based on the clonal selection principle, a novel CLONALG algorithm [45] is proposed to solve the multimodel and combinatorial optimization problems, which fully simulates the affinity maturation procedures of immune response.

Motivated by the outstanding performance of immune algorithm in solving SOPs, immune algorithm is designed to solve MOPs. Generally speaking, most of the current MOIAs can be classified into two categories, such as pure MOIAs and hybrid MOIAs. Pure MOIAs fully mimic the information processing procedures of immune principles, such as clonal selection and immune network theory, while hybrid MOIAs cooperate with other nature-inspired heuristic algorithms and combine their respect advantages to enhance the comprehensive performance. Some representatives from the two kinds of MOIAs are introduced as follows.

Based on the clonal selection principle, multi-objective immune system algorithm (MISA) [46] is proposed, in which an affinity evaluation strategy selects antibodies to perform clonal propagation and an external memory archive collects the previously founded non-dominated solutions. The enhanced performance of MISA is achieved by two mutation approaches, i.e., uniform mutation for cloning population and non-uniform mutation for the not-so-good solutions [46]. In [47], a novel non-dominated neighbor immune algorithm (NNIA) is presented, which uses two heuristic search operators, elitist strategy and a novel selection approach based on the non-dominated neighbors. NNIA selects a few of non-dominated individuals with larger crowding-distance values as active population and only performs proportional cloning, recombination and mutation to the active population. This elitist strategy greatly improves the convergence rate, but may lead to the loss of population diversity in some complex MOPs. This weakness of NNIA is repaired by using a multi-population co-evolutionary strategy developed in [48]. A multi-objective immune algorithm based on a multiple-affinity modal (MAM-MOIA) is designed in [49], where six affinity models are put forward. The clonal selection, reproduction and mutation operators are correspondingly performed based on the specific affinity measure strategy. In [50], a multi-objective optimization immune algorithm using clustering (CMOIA) is reported, which performs a clustering-based clonal selection strategy to maintain the balance between exploration and exploitation. A weight-based multi-objective artificial immune system

(WBMOAIS) is proposed [51], which follows the basic framework of opt-aiNet [44]. It assigns a random weighted sum of multiple objectives as the fitness and develops a new truncation method to eliminate similar individuals in memory. Thus, a well distributed set of non-dominated solutions can be obtained. In [52], a novel immune clonal algorithm (NICA) is presented for MOPs, which performs an entire cloning operator and pure hyper-mutation operator for affinity maturation. Moreover, a novel archive updating operation is also operated for maintaining the population diversity.

For hybrid MOIAs, a new hybrid genetic/immune strategy for multi-objective optimization (GISMOO) [53] is proposed, which combines the advantages of Pareto GA and AIS. The evolutionary procedures of GISMOO have two stages, where the genetic operations are adopted to generate child population in the first stage and then the immune operators are employed to produce offspring from the non-dominated individuals in the second stage. A novel evolutionary artificial immune algorithm is presented in [54] for MOPs, which combines the global searching capability of EA and the immune learning ability of AIS. The concept of clonal selection principle is exploited in a new selection strategy that can maintain the balance of exploration and exploitation. In [55], a hybrid immune algorithm for MOPs (HIMO) is designed by combining the advantages of Gaussian and polynomial mutations. An adaptive switch control factor is used to tune the executions of two mutations. The performance of HIMO is further enhanced by a micro-population immune multi-objection algorithm (MIMO) [37], which presents an efficient adaptive mutation strategy and a fine-grained selection method.

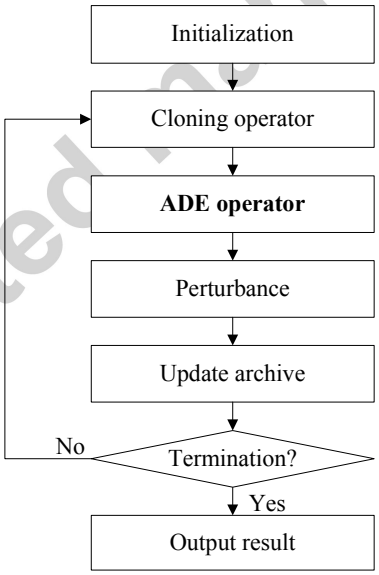


Fig. 1 The algorithmic flowchart of ADE-MOIA

3. The Proposed Algorithm ADE-MOIA

In this section, the details of our proposed algorithm ADE-MOIA are introduced. The major contribution of ADE-MOIA is to embed an ADE operator into the framework of MOIAs. This is the first try to investigate the performance of DE-based MOIAs. To clearly describe ADE-MOIA, the algorithmic flowchart of ADE-MOIA is demonstrated in Fig. 1, where the antibody population is sequentially performed by the procedures of cloning, ADE, perturbance and archive update after the initialization phase. Once the termination condition, usually set to the maximal running generations, is

satisfied, the non-dominated antibodies in external archive are exported as the final result.

When initialization, an initial population $P=\{x_1,x_2,...,x_N\}$ is created, where the decision variables of x_i $(i=1,2,...,N)$ are randomly generated in Ω and the corresponding objective values are evaluated. After performing the fast non-dominated sorting [16] in *P*, the non-dominated and dominated individuals are obtained and respectively preserved in the external archive (*EXA*) and dominated archive (*DA*). The pseudo-code of initialization, named **Algorithm 1**, is described in Fig. 2.

Algorithm 1: Initialization

```
for i = 1 to N
    randomly generate an individual x_i
    evaluate the objectives of x_i
    add x_i to the population P
end for
EXA = Find_nonDominated( P ); //find non-dominated solutions in P
DA = Find_Dominated ( P ); //find the dominated solutions in P
```

Fig. 2 The pseudo-code of initialization

3.1 Cloning Operator

It is assumed that the evolutionary population is P with size *N* and the cloning population is $P_C=\{a_1,a_2,...,a_{NC}\}$ with size *NC*. At first, *NC* antibodies having the larger crowding-distance values are selected from *EXA*. Afterward, the cloning is activated and the evolutionary population will be

$$P=\bigcup_{i=1}^{NC} q_i \times a_i \tag{6}$$

where q_i stands for the number of clones corresponding to each antibody a_i $(i=1,2,...,NC)$, calculated by

$$q_i=\left\lceil N\times\frac{CD(a_i)}{\sum_{j=1}^{NC}CD(a_j)}\right\rceil \tag{7}$$

where $CD(a_i)$ is the fitness value of antibody a_i $(i=1,2,...,NC)$ and set as the crowding-distance value, computed by the following equation.

$$CD(a_i)=\sum_{j=1}^{m}\frac{CD_j(a_i)}{f_j\max-f_j\min} \tag{8}$$

where m is the number of objectives, $f_j\max$ and $f_j\min$ are respectively the maximal and minimal values of the *j*-th objective, $CD_j(a_i)$ is the crowding distance of the *j*-th objective for the antibody a_i $(i=1,2,...,NC)$ and obtained by

$$CD_j(a_i)=\begin{cases}\infty,\text{if}\,(f_j(a_i)==f_j\min\ \text{ or }\ f_j(a_i)==f_j\max);\\ \min\{f_j(a_k)-f_j(a_l)\},\ f_j(a_k)>f_j(a_i)>f_j(a_l)\ (k,l\in[1,N]),\text{otherwise}\end{cases} \tag{9}$$

It is noted that when the antibody locates in the boundary, its crowding distance is ∞ and the number of clones can't be obtained by Eq. (7). In this case, the crowding distance of the boundary antibody is

set as twice of the maximal crowding distance except for the boundary solutions. The pseudo-code of this operator named **Algorithm 2** is shown in Fig. 3, where *NC* is the size of clone population and the function **CrowdingDistanceAssignment**(***P***) calculates the crowding distance value of each antibody in population *P*.

Algorithm 2: Cloning Operator

```
P_C = EXA;
if ( |P_C| > NC )
    CrowdingDistanceAssignment(P_C);
    P_C = Sort( P_C );// sort P_C according to the crowding distance in descending order
    P_C = SelectforClone( P_C );//select the first NC antibodies in P_C
end if
for i = 1 to |P_C|
    calculate the number q_i of clones for x_i according to Eq. (7)
    clone q_i individuals of x_i and add them to P
end for
```

Fig. 3 The pseudo-code of cloning operator

3.2 Evolutionary Operator

3.2.1 DE Operator

Differential evolution has very strong global search capability and shown pretty good convergence ability [27-34]. Thus, DE operator is widely adopted in MOEAs [29-34]. In ADE-MOIA, a novel ADE operator is presented, which is tailor-made to effectively cooperate with MOIA. The weakness of MOIAs in population diversity can be repaired by the strong global search capability of DE. Our proposed ADE operator includes a suitable parent selection strategy and a novel adaptive parameter control approach. This makes ADE operator become more effective for MOIAs. The ADE operation is given as follows.

$$v_i = x_{r_1} + F \times (x_{r_2} - x_{r_3}) \tag{10}$$

where $F \in (0, 1.0]$ is a scaling factor. It is important to note that the selected parent vectors used in Eq. (10) have great impact on the optimization performance. The most popular DE operators includes DE/rand/1/bin, DE/best/1/bin, DE/current-to-best/1/bin, DE/rand/2/bin, DE/best/2/bin [28], which are mostly adopted to solve SOPs and especially effective for some SOPs with certain complex features. When solving MOPs, these popular DE operators are not perfectly fit as all the non-dominated solutions are regarded as the best candidate for MOPs. The selection of parent vectors in MOPs is challenging. In our proposed ADE operator, in order to promote the population diversity, three parent vectors picked from the *EXA* and *DA* are aimed at providing a correct evolutionary direction, where x_{r_1} and x_{r_2} different from x_i are randomly chosen from *EXA* while x_{r_3} is randomly selected from *DA*.

After that, a trial point y_i will be obtained from its parents x_i and v_i using the following crossover rules.

$$y_i^j = \begin{cases} v_i^j, & \text{if } (R_j < CR \text{ or } j = I_i) \\ x_i^j, & \text{otherwise} \end{cases} \tag{11}$$

where the superscript j represents the j-th variable of decision vectors; I_i is a random integer number in [1, n] that ensures at least one dimension of trial vector is inherited from the mutant vector; R_j is a random real number from (0, 1.0) for each j-th variable; the crossover rate CR is a pre-defined real number in [0, 1.0], which controls how much information the offspring will inherit from the parents.

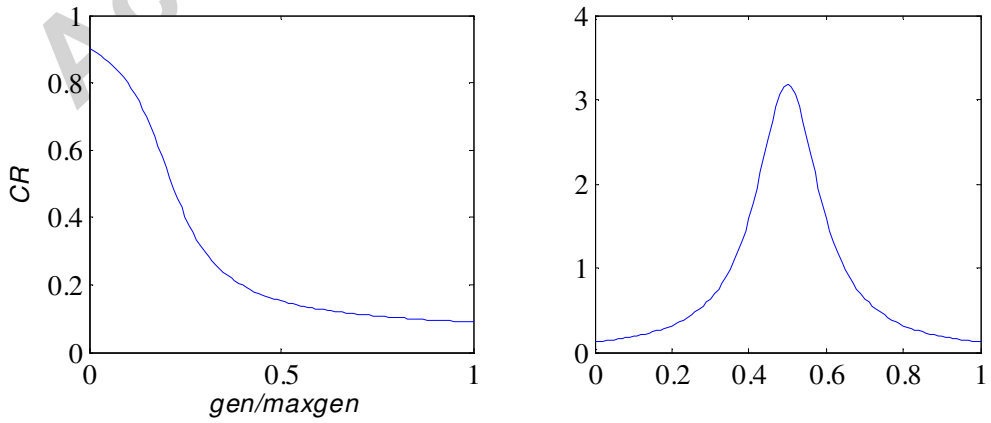
3.2.2 Adaptive Parameters Control

As introduced above, DE operator generates a new individual depending on the mutation and crossover. Thus, the parameter settings of scaling factor (F) in Eq. (10) and crossover rate (CR) in Eq. (11) are quite important for DE-based algorithms. On the one hand, the scaling factor F can affect the convergence speed and population diversity by changing the search step size. Usually, the DE-based algorithms favor a global search to explore the entire feasible region in the early stage of the evolutionary process, and then go with a local search to speed up the convergence. On the other hand, the crossover rate CR decides how much information of the parents will pass to offspring, which has obvious impact on the convergence speed and population diversity.

