A Novel Multi-objective Immune Algorithm with A Decomposition-based Clonal Selection

Lingjie Li[1], Qiuzhen Lin[1]*, Songbai Liu[1], Dunwei Gong[2], Carlos A. Coello Coello[3], Zhong Ming[1]*

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

[2]*School of Information and Control Engineering, China University of Mining and Technology*, Xuzhou 221116, Jiangsu, *PR. China,*

[3] *CINVESTAV-IPN, Department of Computer Science, Mexico, D.F., 07360, Mexico*

Abstract:

In recent years, a number of multi-objective immune algorithms (MOIAs) have been proposed as inspired by the information processing in biologic immune system. Since most MOIAs encourage to search around some boundary and less-crowded areas using the clonal selection principle, they have been validated to show the effectiveness on tackling various kinds of multi-objective optimization problems (MOPs). The crowding distance metric is often used in MOIAs as a diversity metric to reflect the status of population's diversity, which is employed to clone less-crowded individuals for evolution. However, this kind of cloning may encounter some difficulties when tackling some complicated MOPs (*e.g.*, the UF problems with variable linkages). To alleviate the above difficulties, a novel MOIA with a decomposition-based clonal selection strategy (MOIA-DCSS) is proposed in this paper. Each individual is associated to one subproblem using the decomposition approach and then the performance enhancement on each subproblem can be easily quantified. Then, a novel decomposition-based clonal selection strategy is designed to clone the solutions with the larger improvements for the subproblems, which encourages to search around these subproblems. Moreover, differential evolution is employed in MOIA-DCSS to strength the exploration ability and also to improve the population's diversity. To evaluate the performance of MOIA-DCSS, twenty-eight test problems are used with the complicated Pareto-optimal sets and fronts. The experimental results validate the superiority of MOIA-DCSS over four state-of-the-art multi-objective algorithms (i.e., NSLS, MOEA/D-M2M, MOEA/D-DRA and MOEA/DD) and three competitive MOIAs (i.e., NNIA, HEIA, and AIMA).

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

1. Introduction

In real-world engineering problems [1], [2], it is very often to encounter the optimization problems related to several (often conflicting) objectives, which are called multi-objective optimization problems (MOPs), as formulated by

$$\underset{x\in\Omega}{\mathrm{Min}}\, F(x)=\left(f_1(x), f_2(x),..., f_m(x)\right)^T, \tag{1}$$

where $x=(x_1,x_2,...,x_n)$ is a decision vector in Ω (the *n*-dimensional decision space), and the objective function $F:\Omega\rightarrow R^m$ gives the mapping from Ω to R^m (the *m*-dimensional objective space). The final

---

*Corresponding Author: Qiuzhen Lin (Email: qiuzhlin@szu.edu.cn) and Zhong Ming (Email: mingz@szu.edu.cn)

result in tackling MOPs usually gives a Pareto-optimal set (***PS***), which has the best trade-off for all the objectives. The objective function values of ***PS*** are often termed Pareto-optimal front (***PF***). When the preference information of different objectives is unavailable, there have two goals for solving MOP, i.e., a set of solutions with good convergence that can closely approach the true ***PF*** and with good diversity that can be evenly dispersed along the true ***PF***. By this way, these final solutions can support the decision maker to select the appropriate solutions for different practical cases.

In the last decades, there are a number of multi-objective evolutionary algorithms (MOEAs) proposed to tackle various kinds of MOPs. NSGA-II [3], SPEA2 [4], and MOEA/D [5] are widely acknowledged as the three well-known state-of-the-art MOEAs. NSGA-II [3] was designed with a fast nondominated sorting approach to ensure the convergence first and then using the crowding-distance metrics to guarantee the population's diversity. SPEA2 [4] was proposed with the aim to balance convergence and diversity by using a fine-grained fitness assignment strategy. MOEA/D [5] was presented to decompose the target MOP into a set of subproblems and then to optimize them simultaneously on a cooperative manner. These well-known MOEAs have inspired many research studies [6], [7], [8], [9], [10], such as a new definition of dominance relation [11] and an integrated weight assignment strategy [12] for NSGA-II, a shift-based density estimation strategy [13] and an efficient reference direction-based density estimator [14] for SPEA2, an indicator-based method [15] and an acute angle based approach [16] for MOEA/D. Some recent research studies have extended MOEAs to solve many-objective optimization problems [17], [18], [19], [20]. For more detailed review of MOEAs, please refer to [21], [22].

On the other hand, a number of multi-objective immune algorithms (MOIAs) have been proposed as inspired by the clonal selection principle in biologic immune system, showing the superiority over some state-of-the-art MOEAs [11], [12], [13], [14], [15], [16]. Only a small ratio of individuals showing good convergence and diversity capabilities are selected for clonal proliferation and then a number of clones are generated in MOIAs. Then, each clone is evolved by hyper-mutation, expecting to produce the superior offspring. This way, the individuals with high potentiality will have more clones to be evolved, aiming to speed up convergence or extend diversity. The first real-coded MOIA may retrospect to a nondominated neighbor-based immune algorithm (NNIA) [23] based on the clonal selection principle, which was experimentally validated to show some advantages over NSGA-II and SPEA2. After that, a larger number of MOIAs were also designed based on the clonal selection principle, such as HEIA [24], AIMA [25], theta-MCSA [26], and CMIGA [27]. Most of them have demonstrated the superiorities on solving the simple MOPs (like ZDT [28] and DTLZ [29]). However, the experiments in [24], [25] showed that most MOIAs were difficult to handle the UF test problems [30] with the complicated ***PF*** or ***PS***. This is mainly because most MOIAs implement the clonal selection operators only on nondominated individual according to their crowding distance values [23], which may cause the difficulties on these complicated MOPs [30]. This observation motivates us to study whether a novel clonal selection strategy can be implemented in MOIAs to alleviate the above problem. Therefore, in this paper, we propose a novel MOIA with a decomposition-based clonal selection strategy, called MOIA-DCSS. Instead of using the crowding distance metric in clonal selection, the proposed MOIA-DCSS exploits the decomposition approach to realize the clonal selection approach, which has some advantages in selecting the potential solutions for cloning and evolution. By this way, our algorithm is more able to maintain the balance of convergence and diversity, especially on tackling some complicated MOPs. Moreover, following the design of some recent MOIAs [24], [25], differential evolution is also used in MOIA-DCSS to improve the exploration ability and the population's diversity. To have a comprehensive evaluation on the performance of MOIA-DCSS, three different test suites are used, i.e., the walking fish group (WFG) [31], the UF [30], and the F [32] test suites. When compared to four state-of-the-art multi-objective algorithms (i.e., NSLS [33], MOEA/D-M2M [34], MOEA/D-DRA [30], and MOEA/DD [35]) and three competitive MOIAs (i.e., NNIA [23], HEIA [24], and AIMA [25]), the performance of MOIA-DCSS is superior when considering the convergence speed and the population's diversity. Moreover, the effectiveness of our clonal selection strategy is also experimentally studied to confirm its superiority.

The rest of this paper is organized as follows. Section 2 gives some background information, such as some related work of MOIAs and the clonal selection principle in MOIAs. Section 3 provides the details of the proposed clonal selection strategy and MOIA-DCSS. Section 4 lists the experimental results of MOIA-DCSS with other MOEAs and MOIAs. Section 5 presents the conclusions and the future work.

2. Background

2.1 The related work of MOIAs

The concept of antibody-antigen affinity in biologic immune system was firstly used as a fitness assignment mechanism for a standard genetic algorithm [36], which may be a first attempt to present an MOIA. Since then, a large number of MOIAs were designed in order to further enhance the performance. Based on the features inspired from the biologic immune system, most MOIAs can be categorized into three main kinds. The first class of MOIAs simulates the clonal selection principle [37] and clones the superior individuals with the highest affinity values, e.g., NNIA [24] and MAM-MOIA [38]. The second type of MOIAs maintains the population's diversity as inspired from the immune network theory, such as VAIS [39] and WBMOAIS [40]. The last category of MOIAs embeds other heuristic operators into MOIAs, like MOGAIS [41] and MOBAIS [42] which replaces the mutation and cloning operators with a probabilistic model, i.e., Gaussian network and Bayesian network, respectively.

In recent years, some competitive MOIAs have been proposed with more promising performance. For example, CMIGA [43] based on the model of biological immune system was presented to solve the MOPs with multimodel nonlinear constraints; IMADE [44] was proposed to combine a newly designed DE operator and simulated binary crossover (SBX); mcDMOA [45] was designed with an adaptive change reaction strategy to track the changing ***PFs***; IDSMOA [46] was introduced by using various immune operators in two co-evolutionary populations. To combine the advantages of different evolutionary strategies, a novel hybrid evolutionary framework was designed for MOIAs, which implements a hybrid evolutionary MOIA called HEIA [24]. The cloned individuals in HEIA are separated into several sub-populations and then independently evolved by different evolutionary strategies (e.g., SBX and DE). More recently, AIMA [25] was proposed by dividing the process of evolution into three main stages (the early, middle and last stages). Three different DE strategies showing distinct search capabilities are sequentially used on these stages, as controlled by an adaptive selection strategy. Besides that, MOIAs were also studied to solve some constrained MOPs in [47], [48], [49], and extended to solve some real-world applications, such as [50], [51] for the traffic environmental problems, [52], [53] for job-shop scheduling problem, and [54], [55] for the dynamic optimization problem.

2.2 Clonal selection in MOIAs

Most of MOIAs [24], [25], [26], [27], [41], [44], [45], [46], [56] were designed based on the clonal selection principle. In order to show the running of clonal selection in MOIAs, the population's evolution in one generation t of MOIAs is illustrated in Fig. 1. At first, the population P_t is evolved by the meta-heuristic operators (e.g., SBX and polynomial-based mutation [57]) to produce the offspring population D_t. Then, the populations P_t and D_t are combined to update external archive, as marked by E_t. At last, the clonal selection is further run to select some promising individuals (A_t) for cloning a new population (P_{t+1}) for the next generation, as shown in the procedures of selection and cloning from Fig. 1.

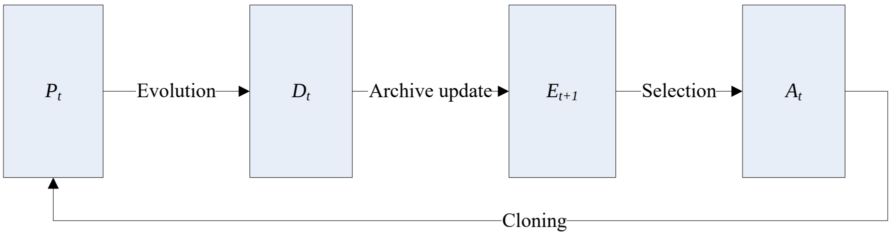


Fig. 1 The population evolution of MOIAs

Here, the details of clonal selection in [23], [24], [25], [26], [27] are introduced. First, clonal selection is run, which will pick up n_A solutions having the largest crowding-distance values from the external archive [4]. These n_A solutions build the population A_t to be cloned. Then, based on the crowding-distance value, each individual $a_i \ (i = 1,2,...,n_A)$ from A_t is cloned proportionally, as defined by

$$P_{t+1} = \bigcup_{i=1}^{n_A} \{ q_i \otimes a_i \}, a_i \in A_t , \tag{2}$$

where q_i is the number of clones for a_i and $\otimes$ indicates to copy the individual. In most MOIAs, q_i is computed by

$$q_i = \left\lceil N \times \frac{cd(a_i)}{\sum_{j=1}^{n_A} cd(a_j)} \right\rceil , \tag{3}$$

where $cd(a_i)$ indicates the crowding-distance value [4] of a_i. This way, more clones will be copied for the solutions with larger crowding-distance values, according to the values of q_i assigned in Eq. (3). In original definition [23], the crowding-distance values of boundary solutions are assigned positive infinity, which cannot be used in Eq. (3). Thus, it is suggested in [23] that their crowding-distance values should be reset to be the double of the maximum value of solutions in A_t except for the boundary solutions. Based on the above process of clonal selection, MOIAs produce more clones to search the less-crowded and boundary areas, thus they have shown the improved convergence speed and population's diversity, as experimentally validated in [23], [24], [25], [26], [27].

However, when solving some complicated MOPs (e.g., the UF test problems [30] and F test problems [32]), this kind of clonal selection operator based on the crowding distance metric is not so effective due to the complicated ***PSs*** and ***PFs***. Therefore, this paper presents a novel MOIA with a clonal selection strategy based on decomposition approach (MOIA-DCSS), which is expected to have a stronger exploration capability on tackling these complicated MOPs.

