Two_Arch2: An Improved Two-Archive Algorithm for Many-Objective Optimization

Handing Wang, *Student Member, IEEE,* Licheng Jiao, *Senior Member, IEEE,* and Xin Yao, *Fellow, IEEE*

***Abstract*—Many-objective optimization problems (ManyOPs) refer, usually, to those multiobjective problems (MOPs) with more than three objectives. Their large numbers of objectives pose challenges to multiobjective evolutionary algorithms (MOEAs) in terms of convergence, diversity, and complexity. Most existing MOEAs can only perform well in one of those three aspects. In view of this, we aim to design a more balanced MOEA on ManyOPs in all three aspects at the same time. Among the existing MOEAs, the two-archive algorithm (Two_Arch) is a low-complexity algorithm with two archives focusing on convergence and diversity separately. Inspired by the idea of Two_Arch, we propose a significantly improved two-archive algorithm (i.e., Two_Arch2) for ManyOPs in this paper. In our Two_Arch2, we assign different selection principles (indicator-based and Pareto-based) to the two archives. In addition, we design a new L_p-norm-based ($p < 1$) diversity maintenance scheme for ManyOPs in Two_Arch2. In order to evaluate the performance of Two_Arch2 on ManyOPs, we have compared it with several MOEAs on a wide range of benchmark problems with different numbers of objectives. The experimental results show that Two_Arch2 can cope with ManyOPs (up to 20 objectives) with satisfactory convergence, diversity, and complexity.**

***Index Terms*—Evolutionary algorithm, L_p-norm, many-objective optimization, two-archive algorithm (Two_Arch).**

I. Introduction

IN the real world, many problems [1], [2] can be regarded as many-objective optimization problems (ManyOPs) [3], [4], a category of multiobjective optimization problems (MOPs) [5] with more than three objectives. Multiobjective evolutionary algorithms (MOEAs) [6] have become useful tools for MOPs with low-dimensional objectives. However, ManyOPs are not easy to be handled as MOPs with low-dimensional objectives, their high-dimensional objectives bring troubles to Pareto-based MOEAs. Experimental results in [3], [4], and [7]–[10] have shown poor performance of Pareto-based MOEAs on ManyOPs. The main reason why Pareto-based MOEAs lose their good performance on ManyOPs is that too many nondominated solutions in the population decrease the effectiveness of their selection processes [11], [12]. With the increasing number of objectives, the percentage of nondominated solutions increases rapidly, which even reaches almost 100% in a population when the number of objectives is more than 12 according to the experimental result in [13]. In addition to the difficulty in the convergence, the diversity is another challenge in many-objective optimization, because the similarity in a high-dimensional space is more difficult to estimate. There have been several attempts at the diversity improvement of ManyOPs [14], [15], but the issue is not solved completely. In view of the two challenges for ManyOPs mentioned above, existing works can be classified into four kinds i.e., non-Pareto-based approaches, objective reduction, preference-based approaches, and dominance relation.

As the Pareto dominance is ineffective on ManyOPs, non-Pareto-based approaches are considered as a better option. There are two kinds of non-Pareto-based approaches, the approaches based on aggregation functions and on indicators. The approaches based on aggregation functions use aggregation functions to decompose a ManyOP into a series of single-objective optimization problems. Among them, MOEA/D [16] is the most well-known. However, assigning weight vectors in the high-dimensional space for ManyOPs is a real challenge for MOEA/D [17]. To tackle this challenge, Giagkiozis *et al.* [18] proposed a generalized decomposition for MOEA/D specifically aimed at ManyOPs. Although it worked well on ManyOPs, it still needs to allocate weight vectors manually in advance.

Indicator-based approaches, which use an indicator as the fitness function in MOEAs, represent a different type of non-Pareto-based approaches to ManyOPs. Indicator-based evolutionary algorithm (IBEA) [19] is the first indicator-based MOEA. Although it performs well in terms of convergence on ManyOPs, it has poor diversity due to the lack of diversity maintenance in its indicator. Therefore, some researchers focus on hypervolume-based MOEAs [20], [21] for ManyOPs, because hypervolume is a metric to evaluate both convergence and diversity [22]. However, the hypervolume computation has

Manuscript received January 3, 2014; revised April 23, 2014; accepted August 6, 2014. Date of publication August 26, 2014; date of current version July 28, 2015. This work was supported in part by the National Basic Research Program (973 Program) of China under Grant 2013CB329402, in part by the EU FP7 IRSES Grant on "Nature Inspired Computation and its Applications (NICaiA)" under Grant 247619, in part by the EPSRC Grant on "DAASE: Dynamic Adaptive Automated Software Engineering" under Grant EP/J017515/1, in part by the Program for Cheung Kong Scholars and Innovative Research Team in University under Grant IRT1170, in part by the National Natural Science Foundation of China under Grant 61329302, in part by the National Research Foundation for the Doctoral Program of Higher Education of China under Grant 20100203120008, and in part by the Fundamental Research Funds for the Central Universities under Grant K5051302028. The work of X. Yao was supported by the Royal Society Wolfson Research Merit Award.

H. Wang and L. Jiao are with Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, International Research Center for Intelligent Perception and Computation, Xidian University, Xi'an 710071, China (e-mail: wanghanding@163.com).

X. Yao is with the Department of Computer Science, University of Birmingham, Birmingham, B15 2TT, U.K.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEVC.2014.2350987

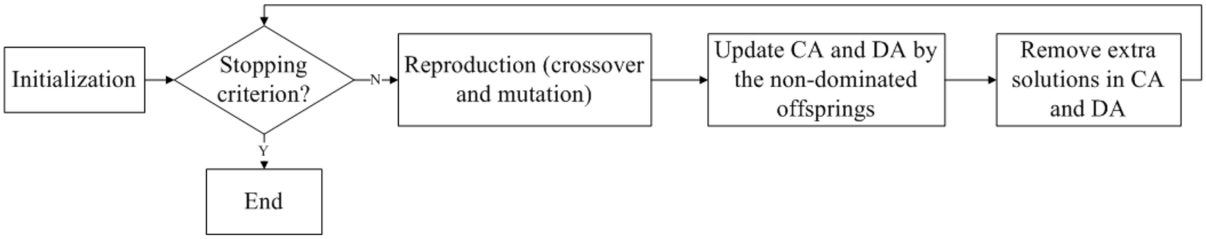
Initialization
Stopping criterion?
N
Y
Reproduction (crossover and mutation)
Update CA and DA by the non-dominated offsprings
Remove extra solutions in CA and DA
End

Fig. 1. Flow chart of Two_Arch.

high complexity ($O(n^{c_d m \pm O(1)})$, $c_d \in [1/4, 1/3]$, where m is the number of objectives) [23].

For some ManyOPs where there is redundancy among objectives [24], objective reduction [25] can be an effective approach to convert a ManyOP to an MOP with only a few objectives so that the existing MOEAs could be used. The algorithm in [24] adopts a metric to detect the changes of the dominance structure in the population for objective reduction. Feature selection is used for objective reduction in [26]. Some dimension reduction techniques in machine learning such as principal component analysis (PCA) and maximum variance unfolding (MVU) are also adopted in objective reduction [27]–[29]. Pareto corner search evolutionary algorithm (PCSEA) [30] is an off-line objective reduction method.

In some specific applications, decision makers only care about a small region of the entire Pareto front (PF) [31]. Hence, preference-based approaches can help decision makers to find parts of PFs [32]. Such studies can be divided into two classes, interactive [33]–[35] and noninteractive [36]–[38].

Due to the ineffectiveness of the Pareto dominance in many-objective optimization [39], many modifications have been made to improve the effectiveness of the Pareto dominance, such as the ε-dominance [40]–[42], the changing dominance area [43], the ranking method [44], and the fuzzy-Pareto-dominance [45]. However, all above mentioned dominance still work unsatisfactorily on ManyOPs. Furthermore, as the complexity of nondominated sort increases with the number of objectives, several improved sorts (i.e., nondominated rank sort [46], corner sort [47], and deductive sort [48]) have been proposed to decrease that complexity.

Recently, nondominated sorting genetic algorithm (NSGA)-III [49], an improved NSGA-II for ManyOPs, was proposed. In order to improve the disadvantages of NSGA-II from the Pareto dominance, a set of reference points is used in the diversity maintenance of NSGA-III. NSGA-III adopts minimal perpendicular distances to these reference points as a measure in selecting individuals. In other words, NSGA-III assigns a set of uniformly distributed reference points in advance, which act as a reference distribution for the final output. Although the experimental results [49] showed that NSGA-III had good convergence, satisfactory diversity, and acceptable complexity on the ManyOPs whose objectives cannot be reduced, it relies crucially on the reference points, which have to be provided in advance.

In this paper, we aim to design an MOEA for ManyOPs with good performance on convergence, diversity, and complexity without any reference points or other manual settings in advance. One of the main ideas is to maintain two archives during evolutionary search [50], where each archive promotes convergence (CA) and diversity (DA) separately. However, the original Two-Archive algorithm (Two_Arch) [50] did not work as well as expected on ManyOPs with a large number of objectives though the idea of two archives is attractive. We improve substantially the original algorithm and introduce two main innovations, including assigning different selection principles to the two archives (CA is indicator-based, and DA is Pareto-based), designing a new L_p-norm-based ($p < 1$) diversity maintenance scheme for ManyOPs. These new techniques have been incorporated into Two_Arch2, the new algorithm in this paper.

The remainder of the paper is organized as follows. We firstly introduce Two_Arch in Section II. After that, we describe our algorithm Two_Arch2 in Section III. In Section IV, several different kinds of MOEAs for ManyOPs are included in our experimental studies and comparisons. Detailed discussions and analyses are also given in this section to explain why those MOEAs behaved the way we observed in the experiments. Finally, Section V gives the conclusion and future work.

II. Two-Archive Algorithm

Two_Arch [50] is the first MOEA that divides the nondominated solution set into two archives for convergence and diversity separately. Thus, two main goals, convergence and diversity, are both emphasized. We will show how these archives work in Two_Arch in the following subsections.

