Adaptation of Reference Vectors for Evolutionary Many-objective Optimization of Problems with Irregular Pareto Fronts

Qiqi Liu, Yaochu Jin, Martin Heiderich, Tobias Rodemann

2019

Preprint:

This is an accepted article published in Proceedings of IEEE Conference on Evolutionary Computation (CEC). The final authenticated version is available online at: https://doi.org/ 10.1109/CEC.2019.8790214

Adaptation of Reference Vectors for Evolutionary Many-objective Optimization of Problems with Irregular Pareto Fronts

Qiqi Liu, Yaochu Jin
Department of Computer Science,
University of Surrey,
Guildford, Surrey, GU2 7XH, UK.
Email: qiqi.liu,yaochu.jin@surrey.ac.uk

Martin Heiderich
Honda R&D Europe (Deutschland) GmbH,
Carl-Legien-Strasse 30,
D-63073 Offenbach/Main, Germany.
Email: martin_heiderich@de.hrdeu.com

Tobias Rodemann
Honda Research Institute Europe,
Carl-Legien-Strasse 30,
D-63073 Offenbach/Main, Germany.
Email: tobias.rodemann@honda-ri.de

***Abstract*—For problems with irregular Pareto fronts, only part of the objective space is covered by optimal solutions. Most decomposition based evolutionary many-objective algorithms, however, predefine uniformly distributed weight or reference vectors, making them less suited for problems with irregular Pareto fronts, since many weight or reference vectors will be wasted. To address the above issue, this paper proposes a variant of the reference vector guided evolutionary algorithm by adjusting reference vectors according to the distribution of the solutions in the current population to make sure that most reference vectors are associated with solutions. A secondary selection criterion based on the dominance relationship is adopted in addition to the angle penalized distance based selection so that a sufficient number of solutions can survive and be passed to the next generation. Experiments on 12 irregular test problems with 60 instances show that the proposed algorithm is competitive compared to the state-of-the-art algorithms for solving problems with irregular Pareto fronts.**

***Index Terms*—Evolutionary many-objective optimization, reference vector, irregular Pareto front**

I. Introduction

Many real-world optimization problems have more than three objectives, which are known as many-objective optimization problems (MaOPs). Over the past decade, a large number of evolutionary algorithms have been developed for solving MaOPs, which can be generally categorized into three groups. Since the original Pareto dominance relationship becomes less suited for distinguishing solutions when the number of objectives is large, the first group aims to propose a modified dominance relationship, like fuzzy dominance [1], α-dominance [2], grid dominance [3], L-dominance [4], and θ-dominance [5]. The second group is known as performance indicator based algorithms. For instance, the epsilon indicator based algorithm [6] and the hypervolume (HV) indicator based algorithm [7] take both convergence and diversity into consideration, which are able to achieve a set of non-dominated solutions regardless of the number of objectives. However, calculation of performance indicators, e.g., the hypervolume, will become computationally rather expensive as the number of objectives increases. Therefore, fast and simple hypervolume calculation methods have been proposed [8]. Decomposition based algorithms shall belong to the third category. Two kinds of decomposition strategies are widely used in decomposition based algorithms. One is to convert an MaOP into a number of single-objective optimization problems using a set of weights [9], reference points [10], or reference vectors [11], while the other is to decompose an MaOP into a number of multi-objective optimization problems [12].

Most decomposition based algorithms are based on the assumption that the Pareto front (PF) has a regular geometrical structure, i.e., the PF is smooth, continuous, and well distributed over the whole objective space. Irregular problems have discontinuous, degenerated or inverted Pareto fronts, which only cover part of the objective space. So decomposition based algorithms become less efficient when they are used to solve irregular problems, since not all weight or reference vectors are used to guide the search of solutions. In the reference vector guided evolutionary algorithm (RVEA) [11], every individual is associated with the nearest reference vector based on the angle between reference vectors and individuals in the population. So it is very likely that multiple individuals are associated with one reference vector. As a result, it can happen that some reference vectors have no associated solutions. In this case, an evenly distributed solution set cannot be guaranteed.

In RVEA, a reference vector is called active if at least one solution is associated with it. Otherwise, the reference vector is inactive. To tackle problems with irregular PFs, in the original publications [11], a variant of RVEA, termed RVEA$*$, was proposed, where inactive reference vectors are replaced by random vectors generated in the whole objective space according to the range of solutions in the population. Unfortunately, randomly regenerated reference vectors may fail to keep promising solutions in less well explored regions, leading to the loss of whole segments of a discontinuous Pareto front. To address this problem, this study proposes an improved version of RVEA, termed iRVEA for MaOPs with irregular Pareto fronts. Main contributions of this study can be summarized as follows.

- The whole objective space is divided into a small number of large subspaces so that at least one solution is selected in each subspace no matter whether a solution is dominated or not to prevent the loss of whole segments of the PF.
- Besides the predefined reference vector set, we generate another set of adaptive reference vectors. In each generation, one inactive reference vector in the adaptive reference vector set is replaced by a new one defined by a solution in the population, which has the least similarity to current active reference vectors in both the predefined and adaptive reference vectors.
- In contrast to the loss of whole Pareto segments, many solutions within a segment of the PF are lost due to the fact only a small number of reference vectors exist in a particular region of the objective space. This can be attributed to the fact that the angle penalized distance (APD) based selection is insufficient. To fix this issue, we use Pareto dominance as an additional selection criterion for problems having up to six objectives, and the strengthened non-dominance relationship (SDR) [13] for problems with more than six objectives to keep promising solutions.

The remainder of this paper is organized as follows. Section II briefly discusses the related work on tackling problems with irregular PFs and describes the details of RVEA. The proposed algorithm and empirical results are presented in Section III and Section IV, respectively. Lastly, conclusions are summarized in Section V.

II. Background

A. Related Work

Existing evolutionary algorithms dealing with irregular problems can be divided into two kinds, i.e., reference vector adaption based and clustering based algorithms. It is intuitive to adjust the reference vectors during the course of evolution for weight vector and reference vector based algorithms when most predefined reference vectors have no solutions associated with them, or many solutions are associated with one vector. In [14], a reference vector will be regenerated if the number of solutions it is associated with is the largest among all reference vectors in the exploration stage, while a reference vector associated with the least number of solutions will be regenerated in the exploitation stage. In [15], Shannon's entropy is used to judge whether reference vectors need to be adjusted and maximum entropy is achieved when only one reference vector is associated with one solution in the population. In [16], reference vectors in each subproblem are adjusted according to the distribution of solutions in the population. NSGA-III [10] was extended for handling irregular problems in [17]. A recently proposed algorithm [18] adjusts its reference vectors based on collected information about feasibility and non-dominance of solutions associate with the reference vectors over a learning period. Clustering based methods for handling irregular problems have also been shown to be effective since they guide the search based on the distribution of the individuals instead of a set of predefined weight or reference vectors. For irregular problems, it is likely that the distribution of individuals changes frequently during the course of evolution. Thus, clustering based algorithms are able to select individuals from different clusters, thereby preventing promising solutions from getting lost. In [19], the Pareto dominance relationship is used as first selection criterion and individuals in the last front are selected based on clusters, making it suited for handling problems with irregular PFs. In [20], a set of cluster centers are adaptively generated using hierarchical clustering for guiding selection at each generation and show competitive performance for multi-objective optimization. Different from [19] and [20], an acute angle is used in [21] as the similarity metric in clustering instead of the Euclidean distance, which adopts partitional clustering to divide the whole population into M clusters at first (M is the number of objectives) and then uses hierarchical clustering to cluster the whole population into N final clusters (N is the population size). There are also some algorithms which do not belong to these two classes. A set of reference points are adaptively adjusted according to the contribution of candidate solutions in an external archive in terms of IGD-NS [22]. In [23], a 'maximum-vector-angle-first' principle and a 'worst-elimination' principle are adopted to guarantee the convergence and diversity, in which reference vectors are not needed. Most recently, incremental learning is used to train reference vectors [24].

