

AREA: An adaptive reference-set based evolutionary algorithm for multiobjective optimisation

Shouyong Jiang^{a,*}, Hongru Li^b, Jinglei Guo^c, Mingjun Zhong^a, Shengxiang Yang^d, Marcus Kaiser^e,
Natalio Krasnogor^e

^a*School of Computer Science, University of Lincoln, Lincoln LN6 7TS, UK*

^b*College of Information Science and Engineering, Northeastern University, Shenyang, 110819, China*

^c*Department of Computer Science, Central China Normal University, Wuhan, China*

^d*School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, UK*

^e*School of Computing, Newcastle University, Newcastle upon Tyne NE4 5TG, UK*

Abstract

Population-based evolutionary algorithms have great potential to handle multiobjective optimisation problems. However, the performance of these algorithms depends largely on problem characteristics. There is a need to improve these algorithms for wide applicability. References, often specified by the decision maker's preference in different forms, are very effective to boost the performance of algorithms. This paper proposes a novel framework for effective use of references to strengthen algorithms. This framework considers references as search targets which can be adjusted based on the information collected during the search. The proposed framework is combined with new strategies, such as reference adaptation and adaptive local mating, to solve different types of problems. The proposed algorithm is compared with state-of-the-arts on a wide range of problems with diverse characteristics. The comparison and extensive sensitivity analysis demonstrate that the proposed algorithm is competitive and robust across different types of problems studied in this paper.

Keywords: Multiobjective optimisation, reference set, search target, Pareto front, local mating

1. Introduction

Evolutionary multiobjective optimisation (EMO) is an important approach to multiobjective optimisation problems (MOPs) that frequently arise from both theoretical studies [5] and real applications [14]. For MOPs, there is unlikely a single solution that is optimal for all objective functions simultaneously, but instead a set of trade-off solutions that compromise between the objectives. Accordingly, the solution set is called the Pareto-optimal set (PS) and its image in objective space is the Pareto-optimal front (PF). EMO employs a population of individuals and therefore can search for multiple distinct points on the PF in a single run.

After several decades of advances, many EMO algorithms are available in literature as the No Free Lunch theorem [35] for optimization proves that no single EMO algorithm solves all MOPs. These algorithms use different techniques to organise population and preserve promising solutions for further evolutionary iterations. For example, Pareto-based algorithms (e.g. NSGA-II [5] and SPEA2 [47]) and indicator-based algorithms (e.g. IBEA [45]) use distinct

*Corresponding author

Email address: math4neu@gmail.com (Shouyong Jiang)

environmental selection strategies, either by Pareto dominance or contribution to indicators, for population preservation for the next generation [22]. In contrast, decomposition-based algorithms [41] employ novel population structure to conduct solution collaboration easily.

Decomposition-based EMO algorithms are effective because subproblems derived from decomposition are often easier to solve than the original MOP. MOEA/D uses references, which can be weight vectors [41], reference points [6, 10], reference vectors [2], or reference directions [16]. While references in MOEA/D are used for problem decomposition to create scalar subproblems, they can be used for objective space decomposition to approximate a small PF segment for each subspace such that the segments form a complete PF [16, 20].

However, the selection of proper references is a difficult task. The use of references raises two open issues. Firstly, requirements on the uniformity of references vary from one algorithm to another. For example, MOEA/D requires a set of well-diversified weight vectors (ideally evenly-spaced) to obtain a good distribution of solutions on the PF. Other algorithms based on objective space decomposition, however, may not necessarily require uniform references. Instead, the reference set that makes all the PF segments easy to approximate is more desirable. Secondly, optimal reference settings depend on problem properties [12]. Many existing decomposition-based algorithms use points on a unit simplex as references [6, 41]. This reference setting works well for problems with a PF similar to the unit simplex, but has been increasingly reported to be ineffective for problems with other types of PFs, such as degeneracy [12] and disconnectedness [18].

Much effort has been devoted to the first issue – reference uniformity. Qi *et al.* [24] investigated the relationship between references and the resulting subproblems after decomposition, and found that evenly-spaced references do not necessarily lead to a uniform distribution of subproblems or solutions. The authors of [24] introduced an improved reference setting method, called WS-transformation, to alleviate the problem. While it is not easy to have an optimal reference layout, other studies turn to optimising decomposition strategies for subproblems since decomposition strategies essentially establish a mapping between references and subproblems [17]. The basic idea is that uniform subproblems can be achieved by a proper mapping, even if a sub-optimal reference set was provided. Decomposition approaches, such as the penalty-based boundary intersection (PBI) [41], inverted PBI [28], adaptive PBI [38] and enhanced scalarizing functions [17], have shown improved solution distribution. Hybrid decomposition approaches have also been explored in a few recent studies [15].

In parallel with the above, there have been some studies in addressing the issue that optimal reference layout is geometrically PF dependent. The primary strategy in these studies is to evolve a reference set and population together – the reference set is dynamically adjusted to best fit the PF shape represented by the population. In [13], the reference set is dynamically adjusted by adding new reference points and removing non-useful ones. A similar idea was used in the MOEA/D with adaptive weight adjustment (MOEA/D-AWA) algorithm [24], with a difference in reversing the order of addition and removal. Farias *et al.* [8] modified MOEA/D-AWA by adopting a uniformly random method to update references every a few generations, and they reported that adaptive weight adjustment with the random method yields improved results in most of test cases. The reference vector guided evolutionary algorithm (RVEA) [2] proposed to maintain two reference sets when handling irregular PF shapes. It keeps a uniformly distributed reference set over

time as suggested by NSGA-III [6], and adaptively updates the other reference set by randomly generated reference vectors.

The study of [1] suggested two types of reference adjustments. The first type is to dynamically increase the number of references as the evolution precedes so that a complete PF can be approximated. The second type of adjustment deals with the effectiveness of the references in use, in which the ineffective ones are re-arranged to better fit the shape of irregular PFs. The use of performance indicators to guide the adaptation of references has recently been studied in [30]. Other adaptive approaches include improved reference set adaptation [18] and sampling [37]. Note that reference utilisation is similar to preference articulation [25, 26], where preference information from the decision maker is incorporated into algorithms to find only preferred regions on the PF. However, the former is intended to find a good representative of the PF (or a set of well-diversified solutions) whereas the latter often expects solutions in a region of interest, which is a small part of the PF preferred by the decision maker.

Besides the above methods, the use of machine learning models to select reference sets has been reported in recent years. For example, a linear interpolation model was used in an approach specifically for problems with discontinuous PFs [40]. Wu *et al.* introduced a Gaussian process regression model to aid reference sampling [36]. Similarly, an incremental learning model was employed for reference adaptation [9]. It should be noted that, however, the use of models for reference setting increases computational costs.

Despite a variety of success in reference-based evolutionary algorithms, the use of references is not fully explored. Many existing studies use references as a tool to form scalar subproblems [41] or estimate solution densities [16]. These studies place references in the attainable objective space for space decomposition. Positioning a few reference points outside the attainable objective space has also been studied, showing effectiveness in finding solutions in regions of interest [27]. However, little to none has been discussed if doing so with a large set of references for finding a good representative of the whole PF.

In this paper, we explore effective use of references by a new EMO framework, called AREA. Different from most decomposition-based algorithms where references are used for space decomposition, AREA considers a set of reference points as targets for population members to search. The reference set is adaptively adjusted based on the PF estimated by the solution set found so far. This way, the population can be robust to various PF geometries. Extensive sensitivity analysis carried out in this paper verifies the importance of each component of AREA. Furthermore, AREA is examined by comparing with state-of-the-arts on 35 test problems with various characteristics, demonstrating its effectiveness and robustness on diverse problem types.

The rest of the paper is organised as follows. Section 2 introduces the motivation of this research and a detailed description of the proposed AREA framework. Section 3 presents the sensitivity analysis of key components of the proposed approach, followed by experimental design in Section 4. Section 5 shows empirical results of comparison between the proposed approach and the state-of-the-arts on a variety of problems. Finally, Section 6 concludes the work.

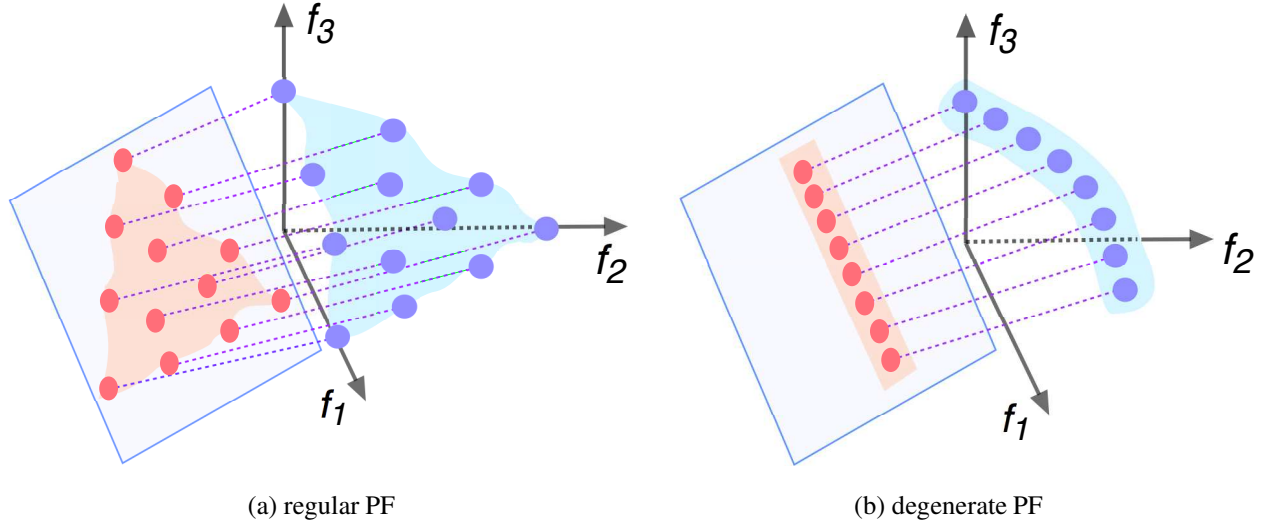


Figure 1: Illustration of EAs guided by reference set for different PF geometries. Orange dots are reference points (targets) and blue ones are solutions on the PF (paleblue) after normalisation, whose projection on a squared plane is in peach.

2. AREA: The proposed approach

In this section, we explain the motivation of this work, followed by a detailed description of the proposed algorithm.

2.1. Motivation

It is widely recognised that a proper reference set can improve population diversity and guide the search of the population effectively toward the PF [41, 6]. A good example is MOEA/D [41]. It demonstrated that a good reference set not only generates well-diversified subproblems to enhance solution distribution but also easily structures neighbourhood to encourage collaboration leading to a fast approximation to the PF. The importance of good reference sets is also seen in NSGA-III variants [13, 6] where a reference set helps to improve environmental selection. Generally, a good reference set can be composed of preferences of the decision maker (DM) represented, in most cases, by the weights of objective functions. While the DM’s preferences are hardly known ahead, preferences can be built progressively so that a good decision is achieved at the end of optimisation [25].

Many existing studies use reference sets to decompose the first orthant of the objective space in order to form either scalar subproblems [41] or search directions [2, 16]. We argue that a reference set, if placed properly in the objective space, can be search targets for population. For example in Fig. 1(a), a diversified reference set outside the orthant of the PF can be used as targets for population individuals to approach, and one can expect the final solution set, if well converged, can be as well diversified as the reference set. In other words, an optimisation problem can be solved by minimising the distance between each pair of search targets and individuals.

In irregular cases, e.g., problems with a degenerate PF in Fig. 1(b), a fixed reference set like Fig. 1(a) is no longer a good target set since such reference set is not similar to the PF. In this situation, the reference set needs to be properly adjusted to suit the degenerate PF. Likewise, it is expected that the final population approximates the degenerate PF well if the reference set is well adjusted.

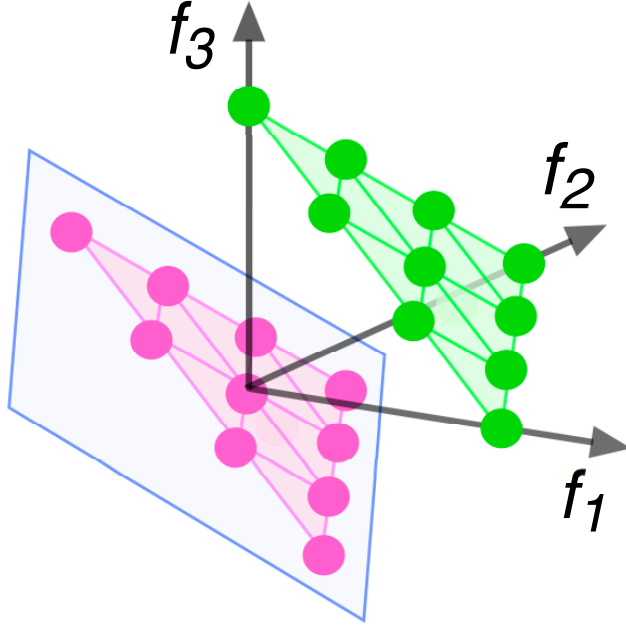


Figure 2: Shifting reference points from a unit simplex to the predefined reference plane (shown in light blue parallelogram). The reference set is in green before shifting and in pink after shifting.

The above two examples clearly show two facts: (a) reference sets can be used directly as search targets if placed in good position and (b) reference sets need adjustment for irregular PF shapes. The adaptation of reference sets has been considered in a few studies [2, 18], and we will discuss it in great detail in later sections. However, how to use reference sets as search targets has been little studied in literature. In the following, we focus on how to create a good reference set as search targets.

Considering an M -objective problem, f_i ($1 \leq i \leq M$) denotes the i -th objective value. We define in the M -dimensional objective space a reference hyperplane to help us generate a good set of reference points. Since we want reference points to be search targets, the reference plane must not be inside the attainable objective space. For simplicity, the reference plane is made to pass through the origin, as shown in Fig. 2, and is written as:

$$f_1 + f_2 + \dots + f_M = 0. \quad (1)$$

Ideally, if the PF is known, we can sample uniformly from the PF. We can then consider the projection of the sampled points (they need normalisation if their objectives are at different scales) on the reference plane as a perfect reference set, as the projection has the minimum euclidean distance from a PF point to the reference plane. However, the PF is often not known in advance, so it is difficult to preset such a perfect reference set. However, information about the PF can be gained as the search proceeds, which can be used to form a good reference set. This is discussed in the following sections in more detail.

Given a reference set, the optimisation task is to minimise the distance between a solution x and a reference point r . To reflect proximity of x to r , this paper uses the following distance-based measure

$$D_{\langle t \rangle}(x, r), \quad (2)$$

where $D_{\langle t \rangle}(x, r)$ measures the distance between x and r using t -type distance. Options for t in this paper can be ‘e’ and ‘c’, representing the Euclidean and Chebyshev distance, respectively. The Chebyshev distance is primarily used when discriminating two candidates, as it is found more effective than the Euclidean distance in our preliminary tests. A possible explanation is that the Chebyshev distance rewards a solution x_1 that approaches a target r in the direction that is perpendicular to the reference plane. In other words, the Chebyshev distance can further distinguish solutions that have the same Euclidean distance. In other cases, the Euclidean distance is used.

2.2. The AREA framework

Algorithm 1 presents the framework of AREA. It consists of four key components: initialisation (lines 1–3), reference set selection (lines 5–9), local mating probability (line 10), and population update (11–17). They are detailed as follows.

2.2.1. Initialisation

Initialisation creates a well-distributed reference set R_0 . This is done by shifting design points of simplex-lattice [4] to the predefined reference plane defined in Section 2.1. Let f_j be the j -th element of a reference point on a unit simplex, the corresponding element f'_j when the point is shifted to the reference plane is calculated by

$$f'_j = f_j - \frac{1}{M}. \quad (3)$$

This ensures that a unit simplex is shifted so as to pass through the origin. That is, any design point on $\sum_{j=1}^M f_j = 1$ can be shifted to the reference plane defined in Eq. (1) since $\sum_{j=1}^M f'_j = 0$. Fig. 2 illustrates how a reference set is created in 3-D objective space.

After the generation of R_0 , a neighbourhood structure storing the T closest neighbours for each reference point is created. Meanwhile, an initial population P is created and the nondominated set of P is saved in an archive A . The ideal and anti-ideal points, i.e. z^l and z^u , refer to two vectors having the best and worst value for each objective, respectively. They are used for objective normalisation when calculating $D_{\langle t \rangle}(x, r)$ between a solution x and a reference point r , although we recognise the identification of these two points are not easy in certain situations [33].

2.2.2. Reference Set Selection

Reference set selection occurs with frequency f_r . That is, the reference set R is maintained every f_r generations or an equivalent number of function evaluations. The setting of f_r is important and its sensitivity is discussed later in the experiments. The working R alternates between the initial R_0 and an updated reference set. The alternation has an advantage over a single mode of reference set: it enables reference set adjustment while stopping the reference set from being over-adjusted.

If it is the turn to update the reference set R , **Algorithm 2** is used. This procedure includes adding K promising points to and then removing the same number of unpromising points from the reference set. The points to be added come from archive A as A is the best representative of the PF so far. Thus K should be smaller than $|A|$, which is the size of A . In addition, when K reference points are added to R , K solutions have to be added to P to guarantee

Algorithm 1: The AREA Framework

Input: stopping criterion, population size (N), neighborhood size (T), reference set update frequency (f_r);

Output: Pareto-optimal approximation A ;

- 1 Generate initial reference set: $R_0 = \{r^1, r^2, \dots, r^N\}$ and then compute for each r^i the T closest neighbours (using Euclidean distance): $B_0(i) = \{i_1, \dots, i_T\}$;
 - 2 Generate an initial population $P = \{x^1, \dots, x^N\}$, and store nondominated members in archive A ;
 - 3 Initialise *ideal* and *anti-ideal* points, i.e., z^l and z^u ;
 - 4 **while** *Stopping criterion not met* **do**
 - 5 **if** *Reference set update condition is met* **then**
 - 6 | Update reference set and neighbourhood structure by **Algorithm 2**;
 - 7 **else**
 - 8 | Use reference set $R = R_0$ and neighbourhood structure $B = B_0$;
 - 9 | Match population to R by **Algorithm 3**;
 - 10 Calculate local mating probability $Prob(i)$ for each individual x^i in P by **Algorithm 4**;
 - 11 Initialise an empty offspring population Q ;
 - 12 **for** *each individual x^i in P* **do**
 - 13 | Select randomly an individual x^k from neighbourhood $B(i)$ with probability $Prob(i)$; Otherwise, select randomly x^k from P ;
 - 14 | Create offspring y from x^i and x^k by a genetic operator;
 - 15 | Add y to Q and update z^l with y ;
 - 16 | Identify the nearest reference r^s to y in term of Chebyshev distance, and replace individual x^s in P with y if y has a smaller Chebyshev distance to r^s than x^s ;
 - 17 Update z^u and A using the merged $P \cup Q$;
-

that each reference point is a target for a solution. Associating K reference points with K solutions has a complexity of $O(K^2)$. For computational efficiency, the complexity is restricted to $O(N)$, i.e. $K \propto \sqrt{N}$. Here, K is simply calculated by the line 1 of **Algorithm 2**.