A. Basic Flow

The flow-chart of Two_Arch is shown in Fig. 1, whose basic framework is similar to general MOEAs (reproduction and iteration). The nondominated solution set is divided into two archives (CA and DA). For the reproduction step (crossover and mutation), the union of CA and DA can be viewed as a parent population comparable to the one of any other MOEA. There is no special parent selection in Two_Arch. Additionally, CA and DA have different updating rules because of different aims (convergence and diversity). After obtaining the offspring by crossover and mutation, Two_Arch uses the nondominated offspring to update both CA and DA. The nondominated offspring that dominate any solution in either CA or DA (the nondominated solution with domination) are added to CA, and nondominated offspring that dominate no solution in both CA and DA (the nondominated solution without domination)

Algorithm 1 Pseudo-Code of the Update of CA and DA in Two_Arch

```
Parameters: $O$-offspring, $A_C$-CA, $A_D$-DA, $N$-the total size of $A_C$ and $A_D$.
Set $O_{nd}$ the non-dominated set of $O$.
For $i = 1{:}|O_{nd}|$
    If $O_{nd}[i]$ cannot be dominated by any solution in either $A_C$ or $A_D$
        If $O_{nd}[i]$ dominates any solution in either $A_C$ or $A_D$
            The dominated solutions in $A_C$ and $A_D$ are deleted.
            $O_{nd}[i].flag = 1$. the non-dominated solution with domination
        Else
            $O_{nd}[i].flag = 0$. the non-dominated solution without domination
        End
    Else
        $O_{nd}[i].flag = -1$. the dominated solution
    End
End
Add the solutions with $flag == 1$ to $A_C$ and the solutions with $flag == 0$ to $A_D$.
If $|A_C| + |A_D| > N$ the total size of $A_C$ and $A_D$ overflows
    Delete the extra solutions from $A_D$ with minimal distances to $A_C$.
End
```

are added to DA. In this process, any dominated solution in CA, DA, and the nondominated offspring set is deleted. The details of the update of CA and DA are shown in Algorithm 1. Two_Arch removes extra solutions from DA according to their distances to CA. In Two_Arch, CA and DA are regarded as a whole for the solution set, and there is only a fixed size for the union of CA and DA. In other words, the sizes of CA and DA are flexible. The reason why Two_Arch never removes any extra solution from CA is that the size of CA never overflows the limit of the union of CA and DA, because when the number of solutions in CA reaches the total size of DA and CA, the whole DA is abandoned, thus any solution that can be added to CA would dominate and eliminate at least one solution in CA.

B. Strengths and Drawbacks

Two_Arch is the first MOEA to separate the two goals of MOPs (convergence and diversity) directly. CA encourages the convergence, and DA maintains the diversity. As the idea is very simple, almost no additional complexity is introduced. However, it is a Pareto-based MOEA after all, it is ineffective in handling ManyOPs with a large number of objectives. Another drawback of Two_Arch is that there is no diversity maintenance within CA, though the role of CA is for convergence. In the case that all the solutions in CA are on the true PF and the size of CA has reached the limit of the union of CA and DA, there is no room for any additional member of DA. It is impossible to find solutions to dominate any solution in CA (because they are nondominated solutions). Thus, all the new offspring are nondominated solutions without domination, which are assigned to DA. However, the total archive of CA and DA is full, no solution in CA can be deleted, the entire DA is deleted. Two_Arch is stuck here without any update of CA and DA. The final output would be a CA with less satisfactory diversity.

III. Proposed New Algorithm : Two_Arch2

A. Main Idea

In order to solve ManyOPs with good convergence, satisfactory diversity, and acceptable complexity, only adopting the Pareto dominance relation is insufficient. We find that the quality indicator $I_{\varepsilon+}$ in IBEA [19] can encourage a good performance on convergence and the Pareto dominance can promote a good performance on diversity. Therefore, we follow the idea of two archives in Two_Arch, but update CA and DA independently by different dominance relations, which makes our Two_Arch2 a hybrid MOEA shown in Fig. 2. In Two_Arch2, the roles of CA and DA are clearer than those in Two_Arch. CA aims to guide the population to converge to the true PF at a fast speed, and DA aims to add more diversity to the population in a high-dimensional objective space. That is why Two_Arch2 makes crossover between CA and DA but mutation on CA only during the process of reproduction (crossover and mutation are independent in Two_Arch2). The sizes of CA and DA are individually fixed and no longer flexible. The selection of offspring to CA and DA are independent of each other by different methodologies, whose details are shown in the following subsections. As the diversity of CA is too poor, we use DA as the final output, which is different from the union set of CA and DA in Two_Arch.

B. Convergence Archive

Since the Pareto dominance loses its performance of convergence on ManyOPs, we choose the quality indicator $I_{\varepsilon+}$ in IBEA [19] as the selection principle for CA in our Two_Arch2. $I_{\varepsilon+}$ is an indicator that describes the minimum distance that one solution needs in order to dominate another solution in the objective space, which is shown by (1) (where m is the number of objectives). According to this indicator, we assign the fitness [shown in (2)] to individuals as done in IBEA. Equation (2) is a measure of the $I_{\varepsilon+}$ if x_1 is deleted from the population

$$I_{\varepsilon+}(x_1, x_2) = \min_{\varepsilon}(f_i(x_1) - \varepsilon \leq f_i(x_2),\ 1 \leq i \leq m) \quad (1)$$

$$F(x_1) = \sum_{x_2 \in P \setminus \{x_1\}} -e^{-I_{\varepsilon+}(x_2, x_1)/0.05}. \quad (2)$$

In the step of updating CA in Two_Arch2, offspring are firstly added to CA. Then, Two_Arch2 deletes the extra solutions in CA according to the fitness. In each iteration, the solution with the smallest $I_{\varepsilon+}$ loss is removed from CA, and the $I_{\varepsilon+}$ values of the remaining members of CA are updated as shown in Algorithm 2. Finally, an updated CA with a fixed number of solutions can be obtained.

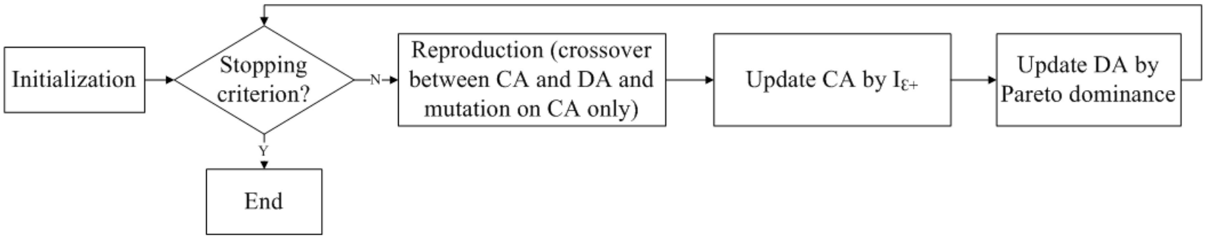
Initialization
Stopping criterion?
N
Y
Reproduction (crossover between CA and DA and mutation on CA only)
Update CA by $I_{\varepsilon+}$
Update DA by Pareto dominance
End

Fig. 2. Flow chart of Two_Arch2.

Algorithm 2 Pseudo-Code of the Selection in an Overflowed CA

```
Parameters: $A_C$-CA, $n_{CA}$-the fixed size of $A_C$.
While $|A_C| > n_{CA}$
    Find the individual $x^*$ with the minimal $F(x^*)$.
    Delete $x^*$ from $A_C$.
    Update the remaining individual by $F(x) = F(x) + e^{-I_{\varepsilon+}(x^*,x)/0.05}$.
End
```

Algorithm 3 Pseudo-Code of the Selection in an Overflowed DA

```
Parameters: $A_D^*$-the overflowed DA, $n_{DA}$-the fixed size of DA, $A_D$-output.
Set $A_D$ empty.
Find solutions with maximal or minimal objective values in $A_D^*$, and move them to $A_D$ from $A_D^*$.
While $A_D$ is not full
    For each member $i$ in $A_D^*$
        $Similarity[i] = \min(distance(i, j)), j \in A_D$
    End
    $I = argmax(Similarity)$, move solution $I$ from $A_D^*$ to $A_D$
End
```

C. Diversity Archive

The diversity of ManyOPs are hard to be maintained. The diversity of ManyOPs is two-fold. On one hand, solutions should be distributed in the whole high-dimensional objective space to provide the information of PF as much as possible. On the other hand, the differences among solutions should be maximized when those solutions are projected to a low-dimensional objective space [51]. Therefore, DA is key to the diversity in Two_Arch2.

1) Selection in DA: In Two_Arch2, the updating of DA is based on the Pareto dominance. That means only nondominated solutions can be added to DA. The extra solutions are removed according to similarity if DA overflows, which is the common idea of Pareto-based MOEAs. In our Two_Arch2, we adopt the inversion of that idea, selecting rather than deleting. The pseudo-code of the selection in an overflowed DA is shown in Algorithm 3. When DA overflows, boundary solutions (solutions with maximal or minimal objective values) are firstly selected. Then, it comes to the iterative process where the most different solution to the selected solutions is added to DA in each iteration. We also adopt distances as the similarity measure in DA, which is the same as most diversity maintenance approaches. As we know, such diversity maintenance approaches based on distances cannot work well on ManyOPs. We find that the reason of their failure is the improper use of distances (the Euclidean or Manhattan distances) for ManyOPs [52]. Therefore, we need to choose suitable distances (L_p-norm-based distances) to improve the diversity of ManyOPs.

2) Similarity in High-Dimensional Space: The Euclidean distance (L_2-norm) in a high-dimensional space has been proved qualitatively less meaningful by both theories and experiments in [53] and [54]. The Euclidean distance degrades its similarity indexing performance in a high-dimensional space. In contrast, the Manhattan distance (L_1-norm) and fractional distances (L_p-norm, $p < 1$) perform better in a high-dimensional space. Following [53], we increase the number of dimensions of a uniform data set and calculate the difference $d_{\max} - d_{\min}$ between the nearest and farthest neighbors. The results of $d_{\max} - d_{\min}$ under L_p-norm-based ($p = 2$, $p = 1$, $p = 0.75$, $p = 0.5$, $p = 0.25$, and $p = 0.1$) distances with increasing dimensions are shown in Fig. 3. Under L_2-norm, $d_{\max} - d_{\min}$ does not increase with the number of dimensions. It is bounded by a constant. Whereas in all other cases $d_{\max} - d_{\min}$ increases with the increasing dimensions. It is theoretically proved that the dependency of $d_{\max} - d_{\min}$ on $Dimension^{1/p-1/2}$ is generally unchanged in [53]. That is the reason why L_2-norm is bounded by a constant in a high-dimensional space. As the conclusion in [53] shows, a smaller parameter p provides a higher contrast between the furthest and nearest neighbors. Therefore, we use the L_p-norm-based ($p < 1$) distance instead of the L_2-norm or L_1-norm-based distances in the updating step of DA in our Two_Arch2. For ManyOPs with different numbers of objectives, an L_p-norm-based ($p < 1$) distance with a fixed p may not suit all the cases. Hence, we set $p = 1/m$ (m is the number of objectives) in Two_Arch2 based on the experiment result in Section IV-B.