In our ADE operator, the parameter of CR is set as follows.

$$CR_{gen} = 0.55 + \frac{1}{\pi} \times \arctan\left(\frac{1 - gen/maxgen - 0.8}{0.1}\right) \tag{12}$$

where gen and $maxgen$ are respectively the current and maximum generation times. The CR value in each generation is gradually reduced with the increase of generation times. Figure 4(a) shows the tendency of the CR value, where the CR value gradually reduces from 0.9 and stagnates almost in the range (0.1, 0.2) after half of $maxgen$ generations. Therefore, as the CR value is large, the child population is generated more randomly and resultantly the ADE operator encourages global search in the first stage. After the running by half of $maxgen$ generations, the child population will inherit more information from the parents and thus the child population only changes little. As a result, the local search is activated at the last stage.



(a) (b)

Fig. 4 The tendency of CR in (a) and the distribution of $Cauchy(0.5, 0.1)$ in (b)

Besides that, the scaling factor F for each x_i in Eq. (10) is adaptively set as follows.

$$F_i = Cauchy(F_m, 0.1), F_i \in (0.1, 0.9) \tag{13}$$

where F_m is initialized to 0.5 at first. As an example, the distribution of *Cauchy*(0.5, 0.1) is shown in Fig. 4(b). With the running of generations, F_m is updated according to the improvement situation of child population, as follows.

$$F_m = average(F_{success}) = \sum_{x \in F_{success}} (x / | F_{success} |) \tag{14}$$

It is noted that the F value is collected into the set $F_{success}$ only when the child generated by this F value is a non-dominate solution to its parent. When the size of $F_{success}$ is larger than $p\%$ of the population, the Cauchy distribution center F_m is recalculated by using Eq. (14) and each F_i in Eq. (13) is generated using the updated value of F_m. In our paper, the value of $p\%$ is set to 10% based on the observation of various experiments.

3.3 Perturbance

After ADE operator, polynomial mutation (PM) operator is further performed to generate new individuals as the perturbance operator. For antibody $y = (y_1, y_2, ..., y_n)$, PM is operated as follows.

$$z_k = \begin{cases} y_k + \sigma_k \times (ub_k - lb_k) & \text{, if } r < p_m \\ y_k & \text{, otherwise} \end{cases} \tag{15}$$

where z_k and y_k are respectively the k-th decision variables after and before mutation; ub_k and lb_k are respectively the upper and lower bounds of the k-th decision variables; r is a random real number in [0, 1.0]; σ_k is a small variation that is obtained by

$$\sigma_k = \begin{cases} (2 \times r)^{\frac{1}{\eta+1}} - 1 & \text{, if } r<0.5 \\ 1 - (2 - 2 \times r)^{\frac{1}{\eta+1}} & \text{, otherwise} \end{cases} \tag{16}$$

where r is a uniformly random number in [0, 1.0] that compares with the threshold value 0.5; η is the mutation distribution parameter, which controls the magnitude of the expected mutation of the solution variable. Generally, larger value of η generates smaller variance on average. The pseudo-code of PM operator is illustrated in Fig. 5, named **Algorithm 3**, where n is the dimension of variables and p_m is the probability for perturbance.

Algorithm 3: PM operator

```
for k = 1 to n
    if random(0,1) < p_m
        calculate σ_k according to Eq. (16)
        calculate z_k according to Eq. (15)
        if z_k > ub_k
            z_k = ub_k
        else if z_k < lb_k
            z_k = lb_k
        end if
    end if
end for
```

Fig. 5 the pseudo-code of PM operator

3.4 Archive Update

After the evolutionary operations such as ADE and PM operators are completed, all the offspring and the antibodies in *EXA* are combined. The same individuals in the mixed population are moved and only keep one sample. Then, the fast non-dominated sorting is performed on the mixed population to record all the non-dominated individuals in *EXA* and other dominated individuals in *DA*. Once the size of *EXA* exceeds the predefined size *N*, a fine-gained selection mechanism introduced by us [37] is executed to remain the population diversity. This is achieved by continuously deleting the most crowded individual and then updating its neighbor's crowding distance until the size of *EXA* is *N*. At last, only the individuals with lager crowding distances will be retained in *EXA*. The pseudo-code of archive update is described in Fig. 6, named **Algorithm 4**.

Algorithm 4: Archive Update

```
1   A = Union( EXA,P_child ); // merge the offspring and EXA to A
2   EXA = Find_nonDominated( A ); //find nondominated antibodies in A
3   DA = Find_Dominated ( A ); //find the dominated solutions in A
4   while ( |EXA| > N )
5       CrowdingDistanceAssignment(EXA); //calculate the crowding distance
6       EXA = Sort(EXA); //sort EXA according to the crowding distance
7       EXA = DeleteCrowded(EXA); //delete the most crowded one in EXA
8   end while
```

Fig. 6 The pseudo-code of archive update

3.5 The Complete Algorithm ADE-MOIA

The above subsections have described the procedures of initialization, cloning, ADE, perturbance and archive update, which compose the main components of ADE-MOIA. The pseudo-code of the complete algorithm ADE-MOIA is demonstrated in Fig. 7, named **Algorithm 5**. After the initialization process (**Algorithm 1**), the initial *EXA* and *DA* are obtained correspondingly. Then, ADE-MOIA turn into the loop of evolutionary process until the generation times *gen* reaches the predefined maximum times *max_gen*. During the evolutionary phase, the cloning operator (**Algorithm 2**) is performed first, which generates the evolutionary population *P*. Afterward, the ADE operator that is introduced in Section 3.2 and the perturbance operator (**Algorithm 3**) are executed on the evolutionary population *P*. In order to perform ADE operator, the values of *CR* and *F* are respectively determined according to Eqs. (12) and (13). Then, the intermediate individual y_i is generated based on Eqs. (10) and (11), and further disturbed to get the child z_i as described in **Algorithm 3**. The objectives of new child are evaluated and collected into the child population P_{child}. Once the parent x_i is dominated by the child z_i, this value of *F* will be recorded into the success value set $F_{success}$. If the size of $F_{success}$ is larger than *p*% of *N*, the value of F_m is updated by Eq. (14). At last, the archive update process in **Algorithm 4** is executed. The above evolutionary phase will repeat until the predefined maximum generation times are achieved. At the end of algorithm, the non-dominated solutions in *EXA* are reported as the final result.

Algorithm 5: The pseudo-code of ADE-MOIA

Initialization (Algorithm 1);
$gen = 0$;
$F_{success} = \{\}$;
while $gen < max\_gen$
  Cloning Operator (Algorithm 2);
  set the value of CR according to Eq. (12);
  for i = 1 to N
    set F_i according to Eq. (13);
    generate v_i according to Eq. (10);
    generate y_i according to Eq. (11);
    perform **perturbance** on y_i to get z_i **(Algorithm 3)**;
    evaluate z_i;
    add z_i to the child population P_{child}.
    if ($z_i \succ x_i$)
      $F_{success} = F_{success} \cup F_i$
    end if
  end for
  if $| F_{success} | > p\% \cdot N$
    update F_m according to Eq. (14)
    $F_{success} = \{\}$;
  end if
  Archive Update (Algorithm 4).
  set $gen = gen + 1$
end while
Output *EXA*

Fig. 7 The pseudo-code of complete ADE-MOIA

4. Experimental Results

In this section, in order to verify the performance of our algorithm, several experimental studies are launched. Firstly, the experiment-related knowledge, such as benchmark problems, performance metric and the parameter settings of all the compared algorithms, are introduced. Secondly, the performance of our proposed algorithm ADE-MOIA is compared with several competitive nature-inspired multi-objective algorithms, such as NSGA-II [16], SPEA2 [14], AbYSS [23], MOEA/D-DE [32], D^2MOPSO [26] and MIMO [37]. At last, in order to investigate the advantages of our proposed ADE operator, ADE-MOIA is further compared with its variants SBX-MOIA and DE-MOIA, in which SBX-MOIA and DE-MOIA replace the ADE operator with the SBX and basic DE operators, respectively. Moreover, the advantage of adaptive parameters control in ADE operator is also validated by the comparison with the fixed parameter settings.

4.1 Benchmark Problems

Various types of benchmark problems are used in order to validate the comprehensive performance of our proposed algorithm. First of all, the most widely used ZDT problems [41] are adopted. Due to the lack of characteristics such as variable linkages and objective function modality in the ZDT family, it is not very challenging to solve ZDT problems. Therefore, the bi-objective version of WFG family is included, which is generated by WFG toolbox in [40]. They have the various properties of convexity, concavity, discontinuity, non-uniform and the existence of many local optimal fronts, which are described in TABLE 1. Moreover, the triple-objective test suite, DTLZ [39], is also included to further assess the performance of our algorithm when solving MOPs with more than two objectives. Note that for ZDT1- ZDT3 test problems, the number of decision variables is 30, while the size of decision variables is 10 for ZDT4, ZDT6 and all of WFG and DTLZ test problems. Especially for WFG test problems, 10 decision variables are composed by 8 position parameters and 2 distance parameters. Totally, 21 test functions (ZDT1-ZDT4, ZDT6, WFG1-WFG9 and DTLZ1-DTLZ7) are covered in our experimental studies, which make the comparison results more comprehensive and convincible.

TABLE 1
PROPERTIES OF THE MOPS CREATED BY THE WFG TOOLKIT

Problems	Separability	Modality	Bias	Geometry
WFG1	separable	uni	polynomial, flat	convex, mixed
WFG2	non-separable	f1 uni, f2 multi	no bias	convex, disconnected
WFG3	non-separable	uni	no bias	linear, degenerate
WFG4	non-separable	multi	no bias	concave
WFG5	separable	deceptive	no bias	concave
WFG6	non-separable	uni	no bias	concave
WFG7	separable	uni	parameter dependent	concave
WFG8	non-separable	uni	parameter dependent	concave
WFG9	non-separable	multi, deceptive	parameter dependent	concave

4.2 Performance Measures

(1) The inverted generational distance metric

When comparing the performance of multi-objective algorithms, two evaluation criteria are generally considered. One is the convergence speed, which is observed from the distance between the approximated Pareto front and the true Pareto front. A closer distance generally indicates a faster convergence speed. The other one is the population diversity that is determined by the distribution of approximated Pareto front. The more uniform and smooth approximated Pareto front usually means the better population diversity. Since the inverted generational distance (IGD) metric [32] can simultaneously reflect the capabilities of convergence and diversity, it is adopted to evaluate the performance of all the algorithms in our experimental studies.

Assume that PF^* is a set of points evenly distributed along the true Pareto front and PF is a set of solutions that are found by the multi-objective algorithms. Then, the IGD metric is defined as

$$IGD(PF^*, PF) = \frac{\sqrt{\sum_{v \in PF^*} d^2(v, PF)}}{|PF^*|} \tag{17}$$

where $d(v, PF)$ returns the minimum Euclidean distance from v to the points of PF. If PF^* is distributed uniformly and large enough to represent the complete true Pareto front, the result of $IGD(PF^*, PF)$ not only expresses the diversity of PF, but also reflects the convergence capability. Theoretically, a closer distance of PF to PF^* and a more uniform distribution of PF will result in a smaller value of $IGD(PF^*, PF)$.