During addition (lines 2–5 of **Algorithm 2**), each time an archive member \hat{A}_i is selected from A based on the max-min distance in K-means clustering method [31]. We select \hat{A}_i from A with the largest max-min distance to the population P . The projection of \hat{A}_i on the reference plane can be potentially used as a search target. Thus, \hat{A}_i and its projection (which can be calculated by Eq. (3) in the normalised objective space) are added to P and R , respectively. The above addition procedure is repeated K times.

To aid removal (lines 6–12 of **Algorithm 2**), we define a score for each reference point $r_i \in R$, which is calculated as:

$$\zeta(r_i) = |\{r_j \in R | D_{<c>}(x_i, r_j) < D_{<c>}(x_i, r_i)\}| \quad (4)$$

where $\zeta(r_i)$ quantifies the potential of r_i being a good search target for its associated solution x_i . The larger $\zeta(r_i)$

Algorithm 2: Update reference set

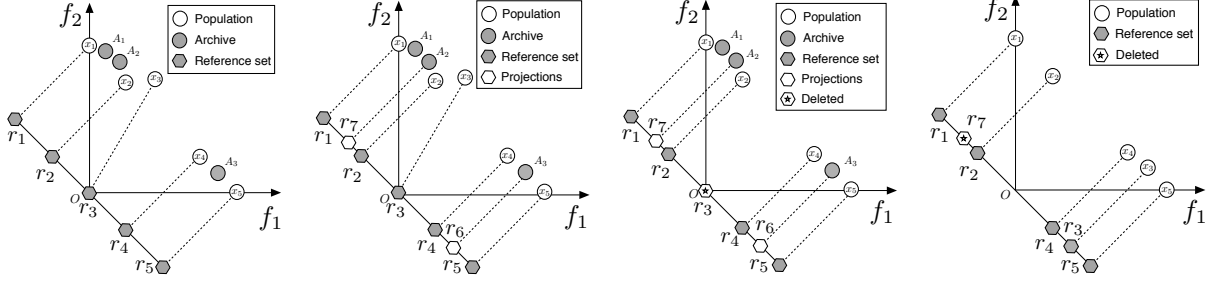
Input: population (P), archive (A), last updated reference set (R), population size (N);

Output: Output R , B and P ;

- 1 Determine the number reference of points to add: $K = \min(\sqrt{N}, |A|)$;
 // Add K reference points
 - 2 **while** $|R| - N < K$ **do**
 - 3 Compute the furthest member \hat{A}_i of A to P by the max-min distance [31];
 - 4 Add \hat{A}_i to P ;
 - 5 Add into R the projection of \hat{A}_i on the reference plane;
 // Remove K reference points
 - 6 Calculate reference score $\zeta(r_i)$ using Eq. (4) for all $r_i \in R$;
 - 7 **while** $|R| > N$ and there exists a reference score larger than 0 **do**
 - 8 // pick \hat{i} (randomly if a tie exists)
 - 9 Identify the reference with the largest score: $\hat{i} = \operatorname{argmax}_{1 \leq i \leq |R|} \zeta(r_i)$;
 - 10 Remove the reference point $r_{\hat{i}}$ from R and the solution associating with it from P ;
 - 11 Subtract one from the reference score for any reference point whose associated solution has shorter Chebyshev distance to $r_{\hat{i}}$ than that reference point;
 - 12 **if** $|R| > N$ **then**
 - 13 Remove $|R| - N$ members from P , using the k -th nearest neighbour [47], and their targets from R ;
 - 14 Calculate B : the neighbourhood structure for R ;
-

is, the less likely r_i renders a good target. The Chebyshev distance is used in Eq. (4) as it was found generally more effective than the Euclidean distance in our tests. Each time the worst target $r_{\hat{i}}$ (with the largest score) is identified ($r_{\hat{i}}$ is randomly chosen among the worst if there is a tie) and then removed from R , and the solution $x_{\hat{i}}$ for this target is also removed from P . This is repeated until the size of R is reduced to N or no more target can have a positive score. If the removal process terminates before $|R|$ reaches N , which happens when all the reference points are the best targets for their associated solutions, an additional operation is performed. The operation employs the k -th nearest neighbour method [47] to truncate P to the size of N , and the targets for those truncated solutions are removed from R . After the reference set update, the neighbourhood for each $r_i \in R$ has to be recalculated. This is to ensure that the new R and its neighbourhood structure are consistent.

Fig. 3 presents an illustrative example of the reference set update procedure of **Algorithm 2** (in objective space). As shown in Fig. 3(a), the reference set has to be adjusted as it does not lead to a good population distribution. Suppose $K = 2$ reference points are allowed for adjustment, which means two archive members are needed in both the reference addition and removal process. In Fig. 3(b), A_3 is first chosen, followed by A_2 , since they have the maximal Euclidean distance to their nearest population members. The projections of the chosen (i.e., r_6 and r_7) on the reference plane are added to the current reference set. Then, each reference point is assigned a ζ score by Eq. (4). This leads to $\zeta = 0$ for



(a) There are five reference points and associated population members and three archived members different from the population. (b) The projections of the top two furthest (to the population) archived members (first A_3 and then A_2) are added to the reference set. (c) The ζ score for all the references are calculated, and the reference point r_3 is first removed as it has the largest score ($\zeta = 1$). (d) The reference point r_7 is removed as a result of the K -th nearest neighbour method [47]. The remaining constitutes a new reference set and population.

Figure 3: Illustration of reference set and population update. Dashed lines show the association relationship between candidates and reference points.

all the reference points except r_3 , for which $\zeta = 1$. So, r_3 is removed from the reference set, as shown in Fig. 3(c). One more reference point is needed for removal. In Fig. 3(d), the k -th nearest neighbour method [47] identifies r_7 as the one having the highest density information. Therefore r_7 is removed, and the remaining reference points and their corresponding solutions are used in the next generation.

If it is the turn to use the initial reference set R_0 as R , then a match between P and R_0 should be computed such that each solution has a correct target. The procedure in **Algorithm 3** is developed to fulfil the task. The population P and archive A are merged to form S , and P is emptied at this point. Then a matrix E is created with its element E_{ij} storing the euclidean distance between $r_i \in R$ and $s_j \in S$. To refill P to the size of N , each time the nearest reference point to each $s_j \in S$ is calculated and the nearest distance is recorded. A solution associated with each nearest reference point is in turn added to P , and the corresponding row and column elements in E are set to infinity. In case that some solutions share the same nearest reference point, the one with the minimum nearest distance is added to P . The operation is repeated until P has exactly N members.

An illustrative example of matching P and R is provided in Fig. 4. Fig. 4(a) shows a population and archive distribution resulting from the use of R_1 . The population resulting from R_1 is obviously unsuitable for uniform R_0 . To update solution association, first, every candidate from both the population and archive identifies the nearest reference point in R_0 , as shown in Fig. 4(b). Then, in Fig. 4(c), each reference point chooses the nearest (Euclidean distance) candidate associated with that reference point. The reference point that does not have any associated candidates repeat the procedure of Fig. 4(b) until all the reference points in R_0 are associated with a candidate, as shown in Fig. 4(d).

2.2.3. Local Mating Probability

When a population member p is considered a parent for offspring production, it is important to select carefully its mating partner q , either locally or globally, from the population. Here, we calculate the probability of selecting q locally for p based on the closeness to its nearest neighbours in archive A (M neighbours is considered for M -

Algorithm 3: Match population to reference set

Input: population (P), archive (A), reference set (R), population size (N);

Output: Updated population;

- 1 Merge P and A to form S , then empty P ;
 - 2 Compute a distance matrix E between pairs of R and S ;
 - 3 **while** $|P| < N$ **do**
 - 4 Identify the nearest reference point for each member of S , and save them in a set I ;
 - 5 **for each** reference point $i \in I$ **do**
 - 6 Select the nearest individual $s_{\hat{k}}$ in S to i , and associate it with i ;
 - 7 Save $s_{\hat{k}}$ to P ;
 - 8 Set to infinity all the elements of the i -th row and \hat{k} -th column of the distance matrix E ;
-

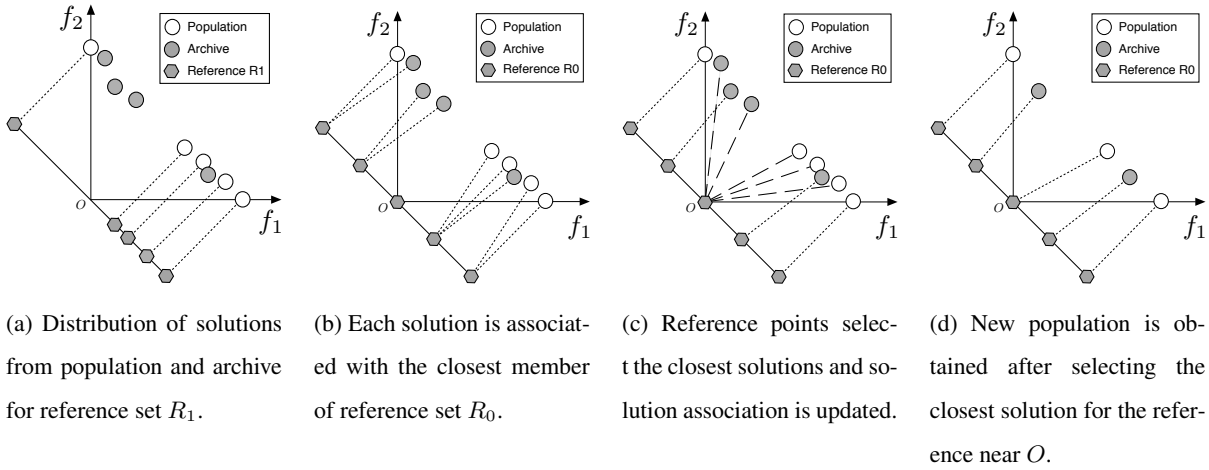


Figure 4: Illustration of updating solution association when R_0 is used. Dashed lines show the association relationship between candidates and reference points.

objective problems). Fig. 5 illustrates four possible scenarios of the distribution of p and its nearest neighbours from A in objective space. When p is a member of A , i.e. p is the same as its nearest neighbour A_2 , p in Fig. 5(a) is expected to mate with a global partner for global exploration since it has very close neighbours. p in Fig. 5(b) is encouraged to have a local partner for local exploitation to fill the gap between A_2 and A_1/A_3 . When p is different from its nearest neighbour A_2 , p in Fig. 5(c) should have a lower probability of selecting a local partner than in Fig. 5(d). For these reasons, we put forward the local mating probability for p as follows: (1) calculating for p the nearest member \bar{p} in A and recording the Euclidean distance $d_{p,1}$ between p and \bar{p} ; (2) calculating the M nearest members of \bar{p} in A and multiplying the corresponding distances to be $d_{p,2}$; (3) summing up $d_{p,1}$ and $d_{p,2}$ to get d_p . Then, the local mating probability for p is

$$Prob(p) = \frac{d_p}{\max_{v \in P}(d_v)}. \quad (5)$$

It may happen that the minimum d value is very small compared with the maximum value, making $Prob(p)$ close to

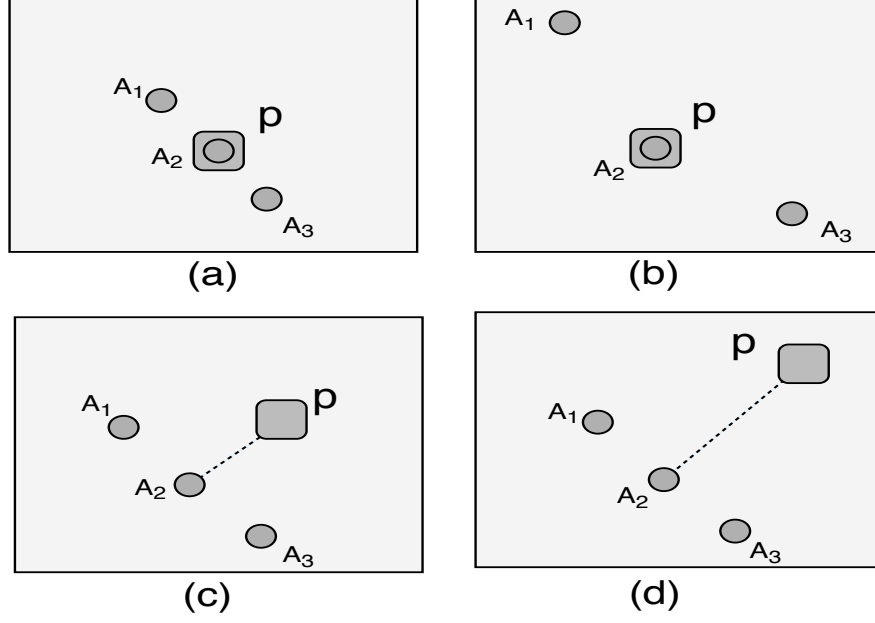


Figure 5: Illustration of the relationship between a solution p (squared) and its neighbours A_1 – A_3 (circled) in the archive. Dashed lines is the distance between p and its closest neighbour. (a) p is identical to A_2 and has two close neighbours; (b) p is identical to A_2 and has two far neighbours; (c) p is close to its nearest neighbour A_2 ; (d) p is far from its nearest neighbour A_2 .

Algorithm 4: Local mating probability

Input: population (P), archive (A), number of objectives (M);

Output: Local mating probability $Prob$;

- 1 **for** each member $i \in P$ **do**
 - 2 Identify the nearest member A_j of A to i and the corresponding nearest distance $d_{i,1}$;
 - 3 Compute the M nearest neighbours around A_j and the corresponding nearest distances: $\bar{d}_1, \dots, \bar{d}_M$;
 - 4 Multiply \bar{d} values: $d_{i,2} = \prod_{j=1}^M \bar{d}_j$;
 - 5 Compute $d_i = d_{i,1} + d_{i,2}$;
 - 6 **Compute** $Prob(i)$ for all $x_i \in P$ according to Eq.(5);
-

zero. In practice, a small value (e.g., 0.2) is added to $Prob(p)$ (the probability is capped at one) to enable p a chance of selecting a local mating partner. **Algorithm 4** presents the procedure for calculating the local mating probability for each member of the population.

2.2.4. Population Update

As described in lines 13–16 of **Algorithm 1**, each population member selects a mating partner based the local mating probability mentioned above. Then, the two members create a new solution y using a genetic operator. y updates the ideal point z^l if it is better than z^l in at least one objective, and y is also saved in an offspring population Q . The best target r^s for y is identified by minimising the Chebyshev distance (better than the Euclidean distance in our testing) to all the reference points. The population member associating with r^s is replaced by y if it is not as close (Chebyshev distance) to r^s as y . At the end of each generation, the anti-ideal point z^u and nondominated archive A

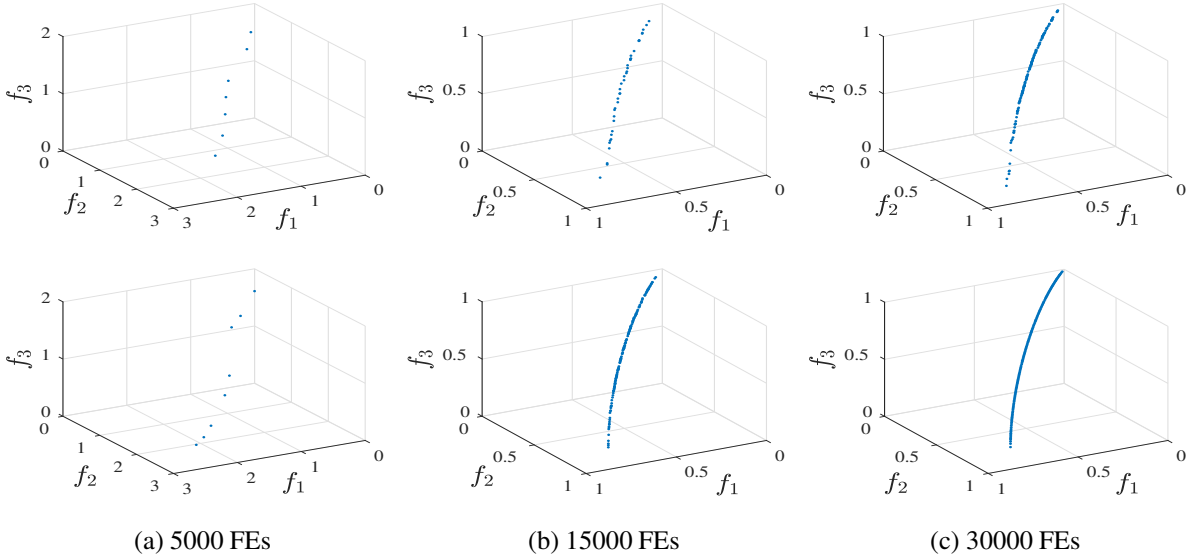


Figure 6: PF approximation of DTLZ5 obtained by $AREA_R_0$ (top) and $AREA$ (bottom) after different number of FEs.

are recalculated from $P \cup Q$. Note that, if A is over a predefined size, then it is truncated by the k -th nearest neighbour method [47].

3. Sensitivity analysis

Sensitivity analysis is conducted to study the robustness of algorithms. This section discusses the impact of different components and parameters on the performance of $AREA$. The test problems used in this section are DTLZ1-7 [7] and F1-F4 [38]. The inverted generational distance (IGD) [16] is used as the performance measurement. These will be detailed, together with parameter settings, in Section 4.

3.1. Influence of reference set

The proposed $AREA$ alternates between a predefined reference set R_0 and an evolving one (denoted R_1). Here we investigate the importance of each of them to $AREA$. We turned off the alternation, and allowed $AREA$ to use one of them only. The following demonstrates the alteration strategy works better than using one of them only.

3.1.1. $AREA$ with a fixed reference set only

This $AREA$ variant, denoted $AREA_R_0$, is compared against the original $AREA$. We found that no significant difference exists between $AREA_R_0$ and $AREA$ for regular problems like DTLZ1-2, for which the projection of the PF on the $\sum_{i=1}^M f_i = 0$ is a simplex. However, $AREA_R_0$ was shown to have poor coverage of the PF for other types of problems. For example, $AREA_R_0$ takes longer than $AREA$ to find a good coverage of the PF for the degenerate DTLZ5, as illustrated in Fig. 6.