B. RVEA

RVEA can be seen as a decomposition based algorithm, which uses a set of reference vectors for maintaining diversity. Reference vectors in RVEA can also be used for articulating user preferences [25]. RVEA adopts the angle between solutions and reference vectors for diversity maintenance and therefore uses the following angle penalized distance for selection to balance convergence and diversity:

$$d_{t,i,j} = (1 + P(\theta_{t,i,j})) \cdot \left\| f'_{t,i} \right\| \tag{1}$$

where $P(\theta_{t,i,j})$ measures the diversity and $\left\| f'_{t,i} \right\|$ measures the convergence. $P(\theta_{t,i,j})$ is a penalty function related to $\theta_{t,i,j}$. $\theta_{t,i,j}$ measures the angle between solution $f'_{t,i}$ and reference vector $v_{t,j}$, t is the generation number.

$$P(\theta_{t,i,j}) = M \cdot (\frac{t}{t_{max}})^{\alpha} \cdot \frac{\theta_{t,i,j}}{\gamma_{v_{t,j}}} \tag{2}$$

and

$$\gamma_{v_{t,j}} = min\langle v_{t,i}, v_{t,j} \rangle, i \in \{1, \cdots, N\} \tag{3}$$

where M is the number of objectives, $\gamma_{v_{t,j}}$ is the smallest angle between all pairs of reference vectors which is used to normalize the angle between reference vectors. In the early stage of the search, the convergence should be emphasized much more than diversity. Parameter α controls the balance between diversity and convergence so that more emphasis is given to convergence in the early search stage while diversity becomes more important in the later stage.

III. Proposed Algorithm

In this section, details of the proposed algorithm, iRVEA, will be given. The overall framework of iRVEA is first presented, followed by its four main components.

A. Overall Framework

The main loop of iRVEA is presented in Algorithm 1. Two new components of iRVEA are the replacement of the inactive reference vectors and environmental selection.

Algorithm 1 Overall Framework of iRVEA

Input: Population size N, population P, maximum generation $MaxGen$, a set of fixed direction vectors $V = \{\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_N\}$, a set of adaptive reference vectors $V^{'} = \{\boldsymbol{\lambda}_1^{'}, \ldots, \boldsymbol{\lambda}_N^{'}\}$, archive A

Output: final population P_{tmax}

```
Initialization: create the initial population P_o with N randomized individuals; A = ∅
while t < MaxGen do
    Q_t = Offspring_Creation(P_t)
    P_t = P_t ∪ Q_t
    V' = Replace_Inactive_Reference_Vectors (P_t, V, V')
    P_{t+1} = Environmental_Selection (t, P_t, V', V)
    V = Reference_Vector_Adaption(t, P_{t+1}, V)
    A = Update_Archive(A, P_{t+1})
    t = t + 1
end while
```

In iRVEA, we replace at most one inactive reference vector in the adaptive reference vector set and those in the predefined reference vector set are kept unchanged. In environmental selection of RVEA$*$, solutions are selected only based on the angle penalized distance (APD). We add a secondary selection criterion based on dominance relationship in addition to APD to ensure that all promising solutions will be selected. Moreover, we use an archive to preserve solutions based on unary additive epsilon indicator [6] in each generation.

B. Preventing Loss of Whole Segments of a PF

When a PF is strongly non-uniform or of mixed shape, it is very likely that optimal solutions in some parts of the objective space are easy to be found while solutions in other regions are extremely difficult to locate. If all computational resources focus on the easily found regions, it can happen that only parts of the Pareto fronts are found. In RVEA$*$, all non-dominated solutions are selected firstly and then APD is used to distinguish solutions in environmental selection. If all non-dominated solutions concentrate on parts of the objective space, then reference vectors will be adjusted to these regions. As a result, dominated but promising solutions in less explored regions have little chance to be selected since no reference vectors have been allocated to these regions. This phenomenon usually happens in a low-dimensional objective space, as in high dimensional objective spaces almost all solutions are non-dominated.

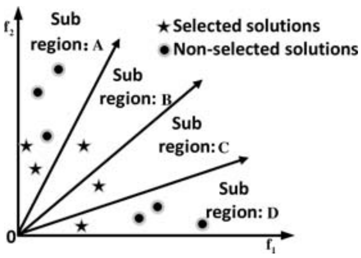


Fig. 1. An illustrative example of preventing the loss of whole segments in a small number of sub regions. In the example, solutions denoted by stars in region A, B, C, D represent solutions being selected.

To address this issue, we divide the entire objective space into a small number of large sub-regions to make sure that promising solutions do not get lost in the early search stage. In Fig. 1, we assume that the whole objective space is divided into four sub regions A, B, C and D, and solutions being selected are denoted by stars and solutions not being selected are denoted by circles. For sub-region A and D, solutions denoted by stars are selected since they belong to the first front according to Pareto dominance. In sub-regions B and C, even though no solutions belong to the first front, the solution which has the smallest Euclidean distance to the ideal point is selected. This way, some certain selection pressure remains in those less explored regions where no reference vectors exist.

C. Replacement of Inactive Reference Vectors

During the search process, active reference vectors can become inactive occasionally even for regular problems. So it is not a good idea to adjust reference vectors frequently as it will slow down the convergence. In iRVEA, we only replace one inactive reference vector at each generation. One inactive reference vector is selected randomly from all inactive reference vectors and is replaced by a new reference vector specified by a solution that has the least similarity to the active reference vectors. The detailed procedure of replacement of inactive reference vectors is summarized in Algorithm 2.

The similarity between a vector and reference vector set is measured by the biggest angle between the vector and all vectors in the reference vector set. At each generation, a solution with the biggest angle to the existing active reference vectors is selected and normalized to replace one randomly chosen inactive reference vector. In Fig. 2, V_1 to V_4 are active reference vectors because they are associated with solutions based on the angle. V_5 is inactive, so we will replace V_5 with a new vector. Θ is the angle between the solution marked with star and reference vector V_4. We find angle Θ is biggest among all pairs of angles between solutions in the population and reference vectors. The solution denoted by star is selected and normalized, that is $V_5^{'}$, to replace the inactive

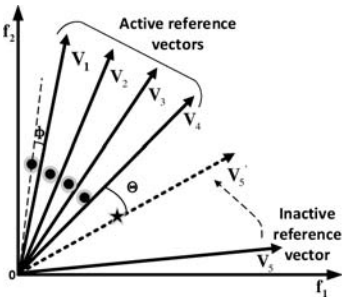


Fig. 2. An illustrative example of replacing an inactive reference vector with a solution vector which has the least similarity to active reference vectors in normalized objective space.

reference vector V_5. When the number of active reference vectors in the population is not sufficient, generating more active reference vectors will be guaranteed. Besides, when the number of active reference vectors is sufficient, selecting a solution which shares the least similarity to the active reference vectors guarantees an even distribution of the active reference vectors, which in turn will encourage the diversity of solutions.

Algorithm 2 Replacement of Inactive Reference Vectors

Input: Population P_t, a set of predefined direction vectors $V = \{\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_N\}$, a set of adaptive reference vectors $V^{'} = \{\boldsymbol{\lambda}_1^{'}, \ldots, \boldsymbol{\lambda}_N^{'}\}$
Output: Adaptive reference vector set $V^{'}$
1: Find a solution P_{ti} which has the least similarity from active reference vectors in both predefined and adaptive reference set and normalize it.
2: Choose an inactive reference vector V_{ran} randomly and replace it by P_{ti}

D. APD and Dominance Based Environmental Selection

The APD selection criterion may not be able to select all promising solutions since the adjustment of reference vectors may not match well with the distribution of solutions. So we use a secondary selection criterion to select more promising solutions on top of the APD selection. The detail of environmental selection is presented in Algorithm 3. After APD based selection, hierarchical clustering is used to cluster solutions into N clusters and one solution is picked up from each cluster which has the smallest Euclidean distance to the ideal point when the number of solutions selected by APD is more than N. Ideal point is defined as the best value for each objective. If the solutions selected by APD cannot fill the population, then we determine whether more solutions should be selected. For problems with up to six objectives, we use Pareto dominance to select the rest of solutions. Since Pareto dominance is no longer suitable when the number of objectives is large, we use SDR, which works well for MaOPs, is adopted as a secondary selection criterion for problems with more than six objectives. In the late stage of evolution, if solutions selected by APD cannot fill the population because the number of active reference vectors are insufficient, then $I_{\epsilon+}$ indicator [6] is adopted to select solutions from the population to make sure that the number of solution being selected to next generation