(2) The coverage of two sets metric

The coverage of two sets metric [56] is another performance metric, which can identify the Pareto-dominance relationship between two compared sets. Assuming that *A*, *B* are two approximated Pareto-optimal sets. The result of the coverage of two sets $I_C(A, B)$ returns the proportion of the individuals in *B* that are equal to or dominated by the individuals in *A*, which can be formulated as follows.

$$I_C(A, B) = \frac{|\{b \in B; \exists a \in A : a \succeq b\}|}{|B|} \tag{18}$$

where the symbol "$a \succeq b$" means *b* is equal to or dominated by *a*. The value $I_C(A, B) = 1$ means that all the individuals in *B* are equal to or dominated by the individuals in *A*, while $I_C(A, B) = 0$ implies no individual in *B* is equal to or dominated by the individuals in *A*. It is noted that both $I_C(A, B)$ and $I_C(B, A)$ are required to be considered as $I_C(A, B)$ is not necessarily equal to $1 - I_C(B, A)$. Therefore, for set *A*, it is preferred to get a larger value of $I_C(A, B)$ and a smaller value of $I_C(B, A)$.

4.3 Parameter Settings

In our simulations, in order to validate the performance of ADE-MOIA, we compare ADE-MOIA with six natural-inspired multi-objective algorithms, including two classical MOEAs (NSGA-II [16] and SPEA2 [14]), an archive hybrid scatter search algorithm (AbYSS) [23], a decomposition-based evolutionary multi-objective algorithm (MOEA/D-DE) [32], a novel PSO multi-objective algorithm (D^2MOPSO) [26] and a newly proposed micro-population immune multi-objective algorithm (MIMO) [37]. All these algorithms have shown very good performance in solving MOPs. Therefore, the comparison of ADE-MOIA with the above-mentioned algorithms can make the results more convincible.

As the parameter settings usually have great impact on the optimization performance, the parameter settings of all the six algorithms are set as suggested by their corresponding authors, which are summarized in TABLE 2. *N* represents the population size and the maximal generation is set to 250. Thus, the maximal numbers of function evaluations (*FEs*) are 25000 for all the benchmark MOPs. p_c and p_m are respectively the crossover and mutation probabilities. η_c and η_m are respectively the distribution indexes of SBX and PM. For AbYSS, $N_{RefSet1}$ and $N_{RefSet2}$ are the sizes of reference set 1 and 2, respectively. In MOEA/D-DE, *T* defines the size of neighborhood in the weight vector and δ controls the probability that chooses parent from the *T* neighbors. n_r is the maximal number that the

individuals are replaced by their neighbors. As introduced in MOEA/D-DE, the population size is $N = C_{H-m-1}^{m-1}$. When the number of objectives is 3 ($m = 3$), there is no exact H value giving $N = 100$. Thus, H is set to 13, which gives the population size 105. C_1 and C_2 are two control parameters in D^{2}MOPSO, which are randomly selected from [1.5, 2.5]. For MIMO, *NA* is the size of antibody population that is selected for cloning proliferation. *A* and *B* are two pre-defined parameters used to control the adaptive mutation operator in MIMO. The size of *EXA* is usually set as the population size *N*.

All the compared algorithms are implemented in the framework of jMetal [57], which facilitates the development of meta-heuristic algorithms for solving MOPs. All the experiments are run by 30 times. The median ($\tilde{x}$) and interquartile-range (*IQR*) of the 30 results are collected for comparison. The best results in the comparison table are marked with bold font. The Wilcoxon rank sum test with significant level $\alpha = 0.05$ is used to test whether the results of ADE-MOIA are statistically different from that of other algorithms.

TABLE 2
THE PARAMETER SETTINGS FOR ALL THE ALGORITHMS

Algorithms	Parameter settings
NSGA-II	$N = 100, p_c = 0.9, p_m = 1/n, \eta_c = 20, \eta_m = 20$
SPEA2	$N = 100, p_c = 0.9, p_m = 1/n, \eta_c = 20, \eta_m = 20$
AbYSS	$N = 100, N_{RefSet1} = 10, N_{RefSet2} = 10, p_c = 0.9, p_m = 1/n, \eta_c = 20, \eta_m = 20$
MOEA/D-DE	$N = 100, CR = 1.0, F = 0.5, p_m = 1/n, \eta_m = 20, T = 20, \delta = 0.9, n_r = 2$
D^{2}MOPSO	$N = 100, w \in [0.1, 0.5], C_1, C_2 = [1.5, 2.5]$
MIMO	$N = 100, NA = 20, p_c = 1.0, p_m = 1/n, \eta_c = 20, A = 1.0, B = 3.0$
ADE-MOIA	$N = 100, NA = 20, F = 0.5, CR = 0.5, p_m = 1/n, \eta_m = 20, p\% = 10\%$

4.4 Experimental Results

4.4.1 The Comparison of ADE-MOIA with Various Multi-objective Algorithms

(1) The comparison using the IGD metric

TABLE 3 shows the experimental results of all the algorithms in ZDT test suite, where the median ($\tilde{x}$), interquartile-range (*IQR*) and rank are respectively shown in the corresponding rows for one specific problem. It is clearly observed that ADE-MOIA performs better than the compared algorithms in ZDT1, ZDT2, ZDT3 and ZDT4 test problems. Although MOEA/D-DE performs best in ZDT6, it is not so good at solving ZDT1-ZDT4 problems. After summing all the ranks in ZDT problems, ADE-MOIA obtains the first rank, whereas D^{2}MOPSO gets the last rank. Moreover, the Wilcoxon rank sum test shows that ADE-MOIA has similar results with AbYSS and MIMO in ZDT6.

In TABLE 4, it has shown the comparison results of all the algorithms in WFG suite problems. It is observed from TABLE 4 that ADE-MOIA performs best in WFG1, WFG2, WFG4, WFG5, WFG6, WFG7 and WFG9, while AbYSS and MOEA/D-DE have the best performance in WFG3 and WFG8, respectively. The Wilcoxon rank sum test also shows that ADE-MOIA gets similar results with AbYSS in WFG4 and with D^{2}MOPSO in WFG4 and WFG9. By adding all the ranks of WFG problems,

ADE-MOIA is found to have obvious advantages as the total rank of ADE-MOIA is much smaller than that of the others. MIMO, D^{2}MOPSO, MOEA/D-DE, SPEA2, NSGA-II and AbYSS respectively get the 2th, 3th, 4th, 5th, 6th and 7th ranks according to the total rank. As WFG test suite is more complicated than ZDT test suite, the excellent performance of ADE-MOIA in WFG problems further confirms its superiority.

From the above experiments, it is found that ADE-MOIA has a very promising performance in ZDT and WFG serial problems. However, these problems only have bi-objective functions. For investigating the performance in solving MOPs with more than two objectives, DTLZ problems are further tested. In TABLE 5, the comparison results of all the algorithms are demonstrated for DTLZ test suite. It is observed that ADE-MOIA performs best in DTLZ1, DTLZ3 and DTLZ6, SPEA2 gets the best results in DTLZ2, DTLZ4 and DTLZ7, and AbYSS is best in DTLZ5. The Wilcoxon rank sum test shows that ADE-MOIA and MOEA/D-DE have similar results in DTLZ1, DTLZ3 and DTLZ4. Moreover, ADE-MOIA is similar with NSGA-II in DTLZ4 and DTLZ7, with ABYSS in DTLZ4 and DTLZ5, and with MIMO in DTLZ3 and DTLZ5. Based on the total rank in the last row of TABLE 5, ADE-MOIA has the best comprehensive performance on DTLZ problems. SPEA2 and MIMO receive the second and third ranks respectively. AbYSS and MOEA/D-DE perform fairly and both get the fourth rank. At last, NSGA-II and D^{2}MOPSO respectively receive the 6th and 7th ranks.

In TABLE 6, the total ranks of the compared algorithms for ZDT, WFG and DTLZ test suits are summarized for evaluating their comprehensive performance. When considering all the test problems, the final rank confirms that ADE-MOIA performs better than the compared multi-objective algorithms, i.e., NSGA-II, SPEA2, AbYSS, MOEA/D-DE, D^{2}MOPSO and MIMO.

TABLE 3
MEDIAN AND INTERQUARTILE RANGE OF THE IGD METRIC FOR ALL THE ALGORITHMS ON ZDT TEST SUITE

		NSGA-II	SPEA2	AbYSS	MOEA/D-DE	D^{2}MOPSO	MIMO	ADE-MOIA
ZDT1	$\tilde{x}$	2.666e-04	2.145e-04	2.006e-04	6.008e-04	5.375e-04	1.982e-04	**1.941e-04**
	IQR	2.05e-05	7.50e-06	6.70e-06	1.80e-04	1.26e-04	3.29e-06	**1.93e-06**
	rank	5	4	3	7	6	2	1
ZDT2	$\tilde{x}$	2.718e-04	2.211e-04	2.049e-04	4.518e-04	3.266e-02	2.051e-04	**2.003e-04**
	IQR	2.04e-05	1.03e-05	6.49e-06	2.32e-04	3.23e-02	3.16e-06	**2.48e-06**
	rank	5	4	2	6	7	3	1
ZDT3	$\tilde{x}$	4.102e-04	3.560e-04	3.331e-04	4.327e-03	1.150e-03	3.237e-04	**3.187e-04**
	IQR	2.10e-05	1.08e-05	5.61e-03	1.66e-03	5.02e-04	4.63e-06	**8.37e-06**
	rank	5	4	3	7	6	2	1
ZDT4	$\tilde{x}$	3.501e-04	4.118e-04	3.336e-04	8.751e-03	1.280e-01	1.975e-04	**1.948e-04**
	IQR	9.92e-05	1.32e-03	1.67e-04	1.46e-02	1.33e-01	3.60e-06	**2.85e-06**
	rank	4	5	3	6	7	2	1
ZDT6	$\tilde{x}$	2.948e-04	5.468e-04	1.288e-04	**1.072e-04**	4.738e-04	1.272e-04	1.274e-04
	IQR	4.72e-05	1.24e-04	7.72e-06	**3.71e-07**	1.67e-04	6.06e-06	7.05e-06
	rank	5	7	4≈	1	6	2≈	3
Total rank		24	24	15	27	32	11	7

"≈" indicates the results obtained by the algorithm are similar to the ones obtained by ADE-MOIA using the Wilcoxon rank sum test with significant level $\alpha = 0.05$.