3. The proposed MOIA-DCSS algorithm

In this section, the details of the proposed MOIA-DCSS algorithm are introduced. The flow chart of MOIA-DCSS is provided in Fig. 2. It starts by initializing the population and setting some relevant parameters. Then, the individuals will undergo three important procedures, i.e., clonal selection, evolution, and population update, to approximate the true ***PF***. To clarify MOIA-DCSS, the implementation details of these procedures are respectively introduced below.

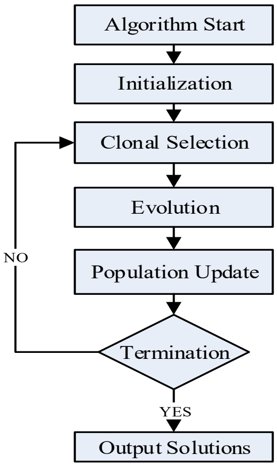


Fig. 2 The flow chart of the proposed MOIA-DCSS algorithm

3.1. Initialization

Algorithm 1 Initialization

```
initialize $t$=0;
initialize the weight vectors $\lambda=\{\lambda^1,\lambda^2,\ldots,\lambda^N\}$;
for each subproblem $i$ =1 to $N$ do
    randomly generate an individual $x_i$ and $n_i$=0;
    add $x_i$ to the population $P_t$ ;
    initialize the neighbors $B(i)=\{i_1,i_2,\ldots,i_T\}$ to $\lambda^i$ ;
end for
initialize the ideal point $z^*=\{z^1,z^2,\ldots,z^N\}$ ;
```

First, we initialize the generation time *t* in line 1 and uniformly generate a set of *N* weight vectors $\lambda = \{\lambda^1, \lambda^2, ..., \lambda^N\}$ in line 2 with the constraints $\sum_{j=1}^{m} \lambda_j^i = 1$ and $\lambda_j^i \geq 0$ for all *i*= {1, 2,..., *N*} (*N* is the size of external archive). Then, in lines 4-5, we randomly produce an initial population $P_t=\{x_1, x_2, ..., x_N\}$ and initialize the cloning number for each solution as zero. In line 6, we calculate the Euclidean distances of each λ^i and other weight vectors, and then find $\lambda^{i_1}, \lambda^{i_2}, ..., \lambda^{i_T}$ as the *T* closest weight vectors for λ^i which compose its neighbor set $B(i) = \{i_1, i_2, ..., i_T\}$. After that, since the true ideal point cannot be known beforehand, an approximated point is used instead to find the minimum value of each objective in line 8, i.e., $z_i^* = \min\{f_i(x) \mid x \in P_t\}$ for all *i*= {1, 2,..., *m*}.

3.2. Clonal selection

In our clonal selection strategy, the affinity value of an individual is assigned as the improvement on its associated aggregated value, which has larger probability to be improved after one generation. The parents $x_i \ (i = 1, 2, ..., n_A)$ in P_t are all cloned proportionally according to the improvement on the associated aggregated values. The mathematical model of proportional cloning has been defined in Eq. (2), while n_i in Eq. (5) is the cloning number for each subproblem at each generation, as defined by

$$n_i = \left\lceil N \times \frac{\Delta_i}{\sum_{j=0}^{N} \Delta_j} \right\rceil, \tag{5}$$

where *N* is the population size and Δ_i is the relative improvement on the aggregated values using a decomposition approach. Here, the Tchebycheff method is used in this paper to construct the aggregated functions, since it is mostly used in many MOEAs [7], [58], [59], [60], as defined by

$$\underset{x \in \Omega}{Min}\ g^{tch}(x \mid \lambda, z^*) = \underset{1 \leq i \leq m}{max}\{\mid f_i(x) - z_i^* \mid / \lambda_i\}, \tag{6}$$

where $\lambda = (\lambda_1, \lambda_2, ..., \lambda_m)$ is the used weight vectors, $z^* = \{z_1^*, ..., z_m^*\}$ is the approximate ideal point with $z_i^* = \min\{f_i(x) \mid x \in P_t\}$ for all *i*={1,2,...,*m*}, and the relative improvement value Δ_i is defined as

$$\Delta_i = \frac{g^{tch}(x_{t-1}^i \mid \lambda^i, z^*) - g^{tch}(x_t^i \mid \lambda^i, z^*)}{g^{tch}(x_{t-1}^i \mid \lambda^i, z^*)}, \tag{7}$$

where x_t^i and x_{t-1}^i respectively indicate the solutions associated to λ^i at *t* and $t-1$ generations, and $g^{tch}(\cdot)$ is the Tchebycheff approach in Eq. (6). Then, a selection probability p_i, which controls the clone number for each subproblem, is defined as

$$p_i = (1 - p_{min}) \times (1.0 + p_{min} \times \exp(20 \times (r_i / T - 0.5)))^{-1}, \tag{8}$$

where p_{min} is a minimum probability for each solution that can be selected for cloning, *T* is the neighbor size, and r_i is the local rank based on the relative improvement aggregated value Δ_i among *B*(*i*). The subproblem with the largest improvement aggregated value Δ_i is λ^{i_1} set as $r_i = 1$, while the subproblem with the smallest improvement aggregated value Δ_i is λ^{i_T} set as $r_i = T$. According to Eq. (8), these subproblems with large improvement aggregated value Δ_i will have more probabilities to be selected for cloning, which can be confirmed from Fig. 3 by plotting the dynamic change of p_i according to different r_i. As shown in Fig. 3, we can easily know that the subproblem with the larger Δ_i value will have higher local rank as well as larger probability value p_i, which means that this solution has the higher probability to be selected for cloning than that with the smaller improvement aggregated values Δ_i.

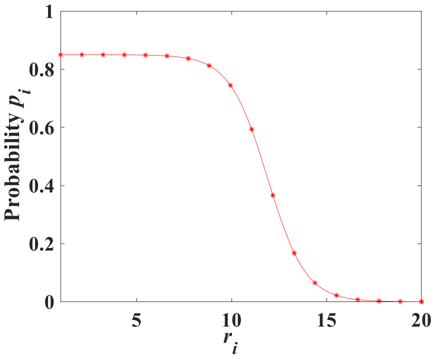


Fig. 3 The dynamic change of selection probability p_i according to different r_i

To clarify this procedure, the pseudo-code of decomposition-based clonal selection strategy (DCSS) is given in **Algorithm 2**, where lines 1-14 introduce the parent's selection procedure and lines 15-17 realize the parent's cloning process.

Algorithm 2 C_t=DCSS (P_t)

```
1.  // Parents Selection
2.  for $i$=1 to $N$ do
3.      calculate the relative improvement value $\Delta_i$ by using Eq. (7);
4.  end for
5.  get the local rank $r_i$ of each subproblem according to the sort of $\Delta_i$ among its neighbor ;
6.  for $i$=1 to $N$ do
7.      calculate selection probability $p_i$ for each subproblem by using Eq. (8);
8.      if ($p_i < rand(0,1)$) then
9.          $\Delta_i$=0;
10.     end if
11. end for
12. for $i$=1 to $N$ do
13.     calculate $n_i$ for each subproblem based on $\Delta_i$ by using Eq. (5);
14. end for
15. // Cloning
16. use Eq. (2) on $P_t$ to generate the cloned population $C_t$;
17. return $C_t$;
```

First, the relative improvement value Δ_i is calculated by using Eq. (7) for each subproblem, as shown in line 3. The local rank r_i will be obtained according to the sorting of Δ_i among its neighbors, as shown in line 5. The value of control parameter p_i will be calculated by using Eq. (8), which is relevant to the relative improvement value Δ_i and will be updated after every generation based on the local rank r_i, as shown in line 7. Then, according to lines 8-10, the selection probability of each solution will be compared to a random real number uniformly sampled from [0, 1] (i.e., *rand* (0, 1)). Once it is smaller than *rand* (0, 1), its relative improvement value Δ_i will be reset to zero. As the solutions with high Δ_i values can be easily enhanced and further optimized, they will be cloned to have more offspring in the next generation.

On the contrary, the solutions with lower Δ_i values will be reset as zero. This way, these solutions with smaller probability value p_i will not be selected in the next generation, as these solutions are hard to be further optimized or may have already converged to the ***PF***. After that, as shown in line 13, the clone number (n_i) of solutions for each subproblem in next generation will be calculated by using Eq. (5). In line 16, the cloning process is run to duplicate the solutions for some subproblems using Eq. (2) based on the value of n_i. At last, the new population C_t will be generated after cloning and returned in line 17.

3.3. Evolution

In MOIAs, evolution is expected to produce the superior offspring. Two evolutionary operators such as SBX [57] and DE [33], [61], [62] are often used in MOEAs and MOIAs. As inspired by some recently proposed MOIAs [24], [25], [26], [27], the DE (rand/1/bin~) operator and polynomial-based mutation [32] are used to produce offspring in this paper. The pseudo-code of **Evolution** is given in **Algorithm 3** and its main process is introduced below.

Algorithm 3 Evolution(C_t, $B(i)$)

```
1. if $rand(0,1) \le \delta$ then
2.     selected two solutions $x^{r1}$, $x^{r2}$ from $B(i)$ randomly;
3. else
4.     selected two solutions $x^{r1}$, $x^{r2}$ from $C_t$ randomly;
5. end if
6. generate an offspring $v^i$ using $x^i$, $x^{r1}$, $x^{r2}$ by DE
7. execute polynomial mutation on $v^i$ to obtain a new solution $y^i$;
8. return $y^i$;
```

As shown in **Algorithm 3**, for each ith subproblem selected for evolution in lines 1-5, one random number is generated to compare with the parameter δ. If it is smaller, the neighbor set $B(i)$ is selected as the candidate set of parents. Otherwise, the entire population C_t will be chosen. Then, in lines 6-8, the selected solutions as the parents are used to run the DE operator and polynomial mutation. Using two selected parents (x^{r1} and x^{r2}) and the original parent (x^i) for the ith subproblem, the DE operator is executed with two control parameters CR and F, which generates a new solution $v^i = (v_1^i, \ldots, v_n^i)$ by

$$v_j^i = \begin{cases} x_j^i + F \times (x_j^{r1} - x_j^{r2}) & \text{if } rand(0,1) < CR \text{ or } j = j_{rand} \\ x_j^i & \text{otherwise} \end{cases}, \tag{9}$$

where $rand(0,1)$ returns a random real number uniformly sampled in [0, 1], j_{rand} is a random integer from $[1, n]$ to ensure that at least one dimension is different from x^i and $j \in \{1, \ldots, n\}$. After that, an offspring solution y^i will be produced by using polynomial-based mutation on v^i, as follows.

$$y_j^i = \begin{cases} v_j^i + \Delta_j \times (u_j - l_j) & \text{if } rand(0,1) < p_m \\ v_j^i & \text{otherwise} \end{cases}, \tag{10}$$

with

$$\Delta_j = \begin{cases} \left[2r + (1-2r)(\dfrac{u_j - u_i}{u_j - l_j}) \right]^{1/(\eta+1)} - 1 & \text{if } rand(0,1) < 0.5 \\ 1 - \left[2r + (1-2r)(\dfrac{u_j - u_i}{u_j - l_j})^{(\eta+1)} \right]^{1/(\eta+1)} & \text{otherwise} \end{cases}, \tag{11}$$

where p_m and η are respectively the mutation probability and the distribution index, while l_j and u_j are respectively the lower and upper bounds of the *j*th decision variable.