Algorithm 3 Environmental Selection

Input: Population P_t, population size N, generation number t, max generation number $MaxGen$, Archive A
Output: Final population P
1: $P_t \leftarrow Fast\_Non-dominated\_Sort(P_t)$
2: **if** $M \leq 3$ **then**
3: $P_t \leftarrow$Prevent Loss of Whole Segments of a PF
4: **end if**
5: $P_{APD} \leftarrow$ Selection Population by using APD criterion
6: $P_{left} \leftarrow P_t - P_{APD}$
7: $SN \leftarrow size(P_{APD})$;$LN \leftarrow size(P_{left})$;$kappa = 0.05$
8: **if** $SN < N \,\&\&\, t > 0.8 \cdot MaxGen$ **then**
9: $[Fit, I, C] \leftarrow$ Calculate fitness of solutions in P_{left} using $I_{\epsilon+}$
10: **for** $i = 1\, to\, LN - N$ **do**
11: $[\sim, d] = min(Fit, [], 2)$
12: $Fitness(d) = 0$
13: **for** $j = 1\, to\, LN$ **do**
14: **if** $Fit(j) \neq 0$ **then**
15: $Fit(j) = Fit(j) + exp(-I(d,j)/C(j) \cdot kappa$
16: **end if**
17: **end for**
18: $P_{left}(d) = \emptyset$;$Fitness(d) = \emptyset$
19: **end for**
20: $P_{t+1} = P_{APD} \cup P_{left}$
21: **else**
22: **if** $SN > N$ **then**
23: $P_{t+1} \leftarrow$ Use hierarchical clustering to delete $SN - N$ solutions in P_{APD}
24: **else**
25: **if** $M > 6$ **then**
26: $P_{t+1} \leftarrow$ Use SDR to select first front solutions from $P_{APD} \cup A$
27: **else**
28: $P_{t+1} \leftarrow$ Use Pareto dominance to select first front solutions from $P_{APD} \cup A$
29: **end if**
30: $P_{t+1} \leftarrow$ Delete crowded solutions in P_{t+1} to make sure the number of solutions is at most N
31: **end if**
32: **end if**

equals to N, which is described from Line 9 to Line 21 in Algorithm 4. At last, if the number of the individuals selected by APD and the secondary selection criterion is more than population size N, the most crowded solutions in terms of angle are deleted one by one to make sure that the number of solutions do not exceed N.

E. Update Archive

To support environmental selection, an archive is created. The archive needs to be updated at each generation and the size of archive is equal to the population size. In our proposed algorithm, we use $I_{\epsilon+}$ indicator to decide which solutions shall be selected. The details of updating archive are presented in Algorithm 4. When the size of archive is exceeded, the individual with the worst fitness value will be discarded. The fitness of other individuals in the combined population will be recalculated, which is shown in Line 10 in Algorithm 4. This way, the size of archive will not exceed N.

IV. Empirical Results

In this section, the performance of the proposed algorithm is examined on 12 benchmarks with irregular PFs, i.e., MaF1 to

Algorithm 4 Update Archive

Input: Population P_{t+1}, population size N, Archive A
Output: Archive A

```
Q = P_{t+1} ∪ A
QN = size(Q); kappa = 0.05
[Fit, I, C] ← Calculate fitness of solutions in Q using I_{ε+}
for i = 1 to QN − N do
    [∼, d] = min(Fit, [], 2)
    Fitness(d) = 0
    for j = 1 to QN do
        if Fit(j) ≠ 0 then
            Fit(j) = Fit(j) + exp(−I(d, j)/C(j) · kappa
        end if
    end for
    Q(d) = ∅; Fitness(d) = ∅
end for
A = Q
```

MaF13 except for MaF2, MaF5, and MaF12 [26] and IDTLZ1 and IDTLZ2 [17]. Two_Arch2 [27], VaEA [23], EMyO/C [19], AR-MOEA [22], RVEA$*$ [11] are adopted to compare with our proposed algorithm. All algorithms are implemented in PlatEMO platform [28] with MATLAB 2017a on Intel Core i7-7700 (2.80GHz).

A. Performance indicators

In this paper, we adopt IGD [29] and HV [8] as the performance indicators to evaluate the performance of algorithms. IGD and HV are believed to be able to account for both convergence and diversity at the same time. IGD calculates the average Euclidean distance between reference points sampled from the real Pareto front and solutions in the population. A smaller IGD value means better performance of algorithm. As the computational complexity of HV is too expensive, a fast way for calculating HV is used in this paper [8]. All reference points used to calculate HV are set to 1.1 times worst value of each objective of Pareto optimal solutions. A higher HV value means better performance. IGD is calculated as follows:

$$IGD\left(P,P^{*}\right)=\frac{\sum_{\mathbf{x}\in P^{*}}d\left(\mathbf{x},P\right)}{\left|P^{*}\right|} \tag{4}$$

where P^* and P represent reference points in the real PFs and solutions achieved by algorithms, respectively, and $d(x,P)$ means the Euclidean distance between x and P. HV can be calculated in the following way:

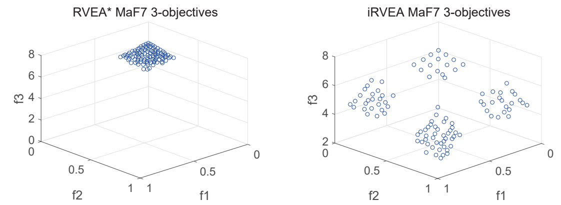
$$HV\left(P,\mathbf{r}\right)=L\left(\underset{\mathbf{x}\in P}{\cup}\left[f_1\left(\mathbf{x}\right),r_1\right]\times\cdots\times\left[f_M\left(\mathbf{x}\right),r_M\right]\right) \tag{5}$$

where $r_1,\cdots,r_M$ is a reference point and $f_1(x),\cdots,f_M(x)$ is a solution x in the Population P.

B. Experimental Settings

To validate the performance and behaviour of the proposed iRVEA, the following two sets of experiments are conducted.

- Comparative studies are conducted to investigate the effectiveness of the strategy for preventing the loss of whole segments of a PF using a small number of sub-regions in a low-dimensional objective space.
- iRVEA is compared with five state-of-the-art algorithms in dealing with MaF [26] and IDTLZ [17] benchmark suites. The performance of iRVEA in solving problems with irregular PFs can be examined by comparing the IGD and HV value.



(a) Solutions obtained by RVEA$*$ on MaF7 using the worst run. (b) Solutions obtained by iRVEA on MaF7 using the worst run.

Fig. 3. Illustration of solutions obtained by RVEA$*$ and iRVEA on MaF7 with 3 objectives using the worst run of IGD value.

The number of objectives, population size, and the number of fitness evaluations is set be $\langle 3,105,60000\rangle$, $\langle 6,132,100000\rangle$, $\langle 8,156,150000\rangle$, $\langle 10,275,200000\rangle$, $\langle 15,135,300000\rangle$, respectively. For problems with 3 objectives, the number of sub region is set to 40. The simulated binary crossover (SBX) [30] and polynomial mutation [31] are adopted in iRVEA with a crossover probability $P_c = 1.0$, a mutation probability $P_m = 1/D$, distribution index of crossover $n_c = 20$ and distribution index of mutation $n_m = 20$, respectively.

C. Results and Discussions

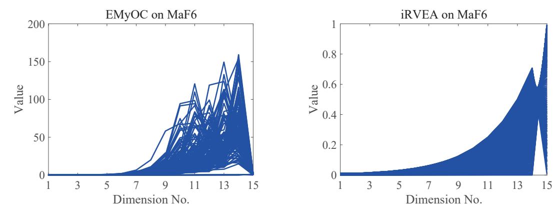
We surmise that only part of the PF can be found by RVEA* because the reference vectors will be allocated to a small region where the non-dominated solutions agglomerate. This can be confirmed by the different solution sets obtained by RVEA* and iRVEA on MaF7, as shown in Fig. 3, where the worst results among 20 independent runs are plotted. It can be seen in Fig. 3(a), only one of four segments of PF has been found by RVEA$*$. In Fig. 3(b), we can see that iRVEA is able to find all segments of the PF even in the worst run. These results demonstrate the effectiveness of the proposed strategy for preventing loss of whole segments of a PF.

Table I and Table II list the IGD and HV values of the solutions obtained by iRVEA and other five state-of-the-art MOEAs proved to perform well in tackling problems with irregular PFs with 3 to 15 objectives. Overall, iRVEA performs the best on 24 of 60 instances in terms of IGD, while the number achieved by EMyO/C, Two_Arch2, AR-MOEA, VaEA and RVEA$*$ is 11, 15, 6, 2 and 2, respectively. In terms of HV, the number of best performance obtained by iRVEA is 15 out of 60 instances and for other five compared algorithms, the number is 15, 10, 11, 6, and 3, respectively. From the results, it can be found iRVEA has the best performance overall, while RVEA$*$ achieves the worst. Moreover, we find iRVEA

TABLE I

IGD COMPARISONS AND FRIEDMAN TEST OF THE SIX ALGORITHMS ON ALL TEST INSTANCES. THE BEST RESULT ON EACH TEST INSTANCE IS HIGHLIGHTED IN BOLDFACE AND BLUE. "+", "−", AND "=" DENOTE THAT THE COMPARED ALGORITHM PERFORMS SIGNIFICANTLY BETTER THAN, WORSE THAN, AND EQUIVALENT TO IRVEA, RESPECTIVELY.