TABLE 4
MEDIAN AND INTERQUARTILE RANGE OF THE IGD METRIC FOR ALL THE ALGORITHMS ON WFG TEST SUITE

		NSGA-II	SPEA2	AbYSS	MOEA/D-DE	D^2MOPSO	MIMO	ADE-MOIA
WFG1	$\tilde{x}$	7.144e-02	7.947e-02	8.092e-02	4.950e-03	5.561e-03	8.570e-04	**6.113e-04**
	IQR	8.62e-03	3.29e-03	3.63e-03	4.49e-03	3.18e-03	4.06e-03	**1.73e-05**
	rank	5	6	7	3	4	2	1
WFG2	$\tilde{x}$	3.569e-02	1.968e-02	5.172e-02	2.022e-02	2.115e-02	5.171e-02	**1.959e-02**
	IQR	3.21e-02	3.21e-02	1.89e-05	6.19e-04	7.91e-03	7.79e-06	**4.20e-06**
	rank	5	2	7	3	4	6	1
WFG3	$\tilde{x}$	4.381e-02	4.582e-02	**1.285e-03**	4.322e-02	4.356e-02	4.328e-02	4.322e-02
	IQR	5.54e-04	2.76e-03	**1.45e-03**	2.47e-06	3.51e-04	9.84e-05	3.68e-06
	rank	6	7	1	2	5	4	3
WFG4	$\tilde{x}$	5.158e-04	4.559e-04	3.797e-04	6.655e-04	3.764e-04	3.780e-04	**3.743e-04**
	IQR	7.57e-05	5.04e-05	8.34e-05	1.39e-04	4.76e-05	8.33e-06	**8.45e-06**
	rank	6	5	4≈	7	2≈	3	1
WFG5	$\tilde{x}$	1.418e-03	1.391e-03	2.522e-03	1.408e-03	1.307e-03	1.387e-03	**1.385e-03**
	IQR	1.44e-05	5.86e-06	4.66e-06	1.72e-06	2.34e-06	3.49e-06	**3.24e-06**
	rank	6	4	7	5	1	3	2
WFG6	$\tilde{x}$	4.824e-04	4.301e-04	1.984e-03	8.083e-04	5.273e-04	3.850e-04	**3.342e-04**
	IQR	1.05e-04	1.75e-04	2.18e-03	4.62e-04	2.21e-04	1.38e-04	**2.30e-05**
	rank	4	3	7	6	5	2	1
WFG7	$\tilde{x}$	8.812e-04	1.486e-03	2.118e-03	5.621e-04	4.303e-04	5.277e-04	**2.874e-04**
	IQR	3.35e-04	9.30e-04	2.38e-03	4.82e-04	1.49e-05	2.51e-04	**1.17e-05**
	rank	5	6	7	4	2	3	1
WFG8	$\tilde{x}$	8.417e-04	9.780e-04	1.129e-02	**4.925e-04**	1.238e-03	8.330e-04	6.021e-04
	IQR	1.44e-04	1.71e-04	4.02e-02	**8.49e-05**	4.53e-04	1.36e-04	8.61e-05
	rank	4	5	7	1	6	3	2
WFG9	$\tilde{x}$	4.114e-04	3.380e-04	3.781e-04	4.010e-04	3.081e-04	3.290e-04	**3.069e-04**
	IQR	5.39e-05	3.48e-05	1.50e-04	1.66e-05	3.29e-05	5.67e-05	**1.41e-05**
	rank	7	4	5	6	2≈	3	1
Total rank		48	42	52	37	31	29	13

"≈" indicates the results obtained by the algorithm are similar to the ones obtained by ADE-MOIA using the Wilcoxon rank sum test with significant level $\alpha = 0.05$.

TABLE 5
MEDIAN AND INTERQUARTILE RANGE OF THE IGD METRIC FOR ALL THE ALGORITHMS ON DTLZ TEST SUITE

		NSGA-II	SPEA2	AbYSS	MOEA/D-DE	D^{2}MOPSO	MIMO	ADE-MOIA
DTLZ1	$\tilde{x}$	5.495e-03	4.850e-03	1.165e-02	5.365e-04	1.153e-01	3.700e-03	**4.773e-04**
	IQR	4.78e-03	4.72e-03	1.23e-02	3.01e-03	1.12e-01	8.77e-03	**3.21e-03**
	rank	5	4	6	2≈	7	3	1
DTLZ2	$\tilde{x}$	7.836e-04	**5.881e-04**	7.911e-04	8.097e-04	6.939e-04	7.505e-04	7.231e-04
	IQR	5.22e-05	**2.19e-05**	5.41e-05	7.19e-06	2.16e-05	5.39e-05	4.72e-05
	rank	5	1	6	7	2	4	3
DTLZ3	$\tilde{x}$	2.811e-02	3.226e-02	3.636e-02	1.429e-03	6.336e-01	1.396e-03	**1.390e-03**
	IQR	2.13e-02	2.53e-02	7.56e-02	7.25e-03	4.67e-01	1.76e-03	**1.16e-02**
	rank	4	5	6	3≈	7	2≈	1
DTLZ4	$\tilde{x}$	1.167e-03	**7.750e-04**	1.119e-03	1.168e-03	1.229e-03	1.197e-03	1.136e-03
	IQR	1.39e-04	**4.56e-03**	1.40e-04	5.78e-04	1.22e-04	1.57e-04	1.38e-04
	rank	4≈	1	2≈	5≈	7	6	3
DTLZ5	$\tilde{x}$	3.803e-04	2.770e-04	**2.668e-04**	1.044e-03	4.930e-04	2.730e-04	2.698e-04
	IQR	4.28e-05	1.45e-05	**9.91e-06**	1.25e-05	1.66e-04	9.59e-06	8.75e-06
	rank	5	4	1≈	7	6	3≈	2
DTLZ6	$\tilde{x}$	4.373e-02	4.336e-02	2.207e-02	1.571e-03	2.969e-03	4.139e-04	**4.008e-04**
	IQR	8.62e-03	4.03e-03	1.45e-02	6.35e-06	8.16e-04	1.79e-05	**1.41e-05**
	rank	7	6	5	3	4	2	1
DTLZ7	$\tilde{x}$	3.409e-03	**2.558e-03**	1.936e-02	9.520e-03	4.022e-03	3.890e-03	3.470e-03
	IQR	3.93e-04	**9.02e-05**	2.09e-02	7.31e-04	6.52e-04	6.67e-04	3.04e-04
	rank	2≈	1	7	6	5	4	3
Total rank		32	22	33	33	38	24	14

"≈" indicates the results obtained by the algorithm are similar to the ones obtained by ADE-MOIA using the Wilcoxon rank sum test with significant level $\alpha = 0.05$.

TABLE 6
FINAL RANK OF ALL THE ALGORITHMS ON ZDT, WFG AND DTLZ TEST SUITES

Problems \ Algorithms	NSGA-II	SPEA2	AbYSS	MOEA/D-DE	D^{2}MOPSO	MIMO	ADE-MOIA
Total Rank on ZDTs	24	24	15	27	32	11	7
Total Rank on WFGs	48	42	52	37	31	29	13
Total Rank on DTLZs	32	22	33	33	38	24	14
Final Rank	104	88	100	97	101	64	34

(2) The comparison using the coverage of two sets metric

The approximated sets obtained by all the algorithms in 30 independent runs are further compared using the coverage of two sets metric. Figures 8-13 respectively show the box plots of ADE-MOIA against NSGA-II, SPEA2, AbYSS, MOEA/D-DE, D^{2}MOPSO and MIMO based on the coverage of two sets metric. In order to check their statistical difference, TABLE 7 presents the *p*-values of wilcoxon rank test that are performed on their 30 independent values of the coverage of two sets, where the *p*-values larger than the significant level 5% indicate no statistical difference. It is noted that in Figs.

8-13, **I, N, S, A, D, P** and **O** respectively denote the solution sets obtained by ADE-MOIA, NSGA-II, SPEA2, AbYSS, MOEA/D-DE, D^{2}MOPSO and MIMO.

In Fig. 8, it is clearly observed that all the values of I_C(**I**, **N**) for ZDT4, ZDT6, DTLZ1, DTLZ3, DTLZ6, WFG1 are greater than 0.5 and some of them are near to 1.0, while the results of I_C(**N**, **I**) on the corresponding test problems are very small, which indicate that most of the solutions obtained by NSGA-II are dominated by that achieved by ADE-MOIA on these test problems. Thus, ADE-MOIA performs much better than NSGA-II on these test problems. For the other 15 test problems, it shows that only minority solutions are weakly dominated by each other. However, when only considering the results of the coverage of two sets, the median values of I_C(**I**, **N**) are higher than that of I_C(**N**, **I**) on all the test problems except WFG8, which means that the ratios of dominated solutions in NSGA-II are larger than that in ADE-MOIA. In this sense, ADE-MOIA is better than NSGA-II on all the test problems except WFG8.

From Fig. 9, it is also found that ADE-MOIA performs much better than SPEA2 in ZDT4, ZDT6, DTLZ1, DTLZ3, DTLZ6, WFG1, as the values of I_C(**I**, **S**) are greater than 0.5 while the corresponding results of I_C(**S**, **I**) are very small. The second column in TABLE 7 shows that ADE-MOIA obtains the similar result with SPEA2 in WFG9. Besides that, for the other 14 test problems, the box plots and the Wilcoxon rank test indicate that ADE-MOIA performs better than SPEA2 in ZDT1-ZDT3, DTLZ2, DTLZ4, DTLZ5, DTLZ7, WFG2-WFG7, and worse than SPEA2 in WFG8. Therefore, ADE-MOIA can obtain the better results than SPEA2 on all the test problems except WFG8 and WFG9.

The comparison of ADE-MOIA and AbYSS is illustrated in Fig. 10, where the values of I_C(**I**, **A**) for ZDT4, DTLZ1, DTLZ3, DTLZ6, WFG1, WFG2 are greater than 0.5 while the corresponding results of I_C(**A**, **I**) are near to 0. Thus, the majority of the solutions obtained by AbYSS are dominated by that found by ADE-MOIA. For the rest 15 test problems, as it is observed from the box plots and the third column in TABLE 7, ADE-MOIA performs better than AbYSS in ZDT1-ZDT3, ZDT6, DTLZ2, WFG3-WFG7, WFG9, similarly with AbYSS in DTLZ4, and worse than AbYSS in DTLZ5, DTLZ7, WFG8. In other words, ADE-MOIA is better than AbYSS on 17 out of 21 test problems.

In Fig. 11, it is observed that the values of I_C(**I**, **D**) for ZDT1-ZDT4 are greater than 0.5 while the corresponding values of I_C(**D**, **I**) are very close to 0. Thus, ADE-MOIA performs much better than MOEA/D-DE in ZDT1-ZDT4, as most of the corresponding solutions obtained by MOEA/D-DE are dominated by that achieved by ADE-MOIA. For the remaining test problems, the box plots and the forth column in TABLE 7 show that ADE-MOIA performs better than MOEA/D-DE in DTLZ2-DTLZ5, WFG2, WFG4, WFG5, WFG7, WFG9, similarly with MOEA/D-DE in ZDT6, DTLZ1, DTLZ6, DTLZ7, WFG1, WFG3, WFG6, and worse than MOEA/D-DE in WFG8. In other words, when compared with MOEA/D-DE, ADE-MOIA respectively performs better, similarly and worse on 13, 7 and 1 out of 21 test problems.

Observed from Fig. 12, the values of I_C(**I**, **P**) for ZDT1-ZDT4, ZDT6, DTLZ1, DTLZ3, DTLZ6 are greater than 0.5 while the corresponding results of I_C(**P, I**) are close to 0. Thus, ADE-MOIA performs much better than D^{2}MOPSO on these test problems. For the rest test problems, it can be observed from the box plots and the fifth column in TABLE 7 that ADE-MOIA performs better than D^{2}MOPSO in DTLZ2, DTLZ4, DTLZ5, DTLZ7, WFG2, WFG4-WFG7, WFG9, similarly with

D^{2}MOPSO in WFG1, WFG3, and worse than D^{2}MOPSO in WFG8. Therefore, ADE-MOIA is better than or similar with D^{2}MOPSO on 20 out of 21 test problems.