3.4. Population update

As shown in **Algorithm 4**, Δ_{max} is the largest relative improvement rate in P_t. The main process of population update is clarified in this section. After a new solution is generated, a replacement strategy for the subproblem should be run to update the population P_t by remaining the superior one. In MOIA-DCSS, one new solution only replaces one old solution for its matched subproblem, which shows the maximum improvement value on the aggregated function. In line 2 of **Algorithm 4**, the relative improvement rate Δ_i is calculated for each subproblem in P_t by using y^i. Then, in lines 3-9, one subproblem will be found to have the largest improvement rate Δ_i and its associated solution is recorded by x^l, which will be renewed by a new solution y^i. This strategy can assign the offspring to one suitable subproblem with the largest improvement without worsening the diversity [61].

```
Algorithm 4   Population Update($P_t$, $y^i$, $\lambda$)
1.  initialize $\Delta_{max} = 0$, $l = 0$;
2.  calculate $\Delta_i$ for each subproblem in $P_t$ using $y^i$;
3.  for $i$=1 to $N$ do
4.      if ( $\Delta_{max} < \Delta_i$ ) then
5.          $\Delta_{max} = \Delta_i$ ;
6.          $l=i$;
7.      end if;
8.  end for
9.  replace the solution $x^l$ with $y^i$ in $P_t$;
10. return $P_t$
```

3.5. The complete algorithm of MOIA-DCSS

```
Algorithm 5   The Complete Algorithm MOIA-DCSS
Initialization; //( Algorithm 1)
while $t<t_{max}$ do
    //Clonal Selection
    $C_t$ = DCSS($P_t$) ; //(Algorithm 2)
    for $i$=1 to $N$ do
        // Evolution
        $y^i$= Evolution($C_t$,$B(i)$); //(Algorithm 3)
        //Population Update
        evaluate the objective values of $y^i$ ;
        update the ideal point $Z^*$;
        $P_t$= Population Update ($P_t$, $y^i$, $\lambda$) //(Algorithm 4)
    end for
    $t=t+1$;
end while;
Output $P_t$;
```

The above subsections have given the details of four main components in MOIA-DCSS, including initialization, clonal selection, evolutionary strategy, and population update. Here, as shown in **Algorithm 5**, the remaining implementation details are introduced in the pseudo-code of MOIA-DCSS, where t and t_{max} are respectively the current generation times and the maximum generations times. n_i is the clone number for each subproblem in C_t, which is calculated by using Eq. (5), as shown in line 13 of **Algorithm 2**. The main process of our proposed algorithm is introduced below.

In line 1, the initialization phase is executed as introduced in **Algorithm 1**. After the initialization, MOIA-DCSS enters the loop of the evolutionary process. In line 4, the selected parents from P_t is cloned by DCSS in **Algorithm 2**, and then a new cloned population C_t is generated. After that, as shown in line 7, two parent solutions x^{r1} and x^{r2} that are randomly selected from the population C_t or $B(i)$ are respectively permutated using the DE operator and polynomial mutation as described in **Algorithm 3**, and then the mutant solution y^i is generated and evaluated in line 9. In line 10, the ideal point z^* is updated using the value of each objective in y^i. At last, the solution y^i and the population P_t are used as the input to run **Algorithm 4**, which will assign y^i to one subproblem with the largest relative improvements in P_t by updating its old solution. The above evolutionary loop will be repeated until the preset maximum number of generations t_{max} is reached. At the end of MOIA-DCSS, the solutions in P_t are reported as the final approximate ***PF***.

4. Experimental studies

4.1. The used test problems

In our experiments, 28 unconstrained test MOPs including ten UF problems (UF1–UF10) [30], nine F problems (F1-F9) [32], and nine WFG problems (WFG1-WFG9) [31], are used for performance comparison. These used test problems have various complex characteristics and very complicated ***PS*** shapes. UF8–UF10 and F6 include three optimization objectives, while the remaining test problems have two optimization objectives. It is noted that the numbers of decision variables in F1–F5, F9 and all UF test problems are set to 30, while those in F6–F8 and all the WFG problems are set to 10.

4.2. Performance indicators

This paper adopts two well-known performance measures (inverted generational distance (IGD) [32] and hypervolume (HV) [63]) to compare the performance of different algorithms, which can reflect the convergence and the solution's diversity for their final solution sets simultaneously.

1) IGD: Let S be a set of solutions that are uniformly distributed along the true ***PF*** and let S' be a set of solutions that are found by an algorithm. The IGD value of S to S' ($IGD(S,S')$) is defined by

$$IGD(S,S')=\frac{\sum_{i=1}^{|S|}d(S_i,S')}{|S|}, \tag{12}$$

where $d(S_i,S')$ indicates the minimum Euclidean distance between S_i and the individuals in S' regarding the objective space, and $|S|$ returns the size of S. The true ***PF*** of the target problem is assumed available in advance when computing IGD. A lower $IGD(S,S')$ value is regarded to be better as it indicates that S is closer to the true ***PF*** with an even distribution along the true ***PF***.

2) HV: this metric calculates the volume of the objective space that is dominated by the approximation set S and bounded by a reference point $z^r=(z_1^r,z_2^r,...,z_m^r)^T$. When computing HV, the reference point is set as dominated by all the points in the true ***PF***. Then, HV can be computed by

$$HV(S) = VoI\left(\bigcup_{x \in S} \left[f_1(x), z_1^r \right] \times \ldots \times \left[f_m(x), z_m^r \right] \right), \tag{13}$$

where *VoI*(.) denotes the Lebesgue measures. The points worse than the reference point on any objective will be removed for computing HV. In our experiments, $z^r = (2.0, 2.0)^T$ and $z^r = (2.0, 2.0, 2.0)^T$ are respectively set for some bi-objective test problems (UF1-UF7, F1-F5 and F7-F9) and three-objective test problems (UF8-UF10 and F6). Moreover, $z^r = (3.0, 5.0)^T$ is set for the WFG test problems, as they have different scaled values for the objectives. A larger HV value indicates a better performance to approximate the entire true ***PF*** with even distribution.

4.3. Experimental settings

In this paper, in order to assess the performance of MOIA-DCSS, we compare MOIA-DCSS with four state-of-the-art multi-objective algorithms (i.e., NSLS [33], MOEA/D-M2M [34], MOEA/D-DRA [30] and MOEA/DD [35]), and three competitive MOIAs (i.e., NNIA [23], HEIA [24], and AIMA [25]). All these compared algorithms are validated to show the promising performance in solving various kinds of MOPs, thus the comparisons of MOIA-DCSS with these algorithms are very comprehensive and convincing.

In jMetal [64], the implementations of some state-of-the-art MOEAs (NSLS, MOEA/D-M2M, MORA/D-DRA and MOEA/DD) are provided. For the compared MOIAs (HEIA, AIMA, NNIA, and MOIA-DCSS), they are realized by us based on the jMetal framework. In order to have a fair comparison, the parameters of all the compared algorithms are set as suggested in the corresponding references, which are summarized in Table 1.

Table 1
The parameter settings of all the compared algorithms

Algorithms	Parameter settings
NSLS	$N = 100, P_m = 1/n, \eta_m = 20, T = 20, \delta = 0.9, n_r = 2, \mu = 0.5, \sigma = 0.1$
MOEA/DD	$N = 100, P_c = 1.0, \eta_c = 30, P_m = 1/n, \eta_m = 20, T = 20, \delta = 0.9, \theta = 5.0$
MOEA/D-DRA	$N = 100, P_m = 1/n, \eta_m = 20, T = 20, \delta = 0.9, n_r = 2$
HEIA	$N = 100, N_A = 20, P_c = 1.0, P_m = 1/n, \eta_c = 20, \eta_m = 20, CR = 1.0, F = 0.5, \delta = 0.9$
AIMA	$N = 100, N_A = 20, P_c = 1.0, P_m = 1/n, \eta_m = 20$
MOEA/D-M2M	$N = 100$， $P_c = 0.9, P_m = 1/n, \eta_c = 20, \eta_m = 20$
NNIA	$N = 100, N_A = 20, P_c = 1.0, P_m = 1/n, \eta_c = 20, \eta_m = 20$
MOIA-DCSS	$N = 100, N_A = 20, P_c = 1.0, P_m = 1/n, \eta_m = 20, T = 20, \delta = 0.9, n_r = 2, P_{min} = 0.15$

In Table 1, *N* is the population size; p_c and p_m are respectively the probabilities to run crossover and mutation; η_c and η_m respectively indicate the distribution indexes of SBX and Polynomial-based mutation. For MOEA/D-DRA, *T* is the neighborhood size for the used weight vectors, δ controls the probability to select parent solutions from *T* neighbors, and n_r is the maximum number of solution replacement in population update. Regarding MOEA/DD, θ indicates the penalty parameter. In NSLS, μ and σ are two control parameter used in the local search process. In NNIA, HEIA, AIMA and MOIA-DCSS, N_A is the size of selected solutions for clonal proliferation.

It is noted that the value *N* in Table 1 are only set for the WFG test problems with the maximum number of function evaluations as 25000. When tackling other test MOPs, the population size and the maximum number of function evaluations are adjusted based on their difficulty and complexity. Here, for the more difficult bi-objective test MOPs (F1-F5, F7-F9, and UF1-UF7), their population sizes are set to 300, while the population sizes are set to 600 for F6 and UF8-UF10 with three optimization objectives. Moreover, the maximum numbers of function evaluations are set to 150 000 when solving F1-F5 and F7-F9, and set to 300 000 when tackling F6 and UF1-UF10.

For each test problem, 30 independent runs are executed by all the compared algorithms. To easily observe the best performance, the best mean value for each problem is identified by boldface and gray background in all the comparison tables. Moreover, in order to show the statistical significance on the experimental results, Wilcoxon's rank sum test is run with a 5% significance level. It is noted that "–", "+" and "~" respectively indicate that the results obtained by the compared algorithm are worse than, better than or similar to those of MOIA-DCSS.

4.4. Comparisons of MOIA-DCSS and four state-of-the-art multi-objective algorithms

Table 2

Performance comparisons of MOIA-DCSS and four state-of-the-art multi-objective algorithms using IGD

Instances	NSLS	MOEA/D-M2M	MOEA/D-DRA	MOEA/DD	MOIA-DCSS
UF1	2.13e-01(1.63e-2)-	9.79e-03(2.65e-3)-	6.72e-03(1.23e-03)-	5.69e-02(2.03e-02)-	1.71e-03(1.05e-04)
UF2	4.91e-01(3.06e-2)-	6.72e-03(4.08e-4)-	6.70e-03(2.50e-03)~	2.83e-02(2.18e-02)-	5.65e-03(1.64e-03)
UF3	3.81e-01(5.08e-3)-	1.43e-02(5.27e-3)-	3.66e-02(7.46e-02)-	2.20e-01(5.87e-02)-	3.49e-03(3.98e-03)
UF4	5.50e-02(3.37e-3)-	4.37e-02(7.99e-4)+	6.08e-02(5.32e-03)-	3.80e-02(9.77e-04)+	5.36e-02(6.84e-03)
UF5	1.85e+00(1.23e-1)-	2.04e-01(3.48e-2)+	3.84e-01(1.89e-01)-	2.92e-01(1.11e-01)~	2.49e-01(1.76e-02)
UF6	1.01e+00(7.28e-2)-	9.78e-02(1.36e-2)-	1.81e-01(6.66e-02)-	2.06e-01(8.38e-02)-	6.49e-02(1.34e-02)
UF7	2.40e-01(1.87e-2)-	8.32e-03(1.89e-3)-	4.22e-03(6.46e-04)-	1.54e-01(2.26e-01)-	2.06e-03(1.50e-04)
UF8	6.83e-01(4.89e-2)-	1.13e-01(4.65e-3)-	5.95e-02(3.42e-02)~	8.75e-02(2.50e-02)-	5.28e-02(1.25e-02)
UF9	8.21e-01(6.51e-2)-	1.66e-01(5.52e-2)-	8.50e-02(1.06e-01)-	6.33e-02(1.15e-02)-	2.85e-02(4.13e-03)
UF10	5.40e+00(4.32e-1)-	9.94e-01(1.15e-1)-	4.24e-01(9.96e-02)+	1.98e-01(5.99e-02)+	6.10e-01(1.43e-01)
F1	9.36e-02(4.51e-3)-	1.99e-03(7.30e-5)-	1.64e-03(2.16e-04)-	1.62e-03(1.28e-04)-	1.29e-03(4.85e-05)
F2	2.13e-01(1.91e-2)-	1.51e-02(1.80e-3)-	7.10e-02(1.02e-01)-	5.98e-02(4.64e-03)-	2.60e-03(2.38e-04)
F3	4.81e-01(2.11e-2)-	4.44e-03(2.95e-4)-	1.54e-02(4.32e-03)-	3.03e-02(8.00e-03)-	2.62e-03(2.18e-04)
F4	4.37e-01(2.17e-2)-	7.25e-03(1.34e-3)-	1.82e-02(2.49e-02)-	6.31e-02(9.97e-03)-	3.10e-03(1.37e-03)
F5	5.04e-01(3.22e-2)-	6.68e-03(5.34e-4)+	1.69e-02(7.60e-03)-	2.89e-02(8.78e-03)-	1.01e-02(4.79e-03)
F6	3.87e-01(5.05e-2)-	2.60e-01(3.60e-2)-	3.01e-02(8.63e-04)-	5.46e-02(7.23e-03)-	2.19e-02(1.24e-04)
F7	3.92e-01(5.25e-3)-	2.49e-03(9.49e-5)-	2.62e-03(5.90e-04)-	2.81e-01(2.36e-01)-	2.27e-03(1.74e-03)
F8	3.94e-01(1.14e-3)-	4.56e-03(2.95e-3)+	1.27e-01(1.07e-01)~	1.65e-01(5.95e-02)-	9.79e-02(4.01e-02)
F9	1.91e-01(8.68e-3)-	1.70e-02(9.33e-3)-	2.22e-02(2.17e-03)-	7.40e-02(4.41e-03)-	2.97e-03(1.08e-03)
WFG1	1.83e+00(4.13e-2)-	9.53e-01(6.98e-2)+	1.06e+00(1.60e-01)+	1.69e+00(5.20e-02)-	1.07e+00 (1.32e-01)
WFG2	5.15e-01(3.32e-2)-	2.62e-02(3.41e-3)+	7.18e-02(4.85e-02)-	9.03e-02(6.48e-03)-	4.12e-02(4.58e-03)
WFG3	5.25e-01(2.67e-2)-	3.09e-02(3.91e-3)-	1.50e-02(1.08e-03)~	2.27e-02(6.08e-03)-	1.49e-02(6.47e-04)
WFG4	3.59e-01(2.80e-2)-	3.39e-02(4.80e-3)+	4.55e-02(9.68e-03)+	2.85e-02(1.43e-02)+	6.07e-02(5.94e-03)
WFG5	1.65e-01(2.65e-2)-	6.80e-02(7.25e-4)-	6.73e-02(2.56e-04)-	6.89e-02(7.93e-04)-	6.71e-02(1.63e-04)
WFG6	4.16e-01(6.37e-2)-	1.38e-01(9.34e-2)+	6.61e-02(3.54e-02)-	9.01e-02(3.27e-02)-	1.70e-02(1.45e-02)
WFG7	4.53e-01(2.87e-2)-	2.49e-02(1.60e-3)-	1.69e-02(2.64e-04)+	2.79e-02(1.03e-02)-	1.71e-02(2.84e-04)
WFG8	5.34e-01(3.39e-2)-	1.19e-01(5.68e-3)+	2.25e-01(2.68e-02)~	2.78e-01(1.60e-02)-	2.11e-01(4.82e-02)
WFG9	2.50e-01(7.14e-3)-	2.80e-02(2.32e-3)+	1.01e-01(2.52e-01)~	2.75e-02(9.89e-03)~	2.32e-02(2.64e-01)
Total	28-/0~/0+	18-/0~/10+	18-/6~/4+	23-/2~/3+	