Problem	M	EMyO/C	Two_Arch2	AR-MOEA	VaEA	RVEA∗	iRVEA
MaF1	3	4.0695e-2 (4.34e-4) -	4.0457e-2 (4.65e-4) =	4.1397e-2 (4.11e-4) -	4.1051e-2 (4.76e-4) -	4.4659e-2 (2.24e-3) -	**4.0137e-2 (7.07e-4)**
	6	**1.5214e-1 (1.27e-3) +**	1.5784e-1 (1.63e-3) =	1.6996e-1 (2.24e-3) -	1.6209e-1 (1.33e-3) -	2.1860e-1 (1.31e-2) -	1.5801e-1 (1.63e-3)
	8	**2.0056e-1 (9.66e-4) +**	2.2185e-1 (3.19e-3) -	2.1627e-1 (2.02e-3) -	2.1033e-1 (1.35e-3) =	2.9819e-1 (2.29e-2) -	2.1034e-1 (4.06e-3)
	10	**2.0712e-1 (8.18e-4) +**	2.3424e-1 (2.40e-3) -	2.2379e-1 (1.61e-3) -	2.2075e-1 (9.25e-4) -	3.3507e-1 (3.79e-2) -	2.1896e-1 (2.09e-3)
	15	**2.8213e-1 (2.45e-3) +**	3.1283e-1 (4.13e-3) -	3.6882e-1 (1.10e-2) -	3.0513e-1 (1.87e-3) =	4.3517e-1 (2.86e-2) -	3.0596e-1 (6.48e-3)
MaF2	3	2.8801e-2 (4.22e-4) =	2.8360e-2 (3.30e-4) +	2.9863e-2 (1.02e-3) -	2.8538e-2 (4.23e-4) =	**2.7819e-2 (2.53e-4) +**	2.8764e-2 (3.59e-4)
	6	1.5172e-1 (7.12e-3) -	1.2986e-1 (3.14e-3) -	1.3528e-1 (2.88e-3) -	1.3224e-1 (3.54e-3) -	1.6231e-1 (7.15e-3) -	**1.2011e-1 (2.83e-3)**
	8	2.1008e-1 (1.07e-2) -	1.6180e-1 (2.84e-3) -	1.7669e-1 (4.98e-3) -	1.7077e-1 (4.08e-3) -	1.9623e-1 (7.63e-3) -	**1.5426e-1 (4.46e-3)**
	10	2.3118e-1 (1.49e-2) -	1.6311e-1 (2.11e-3) =	1.8001e-1 (5.08e-3) -	1.7952e-1 (3.12e-3) -	2.2530e-1 (1.09e-2) -	**1.6277e-1 (3.53e-3)**
	15	3.8079e-1 (2.21e-2) -	**1.8814e-1 (2.49e-3) +**	2.5586e-1 (2.54e-2) -	2.1656e-1 (2.90e-3) +	3.5602e-1 (2.61e-2) -	2.3473e-1 (1.85e-2)
MaF4	3	5.3819e-1 (5.86e-1) -	4.5076e-1 (6.31e-1) -	3.1389e-1 (7.10e-3) -	3.8202e-1 (7.28e-2) -	3.1454e-1 (1.18e-2) -	**2.8586e-1 (1.22e-2)**
	6	5.2991e+0 (3.25e-1) =	**3.9141e+0 (5.79e-2) +**	5.4622e+0 (2.42e-1) -	5.0175e+0 (6.17e-1) =	5.4297e+0 (5.31e-1) -	5.1055e+0 (5.13e-1)
	8	2.2421e+1 (2.59e+0) -	**1.4921e+1 (5.51e-1) +**	2.3293e+1 (1.81e+0) -	1.6868e+1 (1.37e+0) +	2.3167e+1 (2.83e+0) -	2.0515e+1 (1.93e+0)
	10	8.2524e+1 (7.53e+0) =	5.3443e+1 (6.13e+0) +	9.3870e+1 (6.33e+0) -	**5.0307e+1 (2.35e+0) +**	8.2209e+1 (7.44e+0) =	8.2633e+1 (1.33e+1)
	15	3.4612e+3 (4.26e+2) -	**1.6331e+3 (2.03e+2) +**	4.0051e+3 (4.75e+2) -	1.6987e+3 (1.66e+2) +	3.5594e+3 (2.92e+2) -	3.1792e+3 (6.35e+2)
MaF6	3	4.4075e-3 (7.71e-5) =	5.9721e-3 (5.62e-5) -	**4.3259e-3 (7.88e-5) =**	4.4405e-3 (1.31e-4) =	7.8499e-3 (2.10e-3) -	4.4949e-3 (5.60e-4)
	6	3.9484e-3 (1.26e-4) -	5.2204e-3 (7.26e-4) -	**3.4310e-3 (4.33e-5) +**	4.3682e-3 (1.64e-4) -	2.6773e-2 (8.65e-3) -	3.6440e-3 (1.46e-4)
	8	3.5332e-1 (1.26e-1) -	5.2479e-2 (8.43e-2) -	**2.9147e-3 (3.67e-4) +**	1.3757e-1 (1.78e-1) -	3.9840e-2 (2.21e-2) -	3.1622e-3 (1.13e-4)
	10	6.2929e-1 (2.77e-1) -	4.9132e-1 (2.81e-1) -	1.9749e-2 (7.33e-2) =	4.7708e-1 (1.65e-1) -	5.6690e-2 (8.56e-2) -	**1.7005e-3 (4.27e-5)**
	15	1.9356e-1 (9.00e-2) -	7.4209e-1 (2.47e-7) -	6.4727e-2 (1.95e-2) -	3.2317e-1 (2.02e-2) -	3.9466e-2 (3.79e-2) -	**1.7442e-2 (5.95e-2)**
MaF7	3	6.2422e-2 (2.27e-3) -	9.8140e-2 (1.06e-1) =	1.4745e-1 (1.33e-1) -	5.9283e-2 (1.13e-3) -	9.2942e-2 (1.67e-1) =	**5.4915e-2 (6.98e-4)**
	6	4.1693e-1 (1.28e-2) -	4.5301e-1 (4.25e-2) -	4.5295e-1 (7.89e-3) -	4.5335e-1 (1.14e-2) -	4.1204e-1 (6.67e-2) =	**3.9068e-1 (1.38e-2)**
	8	7.4086e-1 (2.07e-2) -	6.7317e-1 (8.25e-2) -	9.7951e-1 (5.48e-2) -	7.0523e-1 (1.28e-2) -	6.7422e-1 (5.23e-2) -	**6.2132e-1 (2.61e-2)**
	10	9.5290e-1 (2.95e-2) +	**7.5562e-1 (7.79e-3) +**	1.5137e+0 (1.59e-1) -	9.5528e-1 (1.37e-2) +	8.8082e-1 (8.75e-2) +	1.0007e+0 (4.97e-2)
	15	**1.6643e+0 (4.11e-2) +**	1.7003e+0 (6.60e-2) +	5.0287e+0 (8.77e-1) -	2.3971e+0 (1.47e-1) -	2.0552e+0 (5.02e-1) =	2.0853e+0 (2.87e-1)
MaF8	3	8.5601e-2 (5.91e-2) -	8.3264e-2 (8.80e-3) -	7.1398e-2 (1.81e-3) -	6.7356e-2 (2.49e-3) -	1.3402e-1 (2.80e-1) -	**6.2759e-2 (1.88e-3)**
	6	1.1404e-1 (3.16e-3) -	1.2475e-1 (4.63e-3) -	1.2687e-1 (2.60e-3) -	1.1815e-1 (1.95e-3) -	5.6310e-1 (1.27e-1) -	**1.1042e-1 (2.49e-3)**
	8	1.2920e-1 (2.73e-3) -	1.3149e-1 (2.59e-3) -	1.4687e-1 (5.09e-3) -	1.3398e-1 (3.00e-3) -	9.2466e-1 (1.76e-1) -	**1.2048e-1 (1.72e-3)**
	10	1.0905e-1 (1.57e-3) -	1.0766e-1 (1.01e-3) -	1.2128e-1 (3.57e-3) -	1.1428e-1 (2.08e-3) -	1.0310e+0 (2.29e-1) -	**1.0347e-1 (2.06e-3)**
	15	1.9678e-1 (3.67e-3) -	1.8989e-1 (3.19e-3) -	2.2529e-1 (9.10e-3) -	2.1195e-1 (4.43e-3) -	1.5246e+0 (2.35e-1) -	**1.8390e-1 (2.46e-3)**
MaF9	3	2.6935e-1 (2.16e-1) -	1.0082e-1 (1.51e-1) =	**6.1111e-2 (8.55e-3) +**	4.1945e-1 (9.18e-2) -	6.8182e-2 (1.90e-2) -	6.5177e-2 (2.76e-3)