The comparison of ADE-MOIA and MIMO is illustrated in Fig. 13. For DTLZ1 and WFG2, ADE-MOIA is much better than MIMO, as the corresponding values of I_C(**I**, **O**) are much larger than that of I_C(**O** ,**I**). Regarding to the other test problems, by observing from the box plots and the sixth column in TABLE 7, it is very clear that ADE-MOIA performs better than MIMO in ZDT1, ZDT2, ZDT4, DTLZ2, DTLZ4, DTLZ5, DTLZ7, WFG3-WFG7, WFG9, similarly with MIMO in ZDT6, DTLZ3, DTLZ6, WFG1, and worse than MIMO in ZDT3 and WFG8. Thus, ADE-MOIA performs better than or similarly with MIMO on 19 out of 21 test problems.

At last, based on the final comparison results of the IGD metric and the coverage of two sets metric, it is reasonable to conclude that ADE-MOIA performs better than NSGA-II, SPEA2, AbYSS, MOEA/D-DE, D^{2}MOPSO and MIMO on most of the test problems.

TABLE 7
THE p-VALUES OF WILCOXON RANK SUM TEST BETWEEN I_C(**I**,~) AND I_C(~,**I**) ON THE 30 INDEPENDENT RUNS

	ADE-MOIA / NSGA-II	ADE-MOIA / SPEA2	ADE-MOIA / AbYSS	ADE-MOIA / MOEA/D-DE	ADE-MOIA / D^{2}MOPSO	ADE-MOIA / MIMO
ZDT1	8.87e-12(-)	7.47e-12(-)	5.81e-11(-)	3.23e-13(-)	1.19e-12(-)	8.46e-11(-)
ZDT2	5.02e-12(-)	4.05e-12(-)	2.06e-11(-)	1.74e-11(-)	3.23e-13(-)	1.27e-10(-)
ZDT3	2.45e-12(-)	2.36e-11(-)	6.84e-04(-)	1.69e-14(-)	1.18e-12(-)	1.89e-04(+)
ZDT4	5.66e-12(-)	3.15e-12(-)	3.42e-12(-)	1.69e-14(-)	1.68e-14(-)	1.92e-08(-)
ZDT6	1.10e-12(-)	2.71e-14(-)	1.15e-12(-)	1.00 (≈)	8.30e-13(-)	1.00 (≈)
DTLZ1	9.17e-09(-)	2.71e-08(-)	5.42e-12(-)	0.0667 (≈)	4.88e-14(-)	4.13e-05(-)
DTLZ2	1.17e-11(-)	1.34e-11(-)	1.24e-09(-)	1.30e-11(-)	1.21e-11(-)	7.84e-09(-)
DTLZ3	7.25e-13(-)	6.00e-13(-)	4.11e-12(-)	0.0484 (-)	1.68e-14(-)	0.8023 (≈)
DTLZ4	4.99e-11(-)	2.46e-08(-)	0.2311 (≈)	1.42e-11(-)	6.10e-12(-)	1.19e-04(-)
DTLZ5	1.93e-11(-)	2.02e-11(-)	6.55e-10(+)	1.17e-11(-)	1.64e-11(-)	7.39e-04(-)
DTLZ6	1.69e-14(-)	1.69e-14(-)	1.69e-14(-)	0.1097 (≈)	2.24e-12(-)	1.00 (≈)
DTLZ7	2.53e-11(-)	4.83e-10(-)	1.44e-05(+)	0.9350 (≈)	2.88e-11(-)	5.02e-05(-)
WFG1	1.83e-08(-)	5.36e-09(-)	1.69e-14(-)	0.8510 (≈)	1.00 (≈)	0.2949 (≈)
WFG2	1.71e-12(-)	1.21e-12(-)	1.20e-12(-)	1.67e-11(-)	2.45e-11(-)	4.62e-12(-)
WFG3	3.67e-07(-)	8.93e-08(-)	7.76e-11(-)	0.7378 (≈)	0.2887(≈)	4.23e-09(-)
WFG4	1.24e-09(-)	2.57e-07(-)	1.13e-08(-)	4.79e-11(-)	1.87e-11(-)	2.31e-10(-)
WFG5	3.28e-10(-)	1.08e-05(-)	1.98e-05(-)	1.63e-05(-)	1.22e-08(-)	0.0036 (-)
WFG6	3.11e-09(-)	4.34e-09(-)	1.21e-11(-)	0.2757 (≈)	3.94e-08(-)	4.33e-10(-)
WFG7	2.49e-11(-)	3.19e-10(-)	8.76e-10(-)	7.06e-09(-)	1.05e-08(-)	7.00e-11(-)
WFG8	2.92e-11(+)	2.90e-11(+)	0.0068 (+)	4.62e-11(+)	2.89e-11(+)	6.21e-09(+)
WFG9	1.85e-04(-)	0.8523 (≈)	2.11e-09(-)	0.0013 (-)	5.59e-05(-)	0.0184 (-)

"-", "+" and "≈" denote that the performance of the corresponding algorithm is worse than, better than, and similar to that of ADE-MOIA, respectively.

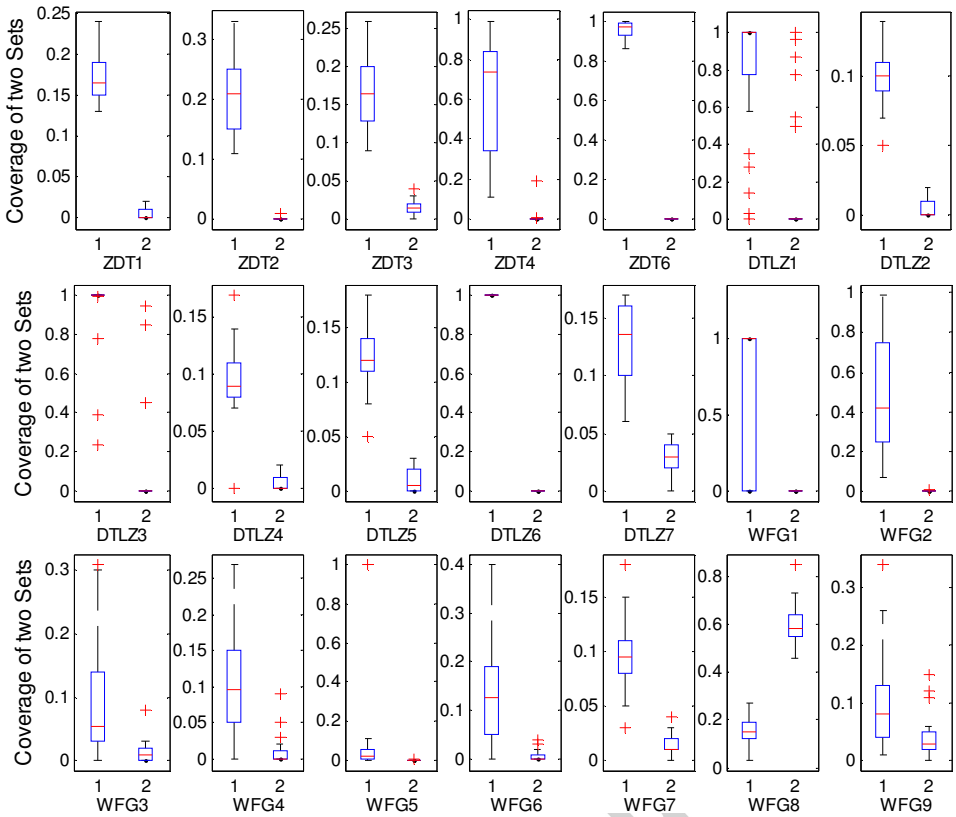


Fig. 8 Box plots of the coverage of two sets obtained by ADE-MOIA and NSGA-II in solving all the test problems. In each plot, the left box represents the distribution of I_C(**I**, **N**) and the right box indicates the distribution of I_C(**N**, **I**)

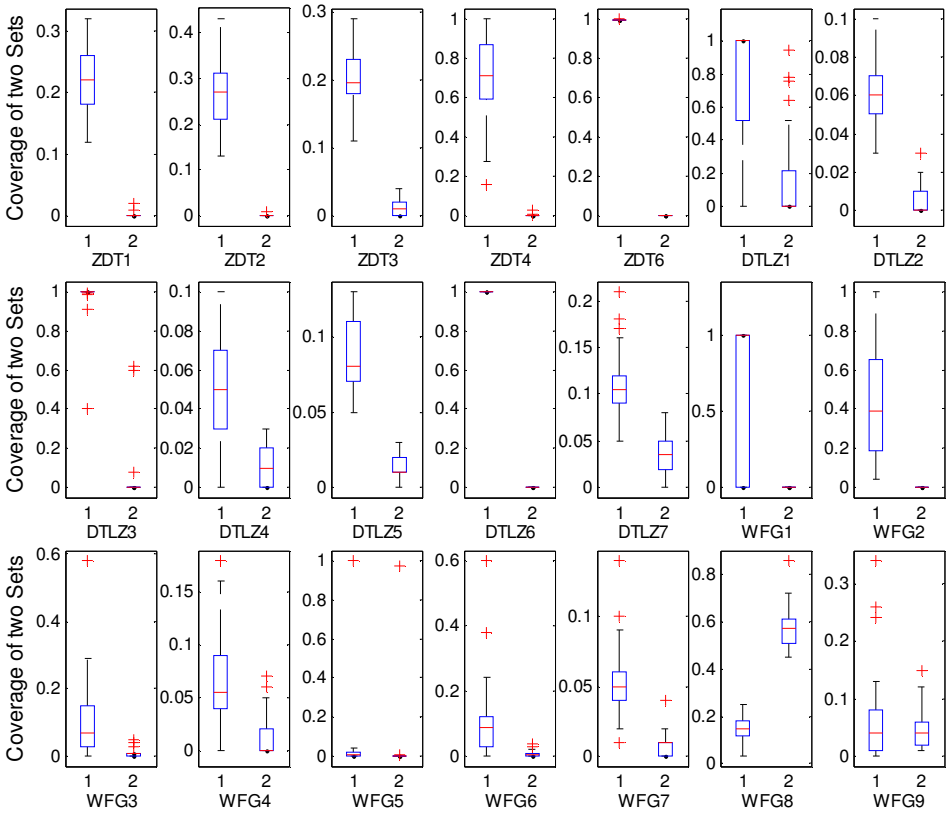


Fig. 9 Box plots of the coverage of two sets obtained by ADE-MOIA and SPEA2 in solving all the test problems. In each plot, the left box represents the distribution of I_C(**I**, **S**) and the right box indicates the distribution of I_C(**S**, **I**)