"+\-\~" denote that the performance of corresponding algorithm is significantly better than, worse than, and similar to MOIA-DCSS respectively by Wilcoxon's sum test with a significance level 0.05.

In this section, the performance of MOIA-DCSS is compared to four state-of-the-art multi-objective algorithms, i.e., NSLS [33], MOEA/D-M2M [34], MOEA/D-DRA [30] and MOEA/DD [35]. Tables 2 provides the IGD results of all the algorithms that are obtained from 30 independent runs when solving UF, F and WFG test problems.

From Table 2, it is observed that MOIA-DCSS shows some advantages over other competitors with respect to IGD, as MOIA-DCSS can perform best on 17 out of all the 28 test problems, while other compared algorithms are only best on 4 cases. Furthermore, the comparison summaries of MOIA-DCSS with other competitors are provided in the last row of Table 2, which indicates the total number of test problems that MOIA-DCSS performs better than (−), similarly to (~), and worse than (+) the corresponding algorithm. By observing the results of Wilcoxon's rank sum test, MOIA-DCSS outperforms NSLS, MOEA/D-M2M, MOEA/D-DRA and MOEA/DD respectively on 28, 18, 18, and 23 out of 28 test problems, while it underperforms NSLS, MOEA/D-M2M, MOEA/D-DRA and MOEA/DD respectively on 0, 10, 4, and 3 test problems. For the F1-F9 test problems with complex ***PSs***, the advantages of MOIA-DCSS are obvious as MOIA-DCSS performs significantly better than other four competitors. Regarding the UF test problems with complicated ***PSs***, MOIA-DCSS performs best on all the test instances except for UF4 and UF10, while MOEA/DD obtains the best result on UF4 and UF10. A similar conclusion can be drawn when considering the results on F test problems, as our proposed problem MOIA-DCSS can outperform the compared algorithm on 7 out of all the 9 F test problems. However, with respect to the WFG test problems with simple ***PSs***, it is found that MOIA-DCSS can only obtain 2 best results and MOEA/D-M2M performs better than other compared algorithms as it can outperform on 5 out of all the 9 WFG test problems.

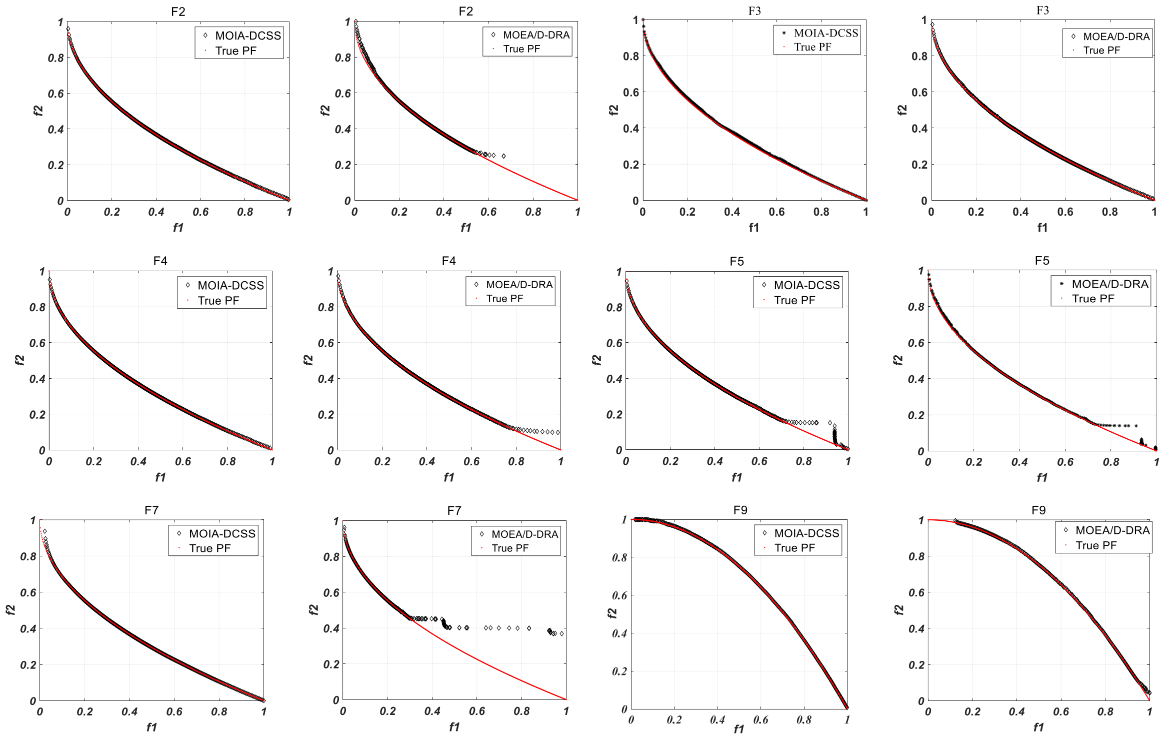


Fig. 4 The plots of the approximation sets obtained by MOIA-DCSS and MOEA/D-DRA on F test problems

Therefore, when considering all the UF, F and WFG test problems, the advantages of the proposed MOIA-DCSS are validated, as MOIA-DCSS presents a superior performance over NSLS, MOEA/D-M2M, MOEA/D-DRA and MOEA/DD on most cases regarding IGD.

To visually show the performance of different compared algorithms, the plots of the approximation sets obtained by two representative algorithms (MOIA-DCSS and MOEA/D-DRA) are shown in Fig. 4 when tackling six test MOPs (i.e., F2-F5, F7, and F9). Although MOIA-DCSS and MOEA/D-DRA use the same decomposition approach, we can observe from Fig. 4 that MOIA-DCSS obviously outperforms MOEA/D-DRA, as MOEA/D-DRA shows some difficulties in searching the boundary solutions. The final solution sets obtained by MOIA-DCSS are closer to the true ***PF*** with more even distribution when compared to that of MOEA/D-DRA.

Table 3 further gives the HV results of all the compared algorithms on all the UF, F and WFG test problems. As observed from Table 3, the conclusions are similar to that observed from the IGD results. First, MOIA-DCSS is able to obtain the best results on 18 out of 28 test problems. When considering the UF and F test problems, the proposed algorithm can obtain the best results on all the problems except UF4, UF10, F7 and F8, while MOEA/DD only performs best on UF4 and UF10. MOEA/D-M2M can get the best results on F7 and F8. Moreover, MOIA-DCSS can get the best results on 4 out of all the 9 WFG problems when considering the HV values (i.e., WFG2, WFG3, WFG6 and WFG7). Thus, MOIA-DCSS presents the obvious advantages when compared to other competitors. Second, MOIA-DCSS is superior on most cases when respectively compared to NSLS, MOEA/D-M2M, MOEA/D-DRA and MOEA/DD, which are summarized in the last row of Table 3. Therefore, it is further confirmed by HV that the proposed MOIA-DCSS shows the superiority in solving all the test problems adopted.

TABLE 3
Performance comparisons of MOIA-DCSS and four state-of-the-art multi-objective algorithms using HV

Instances	NSLS	MOEA/D-M2M	MOEA/D-DRA	MOEA/DD	MOIA-DCSS
UF1	3.17e+00(3.98e-2)-	3.64e+00(6.89e-3)-	3.64e+00(1.29e-02)-	3.41e+00(1.25e-01)-	3.66e+00(1.40e-03)
UF2	2.46e+00(5.19e-2)-	3.64e+00(1.11e-2)-	3.65e+00(1.31e-02)~	3.56e+00(1.15e-01)-	3.65e+00(4.63e-03)
UF3	3.03e+00(7.49e-3)-	3.64e+00(7.85e-3)-	3.53e+00(3.49e-01)~	2.78e+00(2.54e-01)-	3.65e+00(7.95e-03)
UF4	3.19e+00(3.48e-3)-	3.21e+00(2.94e-3)+	3.14e+00(4.59e-02)-	3.21e+00(8.32e-03)+	3.17e+00(1.73e-02)
UF5	1.01e-02(1.97e-2)-	2.93e+00(1.52e-1)+	2.45e+00(5.03e-01)-	2.46e+00(3.89e-01)-	2.84e+00(7.49e-02)
UF6	9.88e-01(1.31e-1)-	3.09e+00(1.53e-1)-	2.79e+00(1.54e-01)-	2.78e+00(2.33e-01)-	3.20e+00(2.01e-02)
UF7	2.93e+00(3.80e-2)-	3.48e+00(7.18e-3)-	3.47e+00(8.67e-03)-	3.00e+00(7.44e-01)-	3.49e+00(2.95e-03)
UF8	3.42e+00(2.41e-1)-	7.02e+00(5.16e-2)-	7.34e+00(5.04e-02)~	7.08e+00(3.53e-01)-	7.35e+00(3.68e-02)
UF9	3.57e+00(2.52e-1)-	7.15e+00(1.91e-1)-	7.44e+00(4.85e-01)-	7.44e+00(1.43e-01)-	7.63e+00(1.19e-01)
UF10	0.00e+00(0.00e+0)-	1.95e+00(4.57e-1)-	3.74e+00(6.56e-01)+	6.06e+00(5.77e-01)+	3.28e+00(6.86e-01)
F1	3.52e+00(1.21e-2)-	3.66e+00(1.25e-4)~	3.66e+00(6.92e-04)-	3.66e+00(4.04e-04)-	3.66e+00(9.55e-05)
F2	3.17e+00(3.71e-2)-	3.62e+00(1.92e-2)-	3.37e+00(3.43e-01)-	3.34e+00(1.73e-02)-	3.66e+00(2.09e-03)
F3	2.47e+00(7.10e-2)-	3.65e+00(4.42e-3)-	3.61e+00(1.36e-02)-	3.52e+00(1.85e-02)-	3.66e+00(2.45e-03)
F4	2.57e+00(5.66e-2)-	3.64e+00(8.65e-3)-	3.58e+00(1.34e-01)-	3.40e+00(1.70e-02)-	3.66e+00(2.58e-03)
F5	2.44e+00(5.73e-2)-	3.64e+00(9.65e-3)-	3.60e+00(5.70e-02)-	3.53e+00(2.64e-02)-	3.65e+00(7.96e-03)
F6	5.91e+00(1.46e-1)-	5.97e+00(2.59e-1)-	7.42e+00(3.25e-03)-	7.32e+00(3.46e-02)-	7.44e+00(6.10e-04)
F7	3.00e+00(6.37e-3)-	3.66e+00(1.57e-4)+	3.65e+00(8.41e-03)+	2.87e+00(4.35e-01)-	3.57e+00(9.70e-02)
F8	3.00e+00(1.37e-3)-	3.65e+00(4.79e-3)+	3.37e+00(1.48e-01)~	3.06e+00(1.77e-01)-	3.42e+00(9.17e-02)
F9	2.86e+00(2.95e-2)-	3.30e+00(1.45e-2)-	3.22e+00(1.60e-02)-	2.79e+00(2.89e-02)-	3.32e+00(4.30e-03)
WFG1	1.39e+00(2.06e-1)-	6.58e+00(3.06e-1)+	6.10e+00(7.87e-01)+	7.04e+00(7.86e-01)+	5.55e+00(5.61e-02)
WFG2	8.29e+00(2.47e-1)-	1.13e+01(3.48e-2)-	1.10e+01(8.27e-01)-	1.04e+01(3.45e-02)-	1.14e+01(1.62e-02)
WFG3	7.82e+00(1.50e-1)-	1.08e+01(3.22e-2)-	1.09e+01(1.53e-02)~	1.08e+01(9.54e-02)-	1.09e+01(6.98e-03)
WFG4	6.69e+00(1.20e-1)-	8.51e+00(3.55e-2)+	8.32e+00(8.45e-02)+	8.53e+00(9.94e-02)+	8.20e+00(4.11e-02)
WFG5	7.65e+00(1.19e-1)-	8.24e+00(6.04e-3)+	8.10e+00(8.45e-03)~	8.09e+00(2.20e-02)-	8.11e+00(1.71e-02)
WFG6	6.65e+00(1.42e-1)-	7.80e+00(6.19e-1)-	8.27e+00(3.06e-01)-	8.00e+00(2.58e-01)-	8.61e+00(3.29e-02)
WFG7	5.97e+00(1.46e-1)-	8.59e+00(1.07e-2)-	8.65e+00(8.72e-03)-	8.36e+00(3.19e-01)-	8.65e+00(5.54e-03)
WFG8	5.48e+00(1.45e-1)-	7.88e+00(3.07e-2)+	6.95e+00(2.34e-01)~	6.64e+00(7.17e-02)-	7.07e+00(2.40e-01)
WFG9	6.93e+00(9.02e-2)-	8.54e+00(1.73e-2)+	7.81e+00(1.55e+00)~	8.15e+00(1.96e-01)~	7.83e+00(1.56e+00)
Total	28-/0~/0+	18-/1~/9+	16-/8~/4+	23-/1~/4+	