IV. Computational Studies

A. Test Problems, Performance Metrics, and Parameter Settings

In order to test the performance of Two_Arch2 on ManyOPs, we choose the DTLZ [55] and WFG [56] benchmark problems that have a tunable number of objectives. Among them, DTLZ1 is a representative hard-to-converge

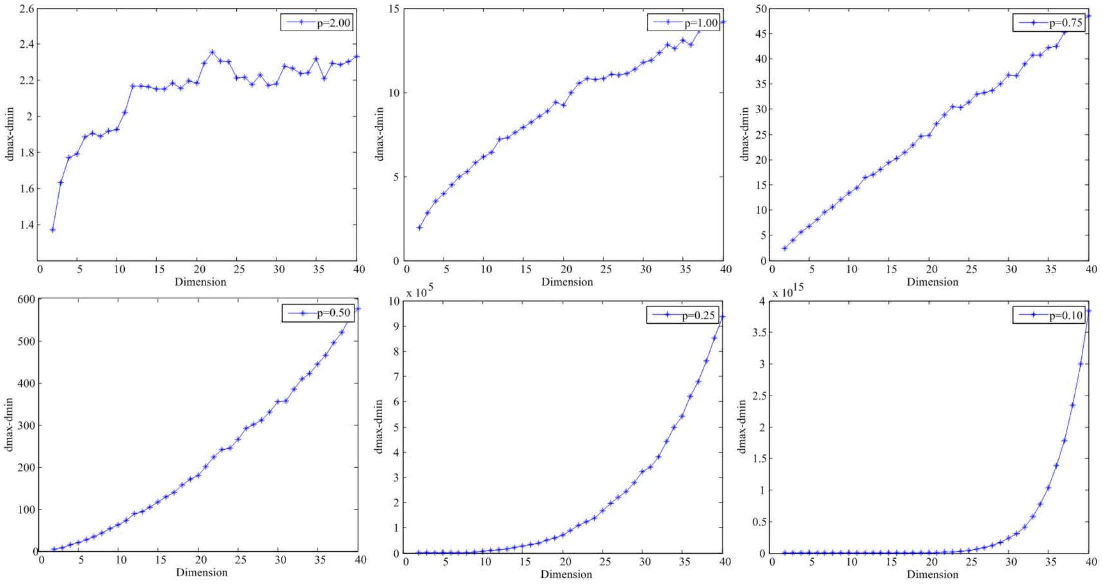

Fig. 3. $d_{\max} - d_{\min}$ for L_p-norm-based ($p = 2$, $p = 1$, $p = 0.75$, $p = 0.5$, $p = 0.25$, and $p = 0.1$) distances in uniform data with increasing dimensions.

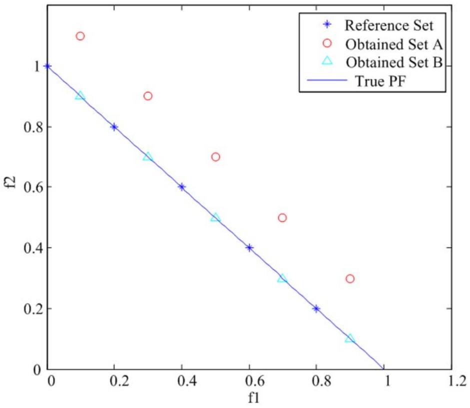

Fig. 4. Examples of effects from a limited size of the reference set on Inverted generational distance (IGD).

problem, that is the reason why we use it as the test instance to tune Two_Arch2 in our experiments. In other words, if we tune Two_Arch2 to a good state on DTLZ1, then similar performance on other problems can be observed.

Although hypervolume [22] can be used as a performance metric for ManyOPs, its computational complexity is too high for the problems with more than ten objectives. Therefore, we use IGD [57] as the performance metric in our experiments. IGD is the average distance from a uniform reference set (samples from the true PF) to the obtained set, whose value expresses both diversity and convergence. Both the size and uniformity of the reference set affect the precision significantly. Taking Fig. 4 as an example, it is clear that the IGD values of the obtained sets A and B are same. However, B is obviously better than A because of its better convergence. In this case, IGD cannot output a reasonable evaluation. Furthermore, if the reference set is an obtained set, we cannot determine the winner between B and the reference set, but the IGD value of the reference set is zero. Therefore, it is unfair to evaluate B by such a reference set. As we know, NSGA-III assigns uniform reference points in the objective space, which is coincidentally similar to the generation method for the reference set in IGD. In order to avoid this effect, all the reference sets for calculating IGD are in the size of 500 000 (actually, the most approximated C^q_{m+q-1} to 500 000 to guarantee the uniformity on the PF [16], [49], where m is the number of objectives, q is the number of divisions on each objective, and C^q_{m+q-1} is the total number of solutions in the reference set). Also, we generated the reference set uniformly on their PFs rather than their Pareto sets. Taking some samples in a small amount in Fig. 5 as examples, the uniformity is based on the structure of the PF (hyper-plane and hyper-sphere). Additionally, Convergence $g(x)$ is a part of the DTLZ problems, hence, we adopt it as a metric to analyze the convergence ability of Two_Arch2 as in [29].

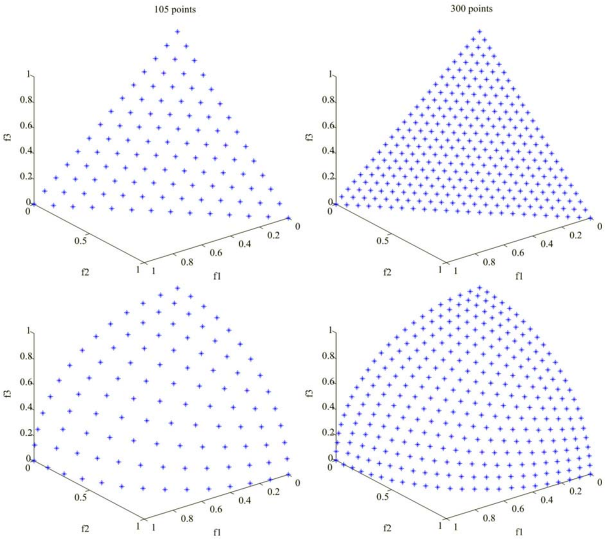

Fig. 5. Samples of the reference set with 105 and 300 points for IGD.

In the following experiments, all algorithms are terminated after 90 000 function evaluations (we deal with the problems

TABLE I

AVERAGE IGD VALUES OF TWO_ARCH2 WITH DIFFERENT L_p-NORM-BASED DISTANCES ($p = 1/m$, $p = 2/m$, $p = 2$, $p = 1$, $p = 0.75$, $p = 0.5$, $p = 0.25$, $p = 0.1$, AND $p = 0.05$) ON DTLZ1 WITH 2–10 OBJECTIVES. THE NUMBERS IN BRACKETS ARE THE RANKING NUMBERS. L_p-NORM-BASED DISTANCES OF RANKS 1 AND 2 ARE ANALYZED BY WILCOXON SIGNED-RANK TEST

Obj #	$1/m$	$2/m$	2	1	0.75	0.5	0.25	0.1	0.05	p-value of algorithms of ranks 1 and 2
2	0.0041(5)	0.0041(6)	0.0042(8)	0.0040(2)	0.0041(4)	0.0041(3)	0.0042(9)	0.0041(7)	0.0040(1)	0.4922
3	0.0446(1)	0.0685(9)	0.0494(2)	0.0594(7)	0.0572(6)	0.0517(3)	0.0553(5)	0.0647(8)	0.0529(4)	0.0273
4	0.0880(1)	0.1178(9)	0.1130(7)	0.1148(8)	0.1007(4)	0.1065(6)	0.0970(3)	0.1037(5)	0.0953(2)	0.0273
5	0.1424(1)	0.1524(5)	0.1631(8)	0.1530(6)	0.1473(2)	0.1652(9)	0.1593(7)	0.1506(4)	0.1475(3)	0.6953
6	0.1781(2)	0.1795(4)	0.1916(7)	0.1791(3)	0.1855(6)	0.1707(1)	0.1920(8)	0.1967(9)	0.1830(5)	0.6250
7	0.2167(2)	0.2224(5)	0.2635(9)	0.2550(8)	0.2335(7)	0.2185(4)	0.2141(1)	0.2263(6)	0.2168(3)	0.6250
8	0.2266(1)	0.2310(2)	0.2866(8)	0.3091(9)	0.2772(7)	0.2598(6)	0.2330(3)	0.2380(5)	0.2343(4)	0.4316
9	0.2458(1)	0.2695(4)	0.3455(8)	0.3531(9)	0.3440(7)	0.3265(6)	0.2785(5)	0.2617(2)	0.2693(3)	0.1602
10	0.2791(1)	0.2859(4)	0.4200(9)	0.3744(8)	0.3525(7)	0.3096(6)	0.2970(5)	0.2792(2)	0.2798(3)	0.4922

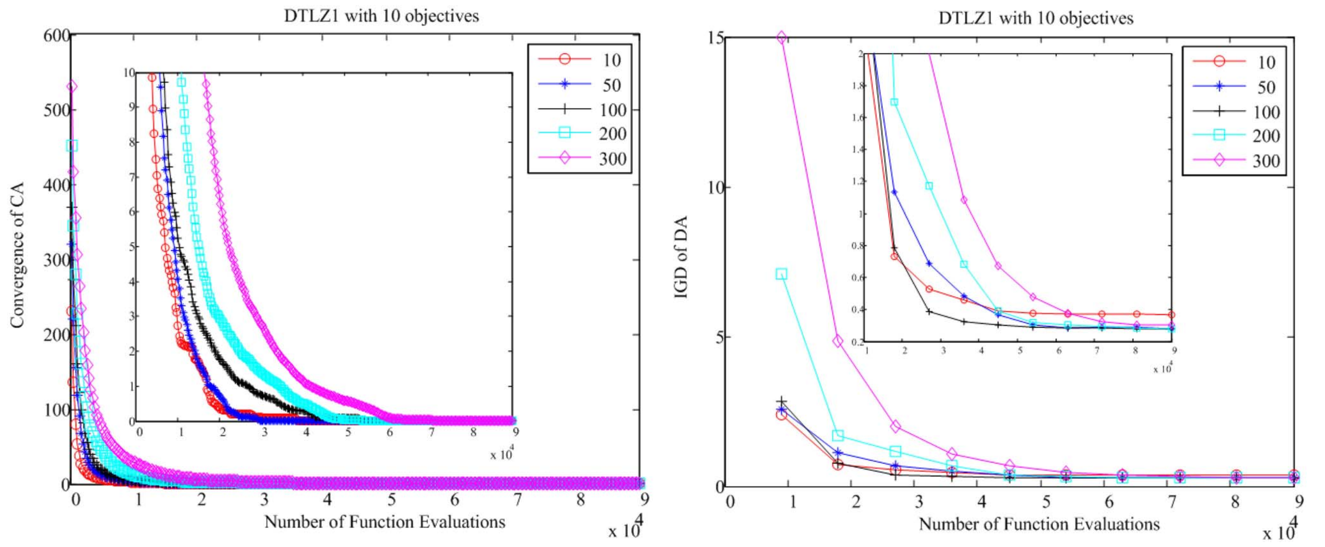

Fig. 6. Average performance of Two_Arch2s with CA of different sizes (10, 50, 100, 200, 300) over generations on DTLZ1 with 10 objectives.