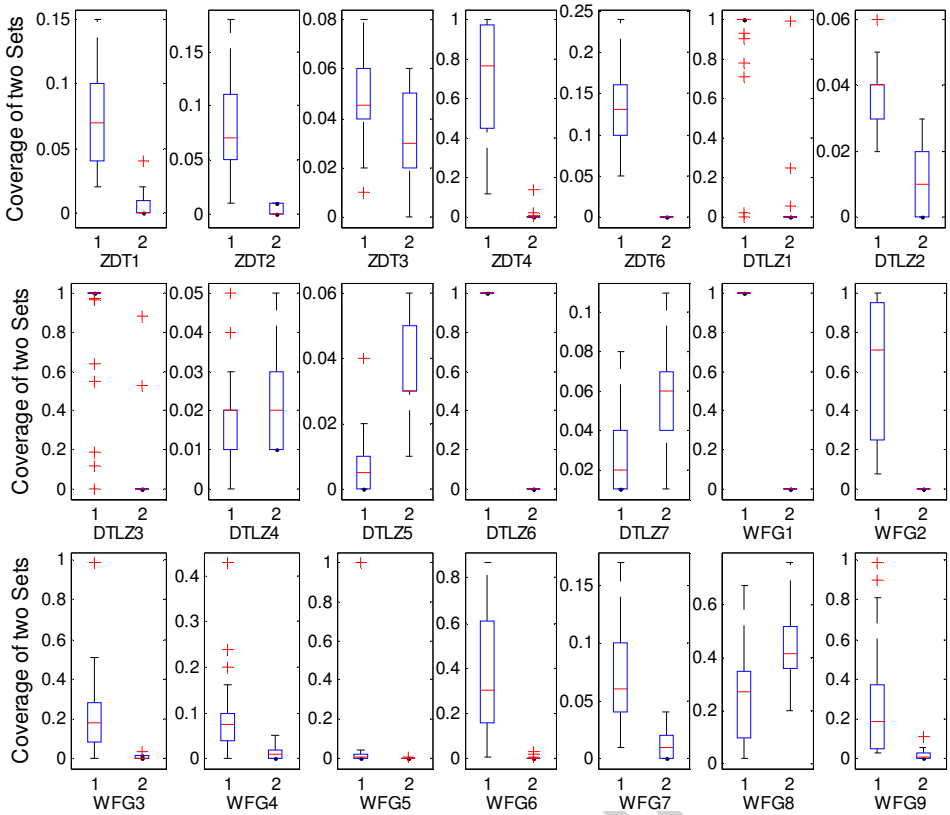


Fig. 10 Box plots of the coverage of two sets obtained by ADE-MOIA and AbYSS in solving all the test problems. In each plot, the left box represents the distribution of $I_C(\mathbf{I}, \mathbf{A})$ and the right box indicates the distribution of $I_C(\mathbf{A}, \mathbf{I})$

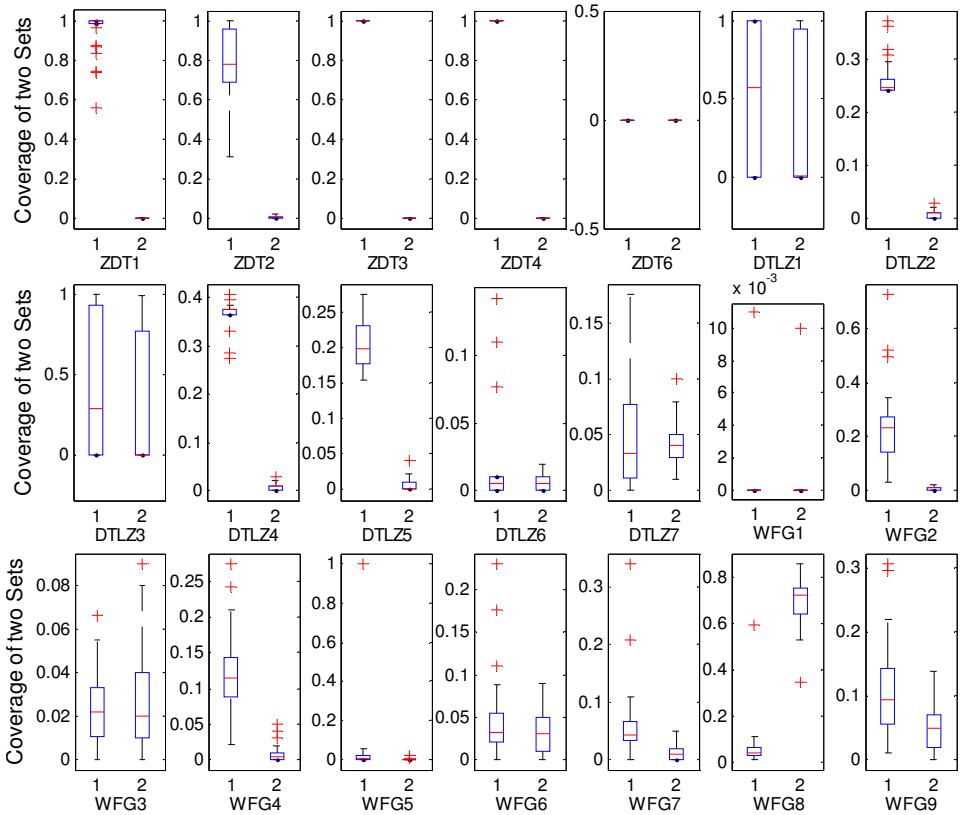


Fig. 11 Box plots of the coverage of two sets obtained by ADE-MOIA and MOEA/D-DE in solving all the test problems. In each plot, the left box represents the distribution of $I_C(\mathbf{I}, \mathbf{D})$ and the right box indicates the distribution of $I_C(\mathbf{D}, \mathbf{I})$

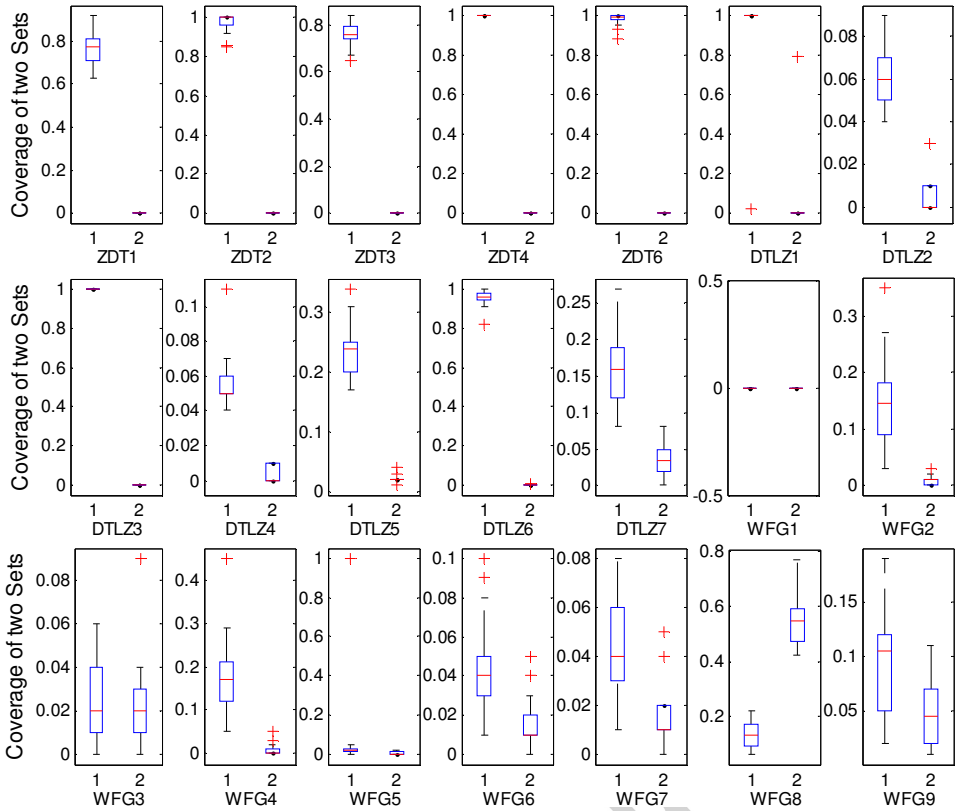


Fig. 12 Box plots of the coverage of two sets obtained by ADE-MOIA and D^{2}MOPSO in solving all the test problems. In each plot, the left box represents the distribution of I_C(**I**, **P**) and the right box indicates the distribution of I_C(**P**, **I**)

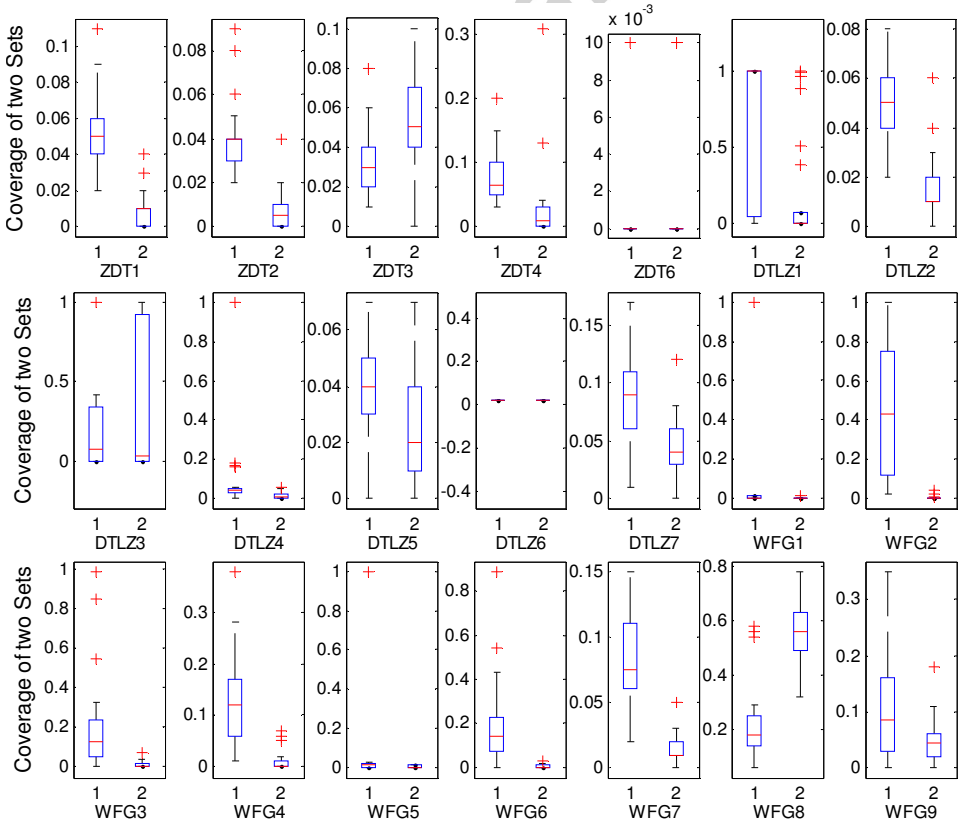


Fig. 13 Box plots of the coverage of two sets obtained by ADE-MOIA and MIMO in solving all the test problems. In each plot, the left box represents the distribution of I_C(**I**, **O**) and the right box indicates the distribution of I_C(**O**, **I**)

4.4.2 The Effectiveness of ADE Operator

(1) The Comparative Studies of Basic DE, SBX and ADE Operators

To demonstrate the advantages of ADE operator, the performance of ADE-MOIA is further compared with the two variants of ADE-MOIA, such as DE-MOIA and SBX-MOIA. DE-MOIA adopts the basic DE operator introduced in MOEA/D-DE [32] instead of ADE operator, while SBX-MOIA replaces the ADE operator with SBX operator, which is mostly exploited in various multi-objective algorithms [27, 29-33, 37]. Their comparison results in all ZDT, WFG and DTLZ problems are shown in TABLE 8, where the best results are identified with boldface.