+\-\~denote that the performance of corresponding algorithm is significantly better than, worse than, and similar to MOIA-DCSS respectively by Wilcoxon's sum test with a significance level 0.05.

To visually show the performance on three-objective test problems, the approximation sets obtained by all the compared algorithms are plotted in Fig. 5 when solving F6, where one run with the 15th best IGD value from 30 runs is selected for illustration. It is observed from Fig. 5 that NSLS and MOEA/D-M2M fail to find the uniformly distributed solutions to approach the true ***PF*** of F6. Due to the use of similar decomposition approach in MOIA-DCSS, MOEA/D-DRA and MOEA/DD, they perform much better to find an approximation set for F6, while MOIA-DCSS performs even better than MOEA/D-DRA and MOEA/DD as its solutions are smoother to cover the entire true ***PF***.

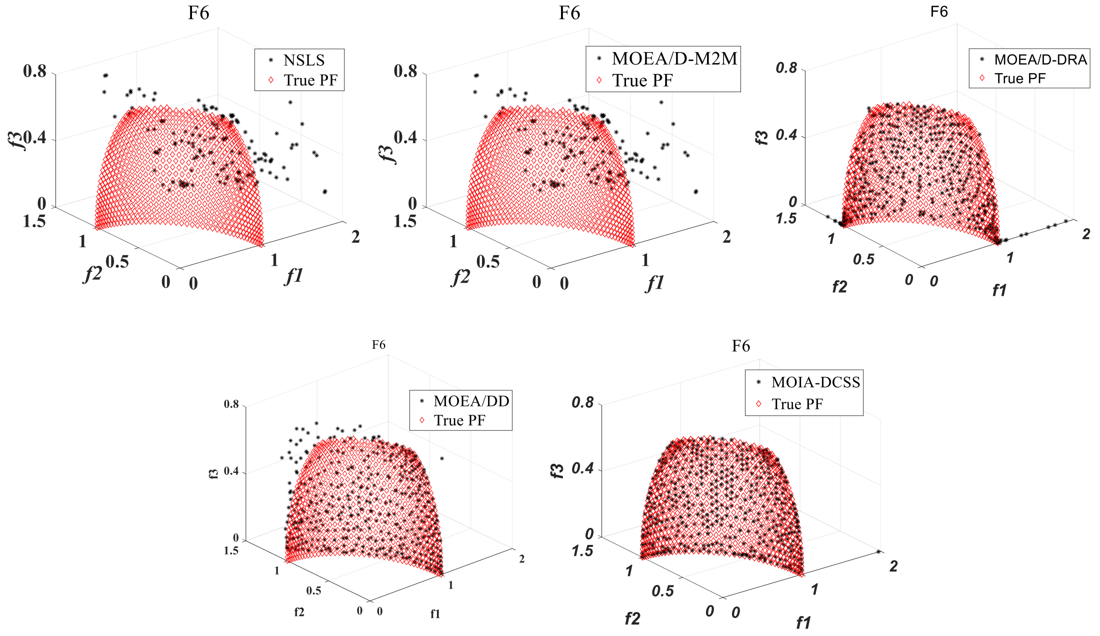


Fig. 5 Comparison of approximation sets obtained by all the compared algorithms on F6

The above experimental results show that MOIA-DCSS is more effective to solve the complicated test MOPs adopted in this paper when compared to NSLS, MOEA/D-M2M, MOEA/D-DRA and MOEA/DD. The outstanding performance of MOIA-DCSS is mainly due to the utilization of DCSS into MOIAs, which enhances the search ability in handling the complicated MOPs.

4.5. Comparisons of MOIA-DCSS with three competitive MOIAs

In this section, the performance of MOIA-DCSS is further compared to three MOIAs, i.e., NNIA [23], HEIA [24], and AIMA [25]. Tables 4 and 5 respectively give the IGD and HV results of all the compared algorithms on UF, F and WFG test problems after running 30 independent times.

When considering the IGD results in Table 4, MOIA-DCSS also shows some advantages over other competitors. Among all the 28 test problems, MOIA-DCSS is the best on 21 test problems, while HEIA gets the best performance on 5 WFG problems and AIMA obtains the best results on UF4 and WFG2. It is worth noting that the proposed MOIA-DCSS performs significantly better than other MOIAs particularly on the UF and F test problems, while HEIA with two search models seems superior on most cases of the

WFG test problems. As revealed by the Wilcoxon's rank sum test, MOIA-DCSS performs better than NNIA, HEIA, and AIMA, respectively on 24, 19, and 18 test problems, while it is outperformed by NNIA, HEIA, and AIMA, respectively on 2, 6, and 6 test problems. Therefore, it is reasonable to conclude that MOIA-DCSS presents the obvious advantages over NNIA, HEIA, and AIMA regarding IGD when solving most of the test problems adopted in this paper. Since all these MOIAs are designed based on the clonal selection approach, this superiority of MOIA-DCSS over NNIA, HEIA, and AIMA is mainly brought by the use of the proposed DCSS.

TABLE 4
Performance comparisons of MOIA-DCSS and three competitive MOIAs using IGD

Instance	NNIA	HEIA	AIMA	MOIA-DCSS
UF1	1.70e-01 (6.68e-02) -	4.76e-02 (1.37e-02) -	2.56e-02 (5.41e-03)-	1.71e-03(1.05e-04)
UF2	4.74e-02 (2.24e-02) -	3.35e-02 (4.90e-03) -	1.68e-02 (2.27e-03)-	5.65e-03(1.64e-03)
UF3	2.41e-01 (7.91e-02) -	1.56e-01 (3.05e-02) -	9.72e-02 (3.33e-02)-	3.49e-03(3.98e-03)
UF4	6.60e-02 (4.72e-03) -	5.17e-02 (6.78e-03) ~	3.86e-02 (1.64e-03)+	5.36e-02(6.84e-03)
UF5	5.31e-01 (3.12e-01) -	4.97e-01 (2.92e-01) -	2.10e-01 (1.60e-01)~	2.49e-01(1.76e-02)
UF6	4.43e-01 (3.34e-01) -	2.83e-01 (1.85e-01) -	2.18e-01 (2.34e-01)-	6.49e-02(1.34e-02)
UF7	4.09e-01 (1.93e-01) -	2.46e-02 (1.80e-02) -	1.22e-02 (9.74e-04)-	2.06e-03(1.50e-04)
UF8	2.73e-01 (9.99e-02) -	2.62e-01 (6.43e-02) -	2.19e-01 (2.88e-02)-	5.28e-02(1.25e-02)
UF9	6.44e-01 (2.63e-01) -	5.57e-01 (1.41e-01) -	4.55e-01 (1.02e-01)-	2.85e-02(4.13e-03)
UF10	1.35e+00 (6.83e-01) -	1.23e+00 (8.44e-01) -	1.44e+00 (7.19e-01)-	6.10e-01(1.43e-01)
F1	9.75e-03 (4.12e-04) -	5.25e-03 (2.06e-04) -	1.92e-03 (5.55e-05)-	1.29e-03(4.85e-05)
F2	1.13e-01 (6.30e-02) -	5.02e-02 (2.89e-02) -	2.58e-02 (6.24e-03)-	2.60e-03(2.38e-04)
F3	6.54e-02 (4.39e-02) -	3.63e-02 (2.24e-03) -	1.87e-02 (2.79e-03)-	2.62e-03(2.18e-04)
F4	5.28e-02 (8.33e-03) -	3.14e-02 (3.46e-03) -	1.34e-02 (1.66e-03)-	3.10e-03(1.37e-03)
F5	4.51e-02 (1.98e-02) -	3.22e-02 (7.15e-03) -	1.59e-02 (1.81e-03)-	1.01e-02(4.79e-03)
F6	3.08e-01 (7.84e-02) -	2.52e-01 (9.94e-02) -	1.01e-01 (1.08e-02)-	2.19e-02(1.24e-04)
F7	2.56e-01 (2.04e-01) -	5.38e-02 (5.17e-02) -	8.01e-02 (3.25e-02)-	2.27e-03(1.74e-03)
F8	2.93e-01 (1.35e-01) -	1.57e-01 (4.04e-02) -	1.75e-01 (7.58e-02)-	9.79e-02(4.01e-02)
F9	1.82e-01 (4.20e-02) -	5.45e-02 (2.02e-02) -	3.28e-02 (8.64e-03)-	2.97e-03(1.08e-03)
WFG1	7.14e-01 (5.86e-01) +	1.24e-02 (4.56e-04) +	2.70e-01 (1.40e-01)+	1.07e+00 (1.32e-01)
WFG2	6.77e-02 (2.23e-03) -	1.03e-02 (8.73e-04) +	1.02e-02 (5.76e-04)+	4.12e-02(4.58e-03)
WFG3	1.79e-02 (2.36e-03) -	1.30e-02 (4.82e-04) +	1.33e-02 (5.18e-04)+	1.49e-02(6.47e-04)
WFG4	1.49e-02 (1.03e-03) +	1.34e-02 (9.09e-04) +	1.74e-02 (3.20e-03)+	6.07e-02(5.94e-03)
WFG5	6.84e-02 (8.33e-04) -	6.72e-02 (3.20e-04) -	6.74e-02 (5.42e-04)~	6.71e-02(1.63e-04)
WFG6	7.97e-02 (4.08e-02) -	1.96e-02 (1.68e-02) ~	2.07e-02 (6.08e-03)~	1.70e-02(1.45e-02)
WFG7	1.66e-02 (1.61e-03) ~	1.34e-02 (6.51e-04) +	1.37e-02 (7.66e-04)+	1.71e-02(2.84e-04)
WFG8	2.51e-01 (1.57e-02) -	2.22e-01 (2.15e-02) ~	2.51e-01 (6.88e-03)-	2.11e-01(4.82e-02)
WFG9	2.29e-02 (4.58e-03) ~	2.07e-02 (2.97e-03) +	2.50e-02 (2.63e-01)~	2.32e-02(2.64e-01)
Total	24-/2~/2+	19-/3~/6+	18-/4~/6+	

+\-\~denote that the performance of corresponding algorithm is significantly better than, worse than, and similar to MOIA-DCSS respectively by Wilcoxon's sum test with a significance level 0.05.