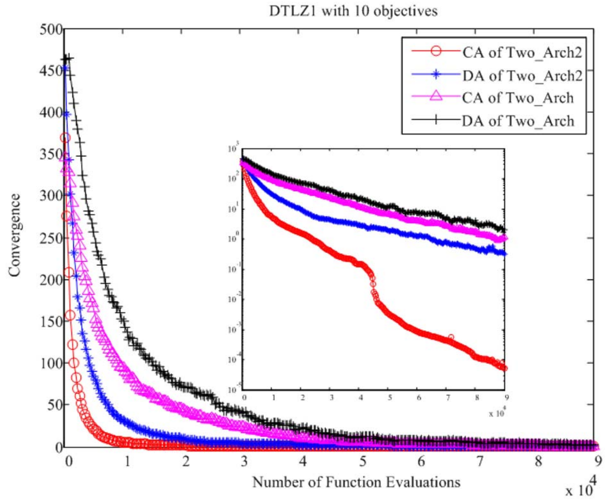

Fig. 7. Average convergence values of CA and DA of Two_Arch2 and Two_Arch during varied function evaluations on DTLZ1 with ten objectives.

with up to 20 objectives, hence, the number of function evaluations needs to be larger than the common setting for MOPs with low-dimensional objectives) and use the same reproduction operators [SBX crossover ($\eta = 15$) and polynomial mutation ($\eta = 15$)]. The population size is set as 200 for the problems with more than ten objectives, and 100 for others. All the experiments are repeated for 30 independent runs.

B. Experiments on L_p-Norm-Based Distances for Diversity Archive Maintenance

In this section, we analyze the setting of p for L_p-norm-based distances in Two_Arch2. We test Two_Arch2 with different L_p-norm-based distances ($p = 1/m$, $p = 2/m$, $p = 2$, $p = 1$, $p = 0.75$, $p = 0.5$, $p = 0.25$, $p = 0.1$, and $p = 0.05$) on DTLZ1 with 2–10 objectives. The results of IGD are shown in Table I. We find that the $L_{1/m}$-norm-based distance performs very well among different L_p-norm-based distances except for the case of DTLZ1 with two objectives. With the increasing number of objectives, the $L_{1/m}$-norm and $L_{0.1}$-norm-based distances perform similarly, because $1/m$ approximates 0.1. Overall, the $L_{1/m}$-norm-based distance is more robust to different ManyOPs. Therefore, it is reasonable to set $p = 1/m$ in our Two_Arch2.

C. Experiments on Archive Size

In Two_Arch2, DA is used as the final output with a fixed size (200 for the problems with more than ten objectives,

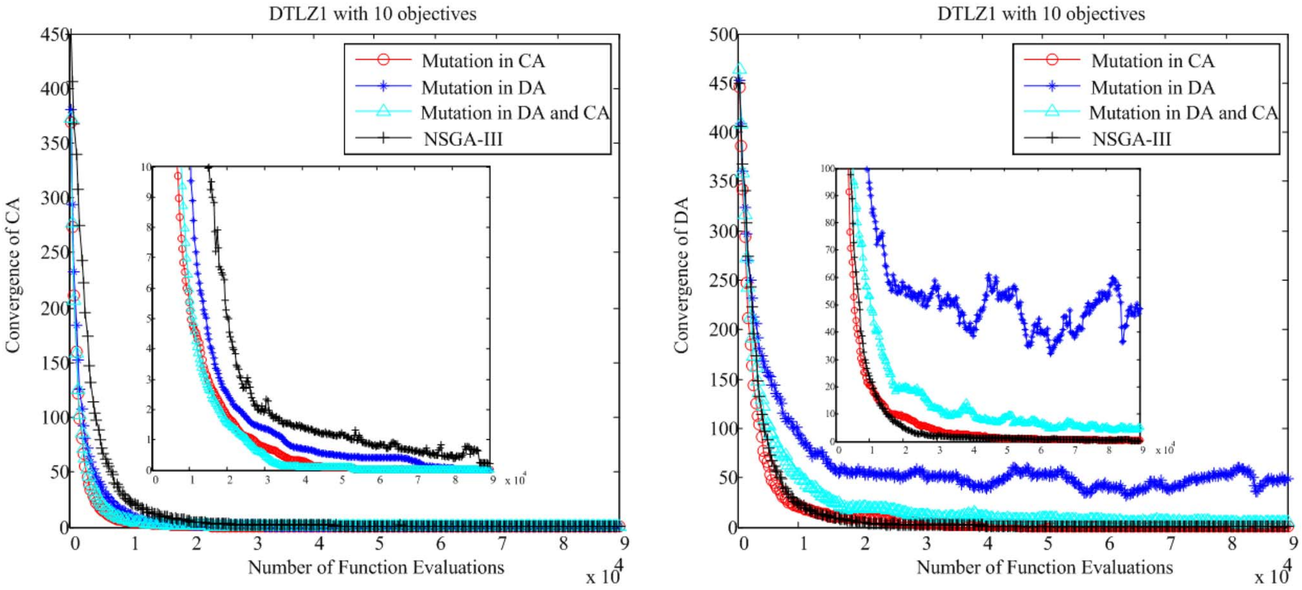

Fig. 8. Average performance of Two_Arch2 with the mutation on different archives (CA, DA, and the union of CA and DA) over generations on DTLZ1 with 10 objectives, where the result of NSGA-III acts as a reference.

and 100 for others), while CA guides DA to converge. Therefore, the size of CA affects the behavior of our algorithm. Two_Arch2s with CA of different sizes (10, 50, 100, 200, 300) are tested on DTLZ1 with ten objectives. All the algorithms repeat for 30 independent times and stop by 90 000 function evaluations. Fig. 6 shows the Convergence values of CA and the IGD values of DA over generations. A smaller size of CA can increase the convergence speed, because the search focuses on a small number of good solutions. However, CA with a small size (10 for instance) cannot result in good diversity in DA, which is reflected by the large IGD value of DA. Therefore, it is suitable to set CA with 100 solutions in Two_Arch2.

D. Experiments on Convergence

In this subsection, we evaluate and analyze the role of CA in Two_Arch2. We run Two_Arch2 and Two_Arch on DTLZ1 with ten objectives for 30 independent times and record the values of Convergence of CA and DA in every generation. The average Convergence values of CA and DA over generations in 30 independent runs are shown in Fig. 7. We find that CA in Two_Arch2 has a faster convergence speed than CA in Two_Arch. This is because that $I_{\varepsilon+}$ in CA of Two_Arch2 works better than the Pareto dominance in CA of Two_Arch in terms of convergence for ManyOPs. Even DA in Two_Arch2 that is Pareto-based has a faster convergence speed than CA in Two_Arch, which is because that DA follows the convergence tendency of CA in Two_Arch2. Therefore, $I_{\varepsilon+}$ works well in promoting convergence in CA, and CA plays a role in guiding DA to converge to the true PF.

E. Experiments on Variation

CA and DA play different roles in Two_Arch2. CA is the guidance of convergence. Thus, we focus on the convergence speed of CA rather than its diversity. DA acts as the final result of Two_Arch2. Hence, both the convergence and diversity of DA are important. The interaction between CA and DA affects the efficiency of Two_Arch2. In this section, we analyze the effects of variation operators (mutation and crossover) in different archives. All the algorithms run on DTLZ1 with ten objectives for 30 independent times.

1) Mutation: The task of the mutation in Two_Arch2 is to cooperate with $I_{\varepsilon+}$ to avoid stagnation in CA. We do the experiment on the Two_Arch2s with the mutation on different archives (CA, DA, and the union of CA and DA) to show the effect.

Fig. 8 shows the average performance of Two_Arch2 with the mutation on different archives over generations on DTLZ1 with ten objectives, where the result of NSGA-III acts as a reference. For the metric Convergence of CA, all the different Two_Arch2s perform better than NSGA-III because of the effective $I_{\varepsilon+}$. The mutation applied to DA only cannot provide a faster convergence speed than other two Two_Arch2s, because it excludes CA and distracts the search of CA. As the mutation on CA can prevent prematurity, the performance of the mutation applied to CA and the union of CA and DA is similar. Fig. 8 also shows the Convergence value of DA to analyze the influence on the final result. The mutation applied to CA only can output a DA with better convergence than that applied to the union of CA and DA. The reason is that the mutation for some members of DA disturbs the guidance of CA to DA. Therefore, we apply the mutation to CA only in Two_Arch2, because it can provide the fastest convergence speed and best guidance to DA.

2) Crossover: The task of the crossover in Two_Arch2 is to lead DA to converge by CA. We do the experiment on the Two_Arch2s with the crossover of different parent selections (between CA and DA, the union of CA and DA, CA only, and DA only) to show the effect.