3.1.2. $AREA$ with an evolving reference set only

$AREA$ with an evolving reference set only is denoted $AREA_R_1$. We found that such a reference set tends to drive the population toward local PF regions, as shown in Fig. 7 for MOPI. This can be simply explained by the fact that the

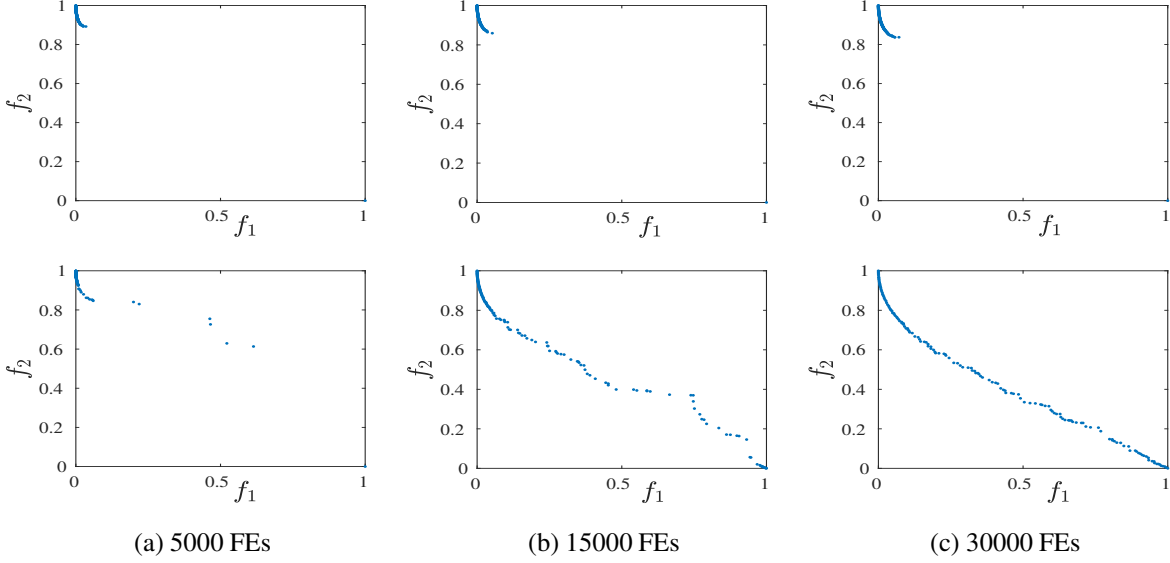


Figure 7: PF approximation of MOP1 obtained by AREA_ R_1 (top) and AREA (bottom) after different number of FEs.

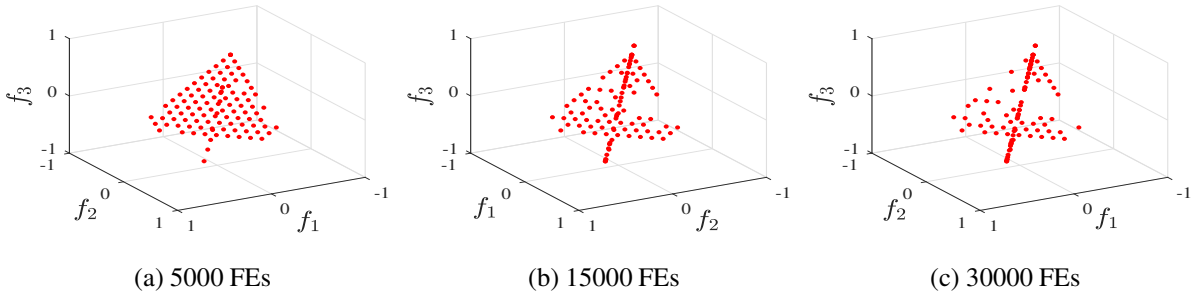


Figure 8: Evolved reference set obtained by AREA_ R_0 after different number of FEs.

nondominated set used to evolve the reference set is not well distributed, particularly at the early stage of the search. Therefore, overcrowded nondominated solutions can create dense reference points as search targets. In contrast, the original AREA avoids this issue by using a fixed reference set periodically. Fig. 8 demonstrates that the evolving reference set is adjusted gradually to fit the PF based on obtained nondominated solutions. Therefore AREA can solve MOP1 whereas AREA_ R_1 cannot.

3.2. Influence of alternation frequency of reference set

The alternation frequency f_r of reference set is an important parameter in AREA and should be properly tuned. Frequent alternation increases significantly computational complexity whereas slow alternation may cause untimely update of the reference set.

We investigated the impact of f_r by varying f_r from 5% to 50% the total number of fitness evaluations on eight selected test problems. f_r below 5% is not considered for computational efficiency. The results are shown in Table 1. It is clear that $f_r = 5\%$ obtains the best results for the DTLZ problems, and no significant improvement is observed for increased f_r on the problems from the F group. $f_r = 5\%$ is specially the best setting for DTLZ5 and DTLZ7, as indicated by the fact that an increase in f_r causes drastic performance deterioration.

Table 1: Mean and standard deviation of IGD obtained from AREA with different f_r values

Problem	AREA+ f_r (5%)	AREA+ f_r (10%)	AREA+ f_r (20%)	AREA+ f_r (30%)	AREA+ f_r (40%)	AREA+ f_r (50%)
DTLZ1	2.0303e-2(5.59e-4)	2.0968e-2(1.10e-3)	2.0748e-2(1.07e-3)	2.1145e-2(1.60e-3)	2.0693e-2(4.68e-4)	2.0862e-2(8.38e-4)
DTLZ2	5.2651e-2(4.95e-4)	5.3351e-2(5.86e-4)	5.3541e-2(4.85e-4)	5.3505e-2(4.64e-4)	5.3689e-2(6.15e-4)	5.3717e-2(5.12e-4)
DTLZ5	4.1568e-3(9.44e-5)	5.1654e-3(3.33e-4)	5.7278e-3(3.60e-4)	5.7360e-3(5.12e-4)	6.5333e-3(7.26e-4)	5.8470e-3(7.84e-4)
DTLZ7	5.6225e-2(1.47e-3)	1.1494e-1(1.18e-1)	1.4391e-1(1.36e-1)	1.8732e-1(1.48e-1)	1.7493e-1(2.38e-1)	1.4371e-1(1.36e-1)
F1	4.6459e-3(3.16e-5)	4.6172e-3(3.46e-5)	4.6103e-3(1.77e-5)	4.6137e-3(3.56e-5)	4.6164e-3(4.32e-5)	4.6147e-3(3.89e-5)
F2	5.1982e-3(1.04e-3)	4.6484e-3(4.40e-5)	4.6585e-3(1.48e-4)	4.6640e-3(1.36e-4)	4.6149e-3(4.34e-5)	4.7884e-3(1.71e-4)
F3	3.1088e-2(1.83e-4)	3.0938e-2(1.71e-4)	3.0803e-2(1.65e-4)	3.0876e-2(1.16e-4)	3.0785e-2(1.60e-4)	3.0774e-2(1.90e-4)
F4	3.7011e-3(3.86e-5)	3.7010e-1(3.93e-5)	3.7012e-1(5.67e-5)	3.7010e-1(7.62e-5)	3.7011e-1(4.90e-5)	3.7011e-1(5.78e-5)

Table 2: Mean and standard deviation of IGD obtained from different archive sizes

Problem	$ A = 1 \cdot N$	$ A = 1.5 \cdot N$	$ A = 2 \cdot N$	$ A = 3 \cdot N$
DTLZ1	1.9877e-2 (4.64e-4)	2.0303e-2 (5.59e-4)	2.0624e-2 (5.58e-4)	2.0743e-2 (7.10e-4)
DTLZ2	5.2804e-2 (5.63e-4)	5.2651e-2 (4.95e-4)	5.3081e-2 (5.32e-4)	5.2769e-2 (5.38e-4)
DTLZ5	5.1165e-3 (4.91e-4)	4.1568e-3 (9.44e-5)	4.7389e-3 (1.63e-4)	4.6006e-3 (1.70e-4)
DTLZ7	1.4356e-1 (1.35e-1)	5.6225e-2 (1.47e-3)	1.3020e-1 (1.29e-1)	1.4350e-1 (1.35e-1)
F1	4.5463e-3 (3.11e-5)	4.6459e-3 (3.16e-5)	4.5895e-3 (3.29e-5)	4.6045e-3 (3.19e-5)
F2	4.4339e-3 (3.23e-5)	5.1982e-3 (1.04e-3)	4.5831e-3 (9.19e-5)	4.5946e-3 (3.39e-5)

3.3. Influence of archive size

The sensitivity to archive size is investigated here. The archive size ($|A|$) should neither be too large nor too small. A too small archive stores very limited information about the PF, which renders insufficient to update the evolving reference set. A too large archive is also not good as it increases computational complexity. It is a common practice that the archive is at least as large as the population size (N). Therefore, we tested $|A| \in \{1, 1.5, 2, 3\}$ times of N on several problems.

The influence of the archive size is reported in Table 2. Surprisingly, a large archive size (i.e. $> 1.5N$) does not improve the performance of AREA. We suspect that this is largely due to non-uniformity caused by the archive truncation method used in AREA. In performance assessment, the archive, if $|A| > N$, needs to be truncated to a size of N . This step likely leads to a non-uniform truncated set even if the archive was uniform. A larger archive requires more solutions to be removed to keep the size as N , and removing more solutions is more likely to reduce the uniformity of solution distribution if solutions in the archive are already well distributed. An archive size of $1.5N$, however, shows to be a wise choice, as it enables remarked performance on some problems, e.g., DTLZ5 and DTLZ7, while staying robust on the other problems and not increasing computational complexity too much compared with $|A| = N$. $|A| = N$ is not good for some problems like the two irregular DTLZ problems, as it probably eliminates potential high-quality solutions due to the size limit.

Table 3: Mean and standard deviation of IGD obtained from AREA with different K values

Problem	K=0%	K=5%	K= \sqrt{N} ($\sim 10\%$)	K=20%
DTLZ1	1.9503e-2 (3.62e-4)	2.0423e-2 (2.28e-3)	2.0303e-2 (5.59e-4)	2.0948e-2 (3.34e-3)
DTLZ2	5.3006e-2 (4.83e-4)	5.2851e-2 (5.66e-4)	5.2651e-2 (4.95e-4)	5.2740e-2 (6.28e-4)
DTLZ5	7.3860e-3 (6.40e-4)	5.5111e-3 (4.86e-4)	4.1568e-3 (9.44e-5)	4.6164e-3 (4.44e-4)
DTLZ7	2.5149e-1 (2.61e-1)	9.5389e-2 (1.02e-1)	5.6225e-2 (1.47e-3)	1.4561e-1 (2.07e-1)
F1	5.0699e-3 (4.14e-4)	4.5311e-3 (3.19e-5)	4.6459e-3 (3.16e-5)	4.5284e-3 (3.98e-5)

3.4. Influence of number of additions/removals

In reference set update, AREA requires adding K potential reference points and then removing the same number of unpromising points. The impact of K on the performance of AREA is investigated through different K values ranging from 0% to 20% of the population size N .

Table 3 shows the performance of AREA with different K values. It is observed that problems with a regular PF (e.g., DTLZ1) prefer no changes in the reference set whereas problems with an irregular PF (e.g., DTLZ5 and DTLZ7) can be better solved by updating around 10% of the reference points. Large K values may be good for some problems but at the expense of high computations. Therefore, The default setting of $K = \sqrt{N}$ ($\sim 10\%$ of N) is a good choice for AREA.

3.5. Influence of local mating probability

It has been widely reported that local mating is an effective strategy to speed up the search toward the PF [19, 16]. AREA adopts this strategy for the same purpose. Dissimilar to the study [19] where the local mating probability (p_r) is set manually, AREA defines the probability based on the closeness of population members to the nearest archive members, thereby eliminating the need for parameter tuning. However, it remains unclear whether such auto-calculation is more beneficial than manual settings. The following compares our method against the p_r of 0.8 and 0.9, which are popular manual settings in a number of studies [19, 32, 44]. In addition, an AREA variant with zero local mating probability (ZLMP) is also investigated.

Table 4 reports the comparison between these approaches. While there is no significant difference between the proposed approach and the manual settings for most of the chosen problems, the proposed adaptive local mating does achieve much better performance in some cases, particularly for irregular problems like DTLZ5 and DTLZ7. Specially, the proposed approach is more than twice better than the manual setting method in obtaining IGD values on DTLZ7. DTLZ7 seems to be very sensitive to the value of p_r as suggested by the difference between $p_r = 0.8$ and $p_r = 0.9$ in Table 4. It is also noticed that ZLMP has no significant effect on AREA except in problems with irregular PFs. ZLMP allows the mating of any two parents, and this seems in particular helpful for the disconnected DTLZ7.

It is not surprising that the proposed approach is effective for irregular problems because it can adaptively determine whether local exploitation or global exploration is needed around a parent solution by calculating the nearness of this solution to the archive. Thus, the comparison here demonstrates the effectiveness of this approach.

Table 4: Mean and standard deviation of IGD obtained from different local mating probability schemes

Problem	AREA	AREA+ $p_r(0.8)$	AREA+ $p_r(0.9)$	AREA +ZLMP
DTLZ1	2.0303e-2 (5.59e-4)	2.0949e-2 (9.14e-4)	2.0979e-2 (1.55e-3)	1.9752e-2 (3.84e-4)
DTLZ2	5.2651e-2 (4.95e-4)	5.3327e-2 (5.76e-4)	5.3269e-2 (5.41e-4)	5.2820e-2 (3.43e-4)
DTLZ5	4.1568e-3 (9.44e-5)	4.6898e-3 (2.29e-4)	4.7078e-3 (2.05e-4)	4.8886e-3 (3.11e-4)
DTLZ7	5.6225e-2 (1.47e-3)	1.4318e-1 (1.36e-1)	2.2449e-1 (1.99e-1)	9.0849e-2 (9.57e-2)
F1	4.6459e-3 (3.16e-5)	4.6417e-3 (4.70e-5)	4.6325e-3 (3.30e-5)	4.5400e-3 (4.72e-5)
F2	5.1982e-3 (1.04e-3)	4.7065e-3 (1.14e-4)	4.6822e-3 (6.06e-5)	4.4528e-3 (2.23e-5)
F3	3.1088e-2 (1.83e-4)	3.0951e-2 (1.99e-4)	3.0987e-2 (2.35e-4)	3.0948e-2 (1.48e-4)
F4	3.7011e-3 (3.86e-5)	3.7015e-1 (5.60e-5)	3.7010e-1 (3.88e-5)	3.7009e-1 (4.57e-5)

4. Experimental design

This section describes experimental design for performance assessment. The following subsections provide details for test problems, compared algorithms, simulation setup, and performance measures.

4.1. Test problems

A total of 35 problems covering a wide range of features, including multimodality, disconnectedness, concavity-convexity, variable linkage, non-simply connectivity, and degeneracy, are selected for experimental studies from literature. Table 5 presents a brief summary of the problems used for study. To be specific, the test problems have four DTLZ [7] instances (three objectives considered here) and some variants whose importance has been recently recognised [3, 12]. Seven problems (MOP1-7) from the MOP test suite [20] are chosen due to their ability to analyse population diversity. UF1-9 [43] involve strong variable linkages and thus are difficult to solve. WFG [11] problems are scalable in terms of objectives and can be used to test the potential of algorithms on many-objective optimisation [2]. Some problems ($F1 - F4$) with irregularly shaped PF are also selected from [38] to test where algorithms are able to find a uniform distribution of solutions on the PF or not. We take into account additionally four test examples, i.e., $F5-F8$, which have new problem features different from the above ones. $F5$ and $F6$ are non-simply disconnected, and the PF of $F6$ consists of 3D curves. $F7$ has two PF regions, one of which is close to the origin and the other not. $F8$ is continuous but its PF has four corners and three of which are not on the objective axes. These new features are largely missing from the literature and their inclusion should provide a more comprehensive analysis on the robustness of algorithms.

Table 5: A summary of test problems. M denotes the number of objectives and n the number of variables which is the same as as DTLZ2.

Category	Problem	Description
Old	DTLZ1 [7]	linear PF, multimodality
	DTLZ2 [7]	concave PF that is a part of hypersphere in the positive orthant
	DTLZ5 [7]	degeneracy with an 1-D PF
	DTLZ7 [7]	disconnectivity with four disconnected PF segments
	IDTLZ1-2 [3]	inverted DTLZ1 and DTLZ2, corner points not on objective axis
	SDTLZ2 [6]	scaled DTLZ2, f_i multiplied by 2^{i-1} for $1 \leq i \leq M$
	CDTLZ2 [6]	convex DTLZ2, $f_i \leftarrow f_i^4$ for $1 \leq i \leq M - 1$ and $f_M \leftarrow f_M^2$
	MOP1-7 [20]	variable linkage, multimodality
	UF1-9 [43]	variable linkage, multimodality, disconnectivity
	WFG2 [11]	convexity, nonseparability, disconnectivity
	WFG4 [11]	concavity, nonseparability
	WFG6 [11]	concavity, nonseparability
	F1 [38]	extremely convex PF, intermediate PF points close to the origin
	F2 [38]	multiple disconnected PF segments
	F3 [38]	disconnectivity and extremely convex geometry
	F4 [38]	a mix of concavity and convexity
New	F5	objective functions are the same as DTLZ2, except $g(x) = \sum_{i=M}^n (x_i - 0.5)^2 + \prod_{i=1}^{M-1} \sin(0.5\pi[4x_i + 1.6]) $
	F6	objective functions are the same as DTLZ2, except $g(x) = (\prod_{i=1}^{M-1} (x_i))^{0.1} \sum_{i=M}^n (x_i - 0.5)^2 + \prod_{i=1}^{M-1} \sin(2\pi x_i) $
	F7	objective functions are the same as DTLZ2, except $g(x) = \sum_{i=M}^n (x_i - \prod_{j=1}^{M-1} x_j)^2 + \sin(0.5\pi[4 \prod_{i=1}^{M-1} (x_i - 2)]) $
	F8	$f_{i \in \{1, M-1\}} = 1 + g(x)((1 + \sum_{j=1}^M x_j)/x_i - 1)$ $f_M = \sum_{j=1}^M x_j$, $g(x) = \sum_{i=M}^n ((x_i - x_1)/n)^2$, $x \in [1, 4]^n$

4.2. Compared algorithms

AREA ¹ is compared with a range of algorithms: MOEA based on decomposition (MOEA/D) [41], adaptive non-dominated sorting genetic algorithm III (ANSGA-III) [13], strength Pareto evolutionary algorithm based on reference directions (SPEA/R) [16], and reference vector based evolutionary algorithm (RVEA) [2]. MOEA/D is used as a baseline algorithm, and ANSGA-III is an upgraded version of NSGA-III for tackling complicated problems. SPEA/R and RVEA are recently developed algorithms and have shown good ability to solve a wide range of problems.