	6	**1.6778e-1 (6.79e-3) +**	9.4214e-1 (4.14e-2) -	1.8441e-1 (6.28e-3) +	3.1370e-1 (2.13e-1) +	6.6410e-1 (2.18e-1) -	3.1644e-1 (4.60e-2)
	8	**1.7762e-1 (8.03e-3) =**	1.1361e+0 (2.15e-1) -	1.9936e-1 (7.38e-3) =	2.0931e-1 (5.59e-2) =	6.7995e-1 (1.39e-1) -	2.2401e-1 (9.40e-2)
	10	1.3742e-1 (2.61e-3) -	1.0350e+0 (6.73e-2) -	1.6440e-1 (5.18e-3) -	1.5598e-1 (1.81e-2) -	8.2310e-1 (1.62e-1) -	**1.2406e-1 (2.98e-3)**
	15	1.8332e-1 (3.41e-3) =	1.8154e-1 (3.75e-3) =	1.9796e-1 (6.84e-3) -	3.8427e-1 (1.80e-1) -	1.2808e+0 (3.77e-1) -	**1.8136e-1 (2.97e-3)**
MaF10	3	1.3997e+0 (7.11e-2) -	**1.5078e-1 (1.49e-2) +**	1.5342e-1 (1.49e-2) +	1.9222e-1 (1.59e-2) +	2.4809e-1 (2.65e-2) =	2.4430e-1 (6.35e-2)
	6	1.7860e+0 (1.34e-1) -	**5.6987e-1 (1.81e-2) +**	6.6104e-1 (3.55e-2) +	7.8369e-1 (8.30e-2) -	8.9799e-1 (8.92e-2) -	6.9361e-1 (3.82e-2)
	8	2.1357e+0 (1.14e-1) -	**8.6531e-1 (1.63e-2) +**	9.3617e-1 (6.02e-2) =	9.8522e-1 (5.94e-2) -	1.1239e+0 (7.51e-2) -	9.1059e-1 (4.25e-2)
	10	2.7431e+0 (1.74e-1) -	**9.5252e-1 (1.46e-2) +**	1.2487e+0 (7.93e-2) -	1.5724e+0 (1.03e-1) -	1.1957e+0 (8.48e-2) -	1.0982e+0 (7.33e-2)
	15	2.9968e+0 (2.04e-1) -	**1.6652e+0 (5.43e-2) +**	1.9095e+0 (8.64e-2) +	1.9455e+0 (7.11e-2) =	2.4454e+0 (1.53e-1) -	2.0467e+0 (1.47e-1)
MaF11	3	3.4025e-1 (4.98e-2) -	**1.4663e-1 (5.25e-3) +**	1.7218e-1 (3.89e-3) -	2.3668e-1 (2.79e-2) -	2.9907e-1 (6.48e-2) -	1.6449e-1 (3.99e-3)
	6	3.0105e+0 (6.79e-1) -	**7.2145e-1 (1.13e-1) +**	7.7975e-1 (4.28e-2) +	1.1523e+0 (1.20e-1) -	2.7109e+0 (5.34e-1) -	9.2761e-1 (1.57e-1)
	8	5.6686e+0 (1.15e+0) -	**1.2084e+0 (1.96e-1) +**	1.4889e+0 (8.27e-2) +	2.2017e+0 (1.50e-1) =	5.9073e+0 (7.00e-1) -	2.0449e+0 (5.39e-1)
	10	1.1483e+1 (2.55e+0) -	**2.3968e+0 (2.67e-1) +**	2.6187e+0 (6.37e-1) +	2.8316e+0 (3.46e-1) +	1.2092e+1 (1.99e+0) -	4.4658e+0 (1.41e+0)
	15	2.1699e+1 (3.74e+0) -	2.5023e-1 (2.24e-1) +	**4.2769e-2 (1.27e-1) +**	7.5461e-1 (4.02e-1) +	2.0253e+1 (3.13e+0) -	4.1620e+0 (3.79e+0)
MaF13	3	7.9816e-2 (9.07e-3) -	9.2337e-2 (5.31e-3) -	7.9019e-2 (7.46e-3) -	1.0002e-1 (1.43e-2) -	**7.1197e-2 (4.57e-3) =**	7.1958e-2 (3.86e-3)
	6	**1.0625e-1 (5.57e-3) =**	1.6257e-1 (1.27e-2) -	1.2569e-1 (5.64e-3) -	2.0330e-1 (3.44e-2) -	6.2696e-1 (2.14e-1) -	1.0869e-1 (1.20e-2)
	8	1.1508e-1 (8.49e-3) -	1.8038e-1 (2.06e-2) -	1.3216e-1 (5.12e-3) -	1.8917e-1 (2.89e-2) -	4.7346e-1 (1.32e-1) -	**1.0880e-1 (1.12e-2)**
	10	9.9863e-2 (7.69e-3) =	1.6272e-1 (1.71e-2) -	1.1035e-1 (4.74e-3) -	1.4784e-1 (2.20e-2) -	4.3761e-1 (1.53e-1) -	**9.6637e-2 (7.32e-3)**
	15	2.0166e-1 (1.84e-2) -	2.4918e-1 (1.48e-2) -	1.8721e-1 (1.23e-2) -	2.3655e-1 (2.92e-2) -	7.5857e-1 (1.49e-1) -	**1.3310e-1 (7.64e-3)**
IDTLZ1	3	2.0311e-2 (2.93e-4) =	2.2238e-2 (1.67e-3) -	2.0675e-2 (4.28e-4) -	2.5631e-2 (6.90e-3) -	2.4052e-2 (2.68e-3) -	**2.0181e-2 (2.24e-4)**
	6	**7.5926e-2 (6.46e-4) +**	8.6633e-2 (9.22e-3) -	8.4693e-2 (1.16e-3) -	8.4000e-2 (9.39e-3) -	1.0714e-1 (5.98e-3) -	7.8780e-2 (1.69e-3)
	8	**9.9827e-2 (6.81e-4) +**	1.1082e-1 (1.10e-2) =	1.0719e-1 (8.77e-4) +	1.1087e-1 (1.80e-2) =	1.4037e-1 (5.15e-3) -	1.0982e-1 (1.98e-2)
	10	**1.0312e-1 (3.91e-4) +**	1.1945e-1 (7.87e-3) -	1.1104e-1 (1.15e-3) =	1.1171e-1 (6.38e-3) +	1.5069e-1 (6.15e-3) -	1.1613e-1 (1.11e-2)
	15	1.4287e-1 (2.76e-3) +	**1.4271e-1 (3.41e-3) +**	1.8135e-1 (7.53e-3) -	1.5080e-1 (1.96e-3) =	1.9567e-1 (1.04e-2) -	1.5038e-1 (2.26e-3)
IDTLZ2	3	6.8498e-2 (4.34e-3) -	5.6821e-2 (1.33e-3) =	7.1036e-2 (1.66e-4) -	6.9546e-2 (2.63e-3) -	6.5556e-2 (2.08e-3) -	**5.6455e-2 (7.65e-4)**
	6	2.8563e-1 (5.92e-3) -	2.7121e-1 (3.22e-3) -	3.0647e-1 (5.26e-3) -	2.6755e-1 (1.96e-3) -	3.2992e-1 (2.10e-2) -	**2.6497e-1 (3.46e-3)**
	8	3.9754e-1 (5.25e-3) -	3.8800e-1 (4.35e-3) =	3.9729e-1 (8.01e-3) -	**3.7221e-1 (2.56e-3) +**	4.7825e-1 (1.93e-2) -	3.9060e-1 (6.99e-3)
	10	4.3158e-1 (2.73e-3) -	4.2660e-1 (3.76e-3) -	4.4144e-1 (6.36e-3) -	4.2326e-1 (1.63e-3) =	4.8985e-1 (1.48e-2) -	**4.2215e-1 (6.78e-3)**
	15	6.1510e-1 (6.73e-3) -	6.2965e-1 (5.92e-3) -	**5.6838e-1 (8.95e-3) +**	5.9801e-1 (3.76e-3) +	7.0653e-1 (1.34e-2) -	6.0519e-1 (1.02e-2)
+/-/=		11/40/9	19/32/9	13/42/5	12/37/11	2/52/6	-
Friedman	3	4.25	3.3333	3.4167	4	4	**2**
	6	3.0833	3.25	3.5833	3.5833	5.3333	**2.1667**
	8	3.6667	2.9167	3.5	3.5	5.1667	**2.25**
	10	3.5833	3.1667	3.9167	3.25	4.75	**2.3333**
	15	3.5833	2.6667	3.6667	3.4167	5.0833	**2.5833**
	overall	3.6333	3.0667	3.6167	3.55	4.8667	**2.2667**