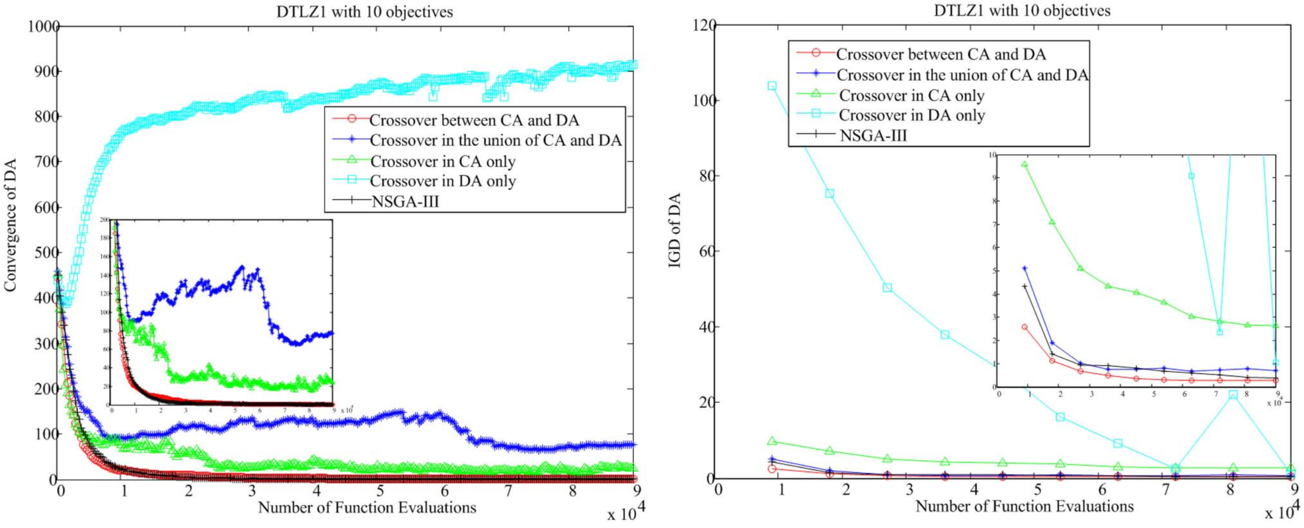

Fig. 9. Average performance of Two_Arch2 with the crossover of different parent selections (between CA and DA, the union of CA and DA, CA only, and DA only) over generations on DTLZ1 with ten objectives, where the result of NSGA-III acts as a reference.

TABLE II
IGD VALUES OF TWO_ARCH2, TWO_ARCH, IBEA, NSGA-III, MOEA/D, AND AGE-II ON DTLZ1 ANALYZED BY WILCOXON SIGNED-RANK TEST. THE SIGNIFICANT RESULTS ARE IN BOLD FACE (SIGNIFICANT LEVEL = 0.05). THE NUMBERS IN BRACKETS ARE THE RANKING NUMBERS

Obj #	Two_Arch2	Two_Arch	IBEA	NSGA-III	MOEA/D	AGE-II	p-value of algorithms of ranks 1 and 2
2	0.0041±0.0001(2)	0.0083±0.0069(3)	1.1338±0.7452(6)	**0.0037±0.0002(1)**	0.7096±0.9932(5)	0.0449±0.0047(4)	0.0000
3	0.0644±0.0482(4)	0.0541±0.0065(2)	0.9325±0.5065(6)	**0.0409±0.0008(1)**	0.6040±0.8815(5)	0.0592±0.0025(3)	0.0000
4	0.1197±0.0452(2)	0.1205±0.0097(3)	0.9079±0.2236(6)	0.1001±0.0126(1)	0.6798±0.8270(5)	0.1930±0.0025(4)	0.1529
5	0.1527±0.0298(1)	0.2095±0.0192(3)	1.0728±0.3842(6)	0.1554±0.0034(2)	0.4192±0.5544(5)	0.2211±0.0023(4)	0.1846
6	**0.1802±0.0327(1)**	0.4297±0.0946(5)	1.0977±0.4282(6)	0.1981±0.0043(2)	0.4237±0.4470(4)	0.2574±0.0068(3)	0.0098
7	**0.2143±0.0397(1)**	0.5437±0.0824(4)	1.1288±0.5626(6)	0.2397±0.0112(2)	0.6106±0.6262(5)	0.2871±0.0062(3)	0.0003
8	**0.2402±0.0246(1)**	0.5586±0.1025(4)	1.1644±0.5625(6)	0.2639±0.0119(2)	0.5804±0.5085(5)	0.3116±0.0043(3)	0.0003
9	**0.2571±0.0212(1)**	0.6387±0.1528(5)	2.3723±5.2262(6)	0.4137±0.2178(3)	0.5793±0.4529(4)	0.3581±0.0070(2)	0.0000
10	**0.2879±0.0247(1)**	0.7517±0.4126(5)	2.1616±4.9046(6)	0.3931±0.2019(2)	0.5004±0.4657(4)	0.4095±0.0040(3)	0.0000

Fig. 9 shows the average performance of Two_Arch2 with crossover of different parent selections over generations on DTLZ1 with ten objectives, where the result of NSGA-III acts as a reference. The crossover in the Pareto-based DA only has the worst Convergence value, because the Pareto dominance cannot help the convergence of ManyOPs. The crossover in CA only has the worst IGD value, because the diversity of CA is too poor to improve IGD. The crossover between CA and DA (one parent from CA and the other from DA) performs much better than that in the union of CA and DA (two parents are chosen randomly from the union) on both the Convergence and IGD values of DA, and the Convergence value of the crossover between CA and DA even closes to that of NSGA-III. The reason is that the crossover between CA and DA can efficiently pass the good convergence of CA to DA. Therefore, we apply the crossover between CA and DA in Two_Arch2.

F. Comparison Experiments

In order to analyze the behavior of our algorithm, we compare Two_Arch2 with Two_Arch, IBEA, NSGA-III, MOEA/D (with penalty-based boundary intersection method and $T = 50$) [16], and AGE-II ($\varepsilon = 0.1$) [42] (an updated version of AGE [41]) on 13 ManyOPs with different numbers of objectives (the DTLZ [55] and WFG [56] benchmark problems).

The compared algorithms are all representatives of MOEAs for ManyOPs. Two_Arch is chosen as a reference to show whether there is any improvement of Two_Arch2 on ManyOPs. IBEA is an MOEA with a fast convergence speed but poor diversity on ManyOPs, it is chosen to show the improvement of Two_Arch2, because both use $I_{\varepsilon+}$ in their algorithms. NSGA-III is a newly-proposed MOEA with reference points for ManyOPs; MOEA/D uses aggregation functions on ManyOPs; AGE-II approximates the objective space by ε-grid. We compare them with Two_Arch2 to analyze the behaviors of their different methodologies. All the experiment settings have been shown in Section IV-A.

1) Experiments on the DTLZ Problems: For the DTLZ problems, IGD is used for evaluating the results. The results of Two_Arch2, Two_Arch, IBEA, NSGA-III, MOEA/D, and AGE-II on the DTLZ problems with 2–10 objectives are shown in Tables II–V, where the algorithms of ranks 1 and 2 are analyzed by Wilcoxon signed-rank test [58]. For the DTLZ problems with low-dimensional objectives (2–3 objectives), NSGA-III has the best IGD value. While Two_Arch2 has the best IGD value when the number of objectives increases.

Table II shows the IGD values of compared algorithms on DTLZ1. IBEA has the worst IGD values. NSGA-III and Two_Arch2 are the top two algorithms for all the DTLZ1. Comparing these two algorithms, NSGA-III has better IGD

TABLE III
IGD VALUES OF TWO_ARCH2, TWO_ARCH, IBEA, NSGA-III, MOEA/D, AND AGE-II ON DTLZ2 ANALYZED BY WILCOXON SIGNED-RANK TEST. THE SIGNIFICANT RESULTS ARE IN BOLD FACE (SIGNIFICANT LEVEL = 0.05). THE NUMBERS IN BRACKETS ARE THE RANKING NUMBERS

Obj #	Two_Arch2	Two_Arch	IBEA	NSGA-III	MOEA/D	AGE-II	p-value of algorithms of ranks 1 and 2
2	0.0044±0.0000(3)	0.0055±0.0008(4)	0.5689±0.2498(6)	0.0042±0.0001(1)	0.0042±0.0000(2)	0.0781±0.0017(5)	0.2289
3	0.0556±0.0008(2)	0.0636±0.0021(4)	0.6812±0.3210(6)	**0.0541±0.0001(1)**	0.0557±0.0008(3)	0.0953±0.0043(5)	0.0000
4	**0.1334±0.0015(1)**	0.1483±0.0044(4)	0.7872±0.2838(6)	0.1496±0.0008(5)	0.1447±0.0026(3)	0.1379±0.0035(2)	0.0000
5	**0.2063±0.0023(1)**	0.2321±0.0058(3)	0.9507±0.2489(6)	0.2340±0.0002(4)	0.2373±0.0041(5)	0.2142±0.0038(2)	0.0000
6	**0.2733±0.0023(1)**	0.3029±0.0053(3)	1.0183±0.1981(6)	0.3112±0.0003(4)	0.3193±0.0033(5)	0.2936±0.0094(2)	0.0000
7	**0.3330±0.0028(1)**	0.3574±0.0078(3)	1.1107±0.1922(6)	0.3951±0.0023(5)	0.3852±0.0046(4)	0.3555±0.0059(2)	0.0000
8	**0.3853±0.0027(1)**	0.4092±0.0097(2)	1.1709±0.0924(6)	0.4313±0.0002(4)	0.4404±0.0033(5)	0.4164±0.0199(3)	0.0000
9	**0.4330±0.0044(1)**	0.4516±0.0075(2)	1.1905±0.0938(6)	0.5301±0.0077(5)	0.4859±0.0042(4)	0.4672±0.0210(3)	0.0000
10	**0.4805±0.0045(1)**	0.4936±0.0081(2)	1.1929±0.0795(6)	0.5551±0.0056(5)	0.5244±0.0054(4)	0.5077±0.0238(3)	0.0000

values than Two_Arch2 on DTLZ1 with two and three objectives, while Two_Arch2 has better IGD values than NSGA-III on DTLZ1 with more objectives. Following Two_Arch2 and NSGA-III, AGE-II, Two_Arch, and MOEA/D have worse performance. However, AGE-II has better performance than Two_Arch and MOEA/D on DTLZ1 with more than five objectives.

Table III shows the IGD values of compared algorithms on DTLZ2. IBEA has the worst IGD values on all the DTLZ2 problems because of poor diversity. For the problems with 2–3 objectives, NSGA-III has the best IGD values, then Two_Arch2 and MOEA/D follow, Two_Arch and AGE-II are in ranks 4 and 5. However, the situation changes when the number of objectives increases. Two_Arch2 becomes the best-performing algorithm, then Two_Arch and AGE-II follow, NSGA-III and MOEA/D are in ranks 4 and 5.

Table IV shows the IGD values of compared algorithms on DTLZ3. NSGA-III and Two_Arch2 are the best-performing algorithms for DTLZ3 with low-dimensional objectives and high-dimensional objectives correspondingly. For DTLZ3 with 4–10 objectives, Two_Arch and AGE-II have slightly worse IGD values than that of NSGA-III and Two_Arch2. IBEA and MOEA/D have the IGD values in ranks 5 and 6.