The settings of other parameters in the compared algorithms are shown in Table 6, as suggested in their original papers. The parameters in AREA are configured based on sensitivity analysis in the previous section. All the algorithms use the same crossover and mutation operators as in MOEA/D [41] for most of the problems except the MOP group. For the MOP group, an adaptive operator [21] is used to speed up the convergence of population.

¹MATLAB implementation is freely available at <https://github.com/chang88ye/AREA> or <https://sites.google.com/view/shouyongjiang/resources>.

Table 6: Configuration of algorithms for comparison. M denotes the number of objectives and N population size. FE is short for function evaluation.

Algorithm	Parameter configuration
MOEA/D	neighbourhood size ($T = 20$)
ANSGA-III	M new points added for a reference point
SPEA/R	restricted mating ($K = 20\%N$)
RVEA	rate of change ($\alpha = 2$), frequency of adaption ($fr = 0.1$)
AREA	neighbourhood size ($T = 20$), update frequency ($f_u = 5\% \text{ #FEs}$), archive size ($1.5N$)

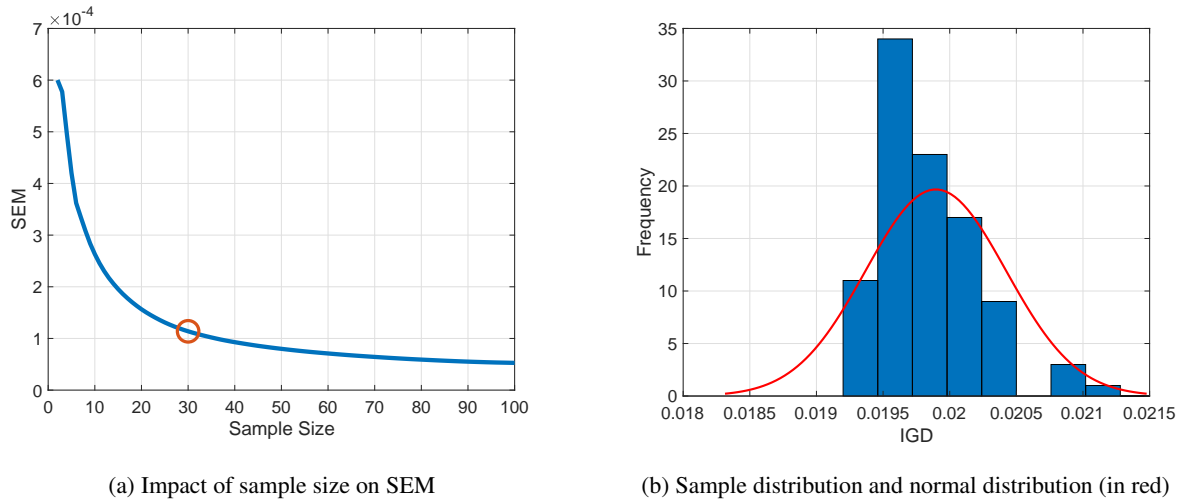


Figure 9: Relationship between sample size and the SEM of IGD and distribution of the IGD values from 100 runs of AREA on DTLZ1.

4.3. Simulation setup

The setting of population size (N) depends on the number of objectives. A population size of 100 is used for 2 objectives, as recommended by [5]. A population size of 105 is used for 3 objectives, according to the suggestion in [3, 18, 37]. For scalable WFG problems, the population size is set according to [3].

All the algorithms are run 30 times for each test problem. This sample size meets the requirement for a non-parametric test like the Wilcoxon rank-sum test [34]. We further justify the sufficiency of this sample size by the method used in [25]. That is, AREA is performed 100 times on DTLZ1. The standard error of the mean (SEM) [25] of IGD is plotted against the sample size in Fig. 9(a). The figure shows that SEM decreases slowly after a sample size of 30. The distribution of the 100 IGD values is shown in Fig. 9(b), suggesting that the samples do not conform well to the normal distribution. Therefore, the non-parametric Wilcoxon rank-sum test [34] is used in this work to indicate significant differences between compared algorithms.

For fair comparison, all the algorithms should use the same termination condition. The termination condition is the maximum number of function evaluations (FEs). 20,000 FEs are used for all the problems except the MOP instances. For MOP1-5, the number is set to 200,000 due to high optimisation difficulty of these problems. MOP6-7 use 300,000 FEs as it takes longer to get population converged toward their PFs. These termination conditions are based on existing studies [5, 20] and also take into account practical demands on runtime. In addition, WFG problems are given the same termination condition as in [3].

4.4. Performance measures

The following three measures are used for performance assessment:

Inverted generational distance (IGD). IGD [20] accounts for both diversity and convergence of approximations to the PF. Let PF be a set of solutions uniformly sampled from the true PF, and PF^* be an approximate solution set in the objective space, the metric measures the gap between PF^* and PF , calculated as follows:

$$IGD(PF^*, PF) = \frac{\sum_{p \in PF} d(p, PF^*)}{|PF|} \quad (6)$$

where $d(p, PF^*)$ is the distance between the member p of PF and the nearest member in PF^* . The calculation of IGD in this paper uses a PF of around 1000 points uniformly sampled from the true PF.

Hypervolume (HV). HV [46] calculates the size of the objective space dominated by the approximation set S and bounded by a reference point $R = (R_1, \dots, R_M)^T$ that is dominated by all points on the PF. In this paper, R is slightly worse than the nadir point by adding 0.1 to each dimension of it. The HV values presented in this paper are all normalised to $[0, 1]$ by dividing $\prod_{i=1}^M R_i$.

Spacing (S). The S measure proposed by Schott [29] is a performance indicator of solution uniformity. It is calculated by

$$S = \sqrt{\frac{1}{|PF^*|} \sum_{p \in PF^*} [d(p, PF^*) - \bar{d}]^2} \quad (7)$$

where $d(p, PF^*)$ has the same definition as in Eq. (6), and \bar{d} is the mean of $d(p, PF^*)$ for all p in PF^* .

5. Experimental results

This section is devoted to reporting the results of comparisons between AREA and other algorithms on a variety of test problems. All the test problems except UF and WFG are somewhat ZDT/DTLZ variants. UF and WFG have different problem characteristics. For example, UF problems have strong variable linkages, and WFG problems involve nonseparability and scalability. Therefore, the following discusses them separately.

5.1. Results on ZDT/DTLZ variants

The comparison between AREA and the other algorithms IS presented in Table 7, Table 8, and Table 9 with respect to IGD, HV, and S, respectively. As mentioned previously, the Wilcoxon rank-sum test [34] is applied to determine the significance of difference between AREA and the compared algorithms. Therefore, “+”, “-”, or “ \approx ” indicates the corresponding algorithm is significantly better than, worse than, or equivalent to AREA, respectively, at 0.05 significance level. The tables show that the algorithms’ IGD values are in good agreement with their HV. The S measure shows that solutions obtained by AREA for most of the problems are more uniform than those by the other algorithms. MOEA/D has some good S values mainly on the MOP group. As will be shown later, small S values by MOEA/D are due to the loss of diversity in the solution set. In what follows, detailed performance analysis of AREA relying on the IGD measure is provided for each type of test problems.

Table 7: Mean (and standard deviation) values of IGD obtained by five algorithms. Best (smallest mean) values are highlighted in boldface. Algorithms that are better than, worse than, or equivalent to AREA on ranksum test with a significance level of 0.05 are indicated by +, -, or \approx , respectively.

Problem	M	MOEA/D	ANSGA-III	SPEA/R	RVEA	AREA
DTLZ1	3	1.9455e-2 (6.26e-4) +	2.4140e-2 (6.56e-3) -	2.7387e-2 (1.47e-2) -	1.9589e-2 (7.57e-4) +	2.0303e-2 (5.59e-4)
DTLZ2	3	5.0856e-2 (3.23e-4) +	5.2590e-2 (1.52e-3) \approx	5.0334e-2 (2.19e-5) +	5.0424e-2 (2.18e-4) +	5.2651e-2 (4.95e-4)
DTLZ5	3	1.8581e-2 (9.82e-6) -	9.3907e-3 (8.52e-4) -	3.0361e-2 (1.59e-3) -	7.1990e-2 (1.03e-2) -	4.1568e-3 (9.44e-5)
DTLZ7	3	2.2484e-1 (1.53e-1) -	7.0095e-2 (2.89e-3) -	8.5436e-2 (2.05e-3) -	1.0412e-1 (4.60e-3) -	5.6225e-2 (1.47e-3)
IDTLZ1	3	3.2832e-2 (3.09e-4) -	2.1707e-2 (2.45e-3) \approx	1.1479e-1 (7.27e-2) -	4.8517e-2 (1.85e-2) -	2.1485e-2 (2.05e-3)
IDTLZ2	3	9.7814e-2 (4.02e-4) -	7.0362e-2 (4.16e-3) -	7.5515e-2 (7.49e-4) -	7.5553e-2 (1.12e-3) -	5.2069e-2 (5.01e-4)
SDTLZ2	3	1.3968e-1 (4.25e-3) -	1.3168e-1 (2.67e-3) -	1.2007e-1 (1.44e-4) -	1.2021e-1 (2.88e-4) -	1.1792e-1 (1.25e-3)
CDTLZ2	3	4.3339e-2 (4.25e-5) -	4.8019e-2 (3.63e-3) -	4.3005e-2 (1.03e-3) -	3.8184e-2 (6.84e-4) -	3.3358e-2 (7.83e-4)
MOP1	2	3.4925e-1 (5.92e-3) -	1.6972e-1 (1.07e-1) -	9.6154e-3 (1.20e-3) +	4.6443e-2 (3.08e-3) -	1.0964e-2 (2.65e-4)
MOP2	2	3.5375e-1 (6.52e-3) -	2.5475e-1 (3.92e-2) -	2.3403e-2 (4.40e-2) +	8.6316e-2 (2.38e-2) \approx	1.1691e-1 (8.64e-2)
MOP3	2	5.3701e-1 (8.60e-2) -	4.0667e-1 (4.28e-2) -	3.0235e-2 (4.61e-2) \approx	1.3239e-1 (5.95e-2) -	1.3660e-2 (3.33e-2)
MOP4	2	2.7318e-1 (1.39e-2) -	2.4355e-1 (2.55e-2) -	7.9912e-3 (3.13e-3) +	3.4180e-2 (1.72e-2) -	2.1709e-2 (3.37e-2)
MOP5	2	3.1330e-1 (1.10e-2) -	2.1310e-1 (6.82e-2) -	2.2083e-2 (1.78e-3) -	7.8166e-1 (3.60e-2) -	1.5858e-2 (7.78e-4)
MOP6	3	3.0983e-1 (1.88e-8) -	2.7121e-1 (3.25e-2) -	2.5371e-1 (3.88e-2) -	1.3391e-1 (1.02e-2) +	2.0143e-1 (4.42e-2)
MOP7	3	3.5765e-1 (1.84e-8) -	3.5410e-1 (1.58e-2) -	3.3627e-1 (2.49e-2) \approx	2.1851e-1 (1.08e-2) +	3.3486e-1 (2.72e-2)
F1	2	2.4318e-2 (2.35e-3) -	4.1538e-2 (1.60e-2) -	2.5016e-2 (2.26e-3) -	2.1641e-2 (1.29e-3) -	4.6459e-3 (3.16e-5)
F2	2	7.5871e-1 (1.38e-1) -	2.0699e-1 (2.18e-1) -	4.7002e-3 (1.11e-6) \approx	2.0577e-2 (4.57e-3) -	5.1982e-3 (1.04e-3)
F3	2	9.1038e-2 (1.30e-1) -	8.0237e-2 (8.03e-2) -	4.0067e-2 (4.49e-4) -	3.8267e-2 (1.02e-3) -	3.1088e-2 (1.83e-4)
F4	2	5.2306e-1 (1.91e-1) -	4.1481e-1 (4.56e-2) -	3.7015e-1 (2.14e-4) \approx	3.7219e-1 (1.32e-3) -	3.7011e-3 (3.86e-5)
F5	3	5.1104e-2 (8.57e-4) -	4.8516e-2 (9.13e-4) +	4.8215e-2 (7.88e-5) +	5.6043e-2 (2.05e-3) -	4.9318e-2 (1.09e-3)
F6	3	1.0757e-1 (7.50e-2) -	2.7622e-2 (2.09e-3) \approx	3.4320e-2 (3.44e-3) -	6.7625e-2 (7.76e-3) -	2.7028e-2 (2.86e-3)
F7	3	8.9180e-2 (9.47e-3) \approx	9.8225e-2 (1.33e-2) -	8.4542e-2 (6.63e-3) \approx	1.0226e-1 (1.54e-2) -	8.5116e-2 (5.93e-3)
F8	3	3.9632e-1 (1.09e-2) -	2.6829e-1 (2.06e-2) -	2.9981e-1 (1.86e-2) -	5.1124e-1 (5.96e-2) -	1.9482e-1 (4.43e-3)
		+ / - / \approx	2/20/1	1/19/3	5/13/5	4/18/1

For the first two DTLZ problems, AREA has no much advantage over the other algorithms, and even is slightly worse than MOEA/D or RVEA, indicating AREA is not the best choice for problems with simple PF geometries. A close visual inspection of the PF approximation for these two problems shows that AREA has a good distribution of solutions, but the distribution is just not as well-organised as those of the other algorithms (see Fig. 10). It is actually the solution organisation that makes the tiny difference between these algorithms. Thus, AREA is indeed a good performer on these two problems. For the other DTLZ variants, AREA is clearly the winner, which demonstrates the proposed approach has the ability to handle complex PF shapes effectively. The second best performer for these irregular DTLZ variants is ANSGA-III. The two top performers suggest that adjusting reference set to guide the search is indeed good for irregular PF shapes. Fig. 11 graphically supports the above conclusion and illustrates that AREA has the advantage in finding more well-diversified solutions covering the whole PF than the other algorithms.

Table 8: Mean (and standard deviation) values of normalised HV obtained by five algorithms. Best (highest mean) values are highlighted in boldface. Algorithms that are better than, worse than, or equivalent to AREA on ranksum test with a significance level of 0.05 are indicated by +, -, or \approx , respectively.

Problem	M	MOEA/D	ANSGA-III	SPEA/R	RVEA	AREA
DTLZ1	3	8.4080e-1 (5.41e-3) +	8.3142e-1 (1.32e-2) -	8.1931e-1 (3.62e-2) -	8.4090e-1 (2.40e-3) +	8.3905e-1 (2.91e-3)
DTLZ2	3	5.6284e-1 (7.32e-5) +	5.5848e-1 (2.98e-3) \approx	5.6265e-1 (1.40e-4) +	5.6247e-1 (4.33e-4) +	5.5970e-1 (1.04e-3)
DTLZ5	3	1.9270e-1 (8.98e-6) -	1.9620e-1 (5.42e-4) -	1.8474e-1 (7.40e-4) -	1.5229e-1 (8.00e-3) -	1.9980e-1 (1.19e-4)
DTLZ7	3	2.5602e-1 (1.55e-2) -	2.7122e-1 (1.33e-3) -	2.7224e-1 (7.37e-4) -	2.5954e-1 (4.19e-3) -	2.7741e-1 (1.06e-3)
IDTLZ1	3	2.0107e-1 (2.01e-3) -	2.1427e-1 (6.58e-3) \approx	1.1620e-1 (5.08e-2) -	1.7373e-1 (2.24e-2) -	2.1434e-1 (6.44e-3)
IDTLZ2	3	5.0858e-1 (2.26e-4) -	5.2272e-1 (3.95e-3) -	5.1722e-1 (7.68e-4) -	5.1456e-1 (1.30e-3) -	5.3789e-1 (7.25e-4)
SDTLZ2	3	5.4925e-1 (3.69e-3) -	5.5087e-1 (1.65e-3) -	5.6261e-1 (1.39e-4) +	5.6230e-1 (2.74e-4) +	5.5887e-1 (1.15e-3)
CDTLZ2	3	9.5974e-1 (5.15e-5) -	9.5725e-1 (1.57e-3) -	9.6013e-1 (4.86e-4) -	9.6001e-1 (7.48e-4) -	9.6224e-1 (5.38e-4)
MOP1	2	2.6452e-1 (1.20e-2) -	5.1353e-1 (1.37e-1) -	7.1162e-1 (1.26e-3) +	5.6704e-1 (3.42e-3) -	7.1006e-1 (3.36e-4)
MOP2	2	1.7408e-1 (2.88e-3) -	2.5167e-1 (3.60e-2) -	4.2335e-1 (3.94e-2) +	3.3901e-1 (3.20e-2) -	3.4022e-1 (7.80e-2)
MOP3	2	9.0909e-2 (7.06e-17) -	1.5771e-1 (3.01e-2) -	3.1959e-1 (4.56e-2) \approx	2.4461e-1 (4.60e-2) -	3.3640e-1 (3.55e-2)
MOP4	2	2.9434e-1 (7.88e-3) -	3.3744e-1 (2.19e-2) -	5.9251e-1 (6.16e-3) +	5.5592e-1 (2.42e-2) -	5.7784e-1 (3.66e-2)
MOP5	2	4.0194e-1 (2.51e-3) -	4.6204e-1 (8.45e-2) -	6.9323e-1 (2.48e-3) -	6.1629e-1 (5.29e-3) -	7.0285e-1 (9.59e-4)
MOP6	3	6.1397e-1 (1.76e-7) -	6.5164e-1 (3.53e-2) -	6.3036e-1 (2.72e-2) -	7.4727e-1 (1.26e-2) +	6.9172e-1 (4.12e-2)
MOP7	3	4.0220e-1 (1.64e-7) -	3.9730e-1 (3.56e-3) -	3.9306e-1 (8.16e-3) -	4.6164e-1 (4.84e-3) +	4.1530e-1 (2.18e-2)
F1	2	9.1326e-1 (2.67e-4) -	9.1057e-1 (3.07e-3) -	9.1275e-1 (6.06e-4) -	9.1259e-1 (5.09e-4) -	9.1470e-1 (8.86e-5)
F2	2	9.6426e-2 (3.02e-2) -	2.0019e-1 (6.28e-2) -	2.5698e-1 (1.52e-7) -	2.5043e-1 (2.11e-3) -	2.5744e-1 (9.82e-5)
F3	2	8.5602e-1 (2.33e-2) -	8.5954e-1 (1.26e-2) -	8.6532e-1 (2.07e-4) -	8.6393e-1 (6.56e-4) -	8.6594e-1 (1.58e-4)
F4	2	2.4826e-1 (1.31e-1) -	2.2656e-1 (8.85e-2) -	3.5264e-1 (4.43e-4) -	3.4757e-1 (2.31e-3) -	3.5336e-1 (7.18e-5)
F5	3	5.6127e-1 (5.17e-4) +	5.6059e-1 (1.78e-3) +	5.6207e-1 (2.11e-4) +	5.5648e-1 (2.27e-3) -	5.5970e-1 (1.02e-3)
F6	3	5.3669e-1 (3.01e-2) -	5.5286e-1 (1.07e-3) -	5.4505e-1 (3.30e-3) -	5.2716e-1 (2.50e-3) -	5.5672e-1 (8.05e-4)
F7	3	8.4265e-1 (1.37e-3) +	8.2664e-1 (6.60e-3) -	8.2016e-1 (6.78e-3) -	8.2307e-1 (4.91e-3) -	8.3293e-1 (2.34e-3)
F8	3	5.0984e-1 (2.33e-4) -	5.0957e-1 (3.25e-3) -	5.1518e-1 (2.38e-3) -	4.9933e-1 (2.09e-3) -	5.2316e-1 (2.01e-3)
+ / - / \approx		4/19/0	1/20/2	6/16/1	5/18/0	