TABLE 8
MEDIAN AND INTERQUARTILE RANGE OF THE IGD METRIC FOR DIFFERENT EVOLUTION OPERATOR

Problems \ Algorithm	SBX-MOIA $\tilde{x}_{IQR}$	DE-MOIA $\tilde{x}_{IQR}$	ADE-MOIA $\tilde{x}_{IQR}$
ZDT1	2.041e-04 4.70e-06 -	1.987e-04 3.24e-06 -	**1.941e-04 1.93e-06**
ZDT2	2.080e-04 3.03e-06 -	2.040e-04 3.70e-06 -	**2.003e-04 2.48e-06**
ZDT3	3.259e-04 7.93e-06 -	3.202e-04 6.75e-06 ≈	**3.187e-04 8.36e-06**
ZDT4	2.750e-04 1.11e-04 -	1.973e-04 5.28e-06 -	**1.948e-04 2.85e-06**
ZDT6	1.313e-04 8.64e-06 -	1.280e-04 7.65e-06 ≈	**1.274e-04 7.05e-06**
WFG1	6.105e-02 1.35e-02 -	6.191e-04 1.73e-02 -	**6.113e-04 1.73e-05**
WFG2	5.171e-02 3.21e-02 -	1.959e-02 3.21e-02 ≈	**1.959e-02 4.20e-06**
WFG3	4.338e-02 1.60e-04 -	4.322e-02 1.22e-05 -	**4.322e-02 3.68e-06**
WFG4	3.808e-04 1.92e-05 -	**3.719e-04 1.12e-05** ≈	3.743e-04 8.45e-06
WFG5	1.389e-03 4.06e-06 -	1.391e-03 1.22e-05 -	**1.385e-03 3.24e-06**
WFG6	4.019e-04 1.48e-04 -	5.184e-04 1.86e-04 -	**3.342e-04 2.30e-05**
WFG7	5.928e-04 4.32e-04 -	2.958e-04 2.26e-05 -	**2.874e-04 1.17e-05**
WFG8	7.419e-04 1.98e-04 -	**4.099e-04 6.56e-05** +	6.021e-04 8.61e-05
WFG9	3.272e-04 2.20e-05 -	3.344e-04 4.94e-05 -	**3.069e-04 1.41e-05**
DTLZ1	1.310e-02 1.13e-02 -	1.243e-02 1.44e-02 -	**4.773e-04 3.21e-03**
DTLZ2	7.721e-04 6.61e-05 -	**6.820e-04 2.69e-05** +	7.231e-04 4.72e-05
DTLZ3	3.362e-02 3.59e-02 -	1.459e-02 2.89e-02 -	**1.390e-03 1.16e-02**
DTLZ4	1.193e-03 1.11e-04 -	1.212e-03 1.16e-04 -	**1.136e-03 1.38e-04**
DTLZ5	2.707e-04 7.88e-06 ≈	2.729e-04 9.91e-06 -	**2.700e-04 8.75e-06**
DTLZ6	3.784e-02 1.11e-02 -	4.039e-04 1.51e-05 ≈	**4.008e-04 1.41e-05**
DTLZ7	3.829e-03 4.93e-04 -	3.662e-03 6.62e-04 -	**3.470e-03 3.04e-04**
-/+/≈	20/0/1	14/2/5	

The symbols of "-", "+" and "≈" denote that the performance of the corresponding algorithm is worse than, better than, and similar to that of ADE-MOIA, respectively.

Considering the ZDT test functions, the experimental results show that ADE-MOIA is better than SBX-MOIA in all ZDT problems, and better than DE-MOIA in ZDT1, ZDT2 and ZDT4. The Wilcoxon

rank sum test indicates that ADE-MOIA and DE-MOIA have similar performance in ZDT3 and ZDT6. Especially in ZDT4 that has many local Pareto fronts, ADE-MOIA performs much better. This is probably because the proposed ADE operator can adaptively adjust its scaling factor, which makes ADE-MOIA jump out from the local optimum.

Concerning the WFG problems in TABLE 8, ADE-MOIA performs better than SBX-MOIA in all WFG serial problems. ADE-MOIA gives pretty well results in the problems of WFG1, WFG2 and WFG6-WFG9. This is probably because the proposed ADE operator is especially suitable for the complex problems with variable linkages. When compared with DE-MOIA, ADE-MOIA performs better in WFG1, WFG3, WFG5-WFG7 and WFG9, and worse in WFG8. The Wilcoxon rank sum test demonstrates that they have comparable results in WFG2 and WFG4.

Regarding the DTLZ problems in TABLE 8, ADE-MOIA performs better than SBX-MOIA in all DTLZ problems except DTLZ5, and better than DE-MOIA in DTLZ1 and DTLZ3-7. The Wilcoxon rank sum test shows that ADE-MOIA performs similarly with SBX-MOIA in DTLZ5, and with DE-MOIA in DTLZ6. These simulations also validate that ADE-MOIA is especially suitable for the MOPs characterized with many local Pareto fronts as it performs very well in DTLZ1 and DTLZ3.

In the last row of TABLE 8, the symbols of "-", "+" and "≈" respectively denote that the performance of the corresponding algorithm is worse than, better than, and similar with that of ADE-MOIA. It is obviously observed that the statistical results of ADE-MOIA are better than or similar with SBX-MOIA in all test problems, and better than or similar with DE-MOIA in 19 out of 21 test problems. Therefore, it is convincible to conclude that our proposed ADE operator is more effective when compared with the commonly used SBX operator and the basic DE operator.

(2) The Advantages of Adaptive Control Parameter

In ADE-MOIA, *CR* is gradually decreased with the running of evolutionary process and the *F* value is dynamically updated by the feedback of successfully produced offspring. When solving SOPs, there are lots of adaptive control mechanisms reported for DE [27, 28]. However, the adaptive control parameter for DE is rare in solving MOPs as most of the DE operators in MOEAs usually use a fixed parameter. As the two parameters of DE operator have great impact on the performance, an adaptive approach to control these parameters is extremely important. To illustrate the effectiveness of adaptive control method, ADE-MOIA is compared with its two variants, i.e., DE1-MOIA and DE2-MOIA, both of which use the popular parameter settings. In our experiments, DE1-MOIA fixedly set *F* as 0.5 and *CR* as 0.5, while DE2-MOIA has the parameter settings with F=0.5 and CR=1.0.

TABLE 9 illustrates the results of ADE-MOIA and its variants (DE1-MOIA, DE2-MOIA) in all ZDT, WFG and DTLZ problems. ADE-MOIA is better than DE2-MOIA in all ZDT test functions. This is probably because the offspring has not inherited the parent's information as the crossover rate (*CR*) is set as 1.0. When compared with DE1-MOIA, ADE-MOIA is better in ZDT1 and ZDT2, and similar in ZDT3, ZDT4 and ZDT6. Looking at the WFG problems, ADE-MOIA is better than both DE1-MOIA and DE2-MOIA in WFG1, WFG5, WFG6, WFG7 and WFG8. Besides that, ADE-MOIA gets the similar results with DE1-MOIA and DE2-MOIA in WFG2. For WFG3, ADE-MOIA performs similarly with DE1-MOIA and slightly worse than DE2-MOIA. Moreover, ADE-MOIA is worse than

DE1-MOIA but better than DE2-MOIA on WFG4. Considering the DTLZ problems, ADE-MOIA performs better in DTLZ5 and DTLZ6, similarly in DTLZ1, DTLZ3, DTLZ4 and DTLZ7, and slightly worse in DTLZ2, when compared with DE1-MOIA. To sum up, ADE-MOIA is better than DE1-MOIA in DTLZ serials. For DE2-MOIA, it is obvious that ADE-MOIA is better in all DTLZ functions except a slightly worse in DTLZ2.

In the last row of TABLE 9, the symbols of "-", "+" and "≈" denote that the performance of the corresponding algorithm is worse than, better than, and similar with that of ADE-MOIA, respectively. From these statistical results, it is shown that ADE-MOIA has 9 problems better than, 3 problems worse than and 9 problems similar with DE1-MOIA. Besides that, ADE-MOIA has 17 problems better than, 3 problems worse than and one problem similar with DE2-MOIA. Based on the above analysis, it is reasonably concluded that the proposed adaptive parameter control is able to get better results than the popular fixed parameter settings in most cases.

TABLE 9
MEDIAN AND INTERQUARTILE RANGE OF THE IGD METRIC FOR DE1-MOIA, DE2-MOIA AND ADE-MOIA

Problem \ Algorithm	DE1-MOIA	DE2-MOIA	ADE-MOIA
	$\tilde{x}_{IQR}$	$\tilde{x}_{IQR}$	$\tilde{x}_{IQR}$
ZDT1	1.980e-04 3.00e-06 -	2.828e-04 1.43e-04 -	**1.941e-04 1.93e-06**
ZDT2	2.044e-04 4.61e-06 -	3.177e-04 1.09e-04 -	**2.003e-04 2.48e-06**
ZDT3	3.202e-04 8.18e-06 ≈	1.338e-03 1.10e-03 -	**3.187e-04 8.37e-06**
ZDT4	1.954e-04 4.35e-06 ≈	4.586e-04 3.85e-04 -	**1.948e-04 2.85e-06**
ZDT6	1.293e-04 9.77e-06 ≈	1.497e-04 2.40e-05 -	**1.274e-04 7.05e-06**
WFG1	2.081e-02 2.29e-02 -	6.170e-04 4.40e-03 -	**6.113e-04 1.73e-05**
WFG2	1.959e-02 3.21e-02 ≈	1.959e-02 1.47e-05 ≈	**1.959e-02 4.20e-06**
WFG3	4.322e-02 1.96e-06 ≈	**4.322e-02 3.36e-07 +**	4.322e-02 3.68e-06
WFG4	**3.679e-04 1.17e-05 +**	3.810e-04 1.78e-05 -	3.743e-04 8.45e-06
WFG5	1.388e-03 4.50e-06 -	1.402e-03 1.27e-05 -	**1.385e-03 3.24e-06**
WFG6	3.437e-04 4.23e-04 -	6.407e-04 2.76e-04 -	**3.342e-04 2.30e-05**
WFG7	2.920e-04 9.62e-06 -	4.370e-04 5.95e-04 -	**2.874e-04 1.17e-05**
WFG8	5.576e-04 7.11e-05 +	**3.183e-04 3.24e-05 +**	6.021e-04 8.61e-05
WFG9	3.308e-04 4.91e-05 -	3.446e-04 5.74e-05 -	**3.069e-04 1.41e-05**
DTLZ1	2.964e-03 2.88e-03 ≈	3.326e-02 5.07e-02 -	**4.773e-04 3.21e-03**
DTLZ2	**6.814e-04 2.25e-05 +**	7.003e-04 2.81e-05 +	7.231e-04 4.72e-05
DTLZ3	**1.181e-03 1.49e-02 ≈**	1.076e-01 1.68e-01 -	1.390e-03 1.16e-02
DTLZ4	**1.132e-03** 1.71e-04 ≈	1.222e-03 2.09e-04 -	1.136e-03 **1.38e-04**
DTLZ5	2.747e-04 8.89e-06 -	2.877e-04 1.32e-05 -	**2.698e-04 8.75e-06**
DTLZ6	4.084e-04 1.57e-05 -	4.058e-04 2.07e-05 -	**4.008e-04 1.41e-05**
DTLZ7	**3.458e-03 3.96e-04 ≈**	4.141e-03 4.46e-04 -	3.470e-03 3.04e-04
-/+/≈	9/3/9	17/3/1	

The symbols of "-", "+" and "≈" denote that the performance of the corresponding algorithm is worse than, better than, and similar to that of ADE-MOIA, respectively.

5. Conclusions

In this paper, we propose a novel hybrid multi-objective immune algorithm with adaptive DE. Its evolutionary strategy employs a novel adaptive DE operation with a suitable parent selecting strategy and a novel adaptive parameter control method. The appropriate parent selection for DE operator is able to provide a correct evolutionary direction. Besides that, the crossover rate (*CR*) is gradually decreasing with the process of evolution and the scaling factor (*F*) is dynamically updated based on the success rate of offspring. The proposed adaptive adjustment of *F* and *CR* provides a good balance between exploration and exploitation. The effectiveness of the proposed adaptive DE operations, including the parent selecting strategy and adaptive parameter control, is validated by the experimental studies. When compared with NSGA-II, SPEA2, AbYSS, MOEA/D-DE, D^2MOPSO and MIMO on 21 benchmark test problems, the proposed algorithm shows better performance on most cases. In our future work, the performance of our algorithm will be further studied by using different DE strategies. The hybridization of various DE strategies in our algorithm may further enhance the exploration capability and thus improve the optimization performance. Besides that, the application of our algorithm for some practical engineering problems will also be investigated.