Table 5 further shows the HV results of all the compared MOIAs on solving all the test problems. Similar conclusions can be observed from Table 5 that MOIA-DCSS performs significantly best on most of the UF and F test problems, while HEIA is the best on most of the WFG test problems. Regarding all the 28 test problems, MOIA-DCSS performs best on 20 cases, while HEIA gets the best performance on 7 WFG problems and AIMA obtains the best results on UF4. As shown by the Wilcoxon's rank sum test, MOIA-DCSS performs better than NNIA, HEIA, and AIMA, respectively on 23, 18, and 17 test problems, while it is outperformed by NNIA, HEIA, and AIMA, respectively on 3, 8, and 8 test problems. Therefore, it is further confirmed by using HV that MOIA-DCSS shows a superior performance over other compared MOIAs and the proposed DCSS is effective.

In order to have a visual observation about the performance of MOIA-DCSS, Figs. 6-8 provide the plots of the final approximation sets obtained by MOIA-DCSS, AIMA, HEIA and NNIA, respectively on tackling UF1, UF3, and UF7, where the true PFs are also plotted for performance comparison. As observed from Figs. 6-8, AIMA, HEIA and NNIA only find a set of solutions to reach some segments of the true PFs, while MOIA-DCSS performs much better to smoothly cover the entire true PFs of UF1, UF3 and UF7.

These experimental results have validated the effectiveness of the proposed DCSS and the advantages of MOIA-DCSS over NNIA, AIMA, and HEIA. Although they are all designed based on the clonal selection principle, MOIA-DCSS runs the cloning based on the decomposition approach, while NNIA, AIMA, and HEIA are all cloned based on the crowding distance metric. By observing the promising performance of MOIA-DCSS, the advantages of DCSS over other traditional clonal selection strategies based on the crowding distance metric can be confirmed, which is more reasonable to assign the evolutionary resources by cloning.

TABLE 5
Performance comparisons of MOIA-DCSS and three competitive MOIAs using HV

Instances	NNIA	HEIA	AIMA	MOIA-DCSS
UF1	3.2185 (1.95e-01) -	3.5752 (7.21e-02) -	3.6200 (6.95e-03)-	3.6614(8.11e-04)
UF2	3.5136 (1.18e-01) -	3.5982 (5.04e-02) -	3.6398 (3.34e-02)-	3.6585(2.53e-03)
UF3	2.7005 (1.87e-01) -	3.3600 (2.29e-01) -	3.4903 (5.76e-02)-	3.6599(7.69e-03)
UF4	3.1726 (1.24e-02) ~	3.1997 (1.48e-02) +	3.2328 (5.26e-03)+	3.1814(2.71e-02)
UF5	1.7880 (8.22e-01) -	1.8927 (6.01e-01) -	2.7350 (7.59e-01)~	2.8959(1.31e-01)
UF6	2.2120 (5.46e-01) -	2.5767 (5.35e-01) -	2.8463 (8.67e-01)-	3.1978(3.87e-02)
UF7	2.3985 (3.18e-01) -	3.4359 (1.42e-01) -	3.4776 (2.33e-03)-	3.4945(1.48e-03)
UF8	6.1222 (4.15e-01) -	6.1777 (1.92e-01) -	6.2790 (4.72e-01)-	7.3789(3.36e-02)
UF9	4.5119 (1.71e+00) -	4.9603 (7.76e-01) -	5.8734 (4.67e-01)-	7.7380(9.99e-03)
UF10	0.6164 (1.48e+00) -	0.8136 (1.47e+00) -	0.0261 (8.40e-01)-	4.4533(2.58e-01)
F1	3.6503 (6.11e-04) -	3.6577 (4.17e-04) -	3.6634 (8.02e-05)-	3.6648(2.65e-04)
F2	3.3054 (1.32e-01) -	3.4993 (1.55e-01) -	3.6133 (6.40e-02)-	3.6580(2.30e-03)
F3	3.4918 (1.23e-01) -	3.5890 (6.29e-02) -	3.6253 (3.07e-02)-	3.6607(1.84e-03)
F4	3.4458 (4.44e-02) -	3.6175 (4.55e-03) -	3.6465 (2.25e-03)-	3.6615(7.55e-04)
F5	3.5355 (1.11e-01) -	3.5940 (5.78e-02) -	3.6403 (7.86e-03)~	3.6482(1.05e-02)
F6	6.3631 (4.12e-01) -	6.8702 (5.01e-01) -	7.2774 (2.05e-02)-	7.4454(6.37e-04)
F7	2.8280 (4.46e-01) -	3.5381 (1.11e-01) -	3.5257 (8.05e-02)-	3.6398(3.18e-02)
F8	2.6757 (3.32e-01) -	3.0611 (2.35e-01) -	3.2218 (1.75e-01)-	3.4096(1.26e-01)
F9	2.9668 (2.52e-01) -	3.2423 (8.74e-02) -	3.2736 (1.19e-02)-	3.3258(1.87e-03)
WFG1	8.8316 (1.80e+00) +	12.0720 (5.83e-04) +	10.4288 (8.15e-01)+	5.8014(7.41e-01)
WFG2	10.5952 (2.17e-02) -	11.4582 (4.13e-03) +	11.4551 (3.35e-03)+	11.4049(9.89e-03)
WFG3	10.8974 (2.44e-02) -	10.9379 (3.91e-03) +	10.9357 (4.73e-03)+	10.9258(8.67e-03)
WFG4	8.6400 (1.65e-02) +	8.6587 (1.00e-02) +	8.5996 (3.10e-02)+	8.2636(3.75e-02)
WFG5	8.1290 (3.06e-02) +	8.1307 (3.02e-02) +	8.1240 (7.70e-03)+	8.1103(8.44e-03)
WFG6	8.1315 (2.35e-01) -	8.6005 (1.41e-01) ~	8.5927 (5.58e-02)~	8.6119(5.78e-02)
WFG7	8.6533 (5.25e-03) -	8.6699 (2.97e-03) +	8.6687 (3.94e-03)+	8.6580(3.41e-03)
WFG8	6.8283 (7.87e-02) -	7.0138 (1.61e-01) ~	6.8618 (4.78e-02)-	7.0164(2.34e-01)
WFG9	8.2870 (2.23e-01) ~	8.3336 (3.21e-02) +	8.3092 (1.56e+00)+	8.3085(2.40e-02)
Total	23-/2~/3+	18-/2~/8+	17-/3~/8+	

+\-\~denote that the performance of corresponding algorithm is significantly better than, worse than, and similar to MOIA-DCSS respectively by Wilcoxon's sum test with a significance level 0.05.

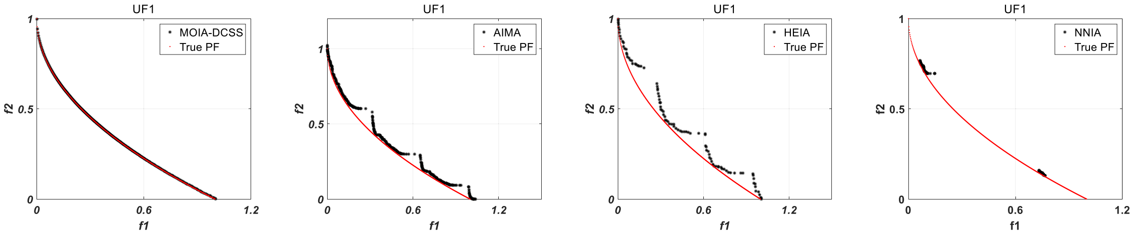


Fig. 6 Comparison of final approximation sets on UF1

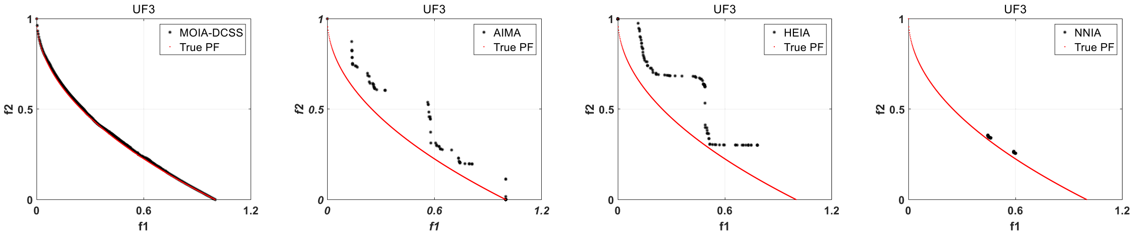


Fig. 7 Comparison of final approximation sets on UF3

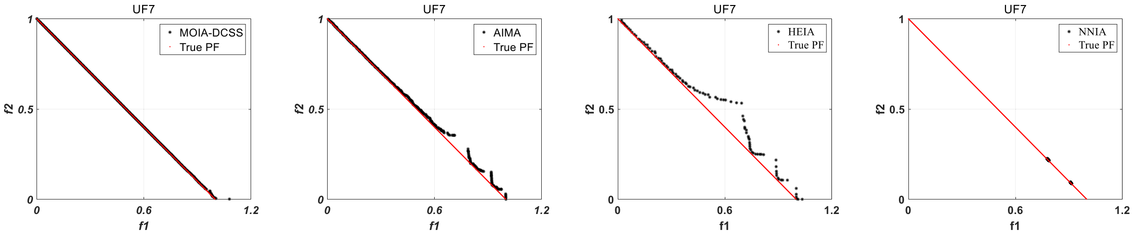


Fig. 8 Comparison of final approximation sets on UF7

4.6. Further discussions and analysis on MOIA-DCSS

4.6.1. A further analysis of all the compered algorithms using the Friedman rank

In this subsection, the Friedman's test based on the KEEL platform [65] was applied to quantify how well each algorithm mentioned above performs overall. Here, Fig. 9 is plotted based on their corresponding average performance ranks in all the 28 test problems (i.e., UF, F and WFG test problems).

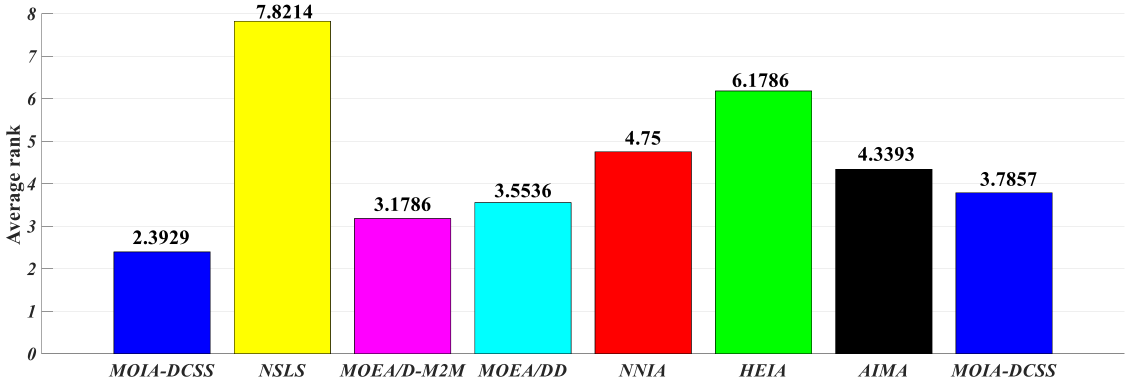


Fig.9. Average rank of Friedman's test for all the compared algorithms.

It is easy to observe from Fig. 9 that the average performance rank of our proposed algorithm MOIA-DCSS is 2.3929, which is much smaller than that of other competitors, while the average performance ranks of NSLS, MOEA/D-M2M, MOEA/D-DRA, MOEA/DD, NNIA, HEIA and AIMA are 7.8214, 3.1786, 3.5536, 4.75, 6.1786, 4.3393 and 3.7857, respectively. Hence, the superiority of our proposed MOIA-DCSS is further confirmed according to the Friedman ranks.

4.6.2. A further discussion of DCSS

This subsection further demonstrates the advantages of applying the proposed DCSS into traditional MOIA. In this section, the performance of MOIA-DCSS is further compared to its variant without using DCSS (MOIA-NO-DCSS). In MOIA-NO-DCSS, each solution is equally evolved without using cloning, while other parameters are set the same as listed in Table 1. Table 6 gives the IGD and HV results of MOIA-DCSS and MOIA-NO-DCSS on UF, F and WFG test problems after running 30 independent times.