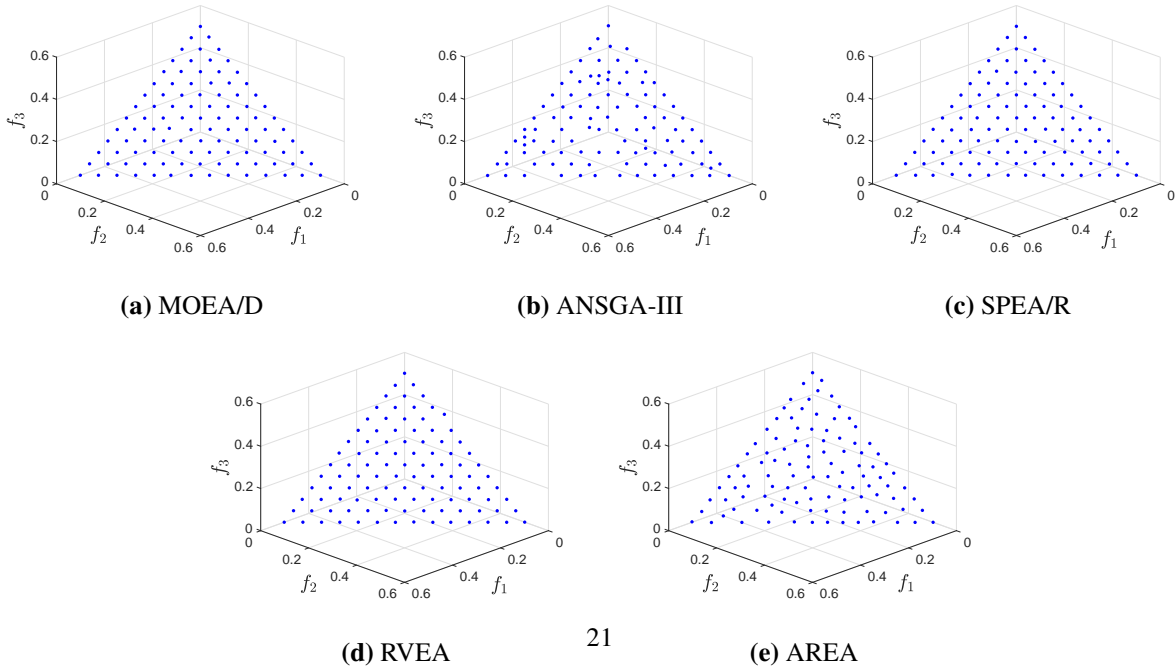


Figure 10: PF approximation of DTLZ1 obtained by different algorithms.

Table 9: Mean (and standard deviation) values of S obtained by five algorithms. Best (smallest mean) values are highlighted in boldface. Algorithms that are better than, worse than, or equivalent to AREA on ranksum test with a significance level of 0.05 are indicated by +, -, or \approx , respectively.

Problem	M	MOEA/D	ANSGA-III	SPEA/R	RVEA	AREA
DTLZ1	3	4.5066e-3 (2.63e-3) +	1.8173e-1 (8.33e-1) -	3.4367e-1 (1.74e+0) -	3.0358e-3 (2.35e-3) +	9.9764e-3 (1.67e-2)
DTLZ2	3	5.5332e-2 (2.16e-3) -	4.9465e-2 (2.68e-3) -	5.2473e-2 (2.71e-4) -	5.2531e-2 (1.68e-4) -	2.1922e-2 (2.01e-3)
DTLZ5	3	1.5355e-2 (2.07e-3) -	1.2224e-2 (1.60e-3) -	1.3130e-2 (1.19e-2) -	1.0213e-1 (2.98e-2) -	4.7130e-3 (4.21e-4)
DTLZ7	3	1.1663e-1 (2.41e-2) -	7.0257e-2 (6.08e-3) -	8.1876e-2 (1.24e-2) -	9.7947e-2 (8.88e-3) -	2.8537e-2 (7.56e-3)
IDTLZ1	3	4.3653e-2 (1.32e-3) -	2.3937e+0 (7.39e+0) -	5.0843e+0 (1.18e+1) -	5.4957e-2 (4.09e-2) -	7.7326e-3 (7.12e-4)
IDTLZ2	3	9.5006e-2 (9.60e-4) -	5.7135e-2 (8.95e-3) -	7.5360e-2 (2.77e-3) -	2.4001e-2 (2.67e-3) -	2.2644e-2 (1.58e-3)
SDTLZ2	3	1.6470e-1 (4.33e-3) -	1.2234e-1 (8.49e-3) -	1.0775e-1 (1.15e-3) -	1.0787e-1 (1.12e-3) -	5.3431e-2 (5.48e-3)
CDTLZ2	3	3.3489e-2 (3.42e-4) -	4.0234e-2 (7.09e-3) -	3.6268e-2 (3.70e-3) -	3.2319e-2 (2.01e-3) -	1.5687e-2 (1.63e-3)
MOP1	2	9.6621e-3 (7.38e-4) -	1.3169e-2 (2.37e-2) \approx	9.8662e-3 (1.61e-4) -	4.4518e-2 (1.41e-2) -	3.5248e-3 (5.51e-4)
MOP2	2	1.2148e-4 (6.65e-4) +	1.1699e-2 (3.84e-2) +	1.5014e-2 (2.98e-2) +	1.7734e-1 (1.07e-1) -	7.2180e-2 (1.85e-2)
MOP3	2	3.8025e-5 (1.74e-4) +	5.8225e-5 (3.02e-4) +	2.4566e-2 (2.84e-2) +	1.7402e-1 (1.32e-1) -	4.0200e-2 (1.21e-1)
MOP4	2	1.1218e-2 (7.54e-4) -	5.2088e-3 (2.38e-3) -	1.1445e-2 (1.63e-3) -	2.8181e-2 (1.41e-2) -	2.6247e-3 (1.43e-3)
MOP5	2	7.9106e-3 (6.45e-3) \approx	8.4370e-2 (3.48e-2) -	1.5577e-2 (4.13e-3) -	5.7877e-2 (4.11e-2) -	9.5701e-3 (1.03e-2)
MOP6	3	1.1480e-2 (1.61e-4) +	1.3729e-2 (4.16e-3) +	7.7551e-2 (3.59e-2) -	4.3294e-2 (2.09e-2) -	4.1046e-2 (4.39e-2)
MOP7	3	1.3615e-2 (1.03e-3) +	1.4848e-2 (3.08e-2) +	1.1384e-1 (7.82e-2) \approx	1.1523e-1 (6.26e-2) \approx	9.3868e-2 (7.62e-2)
F1	2	3.7998e-2 (4.81e-3) -	2.2475e-2 (7.33e-3) -	1.8889e-2 (1.25e-2) -	3.2146e-2 (2.53e-3) -	3.1936e-3 (4.50e-4)
F2	2	4.0123e-4 (2.20e-3) +	5.7417e-2 (5.82e-2) -	7.4123e-3 (1.70e-5) +	3.7215e-2 (1.01e-2) -	1.4676e-2 (5.93e-3)
F3	2	8.8891e-2 (1.98e-3) -	5.9362e-2 (1.79e-2) -	7.4343e-2 (2.17e-2) -	4.4087e-2 (3.34e-3) -	2.1394e-2 (5.25e-3)
F4	2	5.3214e-3 (4.50e-3) \approx	5.2218e-2 (7.32e-2) -	1.3203e-2 (4.62e-3) -	1.3208e-2 (2.83e-3) -	3.3803e-3 (3.92e-4)
F5	3	5.9764e-2 (2.06e-3) -	4.7852e-2 (2.19e-3) -	4.8743e-2 (1.02e-3) -	5.8263e-2 (6.90e-3) -	2.1816e-2 (2.12e-3)
F6	3	4.5513e-2 (1.79e-2) -	5.6561e-2 (5.43e-3) -	6.8635e-2 (1.36e-2) -	6.9962e-2 (5.97e-3) -	2.1303e-2 (1.75e-3)
F7	3	1.2604e-1 (2.97e-3) -	1.4898e-1 (1.25e-1) -	2.1306e-1 (1.25e-1) -	1.1459e-1 (2.93e-2) -	9.9732e-2 (5.38e-2)
F8	3	3.1472e-1 (6.33e-2) -	2.3541e-1 (2.45e-2) -	2.1309e-1 (4.25e-2) -	1.8645e-1 (1.77e-2) -	8.3958e-2 (9.20e-3)
+ / - / \approx		6/15/2	4/18/1	3/19/1	1/21/1	

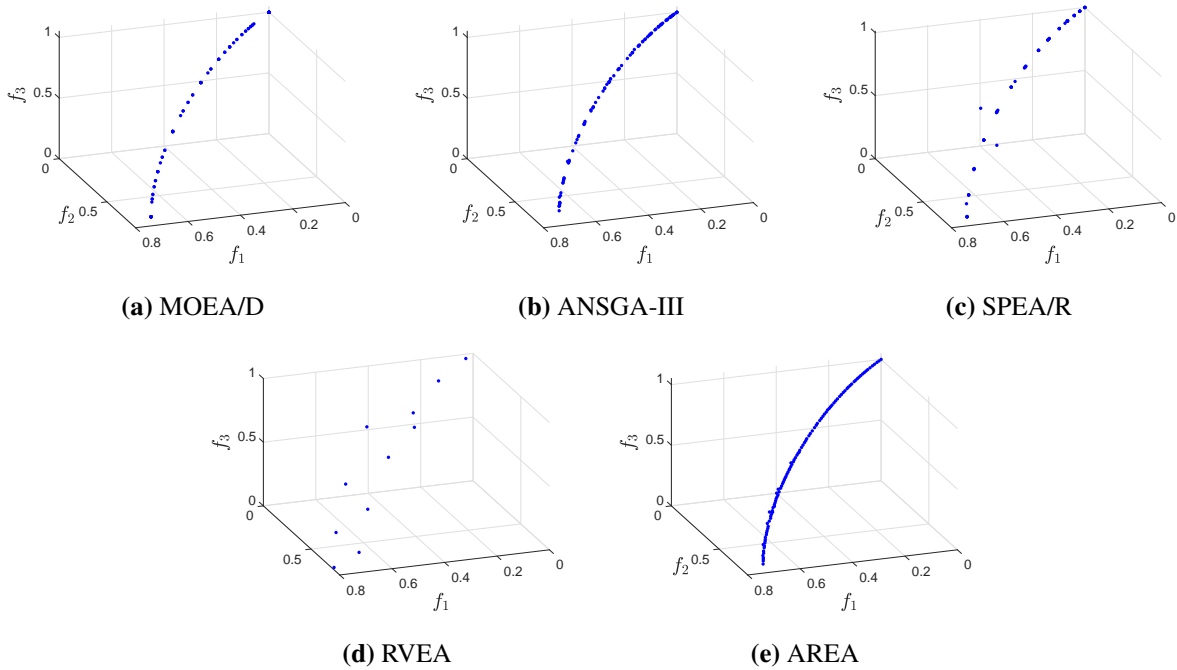


Figure 11: PF approximation of DTLZ5 obtained by different algorithms.

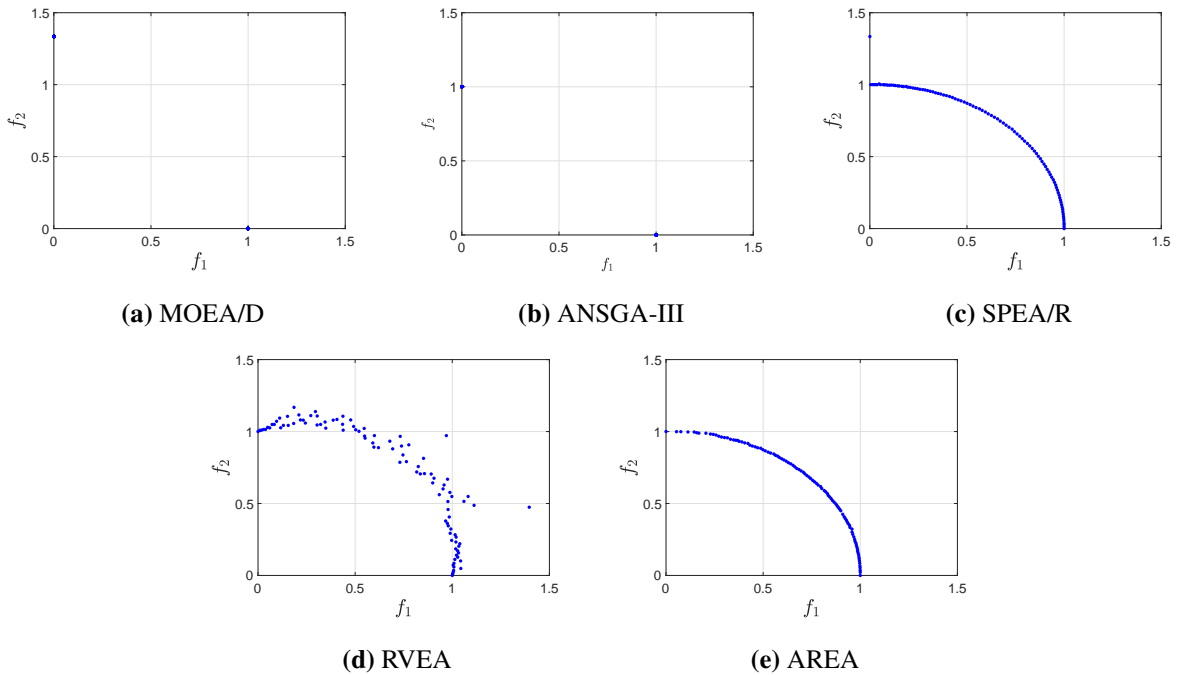


Figure 12: PF approximation of MOP3 obtained by different algorithms.

Although AREA is not always the best algorithm on the MOP group, it has good robustness. Specifically, For the first five biobjective MOP instances, SPEA/R and AREA are the leading approaches: while SPEA/R wins on

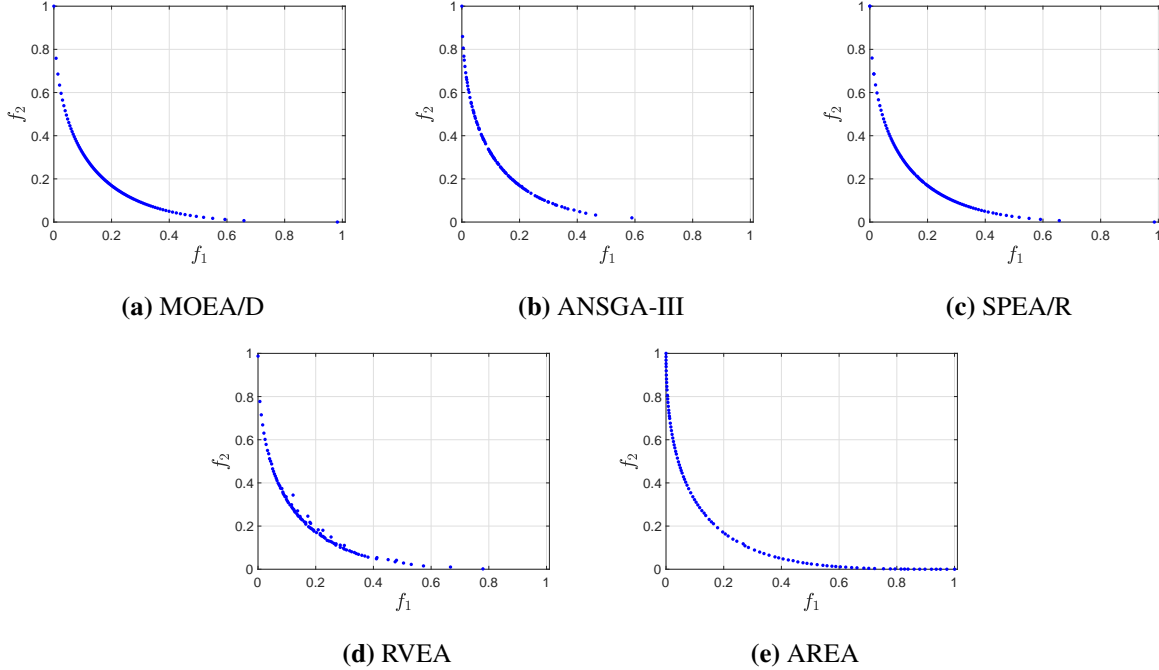


Figure 13: PF approximation of F1 obtained by different algorithms.

some problems, AREA wins on the others. For the two triobjective MOP instances, i.e. MOP6-7, RVEA shows the best performance, followed by AREA. The testing on this test suite, along with a visual example shown in Fig. 12, demonstrates that AREA is capable of dragging the population diversely toward the whole PF when some PF regions are significantly harder to approximate than others. This is actually one of the advantages AREA has over MOEA/D by placing the reference set in opposition to the first quadrant. RVEA seems to have slow convergence for the MOP group illustrated in Fig. 12.

For F1-F8, AREA outperforms the other algorithms on most of the problems except F5, demonstrating again the effectiveness of the proposed search strategies. To be more specific, F1-F4 are biobjective problems, and F5-F8 are triobjective ones. The PF approximations of F1, shown in Fig. 13, are used to show the potential strengths and drawbacks of these algorithms in biobjective cases. It is seen that all the algorithms except AREA are not capable of finding boundary solutions on the PF of F1. AREA has a good coverage of the PF due to effective reference layout. For F5-F8, AREA again shows better performance than the others in terms of solution distribution and coverage of the PF. This is demonstrated by the PF approximations of F8 in Fig. 14. Thus, the proposed AREA is highly competitive and robust for a variety of problem characteristics.