(a) Solutions obtained by EMyO/C on MaF6 using the best run. (b) Solutions obtained by iRVEA on MaF6 using the best run.

Fig. 4. Illustration the performance of EMyO/C and iRVEA in dealing with MaF6 with 15 objectives using the best run of IGD value.

achieves the best among all compared algorithms in terms of both HV and IGD according to the Friedman test.

Here, we take a closer look at the performance of iRVEA and other five compared algorithms in dealing with MaF and IDTLZ are analyzed. MaF1 is easy to solve as its PS is relatively simple. EMyO/C performs the best in terms of IGD while iRVEA ranks second among these six algorithms. MaF2 has a disconnected PF. iRVEA achieves the best performance in terms of IGD on MaF2 having 6, 8, and 10 objectives. EMyO/C, however, performs the best according to HV probably because solutions obtained by EMyO/C have a wider distribution on average. MaF4 is designed to assess whether evolutionary many objective algorithms are capable of handling badly-scaled PFs and the fitness landscape of MaF4 is highly multimodal. We find that Two_Arch2 and VaEA perform better than other four algorithms in dealing with this problem. The poor performance of EMyO/C on MaF4 might be attributed to the fact that environmental selection of

TABLE II

HV COMPARISONS AND FRIEDMAN TEST OF THE SIX ALGORITHMS ON ALL TEST INSTANCES. THE BEST RESULT ON EACH TEST INSTANCE IS HIGHLIGHTED IN BOLDFACE AND BLUE. "+", "−", AND "=" DENOTE THAT THE COMPARED ALGORITHM PERFORMS SIGNIFICANTLY BETTER THAN, WORSE THAN, AND EQUIVALENT TO IRVEA, RESPECTIVELY.