Table V shows the IGD values of compared algorithms on DTLZ4. IBEA and MOEA/D have the worst performance on all the DTLZ4 problems. For DTLZ4 with 2–3 objectives, AGE-II is the best-performing algorithm, followed by NSGA-III (rank 2), Two_Arch2 (rank 3), and Two_Arch (rank 4). For DTLZ4 with 4–5 objectives, NSGA-III is the best-performing algorithm, followed by AGE-II, Two_Arch2, and Two_Arch. Whereas, for DTLZ4 with 6–8 objectives, Two_Arch2 is the best-performing algorithm, followed by NSGA-III. For DTLZ4 with 9–10 objectives, NSGA-III cannot perform better than AGE-II and Two_Arch.

From the above results of compared algorithms on the DTLZ problems, we can conclude the characteristics of those different algorithms. IBEA can obtain the solution set with the best convergence even for hard-to-converge problems DTLZ1 and DTLZ3, but its poor diversity (its solution set concentrates in some small areas), which is reflected by its poor IGD values. AGE-II uses ε-grid to approximate the objective space, which conduces to a better converge speed on ManyOPs. However, it is still a Pareto-based MOEA, its performance is still less than satisfactory. Because of the decomposition of original problems, the diversity of MOEA/D is much better than that of IBEA.

In summary, Two_Arch2 and NSGA-III are the two best-performing algorithms on the DTLZ problems, but Two_Arch2 outperforms NSGA-III on IGD of ManyOPs. It is very hard to explain the reason according to IGD only. Therefore, we show the parallel coordinate plots of the best solution sets obtained by Two_Arch2 and NSGA-III on the DTLZ problems with ten objectives in Fig. 10. We find that both algorithms converge to the true PFs, even for DTLZ1 and DTLZ3. In addition, both algorithms obtain the solution sets which extend to the entire PF rather than in small areas. Especially, NSGA-III can keep extreme points very well, which was pointed out by [49]. In this aspect, Two_Arch2 performs slightly worse than NSGA-III. However, the diversity of Two_Arch2 is much better than NSGA-III. That is the reason why Two_Arch2 can obtain better IGD values than NSGA-III on ManyOPs. Further analysis of the compared algorithms will be shown in Section IV-F3.

2) Experiments on the WFG Problems: Also for the WFG problems, IGD is used for evaluating the results. The results of Two_Arch2, Two_Arch, IBEA, NSGA-III, MOEA/D, and AGE-II on the WFG problems with 2–10 objectives are shown in Tables VI–XIV, where the algorithms of ranks 1 and 2 are analyzed by Wilcoxon signed-rank test [58]. The performance of Two_Arch2 on the WFG problems is worse than that on the DTLZ problems. Two_Arch2 only outperforms other compared algorithms on WFG3, WFG4, and WFG5, while NSGA-III outperforms other compared algorithms on WFG1, WFG7, WFG8, and WFG9. Other compared algorithms have worse performance on the WFG problems than Two_Arch2 and NSGA-III.

WFG1 is the problem with the most transformation functions t among the WFG problems, which makes it hard to achieve good diversity. Table VI shows the IGD values of compared algorithms on WFG1. MOEA/D, AGE-II, and IBEA have larger IGD values because of their poor diversity. NSGA-III has the smallest IGD values, then Two_Arch2 and Two_Arch rank after it. Two_Arch outperforms Two_Arch2 on WFG1 with 2–5 objectives, while Two_Arch2 outperforms Two_Arch on WFG1 with more than five objectives.

WFG2 is the only disconnected problem among the WFG problems. Table VII shows the IGD values of

TABLE IV
IGD VALUES OF TWO_ARCH2, TWO_ARCH, IBEA, NSGA-III, MOEA/D, AND AGE-II ON DTLZ3 ANALYZED BY WILCOXON SIGNED-RANK TEST. THE SIGNIFICANT RESULTS ARE IN BOLD FACE (SIGNIFICANT LEVEL = 0.05). THE NUMBERS IN BRACKETS ARE THE RANKING NUMBERS

Obj #	Two_Arch2	Two_Arch	IBEA	NSGA-III	MOEA/D	AGE-II	p-value of algorithms of ranks 1 and 2
2	0.0046±0.0003(2)	0.0092±0.0170(3)	0.7343±0.0652(5)	**0.0044±0.0007(1)**	7.4998±11.6445(6)	0.1112±0.1013(4)	0.0028
3	0.0735±0.0379(3)	0.0704±0.0091(2)	1.7752±4.4825(5)	**0.0567±0.0112(1)**	8.3691±13.6366(6)	0.1419±0.1103(4)	0.0000
4	**0.1541±0.0781(1)**	0.1679±0.0094(4)	1.0051±0.0090(5)	0.1572±0.0190(2)	10.9347±15.5760(6)	0.1639±0.0730(3)	0.0006
5	0.2497±0.0786(2)	0.3798±0.0662(4)	1.0565±0.0219(5)	0.2377±0.0178(1)	11.8563±20.3757(6)	0.2772±0.0932(3)	0.3286
6	0.3070±0.0610(1)	0.6286±0.0823(4)	1.9352±4.5036(5)	0.3154±0.0119(2)	11.9379±14.0988(6)	0.3610±0.0694(3)	0.1020
7	0.4200±0.0924(2)	0.7522±0.2000(4)	1.9651±4.4866(5)	0.4001±0.0214(1)	7.0684±12.8462(6)	0.4535±0.0550(3)	0.4165
8	0.4528±0.0598(2)	0.7513±0.0417(4)	1.1473±0.0128(5)	0.4427±0.0338(1)	8.8209±15.0124(6)	0.5120±0.0462(3)	0.6583
9	**0.5232±0.0737(1)**	0.8058±0.0795(4)	3.0921±7.3332(5)	0.7458±0.4598(3)	6.4646±10.5022(6)	0.6514±0.0428(2)	0.0000
10	**0.5578±0.0402(1)**	0.9607±0.4142(4)	1.1880±0.0040(5)	0.6112±0.1053(2)	6.6794±12.4615(6)	0.6817±0.0675(3)	0.0073

TABLE V
IGD VALUES OF TWO_ARCH2, TWO_ARCH, IBEA, NSGA-III, MOEA/D, AND AGE-II ON DTLZ4 ANALYZED BY WILCOXON SIGNED-RANK TEST. THE SIGNIFICANT RESULTS ARE IN BOLD FACE (SIGNIFICANT LEVEL = 0.05). THE NUMBERS IN BRACKETS ARE THE RANKING NUMBERS

Obj #	Two_Arch2	Two_Arch	IBEA	NSGA-III	MOEA/D	AGE-II	p-value of algorithms of ranks 1 and 2
2	0.1527±0.3016(3)	0.4002±0.3758(4)	0.7458±0.0000(6)	0.1525±0.3017(2)	0.5481±0.3336(5)	0.1241±0.1757(1)	0.1020
3	0.2207±0.2943(3)	0.3621±0.3024(4)	0.9840±0.0000(6)	0.1700±0.2436(2)	0.6440±0.3756(5)	0.0941±0.0058(1)	0.1714
4	0.3488±0.2740(3)	0.3704±0.2150(4)	1.0989±0.0000(6)	**0.1498±0.0006(1)**	0.6076±0.3606(5)	0.2010±0.1574(2)	0.0243
5	0.2701±0.1332(2)	0.3461±0.1594(4)	1.1579±0.0431(6)	0.2342±0.0001(1)	0.6523±0.2671(5)	0.2812±0.1184(3)	0.3709
6	**0.2711±0.0264(1)**	0.3906±0.1246(4)	1.2019±0.0407(6)	0.3113±0.0001(2)	0.6609±0.1893(5)	0.3631±0.1173(3)	0.0000
7	**0.3261±0.0031(1)**	0.4036±0.0464(3)	1.2398±0.0000(6)	0.3944±0.0020(2)	0.7682±0.2688(5)	0.4196±0.0574(4)	0.0000
8	**0.3784±0.0026(1)**	0.4688±0.0438(4)	1.2623±0.0000(6)	0.4315±0.0001(2)	0.8068±0.2108(5)	0.4685±0.0623(3)	0.0000
9	**0.4265±0.0033(1)**	0.5039±0.0323(3)	1.2734±0.0345(6)	0.5257±0.0053(4)	0.7155±0.1052(5)	0.5031±0.0598(2)	0.0000
10	**0.4715±0.0035(1)**	0.5387±0.0161(2)	1.2934±0.0000(6)	0.5531±0.0048(4)	0.7364±0.1101(5)	0.5446±0.0548(3)	0.0000

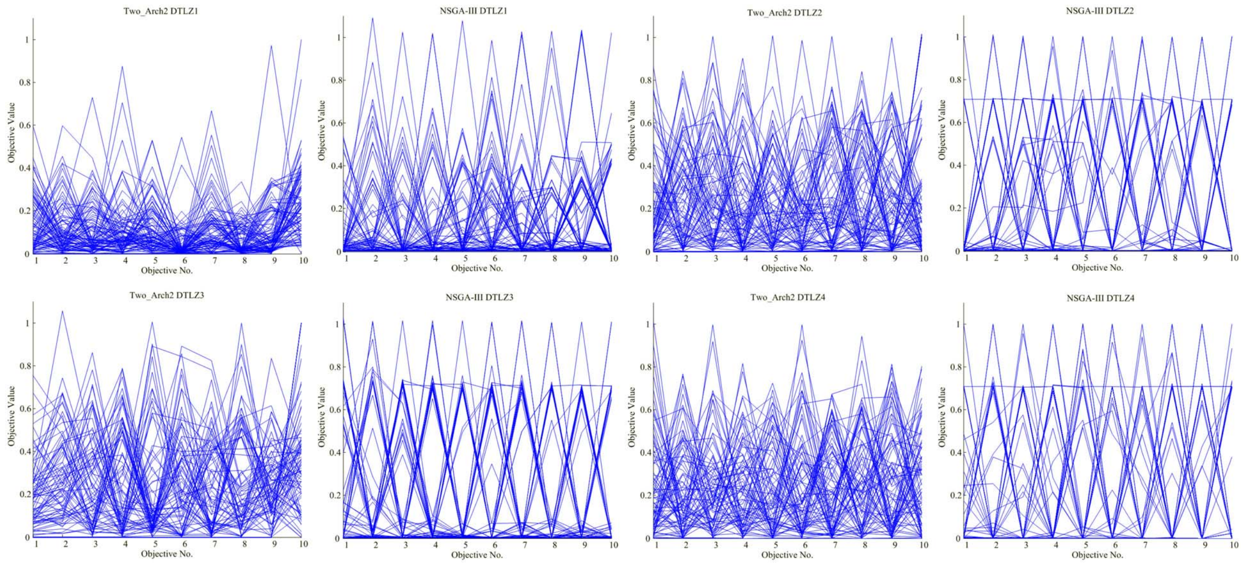

Fig. 10. Parallel coordinate plot of the best solution set obtained by Two_Arch2 and NSGA-III on the DTLZ problems with ten objectives.

compared algorithms on WFG2. MOEA/D, AGE-II, and IBEA have larger IGD values because of their poor diversity. NSGA-III and Two_Arch2 are the best two algorithms according to the IGD values of WFG2. NSGA-III outperforms Two_Arch2 on WFG2 with 2–7 objectives, while Two_Arch2 outperforms NSGA-III on WFG2 with more than seven objectives.