Acknowledge

This work was supported by the National Nature Science Foundation of China under Grants 61170283 and 61402291, National High-Technology Research and Development Program ("863" Program) of China under Grant 2013AA01A212, Ministry of Education in the New Century Excellent Talents Support Program under Grant NCET-12-0649 and Foundation for Distinguished Young Talents in Higher Education of Guangdong under Grant 2014KQNCX129.

Reference:

[1] Silva MM, Subramanian A, Ochi LS. An iterated local search heuristic for the split delivery vehicle routing problem. Computers & Operations Research 2015; 53: 234-49.

[2] Baatar N, Pham M-T, Koh C-S. Multiguiders and Nondominate Ranking Differential Evolution Algorithm for Multiobjective Global Optimization of Electromagnetic Problems. IEEE Transactions on Magnetics 2013; 49(5): 2105-08.

[3] Xue F, Sanderson AC, Graves RJ. Multiobjective Evolutionary Decision Support for Design-Supplier-Manufacturing Planning. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 2009; 39(2): 309-20.

[4] Fattahi M, Mahootchi M, Mosadegh H, Fallahi F. A new approach for maintenance scheduling of generating units in electrical power systems based on their operational hours. Computers & Operations Research 2014; 50: 61-79.

[5] Chiang T-C, Hsu W-H. A knowledge-based evolutionary algorithm for the multiobjective vehicle routing problem with time windows. Computers & Operations Research 2014; 45: 25-37.

[6] Qing-dao-er-ji R, Wang Y. A new hybrid genetic algorithm for job shop scheduling problem. Computers & Operations Research 2012; 39(10): 2291-99.

[7] Fonseca CM, Fleming PJ. An overview of evolutionary algorithms in multi- objective optimization. Evolutionary Computation 1995; 3(1):1-16

[8] Tan Y-y, Jiao Y-c, Li H, Wang X-k. MOEA/D + uniform design: A new version of MOEA/D for optimization problems with many objectives. Computers & Operations Research 2013; 40(6): 1648-60.

[9] Schaffer JD. Multiple objective optimization with vector evaluated genetic algorithms. In: Proceedings of the 1st international Conference on Genetic Algorithms, 1985. p. 93-100.

[10] Fonseca CM, Fleming PJ. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In: Proceedings of ICGA, vol. 93; 1993. p. 416-23.

[11] Horn J, Nafpliotis N, Goldberg DE. A niched Pareto genetic algorithm for multiobjective optimization. In: Proceedings of the First IEEE Conference on Evolutionary Computation, vol. 1; 1994. p. 82-87.

[12] Srinivas N, Deb K. Muiltiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. Evolutionary Computation 1994; 2(3): 221-48.

[13] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Transactions on Evolutionary Computation 1999; 3(4): 257-71.

[14] Zitzler E, Laumanns M, Thiele L. SPEA2: improving the strength Pareto evolutionary algorithm. Technical report 103. Zurich, Switzerland: Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH); 2001.

[15] Corne DW, Knowles JD, Oates MJ. The Pareto envelope-based selection algorithm for multiobjective optimization. In: Proceedings of Parallel Problem Solving from Nature PPSN VI, 2000. p. 839-48.

[16] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 2002; 6(2): 182-97.

[17] Beausoleil RP. "MOSS" multiobjective scatter search applied to non-linear multiple criteria optimization. European Journal of Operational Research 2006; 169(2): 426-49.

[18] Hung M-H, Shu L-S, Ho S-J, Hwang S-F, Ho S-Y. A Novel Intelligent Multiobjective Simulated Annealing Algorithm for Designing Robust PID Controllers. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans; 38(2): 319-30.

[19] Zhan ZH, Li JJ, Cao JN, Zhang J, Chung HSH, Shi YH. Multiple populations for multiple objective a coevolutionary technique for solving multiobjective optimization problems. IEEE Transactions on Cybernetics 2013: 43(2): 445-63.

[20] Lopez-Ibanez M, Stutzle T. The Automatic Design of Multiobjective Ant Colony Optimization Algorithms, IEEE Transactions on Evolutionary Computation 2012; 16(6): 861-75.

[21] Li K, Fialho A, Kwong S, Zhang Q. Adaptive Operator Selection With Bandits for a Multiobjective Evolutionary Algorithm Based on Decomposition. IEEE Transactions on Evolutionary Computation 2014; 18(1): 114-130.

[22] Fukuda T, Mori K, Tsukiyama M. Immune networks using genetic algorithm for adaptive production scheduling. In: Proceedings of 15th IFAC world congress. vol. 3; 1993. p. 57-60.

[23] Nebro AJ, Luna F, Alba E, Dorronsoro B, Durillo JJ, Beham A. AbYSS: Adapting Scatter Search to

Multiobjective Optimization. IEEE Transactions on Evolutionary Computation 2008; 12(4): 439-57.

[24] Tang L, Wang X. A Hybrid Multiobjective Evolutionary Algorithm for Multiobjective Optimization Problems. IEEE Transactions on Evolutionary Computation 2013; 17(1): 20-45.

[25] Al Moubayed N, Petrovski A, McCall J. D^2MOPSO: Multi-Objective Particle Swarm Optimizer Based on Decomposition and Dominance. Evolutionary Computation in Combinatorial Optimization. Lecture Notes in Computer Science. vol. 7245; Springer Berlin Heidelberg; 2012. p. 75-86.

[26] Al Moubayed N, Petrovski A, McCall J. D^2MOPSO: MOPSO Based on Decomposition and Dominance with Archiving Using Crowding Distance in Objective and Solution Spaces. Evolutionary Computation 2014; 22(1): 47-77.

[27] Gong W, Fialho Á, Cai Z, Li H. Adaptive strategy selection in differential evolution for numerical optimization: An empirical study. Information Sciences 2011; 181(24): 5364-86.

[28] Qin AK, Huang VL, Suganthan PN. Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization. IEEE Transactions on Evolutionary Computation 2009; 13(2): 398-417.

[29] Hernandez-Diaz AG, Santana-Quintero LV, Coello CC, Caballero R, Molina J. A new proposal for multi-objective optimization using differential evolution and rough sets theory. In: Proceedings of the 8th annual conference on Genetic and evolutionary computation, 2006. p. 675-82.

[30] Santana-Quintero LV, Hernández-Díaz AG, Molina J, Coello Coello CA, Caballero R. DEMORS: A hybrid multi-objective optimization algorithm using differential evolution and rough set theory for constrained problems. Computers & Operations Research 2010; 37(3): 470-80.

[31] Zhang Q, Li H. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. IEEE Transactions on Evolutionary Computation 2007; 11(6): 712-31.

[32] Li H, Zhang Q. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. IEEE Transactions on Evolutionary Computation 2009; 13(2): 284-302.

[33] Venske SMS, Goncalves RA, Delgado MR. ADEMO/D: Adaptive Differential Evolution for Multiobjective Problems. In: Proceedings of 2012 Brazilian Symposium on Neural Networks (SBRN). 2012. p. 226-31.

[34] Venske SM, Gonçalves RA, Delgado MR. ADEMO/D: Multiobjective optimization by an adaptive differential evolution algorithm. Neurocomputing 2014; 127: 65-77.

[35] Hofmeyr SA, Forrest S. Architecture for an Artificial Immune System. Evolutionary Computation 2000; 8(4): 443-73.

[36] Harmer PK, Williams PD, Gunsch GH, Lamont GB. An artificial immune system architecture for computer security applications. IEEE Transactions on Evolutionary Computation 2002; 6(3): 252-80.

[37] Lin Q, Chen J. A novel micro-population immune multiobjective optimization algorithm. Computers & Operations Research 2013; 40(6): 1590-1601.

[38] Hashim F, Munasinghe KS, Jamalipour A. On the negative selection and the danger theory inspired security for heterogeneous networks. IEEE Wireless Communications 2012; 19(3): 73-84.

[39] Deb K, Thiele L, Laumanns M, Zitzler E. Scalable multi-objective optimization test problems. In: Proceedings of the Congress on Evolutionary Computation (CEC-2002). 2002. p. 825-30.

[40] Huband S, Hingston P, Barone L, While L. A review of multiobjective test problems and a scalable test problem toolkit. IEEE Transactions on Evolutionary Computation 2006; 10(5): 477-506.

[41] Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary computation 2000; 8(2): 173-95.

[42] Bosman PAN, Thierens D. The balance between proximity and diversity in multiobjective evolutionary algorithms. IEEE Transactions on Evolutionary Computation 2003; 7(2): 174-88.

[43] Hunt JE, Cooke DE. Learning using an artificial immune system. Journal of Network and Computer Applications 1996; 19(2): 189-212.

[44] de Castro LN, Timmis J. An artificial immune network for multimodal function optimization. In: Proceedings of the 2002 Congress on Evolutionary Computation, vol. 1, 2002. p. 699-704.

[45] de Castro LN, Von Zuben FJ. Learning and optimization using the clonal selection principle. IEEE Transactions on Evolutionary Computation 2002; 6(3): 239-51.

[46] Coello CAC, Cortés NC. Solving multiobjective optimization problems using an artificial immune system. Genetic Programming and Evolvable Machines 2005; 6(2): 163-90.

[47] Gong M, Jiao L, Du H, Bo L. Multiobjective immune algorithm with nondominated neighbor-based selection. Evolutionary Computation 2008; 16(2): 225-55.

[48] Shi J, Gong M, Ma W, Jiao L. A Multipopulation Coevolutionary Strategy for Multiobjective Immune Algorithm. The Scientific World Journal 2014: 539128.

[49] Hu Z-H. A multiobjective immune algorithm based on a multiple-affinity model. European Journal of Operational Research 2010; 202(1): 60-72.

[50] Sun F, Chen Y, Wu W. Multi-objective Optimization Immune Algorithm Using Clustering. Computing and Intelligent Systems, Communications in Computer and Information Science, vol. 234, 2011. p. 242-51.

[51] Gao J, Wang J. WBMOAIS: A novel artificial immune system for multiobjective optimization. Computers & Operations Research 2010; 37(1): 50-61.

[52] Shang R, Jiao L, Liu F, Ma W. A Novel Immune Clonal Algorithm for MO Problems. IEEE Transactions on Evolutionary Computation 2012; 16(1): 35-50.

[53] Zinflou A, Gagné C, Gravel M. GISMOO: A new hybrid genetic/immune strategy for multiple-objective optimization. Computers & Operations Research 2012; 39(9): 1951-68.

[54] Tan KC, Goh CK, Mamun AA, Ei EZ. An evolutionary artificial immune system for multi-objective optimization. European Journal of Operational Research 2008; 187(2): 371-92.

[55] Chen J, Lin Q, Ji Z. A hybrid immune multiobjective optimization algorithm. European Journal of Operational Research 2010; 204(2): 294-302.

[56] Zitzler E, Thiele L. Multiobjective optimization using evolutionay algorithms-A comparative case study. Parallel Problem Solving from Nature-PPSN V, Lecture Notes in Computer Science 1998; 1498: 292-301.

[57] Durillo JJ, Nebro AJ, Alba E. The jMetal framework for multi-objective optimization: Design and architecture. In: Proceedings of 2010 IEEE Congress on Evolutionary Computation. 2010. p. 1-8.

Highlights:

- Differential evolution is embedded into the multi-objective immune algorithm.
- A suitable parent selection strategy provides a correct evolutionary direction.
- A novel adaptive control approach enhances the algorithmic robustness.