Problem	M	EMyO/C	Two_Arch2	AR-MOEA	VaEA	RVEA*	iRVEA
MaF1	3	2.9330e-1 (7.13e-4) -	2.9519e-1 (5.48e-4) +	**2.9595e-1 (9.42e-4) +**	2.9152e-1 (7.63e-4) -	2.8421e-1 (3.65e-3) -	2.9471e-1 (1.17e-3)
	6	**3.1845e-3 (6.96e-5) +**	3.0029e-3 (1.04e-4) +	2.5121e-3 (1.09e-4) -	2.5121e-3 (9.35e-5) -	1.4572e-3 (1.63e-4) -	2.7418e-3 (7.08e-5)
	8	**7.5886e-5 (5.31e-6) +**	5.6286e-5 (1.15e-5) =	5.8091e-5 (1.10e-5) =	5.3318e-5 (6.98e-6) =	2.0995e-5 (5.34e-6) -	5.5191e-5 (4.15e-6)
	10	1.8179e-6 (6.11e-7) +	1.0846e-6 (1.65e-6) =	1.5446e-6 (2.11e-6) =	**1.9135e-6 (2.01e-6) =**	2.7483e-7 (1.21e-7) -	1.3564e-6 (5.89e-7)
	15	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	1.2747e-12 (1.11e-12) -	**2.2379e-11 (8.43e-11)**
MaF2	3	2.1494e-1 (1.26e-3) -	**2.1806e-1 (2.72e-4) +**	2.1490e-1 (8.38e-4) -	2.1578e-1 (9.31e-4) -	2.1635e-1 (5.79e-4) -	2.1726e-1 (4.59e-4)
	6	8.2615e-2 (7.09e-4) -	**8.5544e-2 (9.30e-4) +**	8.1896e-2 (4.68e-4) -	8.4882e-2 (9.72e-4) +	8.0372e-2 (1.33e-3) -	8.3448e-2 (7.89e-4)
	8	**2.9734e-2 (4.12e-4) +**	2.8669e-2 (3.37e-4) +	2.7399e-2 (5.55e-4) =	2.9491e-2 (2.44e-4) +	2.5272e-2 (5.24e-4) -	2.7109e-2 (4.81e-4)
	10	**8.2589e-3 (8.37e-5) +**	7.4169e-3 (1.46e-4) +	7.4558e-3 (1.48e-4) +	7.8822e-3 (1.14e-4) +	6.4542e-3 (1.08e-4) -	7.1727e-3 (1.23e-4)
	15	**7.0794e-5 (8.83e-7) +**	4.1598e-5 (1.88e-6) -	6.0958e-5 (1.85e-6) +	6.2541e-5 (1.01e-6) +	4.0960e-5 (3.56e-6) -	5.4186e-5 (2.84e-6)
MaF4	3	4.0789e+1 (1.33e+1) =	4.0891e+1 (1.34e+1) =	4.4361e+1 (4.12e-1) -	4.3718e+1 (1.36e+0) -	4.4763e+1 (5.31e-1) =	**4.4943e+1 (5.10e-1)**
	6	8.0242e+4 (7.65e+3) -	7.0926e+4 (7.15e+3) -	5.0703e+4 (5.76e+3) -	9.3501e+4 (2.27e+4) =	9.1897e+4 (1.60e+4) =	**9.3778e+4 (1.45e+4)**
	8	6.2160e+7 (1.30e+7) -	1.5643e+8 (2.37e+7) =	2.9339e+7 (6.75e+6) -	**2.4332e+8 (6.43e+7) +**	5.8172e+7 (2.04e+7) -	1.4021e+8 (8.25e+7)
	10	5.9895e+11 (1.61e+11) -	4.4010e+12 (1.23e+12) =	2.9011e+11 (6.31e+10) -	**1.0570e+13 (1.54e+12) +**	7.0972e+11 (3.14e+11) -	4.5919e+12 (4.77e+12)
	15	1.1088e+26 (4.53e+25) -	0.0000e+0 (0.00e+0) -	5.1450e+25 (2.81e+25) -	**2.1060e+29 (3.13e+29) =**	2.1990e+26 (2.48e+26) -	4.0198e+28 (8.91e+28)
MaF6	3	1.3291e-1 (6.48e-5) -	**1.3338e-1 (3.24e-5) +**	1.3300e-1 (4.61e-5) -	1.3309e-1 (4.93e-5) =	1.3184e-1 (8.82e-4) -	1.3307e-1 (1.79e-4)
	6	1.6326e-3 (4.19e-6) =	1.6309e-3 (6.40e-6) =	1.6288e-3 (5.16e-6) -	**1.6329e-3 (4.31e-6) =**	1.5547e-3 (2.04e-5) -	1.6324e-3 (4.14e-6)
	8	2.0619e-6 (4.12e-6) -	1.9005e-5 (1.16e-6) =	1.9626e-5 (7.25e-8) =	1.5207e-5 (6.44e-6) =	1.8853e-5 (1.47e-7) -	**1.9652e-5 (5.62e-8)**
	10	1.0895e-9 (4.80e-9) -	5.7156e-8 (2.29e-9) -	6.1176e-8 (4.26e-9) =	4.9222e-9 (1.45e-8) -	5.9270e-8 (4.80e-9) -	**6.2404e-8 (2.13e-10)**
	15	3.7802e-17 (2.60e-17) -	8.4321e-17 (5.62e-22) -	8.5446e-17 (7.70e-19) -	8.1046e-17 (4.34e-18) -	8.6371e-17 (7.17e-19) -	**8.8003e-17 (1.57e-18)**
MaF7	3	1.6399e+0 (8.87e-3) -	1.6159e+0 (7.86e-2) -	1.5795e+0 (9.81e-2) -	1.6397e+0 (3.69e-3) -	1.6288e+0 (1.03e-1) =	**1.6500e+0 (5.12e-3)**
	6	1.5174e+0 (1.70e-1) -	1.7974e+0 (1.97e-1) -	2.0234e+0 (2.42e-2) -	2.0676e+0 (3.13e-2) -	**2.2866e+0 (4.40e-2) +**	2.1584e+0 (6.96e-2)
	8	4.1656e-1 (1.28e-1) -	1.6146e+0 (3.77e-1) -	**2.1948e+0 (3.54e-2) +**	1.9833e+0 (7.63e-2) =	2.1673e+0 (1.69e-1) +	1.9811e+0 (1.90e-1)
	10	1.0780e-1 (7.97e-2) -	1.4853e+0 (1.93e-1) -	**2.0106e+0 (1.15e-1) +**	1.7862e+0 (8.05e-2) =	1.9780e+0 (2.22e-1) +	1.7057e+0 (3.65e-1)
	15	8.3403e-5 (1.84e-4) -	7.5205e-1 (5.97e-1) =	9.8163e-1 (1.97e-1) -	**1.4877e+0 (4.25e-2) +**	1.2958e+0 (3.33e-1) =	1.0799e+0 (5.84e-1)
MaF8	3	1.8367e+0 (1.69e-1) -	1.8688e+0 (2.50e-2) -	1.8743e+0 (5.63e-3) -	1.8872e+0 (1.10e-2) -	1.7749e+0 (3.92e-1) -	**1.9027e+0 (4.48e-3)**
	6	9.6520e+0 (3.51e-2) -	9.1612e+0 (1.41e-1) -	9.4498e+0 (6.29e-2) -	9.6002e+0 (4.91e-2) -	5.4656e+0 (9.78e-1) -	**9.6828e+0 (5.25e-2)**
	8	1.5679e+1 (7.39e-2) -	1.4541e+1 (1.21e-1) -	1.5333e+1 (1.58e-1) -	1.5600e+1 (1.25e-1) -	5.4278e+0 (1.71e+0) -	**1.5743e+1 (9.74e-2)**
	10	**2.7567e+1 (2.83e-1) =**	2.5759e+1 (2.04e-1) -	2.7102e+1 (2.73e-1) -	2.7452e+1 (2.36e-1) =	7.9948e+0 (2.93e+0) -	2.7486e+1 (2.77e-1)
	15	**6.8559e+1 (2.37e+0) =**	5.5677e+1 (2.16e+0) -	6.5064e+1 (2.74e+0) -	6.6379e+1 (2.15e+0) -	9.8092e+0 (4.87e+0) -	6.8061e+1 (2.49e+0)
MaF9	3	2.9441e+0 (7.56e-1) -	3.5037e+0 (5.89e-1) -	**3.6610e+0 (2.46e-2) +**	2.7071e+0 (1.55e-1) -	3.6344e+0 (5.21e-2) +	3.6252e+0 (1.55e-2)
	6	**5.5120e+0 (6.29e-2) +**	5.0305e-1 (9.55e-2) -	5.3595e+0 (7.46e-2) +	4.4076e+0 (1.45e+0) +	2.2975e+0 (1.01e+0) -	4.0119e+0 (4.12e-1)
	8	**1.2782e+1 (1.65e-1) =**	8.2742e-1 (2.13e-1) -	1.2297e+1 (1.82e-1) =	1.1951e+1 (1.42e+0) =	5.4355e+0 (1.07e+0) -	1.1855e+1 (1.97e+0)
	10	2.7044e+1 (2.11e-1) -	1.2121e+0 (1.74e-1) -	2.5766e+1 (3.79e-1) -	2.6044e+1 (1.01e+0) -	9.4136e+0 (2.16e+0) -	**2.8038e+1 (2.46e-1)**
	15	1.2089e+2 (3.54e+0) =	1.1538e+2 (2.91e+0) -	1.1736e+2 (4.28e+0) -	7.8892e+1 (3.61e+1) -	2.3760e+1 (8.41e+0) -	**1.2294e+2 (4.62e+0)**
MaF10	3	2.5748e+1 (3.57e+0) -	5.9800e+1 (1.01e+0) +	**5.9915e+1 (9.58e-2) +**	5.7014e+1 (8.50e-1) -	5.7759e+1 (1.19e+0) -	5.8937e+1 (7.34e-1)
	6	3.5717e+4 (4.62e+3) -	7.8565e+4 (5.69e+1) +	**7.8651e+4 (9.19e+2) +**	6.3268e+4 (4.12e+3) -	5.9852e+4 (5.18e+3) -	7.0926e+4 (2.88e+3)
	8	8.4766e+6 (7.87e+5) -	2.0670e+7 (1.27e+4) +	**2.0730e+7 (8.26e+2) +**	1.7601e+7 (1.17e+6) -	1.7466e+7 (1.50e+6) -	2.0538e+7 (3.91e+5)
	10	2.5414e+9 (3.76e+8) -	8.6455e+9 (4.73e+6) =	**8.6567e+9 (3.70e+6) +**	5.4873e+9 (4.29e+8) -	8.1449e+9 (3.95e+8) -	8.5527e+9 (2.04e+8)
	15	5.2127e+16 (6.78e+15) -	1.3898e+17 (5.45e+13) -	1.3913e+17 (5.95e+12) -	1.3914e+17 (3.31e+12) =	1.3872e+17 (1.12e+14) -	**1.3914e+17 (1.29e+12)**
MaF11	3	5.8953e+1 (2.36e-1) -	**5.9777e+1 (6.91e-2) =**	5.9634e+1 (4.70e-2) -	5.9159e+1 (1.07e-1) -	5.9074e+1 (1.33e-1) -	5.9747e+1 (5.48e-2)
	6	7.9802e+4 (3.15e+2) -	8.1267e+4 (7.75e+1) =	**8.1464e+4 (5.62e+1) +**	8.0975e+4 (1.39e+2) -	8.0594e+4 (1.91e+2) -	8.1213e+4 (7.62e+1)
	8	2.1691e+7 (6.39e+4) -	2.2069e+7 (1.31e+4) +	**2.2089e+7 (1.43e+4) +**	2.2012e+7 (2.75e+4) -	2.1850e+7 (7.01e+4) -	2.2034e+7 (2.36e+4)
	10	9.4823e+9 (2.15e+7) -	**9.6168e+9 (3.52e+6) +**	9.6123e+9 (6.87e+6) +	9.5703e+9 (1.12e+7) =	9.4947e+9 (2.76e+7) -	9.5771e+9 (1.30e+7)
	15	1.7621e+17 (6.68e+14) -	**1.7887e+17 (4.60e+13) +**	1.7879e+17 (1.26e+14) +	1.7848e+17 (1.62e+14) -	1.7754e+17 (5.57e+14) -	1.7865e+17 (1.50e+14)
MaF13	3	**7.0417e-1 (1.53e-2) =**	6.6241e-1 (1.51e-2) -	6.6964e-1 (2.28e-2) -	6.3245e-1 (2.39e-2) -	6.8535e-1 (1.27e-2) -	7.0303e-1 (1.01e-2)
	6	**4.1872e-1 (6.64e-3) +**	3.4264e-1 (3.24e-2) -	3.7678e-1 (7.79e-3) -	3.0042e-1 (2.88e-2) -	1.0166e-1 (9.75e-2) -	4.0280e-1 (8.01e-3)
	8	**3.8586e-1 (3.53e-3) +**	3.2351e-1 (1.47e-2) -	3.4652e-1 (8.31e-3) -	2.8664e-1 (3.35e-2) -	1.6596e-1 (7.33e-2) -	3.7077e-1 (7.04e-3)
	10	**3.8631e-1 (1.46e-3) +**	3.1374e-1 (4.82e-2) -	3.5378e-1 (5.60e-3) -	2.9306e-1 (4.48e-2) -	1.7736e-1 (5.97e-2) -	3.7454e-1 (3.30e-3)
	15	**3.9983e-1 (1.88e-3) +**	3.4313e-1 (9.64e-3) -	3.4743e-1 (1.22e-2) -	2.5466e-1 (2.90e-2) -	2.4728e-1 (3.81e-2) -	3.7810e-1 (7.16e-3)
IDTLZ1	3	3.6676e-2 (1.98e-4) -	3.6619e-2 (3.66e-4) -	**3.6923e-2 (2.64e-4) =**	3.5371e-2 (1.77e-3) -	3.5315e-2 (8.05e-4) -	3.6880e-2 (1.42e-4)
	6	2.6402e-5 (2.59e-5) =	**4.6848e-5 (3.46e-6) =**	2.0102e-5 (2.05e-5) -	2.1708e-5 (2.26e-5) -	1.5331e-5 (1.54e-5) -	4.5007e-5 (2.65e-6)
	8	**3.1732e-7 (2.25e-8) +**	2.9792e-7 (5.17e-8) +	2.4701e-7 (5.06e-8) =	2.1987e-7 (6.56e-8) =	1.3322e-7 (1.95e-8) -	2.4228e-7 (7.79e-8)
	10	1.0709e-9 (1.19e-9) =	**1.3910e-9 (9.45e-10) =**	2.5329e-10 (7.80e-10) -	8.3234e-10 (1.65e-9) -	2.4942e-10 (2.75e-10) -	1.2948e-9 (8.47e-10)
	15	6.0529e-16 (2.71e-15) -	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	1.5705e-16 (5.02e-17) =	**1.2034e-15 (1.62e-15)**
IDTLZ2	3	7.1260e-1 (1.75e-3) -	**7.1985e-1 (7.44e-4) +**	7.0447e-1 (4.86e-4) -	7.0525e-1 (2.18e-3) -	7.1178e-1 (1.87e-3) -	7.1489e-1 (1.57e-3)
	6	4.2979e-2 (2.77e-2) -	**7.0276e-2 (1.31e-3) =**	2.8286e-2 (2.36e-2) -	2.9160e-2 (2.12e-2) -	4.2419e-2 (2.68e-2) -	6.9531e-2 (3.51e-3)
	8	6.8270e-3 (4.20e-4) =	7.0424e-3 (2.93e-4) =	4.0671e-3 (3.37e-4) -	3.4218e-3 (1.31e-4) -	**8.0572e-3 (7.13e-4) +**	7.0474e-3 (6.12e-4)
	10	3.1923e-4 (3.11e-4) -	5.9366e-4 (4.45e-5) =	3.2745e-4 (3.26e-4) =	1.5794e-4 (1.51e-4) -	4.1175e-4 (4.03e-4) =	**6.0159e-4 (5.68e-5)**
	15	3.3935e-7 (1.20e-7) +	3.9248e-7 (1.05e-6) +	6.0607e-7 (2.55e-7) +	4.0728e-7 (3.51e-7) +	**7.0334e-7 (1.05e-7) +**	1.4321e-7 (6.82e-8)
+/-/=		13/36/11	16/27/17	16/34/10	9/36/15	6/48/6	-
Friedman	3	4.25	2.9167	3.25	4.25	4.25	**2.0833**
	6	3.0833	3	3.8333	3.6667	4.9167	**2.5**
	8	3.25	3.25	3	3.9167	4.75	**2.8333**
	10	3.3333	3.5833	2.75	3.8333	4.5	**3**
	15	3.5833	4	3.5	3.4167	3.9167	**2.5833**
	overall	3.4833	3.4833	3.2667	3.75	4.4167	**2.6**