We also provide the PF approximations of the five algorithms for the other test problems in the supplementary material. These additional plots clearly show that AREA achieves competitive performance and in many cases is even better than the other algorithms.

Fig. 15 plots the average IGD values of the five algorithms against the number of function evaluations (FEs) for some selected problems. It is observed that the convergence speed differs significantly in the algorithms. Some algorithms, such as MOEA/D and RVEA, converge quickly toward the PF at the beginning of search but are not able to

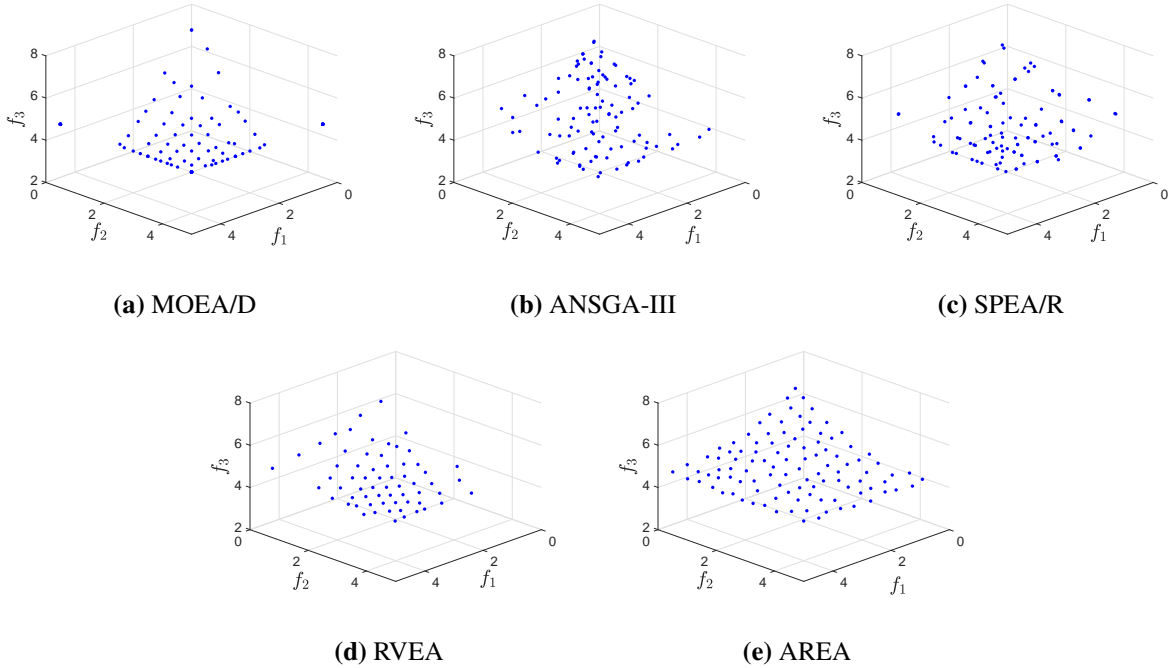


Figure 14: PF approximation of F8 obtained by different algorithms.

further reduce their IGD values as the number of FEs increases. In contrast, AREA is capable of lowering IGD values over function evaluations, therefore obtaining better IGD values at the end of search for the majority of the problems.

5.2. Results on UF/WFG

Here, we extend performance analysis to other test suites: UF [43] and WFG [11]. UF contains a number of 2- and 3-objective difficult problems that have complicated PSs. Therefore, algorithms require a large number of iterations to approximate the PS/PF of UF well. WFG is a set of scalable problems and frequently used for assessing algorithms' scalability in many-objective optimisation. For notational convenience, 'WFGK-M' represents the WFGK instance with M objectives. AREA is compared with MOEA/D [19] and RVEA [2] on these two test suites, and the results are analysed separately in the following paragraphs.

5.2.1. Results on UF

Differential evolution (DE) helps to achieve better results on UF problems than simulated binary crossover (SBX) [19]. Therefore, we use DE in compared algorithms for this test suite. Population size is 600 and 595 for 2- and 3-objective problems [42], respectively. The maximum number of evaluations is 300,000 according to [43].

Table 10 presents the IGD values of the algorithms for nine UF problems. It shows that, while AREA performs the best on approximately half of the problems, MOEA/D wins the rest of comparison. A close inspection of their PF approximations in Figs. 16–17 suggests that both AREA and MOEA/D are effective for the UF problems, although one of them may have better IGD values. On the other hand, RVEA struggles to solve the UF problems, as indicated by both large IGD values and poor PF approximations. RVEA loses all the comparison due to slow convergence performance.

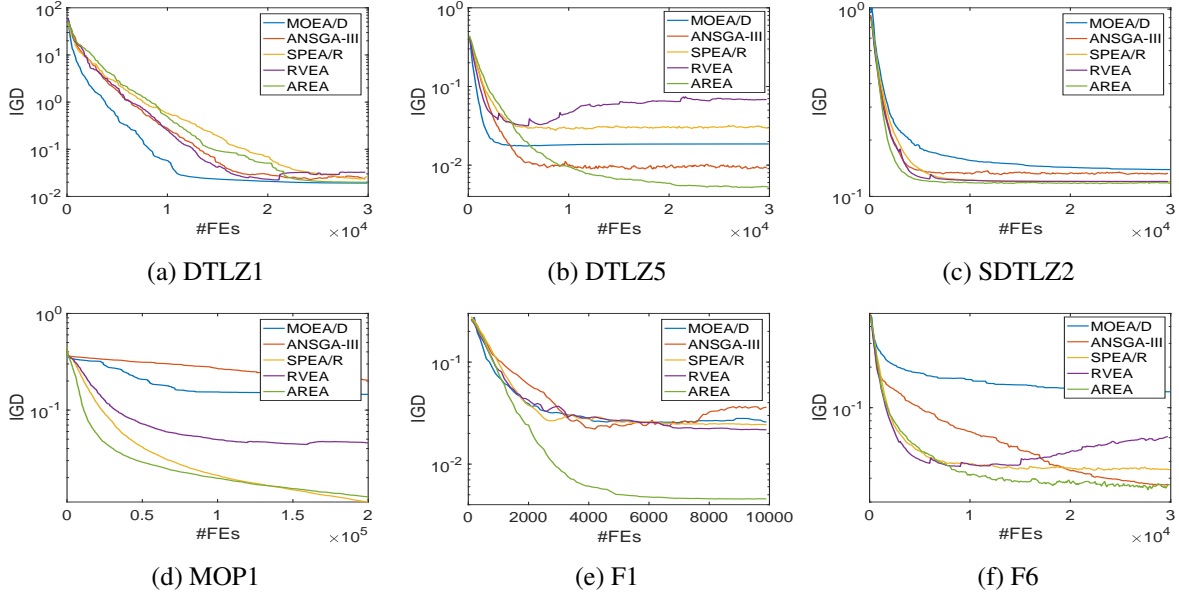


Figure 15: IGD values (averaging 30 runs) of algorithms against the number of FEs.

5.2.2. Results on WFG

All the algorithms use SBX for crossover for the WFG problems, as recommended in [6, 2]. Our experimental design has the same configurations as in [6].

Table 11 reports the IGD values of the three algorithms for selected WFG problems with different number of objectives. It is observed that AREA has the best performance on the 3-objective WFG problems, and MOEA/D is significantly worse than the other two. With an increase in the number of objectives, all the algorithms experience performance degradation, as indicated by larger IGD values. For 8-objective WFG problems, RVEA performs better than the other two algorithms, and AREA is only slightly worse in term of IGD. When the number of objectives reaches 15, the IGD values of all the three algorithms are large, indicating they all suffer from the curse of dimensionality. Despite that, AREA shows better performance particularly for WFG4 and WFG6. We also show in Fig. 18 the parallel coordinate plots of the three algorithms for WFG4 with 8 objectives. This figure suggests that MOEA/D and RVEA tend to find numerous similar solutions as many parallel coordinate lines overlap. In contrast, the solutions obtained by AREA seem diversified as there are fewer overlapping lines. Despite that, we recognise that many-objective optimisation is challenging, and believe AREA can be improved to better handle many-objective problems.

5.3. Comparison with reference adaptation approaches

Here, we would like to compare AREA with the algorithms that are equipped with specific reference adaptation mechanisms. RVEA* [2] (which is another version of the RVEA algorithm) and MOEAD-AWA [24] have proven effective for various PF geometries, and therefore are used as comparison targets for AREA.

Table 12 presents the IGD values obtained by the three algorithms on a number of test problems. It is clear that MOEA/D-AWA is indeed very suitable for irregular PF shapes (from DTLZ5 to CDTLZ2), but its performance on the MOP problems with regular PF shapes is unexpectedly poor. RVEA* does not show much promise on the test problems. The performance of AREA is very similar to that of MOEA/D-AWA on DLTZ-based irregular PF shapes.

Table 10: Mean (and standard deviation) values of IGD obtained by three algorithms for nine UF instances. Algorithms that are better than, worse than, or equivalent to AREA on ranksum test with a significance level of 0.05 are indicated by +, -, or \approx , respectively.

Problem	M	AREA	MOEA/D	RVEA
UF1	2	2.3897e-3 (4.80e-4)	1.5587e-3 (7.35e-5) +	7.7344e-2 (8.70e-3) -
UF2	2	5.7683e-3 (8.01e-4)	6.3921e-3 (1.68e-3) -	9.7725e-2 (4.91e-3) -
UF3	2	1.3792e-2 (9.70e-3)	3.6351e-3 (2.04e-3) +	2.3958e-1 (3.02e-2) -
UF4	2	4.8969e-2 (1.78e-3)	5.6859e-2 (6.58e-3) -	6.5310e-2 (2.17e-2) -
UF5	2	2.3068e-1 (4.88e-2)	2.8797e-1 (5.85e-2) -	1.1732e+0 (1.40e-1) -
UF6	2	2.7811e-1 (1.30e-1)	8.0531e-2 (1.19e-2) +	5.1894e-1 (2.23e-2) -
UF7	2	2.0780e-3 (2.27e-4)	2.1632e-3 (1.85e-4) \approx	7.0801e-2 (4.92e-3) -
UF8	3	6.6235e-2 (8.10e-3)	7.4085e-2 (1.55e-2) -	2.3143e-1 (1.17e-2) -
UF9	3	2.0119e-1 (4.79e-2)	9.6465e-2 (7.76e-2) +	2.0489e-1 (1.52e-2) \approx
+ / - / \approx			4/4/1	8/0/1

Table 11: Mean (and standard deviation) values of IGD obtained by three algorithms for selected WFG instances. Algorithms that are better than, worse than, or equivalent to AREA on ranksum test with a significance level of 0.05 are indicated by +, -, or \approx , respectively.

Problem	AREA	MOEA/D	RVEA
WFG2-3	1.5597e-1 (4.60e-3)	4.1966e-1 (1.54e-2) -	1.9748e-1 (1.57e-2) -
WFG2-8	5.5505e+0 (7.46e-1)	2.9323e+0 (6.93e-1) +	4.3956e+0 (8.30e-1) +
WFG2-15	1.9090e+1 (2.53e+0)	7.1471e-2 (3.49e-2) +	1.5960e+1 (3.52e+0) +
WFG4-3	2.2547e-1 (2.64e-3)	2.7697e-1 (8.36e-3) -	2.2911e-1 (8.38e-3) -
WFG4-8	3.1156e+0 (2.84e-2)	3.7067e+0 (9.05e-2) -	3.081e+0 (1.07e-2) \approx
WFG4-15	9.1102e+0 (4.67e-1)	1.0388e+1 (4.70e-1) -	9.1849e+0 (1.25e-1) -
WFG6-3	2.4813e-1 (8.49e-3)	2.9383e-1 (2.03e-2) -	2.6076e-1 (1.92e-2) -
WFG6-8	3.1376e+0 (2.85e-2)	3.3937e+0 (1.13e-1) -	3.0320e+0 (3.69e-1) +
WFG6-15	9.1245e+0 (6.63e-1)	9.9702e+0 (1.04e+0) -	9.4863e+0 (3.99e-1) -
+ / - / \approx		2/7/0	4/4/1

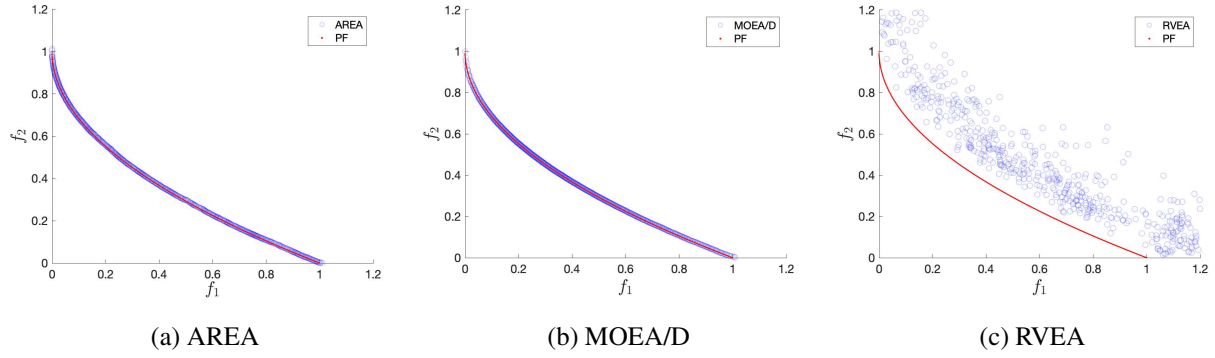


Figure 16: PF approximations obtained by three algorithms for UF1.

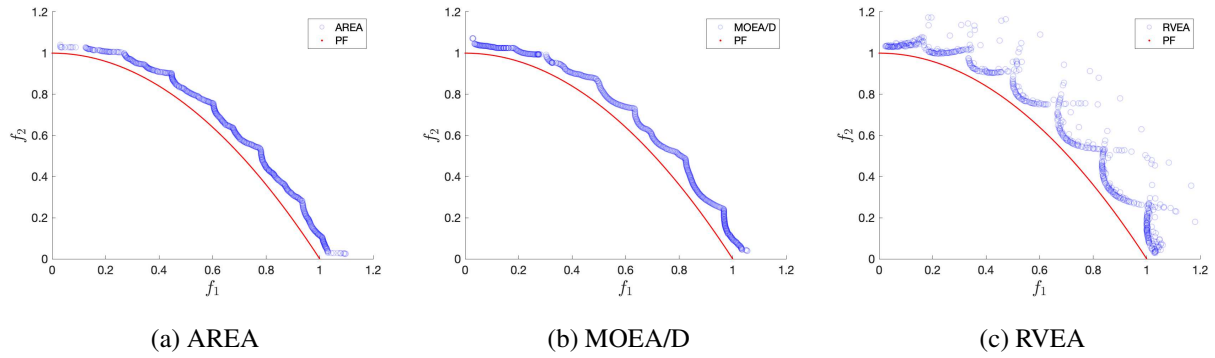


Figure 17: PF approximations obtained by three algorithms for UF4.

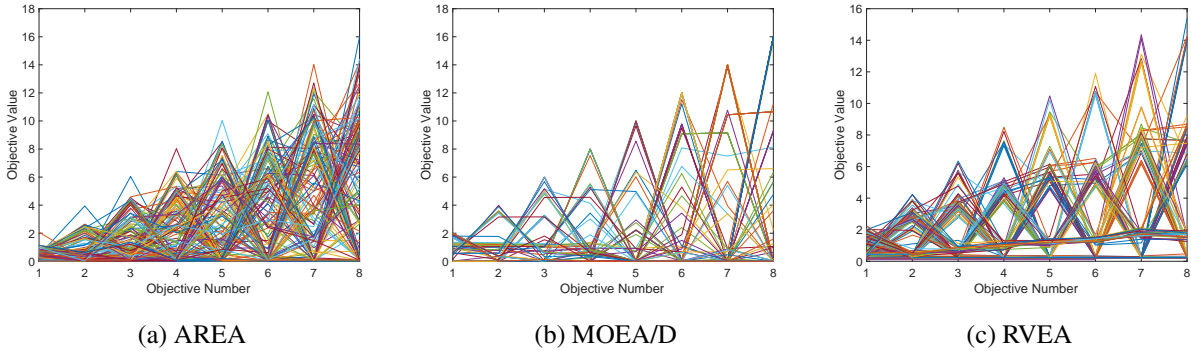


Figure 18: Plots of parallel coordinates of solutions obtained by three algorithms on WFG4 with 8 objectives.

AREA also shows much better IGD values on the MOP problems than the other two algorithms. This demonstrates that AREA works well for both irregular and regular PF shapes. This highlights AREA is not designed to be the best for all types of problems, but should work well on a wide range of problems.

5.4. Comparison with preference articulation approaches

We investigate whether the proposed algorithm can find a preferred region of interest (ROI) or not. We thus compare AREA with three specialist algorithms, which are g-NSGA-II [23], r-NSGA-II [27] and WV-MOEAP [39]. Parameters in these algorithms are configured according to [26]. The ROI is specified through the definition of preferences, and here we focus on the ROI that dominates (is dominated by) a preference vector if the preference vector is inside

Table 12: Comparison with reference adaptation approaches. Mean (and standard deviation) values of IGD are reported. Algorithms that are better than, worse than, or equivalent to AREA on ranksum test with a significance level of 0.05 are indicated by +, -, or \approx , respectively.

Problem	M	RVEA*	MOEA/D-AWA	AREA
DTLZ1	3	2.0124e-2 (5.17e-4) \approx	1.9435e-2 (4.76e-4) +	2.0303e-2 (5.59e-4)
DTLZ2	3	5.1207e-2 (3.43e-4) +	5.0914e-2 (2.40e-4) +	5.2651e-2 (4.95e-4)
DTLZ5	3	8.4923e-3 (4.08e-3) -	4.1110e-3 (6.12e-5) +	4.1568e-3 (9.44e-5)
DTLZ7	3	8.4604e-2 (9.12e-2) -	8.5103e-2 (8.90e-2) -	5.6225e-2 (1.47e-3)
IDTLZ1	3	2.3392e-2 (2.25e-3) -	2.0013e-2 (2.79e-4) +	2.1485e-2 (2.05e-3)
IDTLZ2	3	6.4158e-2 (1.70e-3) -	5.1785e-2 (3.82e-4) +	5.2069e-2 (5.01e-4)
SDTLZ2	3	1.2813e-1 (2.66e-3) -	1.1660e-1 (9.65e-4) +	1.1792e-1 (1.25e-3)
CDTLZ2	3	4.9651e-2 (2.62e-3) -	3.2615e-2 (5.48e-4) \approx	3.3358e-2 (7.83e-4)
MOP1	2	1.2414e-1 (9.37e-2) -	2.0304e-2 (3.01e-2) -	1.0964e-2 (2.65e-4)
MOP2	2	2.9402e-1 (5.93e-2) -	1.5528e-1 (9.17e-2) -	1.1691e-1 (8.64e-2)
MOP3	2	4.6595e-1 (2.92e-2) -	3.7836e-1 (1.33e-1) -	1.3660e-2 (3.33e-2)
MOP4	2	2.4221e-1 (2.31e-2) -	1.9705e-1 (4.64e-2) -	2.1709e-2 (3.37e-2)
MOP5	2	1.2436e-1 (1.21e-1) -	2.7638e-1 (3.88e-2) -	1.5858e-2 (7.78e-4)

Table 13: Comparison with preference articulation approaches. Mean (and standard deviation) values of normalised HV are reported.