TABLE 6
Performance comparisons of MOIA-DCSS and MOIA-NO-DCSS using IGD and HV

	IGD		HV	
Instances	MOIA-DCSS	MOIA-NO-DCSS	MOIA-DCSS	MOIA-NO-DCSS
UF1	1.71e-03(1.05e-04)	2.62e-03 (1.18e-03)-	3.6614(8.11e-04)	3.6559 (7.99e-03)-
UF2	5.65e-03(1.64e-03)	6.12e-03 (2.76e-03)-	3.6585(2.53e-03)	3.6527 (8.94e-03)~
UF3	3.49e-03(3.98e-03)	1.50e-02 (7.76e-02)-	3.6599(7.69e-03)	3.6400 (3.68e-01)-
UF4	5.36e-02(6.84e-03)	6.11e-02 (6.58e-03)-	3.1814(2.71e-02)	3.1528 (2.93e-02)-
UF5	2.49e-01(1.76e-02)	4.59e-01 (3.60e-01)-	2.8959(1.31e-01)	2.2082 (3.67e-01)-
UF6	6.49e-02(1.34e-02)	4.22e-01 (3.80e-01)-	3.1978(3.87e-02)	2.1977 (7.16e-01)-
UF7	2.06e-03(1.50e-04)	1.08e-02 (2.60e-02)-	3.4945(1.48e-03)	3.3605 (2.40e-01)-
UF8	5.28e-02(1.25e-02)	5.06e-02 (1.71e-02)~	7.3789(3.36e-02)	7.3595 (3.00e-02)~
UF9	2.85e-02(4.13e-03)	1.41e-01 (5.13e-02)-	7.7380(9.99e-03)	7.1952 (1.83e-01)-
UF10	6.10e-01(1.43e-01)	4.08e-01 (6.29e-02)+	4.4533(2.58e-01)	3.5208 (7.21e-01)~
F1	1.29e-03(4.85e-05)	1.40e-03 (1.31e-04)-	3.6648(2.65e-04)	3.6646 (5.31e-04)-
F2	2.60e-03(2.38e-04)	3.81e-03 (2.69e-03)-	3.6580(2.30e-03)	3.6449 (2.99e-02)-
F3	2.62e-03(2.18e-04)	7.71e-03 (4.77e-02)-	3.6607(1.84e-03)	3.5935 (2.51e-01)-
F4	3.10e-03(1.37e-03)	2.80e-03 (8.26e-04)~	3.6615(7.55e-04)	3.6562 (3.74e-03)~
F5	1.01e-02(4.79e-03)	1.24e-02 (6.77e-03)~	3.6482(1.05e-02)	3.6448 (2.13e-02)~
F6	2.19e-02(1.24e-04)	2.93e-02 (1.05e-03)-	7.4454(6.37e-04)	7.4257 (2.75e-03)-
F7	2.27e-03(1.74e-03)	2.41e-03 (4.89e-04)+	3.6398(3.18e-02)	3.6501 (7.67e-03)+
F8	9.79e-02(4.01e-02)	1.99e-01 (4.88e-02)-	3.4096(1.26e-01)	3.2779 (1.21e-01)-
F9	2.97e-03(1.08e-03)	3.89e-03 (6.20e-04)-	3.3258(1.87e-03)	3.3182 (5.64e-03)-
WFG1	1.07e+00 (1.32e-01)	1.03e+00 (1.43e-01)+	5.8014(7.41e-01)	6.3008 (8.73e-01)+
WFG2	4.12e-02(4.58e-03)	4.31e-02 (3.41e-02)~	11.4049(9.89e-03)	11.3895 (4.43e-01)~
WFG3	1.49e-02(6.47e-04)	1.52e-02 (9.32e-04)~	10.9258(8.67e-03)	10.9085 (2.03e-02)-
WFG4	6.07e-02(5.94e-03)	4.18e-02 (1.31e-02)+	8.2636(3.75e-02)	8.3337 (8.34e-02)+
WFG5	6.71e-02(1.63e-04)	6.76e-02 (3.06e-04)-	8.1103(8.44e-03)	8.0983 (1.45e-02)-
WFG6	1.70e-02(1.45e-02)	1.93e-02 (2.71e-02)-	8.6119(5.78e-02)	8.5786 (2.64e-01)-
WFG7	1.71e-02(2.84e-04)	1.69e-02 (2.03e-04)+	8.6580(3.41e-03)	8.6557 (2.85e-03)~
WFG8	2.11e-01(4.82e-02)	2.27e-01 (2.91e-02)~	7.0164(2.34e-01)	6.9008 (2.36e-01)~
WFG9	2.32e-02(2.64e-01)	1.21e-01 (2.64e-01)-	8.3085(2.40e-02)	7.6056 (1.54e+00)-
Total		17-/6~/5+		17-/8~/3+

+\-\~denote that the performance of corresponding algorithm is significantly better than, worse than, and similar to MOIA-DCSS respectively by Wilcoxon's sum test with a significance level 0.05.

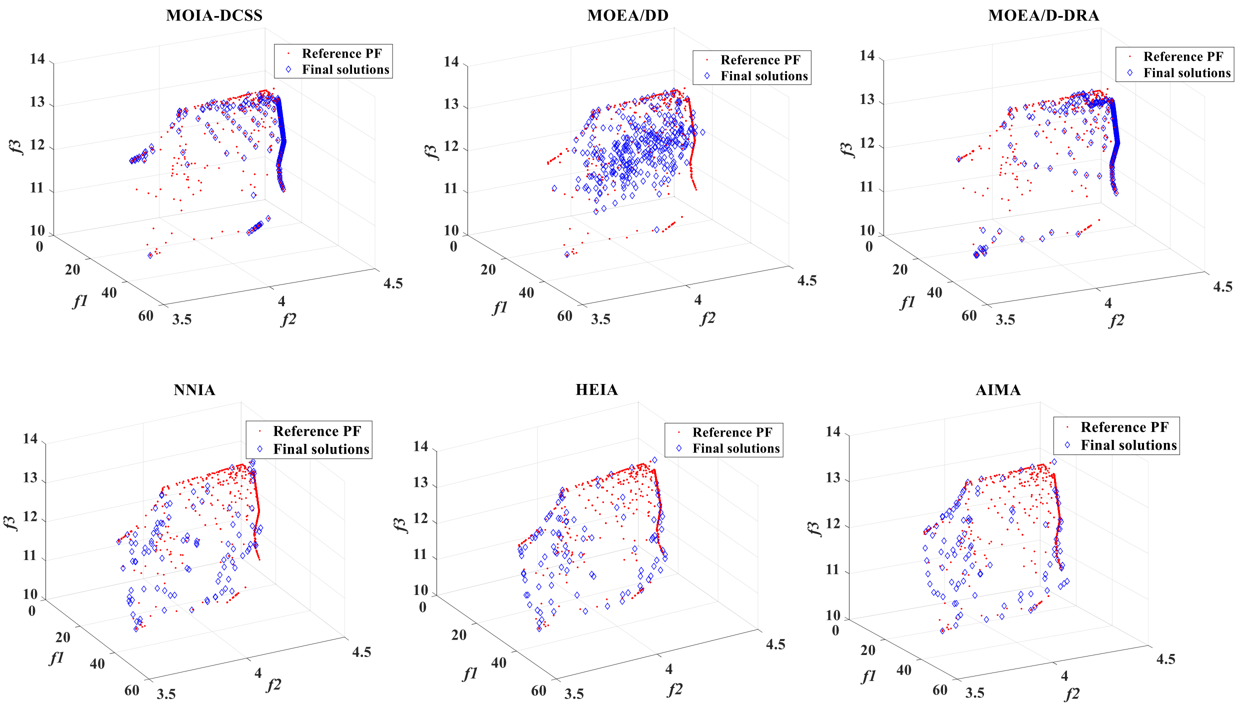
When considering the IGD results in Table 6, MOIA-DCSS shows obvious advantages over MOIA-NO-DCSS, as MOIA-NO-DCSS doesn't use the proposed DCSS. Among all the 28 test problems, MOIA-DCSS is able to perform best on 23 test problems, while the compared algorithm can only get the best performance on 5 test problems. MOIA-DCSS performs much better on most of the UF and F test problems, as the proposed MOIA-DCSS algorithm obtains the better results on all the complex UF and F test problems except UF10 and F7. Similar conclusions are observed from Table 6 when considering the HV results, as MOIA-DCSS performs better on 25 out of 28 test problems, while MOIA-NO-DCSS only obtains the better results on 3 test problems (F7, WFG1 and WFG4). Particularly, MOIA-DCSS shows the advantages on tackling the more complex test problems, as it performs best on all the UF and F test problems except F7.

These experimental results have validated the effectiveness of the proposed DCSS and the advantages of our proposed algorithm MOIA-DCSS.

4.6.3. A further study on solving real-life problem

In this subsection, the performance of the proposed algorithm (MOIA-DCSS) on tackling a real-life engineering problem is performed. The car side-impact problem [66] with three conflict objectives and ten constraints is employed in our experiment, which aims to simultaneously minimizing the weight of a car and the public force experienced by using a passenger and the average velocity of V-Pillar responsibility for withstanding the impact load. More details about the car side-impact real-world problem can refer to [66].

The population size and maximum number of evaluation was set as 300 and 150 000 for the car side-impact problem, respectively. Please note that the Pareto-optimal front is unknown in advance, hence the reference ***PF*** is constructed by all the compared algorithms. First, all the compared algorithms are run on this problem independently with 10 times and 1000 generations. After that, all the non-dominated solutions obtained from each algorithm are combined to construct the reference ***PF***. To visually show the final solution set obtained by all the compared algorithm on the car side-impact problem. Fig. 10 shows the final solution sets of the compared algorithm and the obtained reference ***PF***. It is easy to draw a conclusion that the solutions obtained by our proposed MOIA-DCSS can cover the reference ***PF*** more evenly and closely, especially for the boundary solutions, while other compared algorithms have some difficulties to find a complete solution set that is close to the reference ***PF*** or to obtain the boundary solutions.


MOIA-DCSS

MOEA/DD

MOEA/D-DRA

NNIA

HEIA

AIMA

Fig.10. The final solutions obtained by all the compared algorithms on the Car Side-impact problem

5. Conclusions and future work

In this paper, we present a novel MOIA with a decomposition-based clonal selection strategy (MOIA-DCSS). Based on the relative improvement on aggregated function values, the number of clones for each subproblem can be obtained. By this way, the potential area in objective space can be identified and will be assigned with more clones using the proposed DCSS to search these areas. Moreover, we also use DE in MOIA-DCSS to further enhance the exploration ability and the population's diversity. When compared to four state-of-the-art multi-objective algorithms (NSLS, MOEA/D-M2M, MOEA/D-DRA, and MOEA/DD) and three MOIAs (NNIA, HEIA and AIMA), the experimental results validate that MOIA-DCSS shows the superiority on solving most cases of different MOPs, such as the WFG, UF and F test problems. Moreover, the effectiveness of DCSS is also experimentally confirmed.

Although MOIA-DCSS can obtain the promising performance on most cases of the test problems adopted, there are still several research directions to be studied: 1) further enhance the performance of MOIA-DCSS by combining with other heuristic methods; 2) further extend MOIA-DCSS to tackle more complicated MOPs.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grants 61876110, 61836005, 61672358, the Joint Funds of the National Natural Science Foundation of China under Key Program Grant U1713212, the Natural Science Foundation of Guangdong Province under Grant 2017A030313338, and the Fundamental Research Project in the Science and Technology Plan of Shenzhen under Grants JCYJ20170817102218122 and JCYJ20170302154032530.

References

[1] Z. H. Zhan, X. F. Liu, Y. J. Gong, J. Zhang, H. S. H. Chung, and Y. Li, Cloud computing resource scheduling and a survey of its evolutionary approaches, ACM Computing Surveys., 47 (2015) pp. 1-33.

[2] X. Yu, W. N. Chen, T. L. Zhang, H. Q. Yuan, S. Kwong, J. Zhang, Set-based discrete particle swarm optimization based on decomposition for permutation-based multiobjective combinatorial optimization problems, IEEE Trans. Cybern., 48(2018) pp. 2139-2153.

[3] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multi-objective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput., 6 (2002) pp.182-197.

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

[5] Q.F. Zhang, H. Li, MOEA/D: A Multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput., 11 (2007) pp.712-731.

[6] W. N. Chen, Y. H. Jia, F. Zhao, X. N. Luo, X. D. Jia, J. Zhang, A cooperative co-evolutionary approach to large-scale multisource water distribution network optimization, IEEE Trans. Evol. Comput., in press, 2018.

[7] X.L. Ma, Q.F. Zhang, G.D. Tian, On Tchebycheff decomposition approaches for multiobjective evolutionary optimization, IEEE Trans. Evol. Comput., 22 (2017) pp.226-244.