EMyO/C are not able to cluster solutions based on Euclidean distance in high dimensional space properly.

The PF of MaF6 is degenerated, which is not easy to solve. We can find that EMyO/C does not perform well in high dimensional objective space as the individuals far from the real PF are easy to be selected based on the cluster criterion, which will mislead the evolution. iRVEA performs the best on 10 and 15 objective problems because redundant solutions in the population will not be selected in environmental selection if not enough non-dominated solutions exists in the population. As a result, the algorithm can converge to the real PF, which can be seen from the solution obtained by iRVEA in Fig. 4. Fig. 4 plots the best results among 20 independent runs of EMyO/C and iRVEA on MaF6. In 3,6, and 8 objective problems, iRVEA performs worse than AR-MOEA but performs better than other four algorithms. MaF7 is a disconnected problems whose PF is of 2^{M-1} segments. iRVEA achieves the best in handling 3, 6, and 8 objective problems but can not find whole segments of PF on 10 and 15 objective problems. MaF8 and MaF9 are problems of only two-dimensional decision space. MaF9 poses great challenge to algorithms which only adopt Pareto dominance as selection criterion. Results show iRVEA has the best performance over other five algorithms on all objectives of MaF8 and 10 and 15 objectives of MaF9. iRVEA does not perform well in dealing with MaF10 and MaF11, which have complex and mixed geometries. Results on MaF13 shows iRVEA is capable of handling degenerate PFs with complicated variable linkages and experiments on IDTLZ1 and IDTLZ2 demonstrate the good performance of iRVEA in dealing with problems with inverted PFs. Overall, iRVEA shows competitive performance in solving problems with irregular PFs.

V. Conclusion

In this paper, we have proposed an improved reference vector guided evolutionary algorithm, termed iRVEA, for dealing with problems with irregular PFs. Compared with five state-of-the-art algorithms, iRVEA shows robust performance in dealing with problems having disconnected, inverted and degenerate PFs. Future work will focus on designing more effective mechanisms for generating reference vectors adapted to the estimated geometrical features of PFs. In addition, new reproduction operators such as estimation of distribution algorithms will be investigated to generate more solutions in the less explored regions. We expect that the performance of the algorithms for solving problems with irregular PFs can be further enhanced by combining more specific reproduction strategies with tailored selection methods.

References

[1] Z. He, G. G. Yen, and J. Zhang, "Fuzzy-based Pareto optimality for many-objective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, pp. 269–285, 2014.

[2] K. Ikeda, H. Kita, and S. Kobayashi, "Failure of Pareto-based MOEAs: Does non-dominated really mean near to optimal?" in *Proceedings of the 2001 IEEE Congress on Evolutionary Computation*, 2001, pp. 957–962.

[3] S. Yang, M. Li, X. Liu, and J. Zheng, "A grid-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 5, pp. 721–736, 2013.

[4] X. Zou, Y. Chen, M. Liu, and L. Kang, "A new evolutionary algorithm for solving many-objective optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 5, pp. 1402–1412, 2008.

[5] Y. Yuan, H. Xu, B. Wang, and X. Yao, "A new dominance relation-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 16–37, 2016.

[6] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2004, pp. 832–842.

[7] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.

[8] L. While, P. Hingston, L. Barone, and S. Huband, "A faster algorithm for calculating hypervolume," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 29–38, 2006.

[9] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[10] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints." *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[11] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization." *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 773–791, 2016.

[12] H.-L. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems." *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 450–455, 2014.

[13] Y. Tian, R. Cheng, X. Zhang, Y. Su, and Y. Jin, "A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization," *IEEE Transactions on Evolutionary Computation*, 2018.

[14] R. Cheng, Y. Jin, and K. Narukawa, "Adaptive reference vector generation for inverse model based evolutionary multiobjective optimization with degenerate and disconnected Pareto fronts," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2015, pp. 127–140.

[15] C. Zhou, G. Dai, C. Zhang, X. Li, and K. Ma, "Entropy based evolutionary algorithm with adaptive reference points for many-objective optimization problems," *Information Sciences*, vol. 465, pp. 232–247, 2018.

[16] H.-L. Liu, L. Chen, Q. Zhang, and K. Deb, "Adaptively allocating search effort in challenging many-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 3, pp. 433–448, 2018.

[17] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach." *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2014.

[18] M. Asafuddoula, H. K. Singh, and T. Ray, "An enhanced decomposition-based evolutionary algorithm with adaptive reference vectors." *IEEE Transactions on Cybernetics*, vol. 48, no. 8, pp. 2321–2334, 2017.

[19] R. Denysiuk, L. Costa, and I. E. Santo, "Clustering-based selection for evolutionary many-objective optimization," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2014, pp. 538–547.

[20] Y. Hua, Y. Jin, and K. Hao, "A clustering-based adaptive evolutionary algorithm for multiobjective optimization with irregular Pareto fronts," *IEEE Transactions on Cybernetics*, no. 99, pp. 1–13, 2018.

[21] Q. Lin, S. Liu, K.-C. Wong, M. Gong, C. A. C. Coello, J. Chen, and J. Zhang, "A clustering-based evolutionary algorithm for many-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, 2018.

[22] Y. Tian, R. Cheng, X. Zhang, F. Cheng, and Y. Jin, "An indicator based multi-objective evolutionary algorithm with reference point adaptation for better versatility," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 4, pp. 609–622, 2017.

[23] Y. Xiang, Y. Zhou, M. Li, and Z. Chen, "A vector angle-based evolutionary algorithm for unconstrained many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 1, pp. 131–152, 2017.

[24] H. Ge, M. Zhao, L. Sun, Z. Wang, G. Tan, Q. Zhang, and C. P. Chen, "A many-objective evolutionary algorithm with two interacting processes: Cascade clustering and reference point incremental learning," *IEEE Transactions on Evolutionary Computation*, 2018.

[25] R. Cheng, T. Rodemann, M. Fischer, M. Olhofer, and Y. Jin, "Evolutionary many-objective optimization of hybrid electric vehicle control: from general optimization to preference articulation," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 1, no. 2, pp. 97–111, 2017.

[26] R. Cheng, M. Li, Y. Tian, X. Zhang, S. Yang, Y. Jin, and X. Yao, "A benchmark test suite for evolutionary many-objective optimization," *Complex & Intelligent Systems*, vol. 3, no. 1, pp. 67–81, 2017.

[27] H. Wang, L. Jiao, and X. Yao, "Two_Arch2: An improved two-archive algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 4, pp. 524–541, 2015.

[28] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.

[29] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang, "Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion," in *Proceedings of the 2006 IEEE Congress on Evolutionary Computation*, 2006, pp. 892–899.

[30] R. B. Agrawal, K. Deb, and R. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, no. 2, pp. 115–148, 1995.

[31] K. Deb and M. Goyal, "A combined genetic adaptive search (geneas) for engineering design," *Computer Science and Informatics*, vol. 26, pp. 30–45, 1996.