Problem	g-NSGA-II	r-NSGA-II	WV-MOEAP	AREA
DTLZ1	0.0000e-1 (0.00e-2)	4.0613e-1 (1.64e-2)	3.9252e-1 (3.93e-5)	3.9067e-1 (3.15e-2)
DTLZ2	1.5951e-1 (1.98e-3)	1.2584e-1 (3.85e-3)	1.2995e-1 (3.73e-5)	1.2422e-1 (1.24e-2)
WFG5	1.5841e-1 (1.29e-4)	1.2174e-1 (7.45e-3)	1.0667e-1 (2.39e-4)	1.2982e-1 (1.07e-2)
WFG6	1.5647e-1 (1.46e-3)	1.1599e-1 (7.03e-3)	5.1171e-2 (4.48e-2)	1.2613e-1 (8.17e-3)

(outside) the attainable objective space. The preference vector is set to (0.2,0.2,0.2), (0.5,0.5,0.5), (3.0,2.0,1.0), and (3.0,2.0,1.0) for DTLZ1, DTLZ2, WFG5, and WFG6, respectively. All the algorithms were run with a population size of 105 and a stopping criterion of 100,000 function evaluations.

Table 13 presents the comparison results of normalised HV, which is computed from the solutions in ROI only. It can be observed that AREA is neither the best nor the worst in finding the ROI of the four problems. g-NSGA-II is the best for these problems except DTLZ1 for which r-NSGA-II seems more effective. WV-MOEAP obtains good results for some problems but fails for others. This is probably due to the small value for the extent of preference region parameter in its default setting. We believe a larger value for this parameter improves WV-MOEAP significantly in terms of the HV value. The comparison suggests in general specialist algorithms for ROI work better. Nevertheless, AREA can be improved if preferences defined by the decision maker are integrated into its search process.

6. Conclusion

The use of references has great potential to direct EA population toward the true PF. Weight set or its descendants, e.g. reference directions/vectors, is one of the most popular forms of references and has been widely adopted in a range of algorithms, particularly the well-known decomposition-based algorithms, for solving MOPs. While the weight-like form of reference set continues its success in the field of EMO, potential downsides are increasingly recognised. For example, one key downside is that a fixed weight set has difficulty in identifying the complete PF if the PF is shaped irregularly.

Instead of interpreting the reference set as weights, this paper has treated it as a set of targets, placed away from the attainable objective space, for population individuals to move toward. The reference set is adaptively adjusted by archived solutions. A couple of strategies are proposed to coordinate the evolution of the reference set and therefore optimise the reference-set guided search. The proposed framework, called AREA, is examined on a number of problems with a wide variety of properties, along with extensive sensitivity analysis. A comparison with state-of-the-arts shows that AREA is a powerful engine for solving the problems considered in this paper.

The time complexity of AREA is similar to that of MOEA/D, despite that additional computations are periodically required in the reference alternation process. Thus, AREA is computationally comparable to MOEA/D and has competitive runtime. AREA is also robust to problem types, as demonstrated by the experimental results. It works well particularly for problems with irregular PF shapes. It should be noted that AREA may have reduced performance when solving problems that do not need reference adjustment. However, it is often difficult to know in advance whether a fixed reference set is the best for the problem in question. Thus, it is wise to assume that the problem needs reference adjustment first, and then let AREA to decide how much change the reference set needs.

Besides, there is still space to further improve AREA. AREA currently involves several parameters, although they have low sensitivity to problem characteristics studied in this paper. It is desirable to remove as many of these parameters as possible for the benefit of future application. Another improvement could be made to increase the capability to solve the problems that AREA currently works poorly. It will be also interesting to further study the interdependence between the reference set and archive management. All of these will be investigated in our future work.

Acknowledgement

SJ, MK, and NK acknowledge the Engineering and Physical Sciences Research Council (EPSRC) for funding project “Synthetic Portabolomics: Leading the way at the crossroads of the Digital and the Bio Economies (EP/N031962/1)”.

References

- [1] X. Cai, Z. Mei, Z. Fan, A decomposition-based many-objective evolutionary algorithm with two types of adjustments for direction vectors, *IEEE Trans. Cybern.* 48(8) (2018) 2335–2348.

- [2] R. Cheng, Y. Jin, M. Olhofer, B. Sendhoff, A reference vector guided evolutionary algorithm for many-objective optimization, *IEEE Trans. Evol. Comput.* 20(5) (2016) 773–791.
- [3] R. Cheng, M. Li, Y. Tian, X. Zhang, S. Yang, Y. Jin, X. Yao, A benchmark test suite for evolutionary many-objective optimization, *Complex & Intelligent Systems* 3(1) (2017) 6781.
- [4] I. Das, J. Dennis, Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems, *SIAM J. Optimiz.* 8(3) (1998) 631–657.
- [5] K. Deb, S. Agrawwal, A. Pratap, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Trans. Evol. Comput.* 6(2) (2002) 182–197.
- [6] K. Deb, H. Jain, An evolutionary many-Objective optimization algorithm using reference-point based non-dominated sorting approach, Part I: Solving problems with box constraints, *IEEE Trans. Evol. Comput.* 18(4) (2014) 577–601.
- [7] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable test problems for evolutionary multi-objective optimization, in: *Evolutionary Multiobjective Optimization*, London: Springer, 2005, pp. 105–145.
- [8] L.R.C de Farias, P.H.M. Braga, H.F. Bassani, A.F.R. Araujo, MOEA/D with uniformly randomly adaptive weights, in: *Proc. Ann. Conf. Genetic Evol. Comput. (GECCO 2018)*, 2018, in press.
- [9] H. Ge, M. Zhao, L. Sun, Z. Wang, G. Tan, Q. Zhang, C.L.P. Chen, A many-objective evolutionary algorithm with two interacting processes: Cascade clustering and reference point incremental learning. *IEEE Trans. Evol. Comput.* 23(4) (2019) 572–586.
- [10] F. Goulart, F. Campelo, Preference-guided evolutionary algorithms for many-objective optimization, *Inf. Sci.* 329 (2016) 236–255.
- [11] S. Huband, P. Hingston, L. Barone, L. While, A review of multiobjective test problems and a scalable test problem toolkit, *IEEE Trans. Evol. Comput.* 10(2) (2006) 477–506.
- [12] H. Ishibuchi, Y. Sakane, H. Masuda, Y. Nojima, Performance of decomposition-based many-objective algorithms strongly depends on Pareto front shapes, *IEEE Trans. Evol. Comput.* 21(2) (2017) 169–190.
- [13] H. Jain, K. Deb, An evolutionary many-Objective optimization algorithm using reference-point based non-dominated sorting approach, Part II: Handling constraints and extending to an adaptive approach, *IEEE Trans. Evol. Comput.* 18(4) (2014) 602–622.
- [14] S. Jiang, M. Torres, D. A. Pelta, D. Krabben, R. Daniel, J. T. Luzardo, M. Kaiser, N. Krasnogor, Improving microbial strain design via multiobjective optimisation and decision making, in: *Proc. AI for Synthetic Biology (IJCAI)*, 2018.

- [15] S. Jiang, S. Yang, An improved multi-objective optimization evolutionary algorithm based on decomposition for complex Pareto fronts, *IEEE Trans. Cybern.* 46(2) (2016) 421–437.
- [16] S. Jiang, S. Yang, A strength Pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimisation, *IEEE Trans. Evol. Comput.* 21(3) (2017) 329–346.
- [17] S. Jiang, S. Yang, Y. Wang, X. Liu, Scalarizing functions in decomposition-based multiobjective evolutionary algorithms, *IEEE Trans. Evol. Comput.* 22(2) (2018) 296–313.
- [18] M. Li, X. Yao, What weights work for you? Adapting weights for any Pareto front shape in decomposition-based evolutionary multi-objective optimisation, 2018, Available: arXiv:1709.02679
- [19] H. Li, Q. Zhang, Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II, *IEEE Trans. Evol. Comput.* 13(2) (2009) 284–302.
- [20] H. Liu, F. Gu, Q. Zhang, Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems, *IEEE Trans. Evol. Comput.* 18(3) (2014) 450–455.
- [21] H. Liu, X. Li, The multiobjective evolutionary algorithm based on determined weight and sub-regional search, in: *Proc. Congr. Evol. Comput.*, vol. 15, 2009, pp. 361–384.
- [22] Z. Liu, Y. Wang, P. Huang, AnD: A many-objective evolutionary algorithm with angle-based selection and shift-based density estimation, *Inf. Sci.* 509 (2020) 400–419.
- [23] J. Molina, L. V. Santana, A. G. Hernandez-Daz, C. A. C. Coello, R. Caballero, g-dominance: Reference point based dominance for multiobjective metaheuristics, *Eur. J. Oper. Res.* 197(2) (2009) 685–692.
- [24] T. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, J. Wu, MOEA/D with adaptive weight adjustment, *Evol. Comput.* 22(2) (2014) 231–264.
- [25] S. Rostami, D. O'Reilly, A. Shenfield, N. Bowring, A novel preference articulation operator for the evolutionary multi-objective optimisation of classifiers in concealed weapons detection, *Inf. Sci.* 295 (2015) 494–520.
- [26] S. Rostami, F. Neri, M. Epitropakis, Progressive preference articulation for decision making in multi-objective optimisation problems, *Integrated Computer-Aided Engineering*, 24(4) (2017) 315–335.
- [27] L. B. Said, S. Bechikh, K. Ghedira, The r-dominance: A new dominance relation for interactive evolutionary multicriteria decision making, *IEEE Trans. Evol. Comput.* 14(5) (2010) 801–818.
- [28] H. Sato, Analysis of inverted PBI and comparison with other scalarizing functions in decomposition based MOEAs, *J. Heuristics* 21(6) (2015) 819–849.
- [29] J. R. Schott, Fault tolerant design using single and multicriteria genetic algorithm optimization, M.S. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1995.

- [30] Y. Tian, R. Cheng, X. Yang, F. Cheng, Y. Jin, An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility, *IEEE Trans. Evol. Comput.* 22(4) (2018) 609–622.
- [31] N. K. Visalakshi, J. Suguna, K-means clustering using Max-min distance measure, in: *Proc. Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS 2009)*, IEEE, 2009, pp. 1–6.
- [32] R. Wang, H. Ishibuchi, Y. Zhang, X. Zheng, T. Zhang, On the effect of localized PBI method in MOEA/D for multi-objective optimization, in: *Proc. IEEE Symposium Series on Computational Intelligence*, 2016.
- [33] Z. Wang, Y.S. Ong, H. Ishibuchi, On scalable multiobjective test problems with hardly dominated boundaries, *IEEE Trans. Evol. Comput.* 23(2) (2019) 217–231.
- [34] F. Wilcoxon, Individual comparisons by ranking methods, *Biometrics* 1(6) (1945) 80–83.
- [35] D. H. Wolpert, W. G. Macready, No free lunch theorems for optimization, *IEEE Trans. Evol. Comput.* 1(1) (1997) 67–82.
- [36] M. Wu, K. Li, S. Kwong, Q. Zhang, J. Zhang, Learning to decompose: A paradigm for decomposition-based multiobjective optimization, *IEEE Trans. Evol. Comput.* 23(3) (2019) 376–390.
- [37] B. Yan, Q. Zhao, Z. Wang, J.A. Zhang, Adaptive decomposition-based evolutionary approach for multiobjective sparse reconstruction, *Inf. Sci.* 462 (2008) 141–159.
- [38] S. Yang, S. Jiang, Y. Jiang, Improving the multiobjective evolutionary algorithm based on decomposition with new penalty schemes, *Soft Computing* 21(16) (2017) 4677–4691.
- [39] X. Zhang, X. Jiang, L. Zhang, Parameter multiobjective optimization method based on weight vector, *Acta Electronic Journal* 44(11) (2016) 2639–2645.
- [40] C. Zhang, K.C. Tan, L.H. Lee, L. Gao, Adaptive weight vectors in MOEA/D for bi-objective optimization problems with discontinuous Pareto fronts, *Soft Computing*, 22 (2018) 3997–4012.
- [41] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, *IEEE Trans. Evol. Comput.* 11(6) (2007) 712–731.
- [42] Q. Zhang, W. Liu, H. Li, The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances, in: *Proc. Congr. Evol. Comput.*, IEEE, 2009, pp. 203–208.
- [43] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, S. Tiwari, Multiobjective optimization test instances for the CEC 2009 special session and competition, Technical Report CES-487, The School of Computer Science and Electronic Engineering, University of Essex, Tech. Rep., 2008.
- [44] A. Zhou, Q. Zhang, Are all the subproblems equally important? Resource allocation in decomposition-based multiobjective evolutionary algorithms, *IEEE Trans. Evol. Comput.* 20(1) (2016) 52–64.

- [45] E. Zitzler, S. Kunzli, Indicator-based selection in multiobjective search, in: Proc. 8th Int. Conf. Parallel Problem Solving from Nature (PPSN VIII), LNCS, vol. 3242, 2004, pp. 832–842.
- [46] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach, IEEE Trans. Evol. Comput. 3(4) (1999) 257–271.
- [47] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization, in: Proc. Evol. Methods Des. Optimiz. Contr. Applicat. Indust. Problems, EUROGEN, vol. 3242, no. 103, 2002, pp. 95–100.

AREA: An Adaptive Reference-set Based Evolutionary Algorithm for Multiobjective Optimisation

Shouyong Jiang^{a,*}, Hongru Li^b, Jinglei Guo^c, Mingjun Zhong^a, Shengxiang Yang^d, Marcus Kaiser^e,
Natalio Krasnogor^e

^a*School of Computer Science, University of Lincoln, Lincoln LN6 7TS, UK*

^b*College of Information Science and Engineering, Northeastern University, Shenyang, 110819, China*

^c*Department of Computer Science, Central China Normal University, Wuhan, China*

^d*School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, UK*

^e*School of Computing, Newcastle University, Newcastle upon Tyne NE4 5TG, UK*

Abstract

This supplementary material provides a number of additional plots resulting from performance analysis of the proposed approach. These plots illustrate the comparison between different algorithms and demonstrate the effectiveness of the proposed approach.

*Corresponding author

Email address: math4neu@gmail.com (Shouyong Jiang)

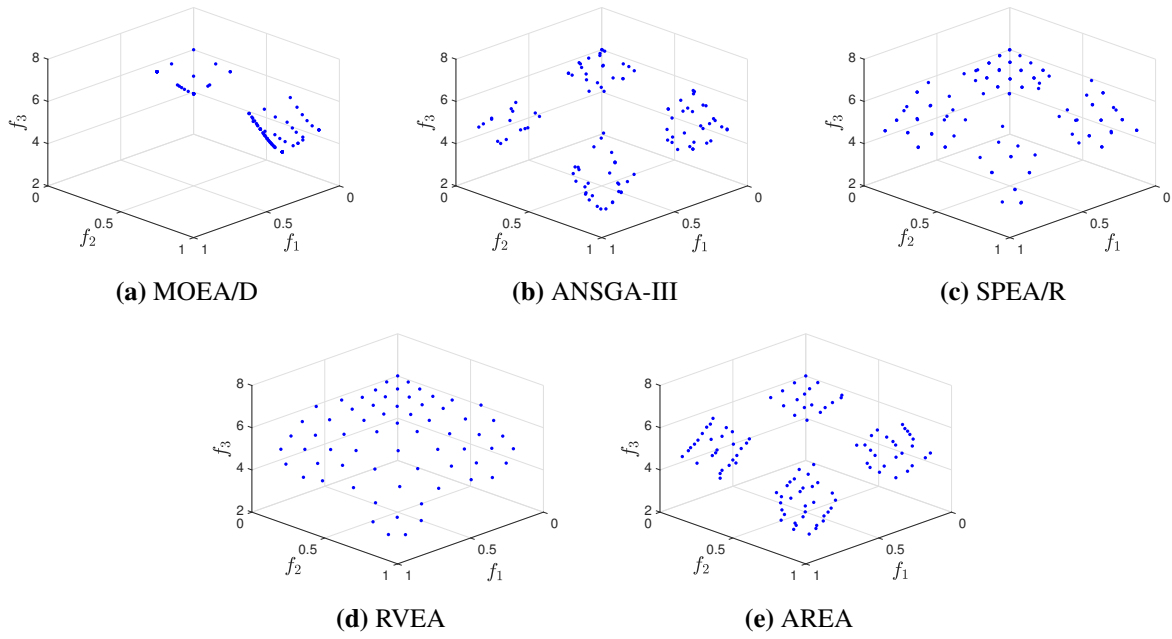


Figure 1: PF approximation of DTLZ7 obtained by different algorithms.