WFG3 is the connected version of WFG2. Table VIII shows the IGD values of compared algorithms on WFG3. MOEA/D, AGE-II, and IBEA have larger IGD values because of poor diversity. Two_Arch has better IGD values than those three algorithms, but worse than Two_Arch2 and NSGA-III. NSGA-III is the best-performing algorithm for WFG3 with two, three, and ten objectives, while Two_Arch2 outperforms NSGA-III for the remaining WFG3 problems.

Table IX shows the IGD values of compared algorithms on WFG4. WFG4 is a multimodal problems with larger "hill sizes." Because of that difficulty, aggregation functions cannot jump out of those local optima, which results in the worst IGD values of MOEA/D, whereas AGE-II and IBEA

TABLE VI
IGD Values of Two_Arch2, Two_Arch, IBEA, NSGA-III, MOEA/D, and AGE-II on WFG1 Analyzed by Wilcoxon Signed-Rank Test. The Significant Results Are in Bold Face (Significant Level = 0.05). The Numbers in Brackets Are the Ranking Numbers

Obj #	Two_Arch2	Two_Arch	IBEA	NSGA-III	MOEA/D	AGE-II	p-value of algorithms of ranks 1 and 2
2	0.7944±0.0407(3)	0.7000±0.0375(2)	0.9436±0.0249(4)	**0.5556±0.0402(1)**	1.8600±0.0571(5)	2.4421±0.0616(6)	0.0000
3	0.7548±0.0278(3)	0.7136±0.0275(2)	0.8725±0.0220(4)	**0.5870±0.0316(1)**	2.5094±0.1938(5)	3.1176±0.1252(6)	0.0000
4	0.7134±0.0485(2)	0.7235±0.0228(3)	0.8676±0.0233(4)	**0.6194±0.0304(1)**	3.7403±0.0956(5)	3.9808±0.2460(6)	0.0000
5	0.7121±0.0378(3)	0.6957±0.0282(2)	0.8507±0.0329(4)	**0.6636±0.0243(1)**	5.0041±0.0597(5)	5.1906±0.2211(6)	0.0001
6	0.6798±0.0432(2)	0.6969±0.0272(3)	0.8601±0.0213(4)	**0.6205±0.0236(1)**	6.2945±0.0567(5)	6.3868±0.2673(6)	0.0001
7	0.6663±0.0525(2)	0.6673±0.0233(3)	0.8515±0.0242(4)	**0.5919±0.0303(1)**	7.6667±0.0261(5)	7.7517±0.0176(6)	0.0000
8	0.6348±0.0432(2)	0.6971±0.0328(3)	0.9002±0.0237(4)	**0.5658±0.0263(1)**	9.2011±0.0184(6)	9.1849±0.3851(5)	0.0000
9	0.6373±0.0355(2)	0.7147±0.0302(3)	0.9117±0.0207(4)	**0.5746±0.0308(1)**	10.7617±0.0183(6)	10.7190±0.4352(5)	0.0000
10	0.6559±0.0454(2)	0.7232±0.0339(3)	0.9047±0.0215(4)	**0.5879±0.0237(1)**	12.4282±0.0108(5)	12.4602±0.0158(6)	0.0000

TABLE VII
IGD Values of Two_Arch2, Two_Arch, IBEA, NSGA-III, MOEA/D, and AGE-II on WFG2 Analyzed by Wilcoxon Signed-Rank Test. The Significant Results Are in Bold Face (Significant Level = 0.05). The Numbers in Brackets Are the Ranking Numbers

Obj #	Two_Arch2	Two_Arch	IBEA	NSGA-III	MOEA/D	AGE-II	p-value of algorithms of ranks 1 and 2
2	0.0940±0.0312(2)	0.1015±0.0236(3)	0.3713±0.0052(4)	0.0850±0.0316(1)	0.8933±0.1113(6)	0.4636±0.1227(5)	0.1204
3	**0.0584±0.0082(1)**	0.0886±0.0170(3)	0.3310±0.0144(4)	0.0605±0.0030(2)	2.0329±0.1769(6)	0.7334±0.2063(5)	0.0030
4	0.1178±0.0125(2)	0.1390±0.0257(3)	0.3630±0.0148(4)	**0.0957±0.0057(1)**	3.6751±0.2831(6)	1.7377±0.2028(5)	0.0000
5	0.1681±0.0186(3)	0.1678±0.0275(2)	0.3808±0.0175(4)	**0.1386±0.0242(1)**	5.3102±0.5261(6)	2.8918±0.3172(5)	0.0004
6	0.1667±0.0316(1)	0.1712±0.0207(3)	0.4256±0.0465(4)	0.1702±0.0515(2)	7.3498±0.6986(6)	4.6957±1.2931(5)	0.7813
7	0.2077±0.0280(3)	0.1907±0.0287(2)	0.4559±0.0591(4)	**0.1633±0.0383(1)**	9.2672±0.6380(6)	6.8285±1.6882(5)	0.0028
8	0.1092±0.0100(1)	0.1800±0.0404(3)	0.5859±0.0489(4)	0.1427±0.0670(2)	13.1278±0.0723(6)	8.9329±2.1716(5)	0.0937
9	0.1108±0.0092(1)	0.1804±0.0440(3)	0.5771±0.0912(4)	0.1196±0.0732(2)	14.9630±0.0721(6)	11.3912±1.9758(5)	0.0598
10	0.0471±0.0152(1)	0.1897±0.0479(3)	0.7710±0.0812(4)	0.0757±0.0813(2)	17.4688±0.3123(6)	14.2573±1.8932(5)	0.4528

TABLE VIII
IGD Values of Two_Arch2, Two_Arch, IBEA, NSGA-III, MOEA/D, and AGE-II on WFG3 Analyzed by Wilcoxon Signed-Rank Test. The Significant Results Are in Bold Face (Significant Level = 0.05). The Numbers in Brackets Are the Ranking Numbers

Obj #	Two_Arch2	Two_Arch	IBEA	NSGA-III	MOEA/D	AGE-II	p-value of algorithms of ranks 1 and 2
2	0.0179±0.0097(2)	0.0251±0.0100(3)	0.6624±0.0256(6)	**0.0059±0.0016(1)**	0.3260±0.0624(5)	0.2512±0.0565(4)	0.0000
3	0.1998±0.0010(2)	0.2073±0.0033(3)	0.2984±0.1097(4)	**0.1984±0.0023(1)**	0.8055±0.0783(6)	0.7744±0.0052(5)	0.0023
4	0.2401±0.0033(1)	0.2419±0.0036(2)	0.3146±0.0351(4)	0.2483±0.0035(3)	1.1796±0.0682(5)	1.2836±0.0503(6)	0.0978
5	**0.2845±0.0029(1)**	0.2880±0.0031(2)	0.3651±0.0105(4)	0.2918±0.0032(3)	1.6331±0.0602(5)	1.9061±0.2098(6)	0.0004
6	**0.3137±0.0037(1)**	0.3234±0.0037(3)	0.4101±0.0057(4)	0.3232±0.0095(2)	2.2044±0.0868(5)	3.6340±0.3489(6)	0.0000
7	**0.3450±0.0030(1)**	0.3616±0.0047(3)	0.4423±0.0149(4)	0.3500±0.0073(2)	2.9480±0.0717(5)	4.9139±0.6207(6)	0.0010
8	**0.3600±0.0024(1)**	0.4081±0.0125(3)	0.4789±0.0140(4)	0.3850±0.0112(2)	3.7460±0.0854(5)	6.3188±0.0949(6)	0.0000
9	0.3816±0.0028(1)	0.4450±0.0119(3)	0.4978±0.0153(4)	0.3824±0.0101(2)	4.4063±0.1051(5)	7.4088±0.3327(6)	0.9426
10	0.4013±0.0025(2)	0.4838±0.0217(3)	0.5398±0.0885(4)	**0.3986±0.0089(1)**	4.9705±0.1074(5)	9.0260±0.0889(6)	0.0428

TABLE IX
IGD Values of Two_Arch2, Two_Arch, IBEA, NSGA-III, MOEA/D, and AGE-II on WFG4 Analyzed by Wilcoxon Signed-Rank Test. The Significant Results Are in Bold Face (Significant Level = 0.05). The Numbers in Brackets Are the Ranking Numbers

Obj #	Two_Arch2	Two_Arch	IBEA	NSGA-III	MOEA/D	AGE-II	p-value of algorithms of ranks 1 and 2
2	0.0048±0.0001(2)	0.0243±0.0091(3)	0.7458±0.0002(6)	**0.0044±0.0002(1)**	0.0396±0.0233(4)	0.1409±0.0044(5)	0.0000
3	0.0596±0.0011(2)	0.0833±0.0082(3)	0.8549±0.1044(5)	**0.0543±0.0003(1)**	1.0963±0.1148(6)	0.3244±0.0060(4)	0.0000
4	**0.1319±0.0012(1)**	0.1533±0.0089(3)	1.0114±0.1318(5)	0.1455±0.0008(2)	3.6984±0.3753(6)	0.7596±0.0194(4)	0.0000
5	**0.2026±0.0018(1)**	0.2379±0.0108(3)	1.0647±0.1248(4)	0.2310±0.0011(2)	5.3457±0.2728(6)	1.3887±0.0617(5)	0.0000
6	**0.2688±0.0025(1)**	0.3006±0.0073(2)	1.1275±0.1148(4)	0.3051±0.0014(3)	6.8048±0.2073(6)	2.2764±0.0981(5)	0.0000
7	**0.3255±0.0034(1)**	0.3711±0.0135(2)	1.1573±0.0856(4)	0.3731±0.0051(3)	8.3484±0.2080(6)	3.0940±0.1025(5)	0.0000
8	**0.3804±0.0028(1)**	0.4261±0.0164(2)	1.2338±0.0551(4)	0.4268±0.0007(3)	9.7015±0.2024(6)	4.0957±0.1115(5)	0.0000
9	**0.4287±0.0052(1)**	0.4775±0.0093(2)	1.2390±0.0588(4)	0.4965±0.0064(3)	11.2488±0.1846(6)	5.2919±0.8755(5)	0.0000
10	**0.4725±0.0054(1)**	0.5245±0.0139(2)	1.2584±0.0459(4)	0.5295±0.0060(3)	12.7737±0.2188(6)	6.3346±0.9050(5)	0.0000

have slightly better IGD values than MOEA/D. NSGA-III is the best-performing algorithm for WFG4 with two and three objectives. However, NSGA-III declines its performance on WFG4 with more than three objectives, Two_Arch2 becomes the best-performing algorithm, even Two_Arch outperforms NSGA-III.