[8] S.Y. Jiang, S.X. Yang, Y. Wang, Scalarizing functions in decomposition-based multiobjective evolutionary algorithms, IEEE Trans. Evol. Comput., 22 (2017) pp.296-313.

[9] K. Li, Q.F. Zhang, S. Kwong, An evolutionary many-objective optimization algorithm based on dominance and decomposition, IEEE Trans. Evol. Comput., 19 (2015) pp.694-716.

[10] X. Cai, Z. Yang, Z. Fan, Efficient nondomination level update method for steady-state evolutionary multiobjective optimization, IEEE Trans. Cybern., 47 (2017) pp.2824-2837.

[11] Y. Yuan, H. Xu, B. Wang, A new dominance relation-based evolutionary algorithm for many-objective optimization, IEEE Trans. Evol. Comput., 20 (2016) pp.16-37.

[12] S. Wang, S. Ali, T. Yue, Integrating weight assignment strategies with NSGA-II for supporting user preference multiobjective optimization, IEEE Trans. Evol. Comput., 22 (2018) pp.378-393.

[13] M.Q. Li, S.X. Yang, X.H. Liu, Shift-based density estimation for Pareto-based algorithms in many-objective optimization, IEEE Trans. Evol. Comput., 18 (2014) pp.348-365.

[14] S.Y. Jiang, S.X. Yang, A strength Pareto evolutionary algorithm based on reference direction for multi-objective and many-objective optimization, IEEE Trans. Evol. Comput., 21 (2017) pp. 329-346.

[15] T. Tian, R. Cheng, X. Y. Zhang, Y. C. Jin, An indicator-based multiobjective evolutionary algorithm with reference point adaption for better versatility, IEEE Trans. Evol. Comput., 22 (2018) pp.14-33.

[16] Y.T. Qi, X.L. Ma, F. Liu, MOEA/D with adaptive weight adjustment, Evol. Comput., 22 (2014) pp.231-264.

[17] X.Y. Cai, Z.X. Yang, Z. Fan, Decomposition-based-sorting and angle-based-selection for evolutionary multiobjective and many-objective optimization, IEEE Trans. Cybern., 47 (2016) pp.2824-2837.

[18] J. Zou, L.W. Fu, J.H. Zheng, A many-objective evolutionary algorithm based on rotated grid, Applied Soft Computing., 67 (2018) pp. 596-609.

[19] T. Santos, Ricardo. H.C. Takahashi,On the performance degradation of dominance-based evolutionary algorithms in many-objective optimization, IEEE Trans. Evol. Comput., 22 (2018) pp.19-31.

[20] J. Zou, Y.P. Zhang, S.X. Yang, Adaptive neighborhood selection for many-objective optimization problems, Applied Soft Computing., 64 (2017) pp. 186-198.

[21] N. Nedjah, L.D.M. Mourelle, Evolutionary multi-objective optimization: a survey, Int. J. Bio-Inspir. Com., 7 (2015) pp.1-25.

[22] A. Trivedi, D. Srinivasan, K. Sanyal, A survey of multiobjective evolutionary algorithms based on decomposition, IEEE Trans. Evol. Comput., 21 (2017) pp.440-462.

[23] M.G. Gong, L.C. Jiao, H.F. Du, L.F. Bo, Multi-objective immune algorithm with nondominated neighbor-based selection, Evol. Comput., 16 (2008) pp.225-255.

[24] Q.Z. Lin, J.Y. Chen, Z.H. Zhan, W.N. Chen, C.A. Coello Coello, Y.L. Yin, C.M. Lin, J. Zhang, A hybrid evolutionary immune algorithm for multi-objective optimization problems, IEEE Trans. Evol. Comput., 20 (2016) pp.711-729.

[25] Q.Z Lin, Y.P Ma, J.Y Chen, An adaptive immune-inspired multi-objective algorithm with multiple differential evolution strategies, Inform. Sci., 430-431 (2018) pp.46-64.

[26] Z. Zareizadeh, M.S. Helfroush, K. Kazemi, A new multiobjective evolutionary optimization algorithm based on theta-multiobjective clonal selection, J. Intell. Fuzzy. Syst., 32 (2017) pp.1685-1696.

[27] S.Q. Qian, Y.Q. Ye, B. Jiang, A constrained multi-objective optimization algorithm based on immune system model, IEEE Trans. Cybern.,46 (2016) pp.2056-2069

[28] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: Empirical results, IEEE Trans. Evol. Comput., 8 (2000) pp.173-195.

[29] K. Deb, L. Thiele, M. Laumanns, Scalable test problems for evolutionary multi-objective optimization, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Tech. Rep., 2001.

[30] Q.F. Zhang, W.D. Liu, H. Li, The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances, IEEE Congr. Evol. Comput., (2009) pp.203-208.

[31] S. Huband, L. Barone, L. While, P. Hingston, A scalable multi-objective test problem toolkit, In: C.A.C. Coello, A.H. Aguirre, E. Zitzler (Eds.), Guanajuato, Mexico, Proceedings of Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science, 3410 (2005) pp.280-295.

[32] H. Li, Q.F. Zhang, Multi-objective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II, IEEE Trans. Evol. Comput., 13 (2009) 284-302.

[33] B.L. Chen, W.H. Zeng, Y.B. Lin and D.F. Zhang, A new local search-based multiobjective optimization algorithm, IEEE Trans. Evol. Comput., 19 (2015) 1100-1113.

[34] H.L. Liu, F.Q. Gu and Q.F. Zhang, Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems, IEEE Trans. Evol. Comput., 18 (2014) 450-455.

[35] K. Li, K. Deb, Q.F. Zhang and S. Kwong, An evolutionary many-objective optimization algorithm based on dominance and decomposition, IEEE Trans. Evol. Comput., 19 (2015) 694-716.

[36] J. Yoo, P. Hajela, Immune network simulations in multicriterion design, Struct. Optim., 18 (1999) pp.85–94.

[37] S.F.M. Burnet. The clonal selection theory of acquired immunity. Nashville: Vanderbilt University Press. (1959).

[38] Z.H. Hu, A multiobjective immune algorithm based on a multiple-affinity model, Eur. J. Oper. Res., 202 (2010) pp.60-72.

[39] F. Freschi, M. Repetto, Vis: an artificial immune network for multi-objective optimization, Eng. Optimiz., 38 (2006) pp.975-996.

[40] J.Q. Gao, J. Wang, WBMOAIS: a novel artificial immune system for multi-objective optimization, Comput. Oper. Res., 37 (2010) pp.50-61.

[41] Castro. P. A. D, Von. Zuben. F. J, A Gaussian artificial immune system for multi-objective optimization in continuous domains, Proc. of the 10th international conference on hybrid intelligent (HIS)., (2010) pp. 159-164.

[42] Castro. P. A. D, Von. Zuben. F. J, MOBAIS: A Bayesian artificial immune system for multi-objective optimization, 7th international conference on artificial immune system (ICARIS)., (2008) pp. 48-59.

[43] S. Q. Qian, Y. Q. Ye, B. Jiang, J. H. Wang, Constrained multiobjective optimization algorithm based on immune system model, IEEE Trans. Cybern., 46 (2016) pp.2056-2069.

[44] Y.T. Qi, Z.T. Hou, M.L. Yin, H.L. Sun, J.B. Huang, An immune multi-objective optimization algorithm with differential evolution inspired recombination, Appl. Soft Comput., 29 (2015) pp.395-410.

[45] S.Q. Qian, Y.Q. Ye, B. Jiang, A micro-cloning dynamic multiobjective algorithm with adaptive change reaction strategy, Soft. Comput., 21 (2017) pp.3781-3801.

[46] J.K. Xiao, W.M. Li, X.R. Xiao, A novel immune dominance selection multi-objective optimization algorithm for solving multi-objective optimization problems, Appl. Intell., 46 (2017) pp.739-755.

[47] J. P. Xiu, Q. He, Z. Q. Yang, Research on a multi-objective constrained optimization evolutionary algorithm, 2016 4th IEEE International Conference on Cloud Computing and Intelligence systems (IEEE CCIS 2016)., (2016) pp.282-286.

[48] P. Boonyopakorn, P. Meesad, The performance evaluation of a hybrid immune genetic algorithm based on mathematical functions, 2016 International Computer Science and Engineering Conference (ICSEC)., pp.1-6.

[49] R.H. Shang, B.Q. Du, H.N. Ma, Immune clonal algorithm based on directed evolution for multi-objective capacitated arc routing problem, Appl. Soft. Comput., 49 (2016) pp.748-758.

[50] S.W. Li, X.L. Fan, S. Gao, Research on two-stage planning method of electric vehicle charging station

based on immune algorithm, 2017 IEEE Conference on, Energy Internet and Energy System Integration (EI2), pp.1-6.

[51] X. H. Liu, M. Y. Shan, R. L. Zhang, Green vehicle routing optimization based on carbon emission and multiobjective hybrid quantum immune algorithm, Math Probl ENG., 7 (2018) pp.1-9.

[52] E. Ahmadi, M. Zandieh, M. Farrokh, A multi objective optimization approach for flexible job shop scheduling problem under random machine breakdown by evolutionary algorithms, Comput. Oper. Res., 73 (2016) pp.56-66.

[53] H. Z. Ren, H. Xu, S. S. Sun, Immune genetic algorithm for multiobjective flexible job-shop scheduling problem, Proceedings of the 28th Chinese Control and Decision Conference (2016 CCDC)., (2016) pp. 2167-2171.

[54] C. Y. Bu, W. J. Luo, L. H. Yue, Continuous dynamic constrained optimization with ensemble of locating and tracking feasible regions strategies, IEEE Trans. Evol. Comput., 21 (2017) pp. 14-33.

[55] W. J. Luo, R. K. Yi, B. Yang, P. L. Xu, Surrogate-assisted evolutionary framework for data-driven dynamic optimization, IEEE TETCI., (2018) pp.1-14.

[56] W. J. Luo, X. Lin, T. Zhu, P. L. Xu, A clonal selection algorithm for dynamic multimodal function optimization, Swarm and Evolutionary Computation., https:// doi.org/10.1016/j.swevo.2018010.010..

[57] K. Deb, R.B. Agrawal, Simulated binary crossover for continuous search space, Complex. Syst., 9 (1995) pp.115-148.

[58] A. Zhou, Q.F. Zhang, Are all the subproblems equally important? Resource allocation in decomposition-based multi-objective evolutionary algorithms, IEEE Trans. Evol. Comput., 20 (2016) pp.52-64.

[59] K. Li, Á. Fialho, S. Kwong, Q.F. Zhang, Adaptive operator selection with bandits for a multi-objective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput., 18 (2014) pp.114-130.

[60] S.Z. Zhao, P.N. Suganthan, Q.F. Zhang, Decomposition-based multi-objective evolutionary algorithm with an ensemble of neighborhood sizes, IEEE Trans. Evol. Comput., 16 (2012) pp.442-446.

[61] X.F. Liu, Z.H. Zhan, Y. Lin, W.N. Chen, Y.J. Gong, T.L. Gu, H.Q. Yuan, and J. Zhang, Historical and heuristic based adaptive differential evolution, IEEE Trans. Syst., Man, Cybern., Syst., DOI: 10.1109/TSMC.2018.2855155. 2018.

[62] Z. J. Wang, Z. H. Zhan, Y. Lin, W. J. Yu, H. Q. Yuan, T. L. Gu, S. Kwong, and J. Zhang, Dual-strategy differential evolution with affinity propagation clustering for multimodal optimization problems, IEEE Trans. Evol. Comput., DOI: 10.1109/TEVC.2017.2769108. 2017.

[63] E. Zitzler, L. Thiele, Multi-objective evolutionary algorithms: a comparative case study and the strength Pareto approach, IEEE Trans. Evol. Comput., 3 (1999) pp.257-271.

[64] J.J. Durillo, A.J. Nebro, E. Alba, The jMetal framework for multi-objective optimization: Design and architecture, 2010 IEEE Congress on Evolutionary Computation (CEC), Barcelona, Spain, (2010) pp.1–8.

[65] J. Alcala-Fdez, L. Sanchez, S. Garcia, M.J. del Jesus, S. Ventura, J.M. Garrel, J.Otero, C. Romero, J. Bacardit, V.M. Rivas, J.C. Fernandez, F. Herrera, KEEL: a software tool to assess evolutionary algorithms for data mining problems, Soft Comput., 13 (2009) 307–318.

[66] H. Jain and K. Deb, An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach, IEEE Trans. Evol. Comput., 18 (2014) 602-622.