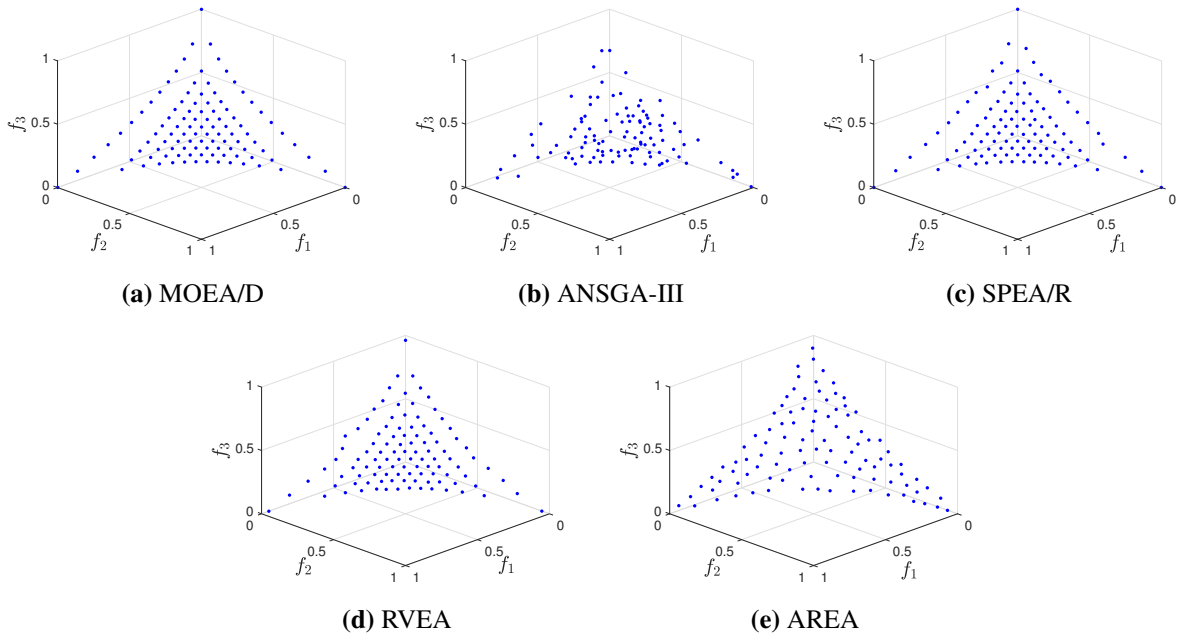


Figure 2: PF approximation of CDTLZ2 obtained by different algorithms.

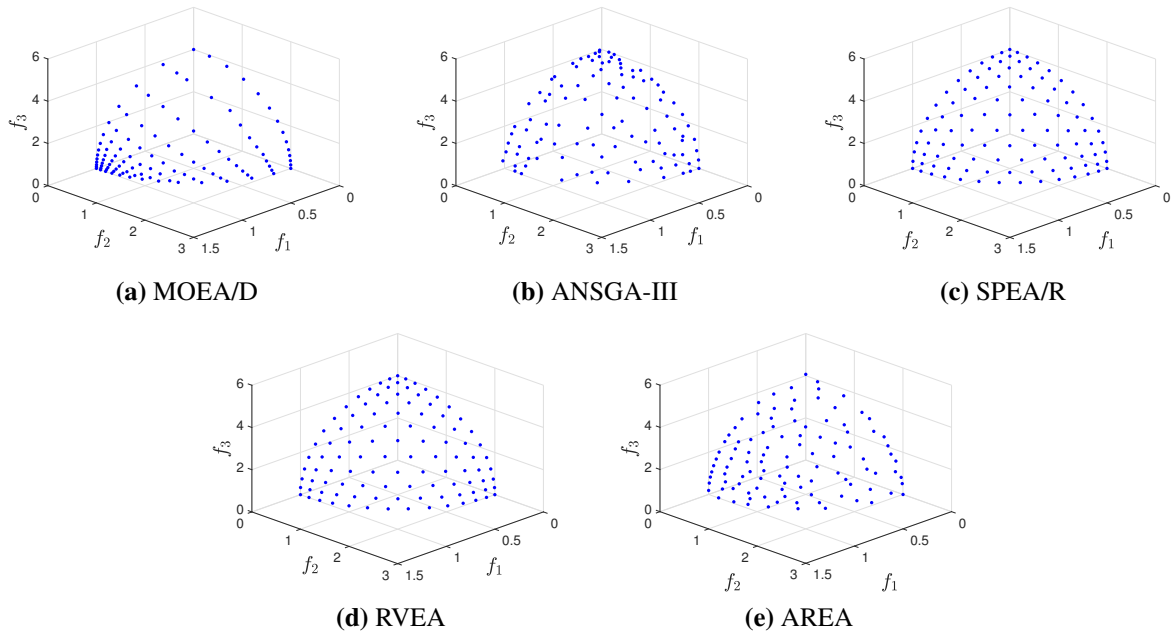


Figure 3: PF approximation of SDTLZ2 obtained by different algorithms.

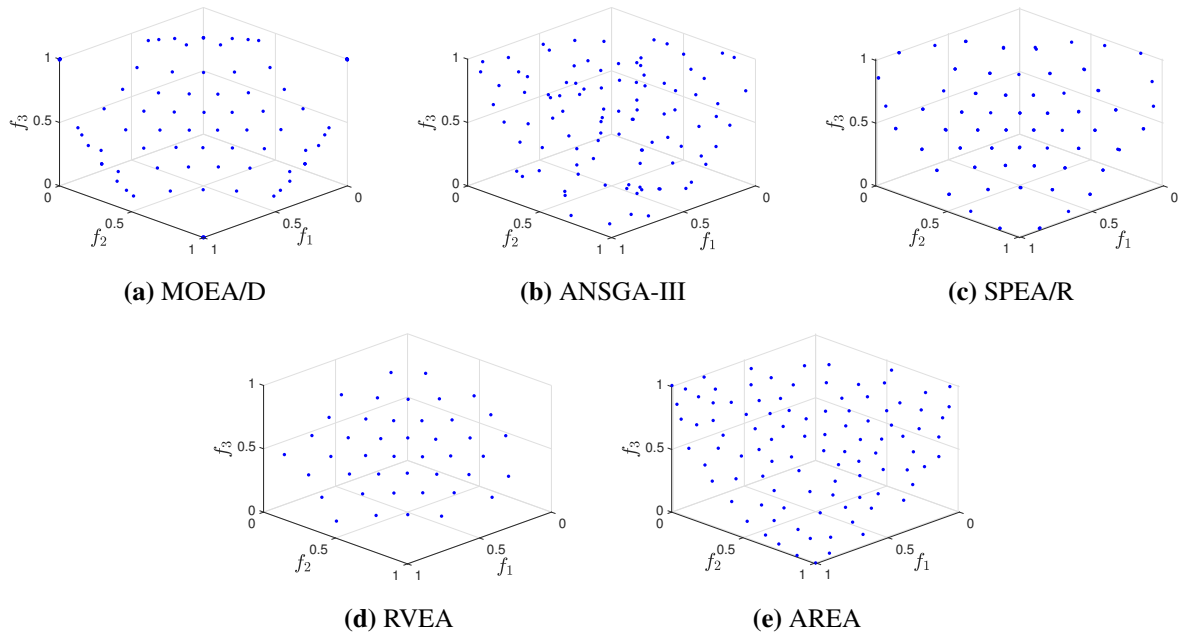


Figure 4: PF approximation of IDTLZ2 obtained by different algorithms.

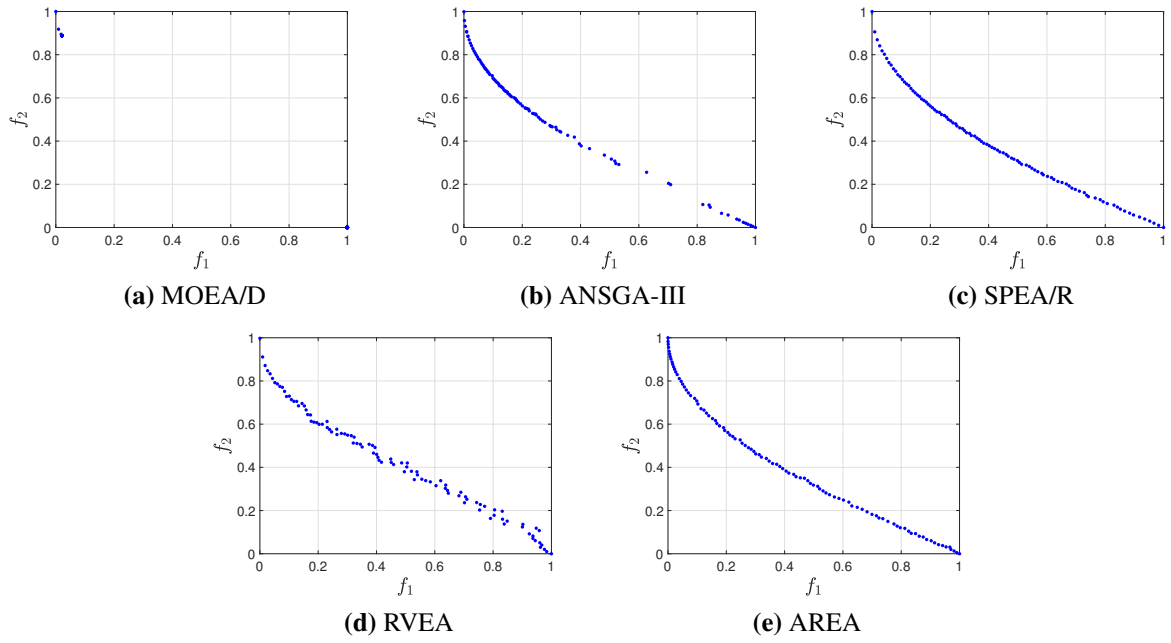


Figure 5: PF approximation of MOP1 obtained by different algorithms.

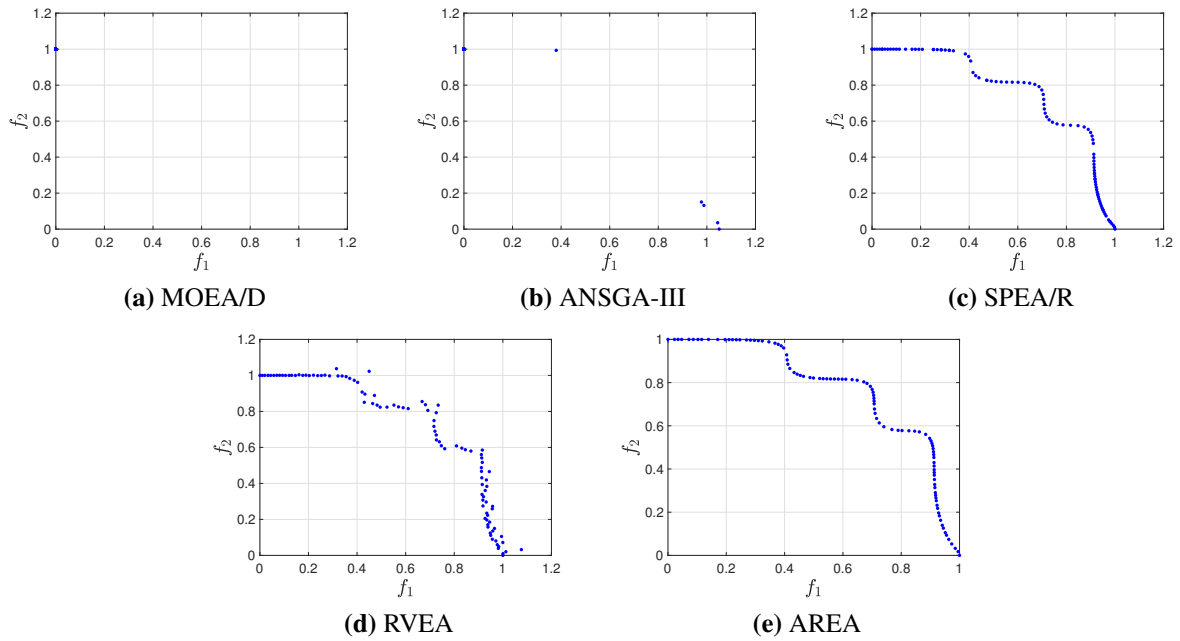


Figure 6: PF approximation of F4 obtained by different algorithms.

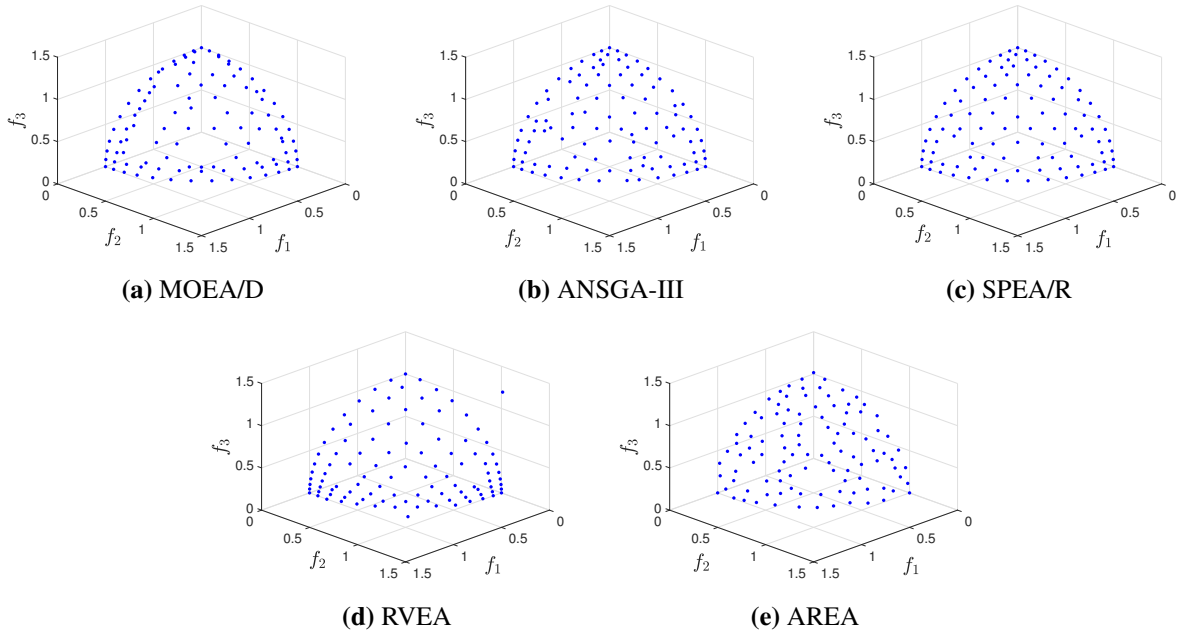


Figure 7: PF approximation of F5 obtained by different algorithms.

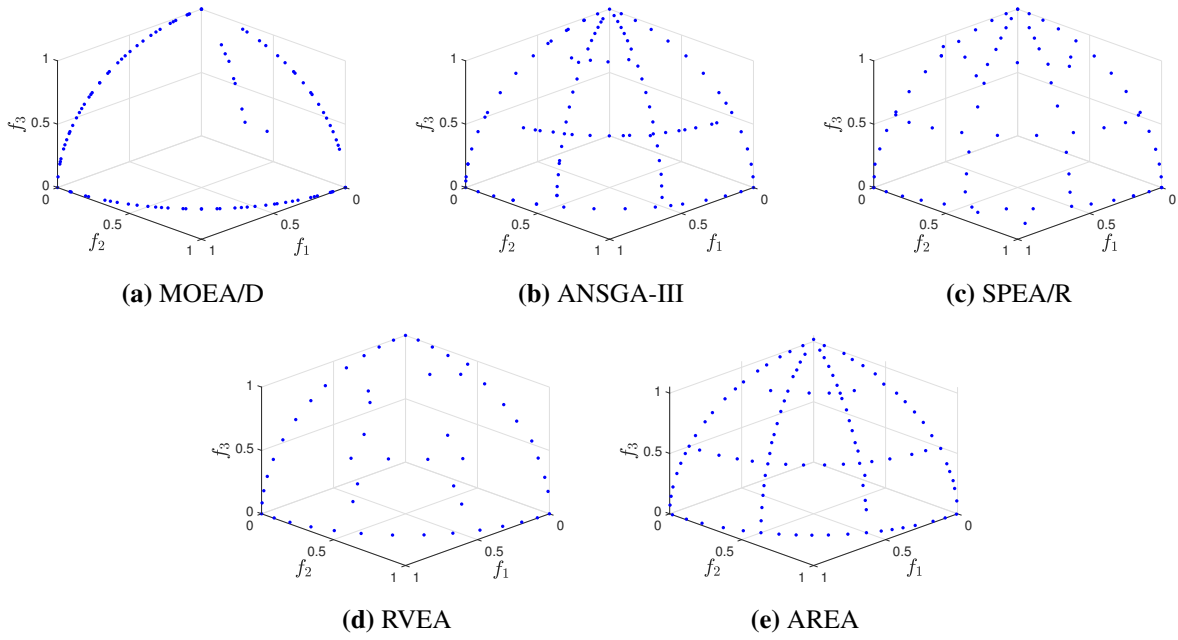


Figure 8: PF approximation of F6 obtained by different algorithms.

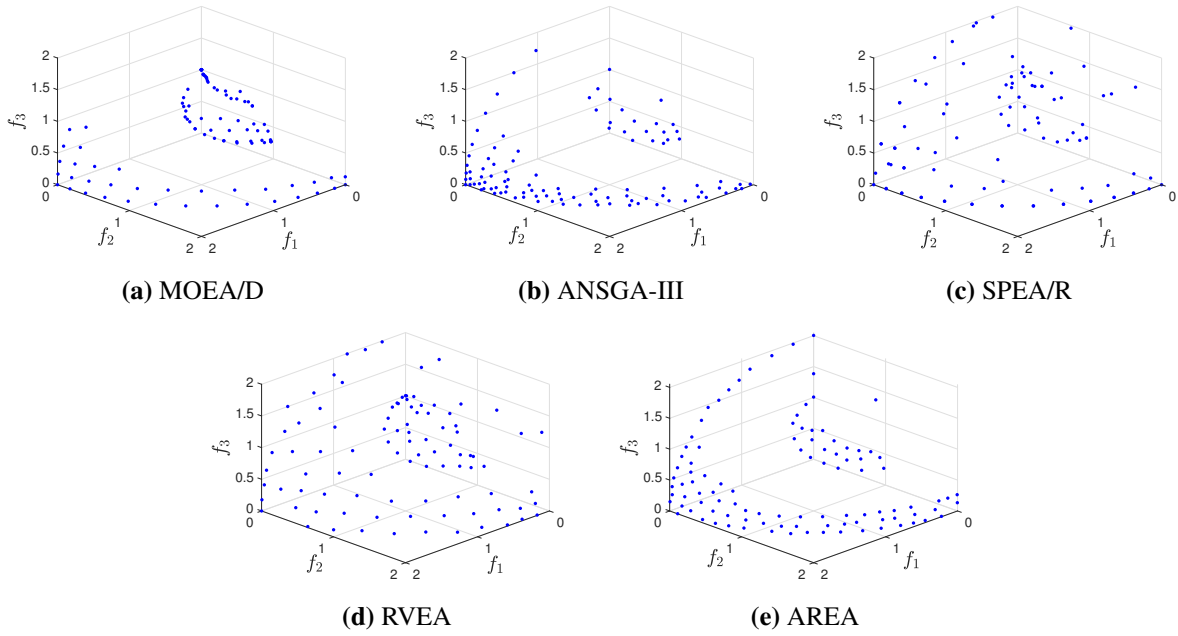


Figure 9: PF approximation of F7 obtained by different algorithms.