Table X shows the IGD values of compared algorithms on WFG5. WFG5 is a deceptive problem that MOEA/D cannot

TABLE X
IGD VALUES OF TWO_ARCH2, TWO_ARCH, IBEA, NSGA-III, MOEA/D, AND AGE-II ON WFG5 ANALYZED BY WILCOXON SIGNED-RANK TEST. THE SIGNIFICANT RESULTS ARE IN BOLD FACE (SIGNIFICANT LEVEL = 0.05). THE NUMBERS IN BRACKETS ARE THE RANKING NUMBERS

Obj #	Two_Arch2	Two_Arch	IBEA	NSGA-III	MOEA/D	AGE-II	p-value of algorithms of ranks 1 and 2
2	0.0271±0.0002(2)	0.0509±0.0100(3)	0.1042±0.1361(5)	**0.0270±0.0001(1)**	0.0814±0.0188(4)	0.2274±0.0043(6)	0.0104
3	0.0641±0.0010(2)	0.0888±0.0079(3)	0.2140±0.1691(4)	**0.0589±0.0003(1)**	0.9220±0.0299(6)	0.3654±0.0094(5)	0.0000
4	**0.1330±0.0011(1)**	0.1524±0.0066(3)	0.4641±0.2136(4)	0.1413±0.0007(2)	2.8262±0.2953(6)	0.7960±0.0190(5)	0.0000
5	**0.2029±0.0022(1)**	0.2215±0.0053(2)	0.7079±0.1109(4)	0.2221±0.0007(3)	5.0810±0.2701(6)	1.3957±0.0617(5)	0.0000
6	**0.2691±0.0025(1)**	0.2725±0.0043(2)	0.7890±0.0835(4)	0.2937±0.0013(3)	6.5609±0.2175(6)	2.2885±0.1112(5)	0.0021
7	**0.3264±0.0023(1)**	0.3297±0.0046(2)	0.8736±0.0349(4)	0.3648±0.0036(3)	8.2166±0.2193(6)	3.3060±0.3982(5)	0.0014
8	**0.3786±0.0030(1)**	0.3819±0.0038(2)	0.8878±0.1062(4)	0.4193±0.0011(3)	9.4119±0.0793(6)	4.2451±0.1545(5)	0.0015
9	**0.4252±0.0027(1)**	0.4286±0.0052(2)	0.9179±0.1131(4)	0.5029±0.0056(3)	10.9330±0.1147(6)	5.1361±0.1347(5)	0.0068
10	0.4691±0.0032(1)	0.4711±0.0050(2)	0.9411±0.1015(4)	0.5354±0.0043(3)	12.3765±0.1560(6)	6.1287±0.1272(5)	0.1254

TABLE XI
IGD VALUES OF TWO_ARCH2, TWO_ARCH, IBEA, NSGA-III, MOEA/D, AND AGE-II ON WFG6 ANALYZED BY WILCOXON SIGNED-RANK TEST. THE SIGNIFICANT RESULTS ARE IN BOLD FACE (SIGNIFICANT LEVEL = 0.05). THE NUMBERS IN BRACKETS ARE THE RANKING NUMBERS

Obj #	Two_Arch2	Two_Arch	IBEA	NSGA-III	MOEA/D	AGE-II	p-value of algorithms of ranks 1 and 2
2	0.0957±0.0325(3)	0.0528±0.0132(2)	0.7456±0.0005(5)	**0.0329±0.0100(1)**	1.3100±0.9289(6)	0.3100±0.0182(4)	0.0000
3	0.0884±0.0144(3)	0.0831±0.0111(2)	0.9862±0.0015(5)	**0.0651±0.0047(1)**	2.3520±0.5433(6)	0.5533±0.0135(4)	0.0000
4	0.1452±0.0066(1)	0.1507±0.0064(3)	1.0792±0.0548(5)	0.1459±0.0009(2)	4.0541±0.2046(6)	0.9266±0.0228(4)	0.0627
5	**0.2200±0.0096(1)**	0.2254±0.0068(3)	1.0861±0.1005(4)	0.2232±0.0011(2)	5.5143±0.1742(6)	1.6021±0.0620(5)	0.0010
6	0.2773±0.0100(2)	0.2769±0.0049(1)	1.0725±0.0281(4)	0.2970±0.0018(3)	6.8400±0.1938(6)	2.5105±0.0804(5)	0.7343
7	0.3373±0.0061(2)	**0.3336±0.0046(1)**	1.0949±0.0078(4)	0.3760±0.0049(3)	8.3621±0.1941(6)	3.3787±0.2640(5)	0.0230
8	0.3858±0.0057(1)	0.3867±0.0051(2)	1.1097±0.0513(4)	0.4175±0.0021(3)	10.0908±0.2721(6)	4.1444±0.1418(5)	0.2452
9	0.4343±0.0113(1)	0.4348±0.0056(2)	1.1340±0.0570(4)	0.5076±0.0084(3)	11.7105±0.2342(6)	5.0481±0.2077(5)	0.3286
10	0.4772±0.0151(1)	0.4778±0.0045(2)	1.1424±0.0716(4)	0.5373±0.0059(3)	13.1932±0.2539(6)	6.4545±0.3883(5)	0.0656

TABLE XII
IGD VALUES OF TWO_ARCH2, TWO_ARCH, IBEA, NSGA-III, MOEA/D, AND AGE-II ON WFG7 ANALYZED BY WILCOXON SIGNED-RANK TEST. THE SIGNIFICANT RESULTS ARE IN BOLD FACE (SIGNIFICANT LEVEL = 0.05). THE NUMBERS IN BRACKETS ARE THE RANKING NUMBERS

Obj #	Two_Arch2	Two_Arch	IBEA	NSGA-III	MOEA/D	AGE-II	p-value of algorithms of ranks 1 and 2
2	0.3605±0.1895(3)	0.3280±0.1092(2)	0.7458±0.0000(4)	**0.1133±0.0310(1)**	1.8779±0.2206(6)	1.8030±0.4499(5)	0.0000
3	**0.0703±0.0141(1)**	0.4261±0.1355(3)	0.8706±0.1053(4)	0.1428±0.0582(2)	3.3829±0.7157(6)	1.3888±0.4468(5)	0.0000
4	**0.1392±0.0181(1)**	0.4254±0.0930(3)	0.9719±0.0774(4)	0.2092±0.0561(2)	3.8985±0.9105(6)	1.6353±0.5692(5)	0.0000
5	**0.2980±0.0640(1)**	0.5439±0.1260(3)	0.9820±0.1006(4)	0.3359±0.0565(2)	5.8867±1.1597(6)	2.4948±0.5012(5)	0.0387
6	0.4624±0.0832(2)	0.6200±0.1692(3)	1.0188±0.1086(4)	**0.3979±0.0413(1)**	6.5829±1.6782(6)	3.2367±0.7267(5)	0.0021
7	0.5248±0.0905(2)	0.6485±0.1620(3)	1.0732±0.0789(4)	**0.4233±0.0230(1)**	8.6578±1.1724(6)	4.1282±0.5790(5)	0.0000
8	0.5868±0.1510(2)	0.6764±0.1849(3)	1.1206±0.0617(4)	**0.4571±0.0380(1)**	8.8782±3.0126(6)	5.9780±1.6429(5)	0.0001
9	0.6129±0.1237(2)	0.7326±0.1944(3)	1.1397±0.0767(4)	**0.5170±0.0072(1)**	11.8538±3.9523(6)	7.1262±1.4564(5)	0.0000
10	0.6738±0.1074(2)	0.7383±0.1552(3)	1.1723±0.0448(4)	**0.5481±0.0105(1)**	12.0360±3.4074(6)	9.0531±2.2319(5)	0.0000

solve, which is reflected by its poor IGD values. The situation is similar to that of WFG4. NSGA-III is the best-performing algorithm on WFG5 with two and three objectives. However, NSGA-III declines its performance for WFG5 with more than three objectives, Two_Arch2 becomes the best-performing algorithm.

Table XI shows the IGD values of compared algorithms on WFG6. WFG6 is a nonseparable-reduced problem. MOEA/D, AGE-II, and IBEA all fail on this problem. NSGA-III is the best-performing algorithm for WFG6 with 2 and 3 objectives, followed by Two_Arch (rank 2) and Two_Arch2 (rank 3). Two_Arch2 is the best-performing algorithm for WFG6 with 4–5 and 7–10 objectives, and Two_Arch is the best-performing algorithm for other WFG6 problems.

Table XII shows the IGD values of compared algorithms on WFG7. WFG7 is both separable and uni-modal. Because of its three transformation functions, MOEA/D and IBEA have the worst IGD values for their poor diversity. Two_Arch and IBEA perform better than MOEA/D and IBEA but worse than Two_Arch2 and NSGA-III. NSGA-III outperforms Two_Arch2 on WFG7 with 2 and 6–10 objectives, while Two_Arch2 outperforms NSGA-III on WFG7 with 3–5 objectives.

Table XIII shows the IGD values of compared algorithms on WFG8. WFG8 is a hard nonseparable problem. MOEA/D, AGE-II, and IBEA all fail on this problem. Two_Arch2 cannot perform better than Two_Arch and NSGA-III. Two_Arch is the best-performing algorithm for WFG8 with 9–10 objectives, and NSGA-III is the best-performing algorithm for other WFG8 problems.

Table XIV shows the IGD values of compared algorithms on WFG9. WFG9 is nonseparable-reduced. For all WFG9 problems, NSGA-III is the best-performing algorithm, Two_Arch2

TABLE XIII

IGD VALUES OF TWO_ARCH2, TWO_ARCH, IBEA, NSGA-III, MOEA/D, AND AGE-II ON WFG8 ANALYZED BY WILCOXON SIGNED-RANK TEST. THE SIGNIFICANT RESULTS ARE IN BOLD FACE (SIGNIFICANT LEVEL = 0.05). THE NUMBERS IN BRACKETS ARE THE RANKING NUMBERS

Obj #	Two_Arch2	Two_Arch	IBEA	NSGA-III	MOEA/D	AGE-II	p-value of algorithms of ranks 1 and 2
2	0.3431±0.0287(3)	0.2611±0.0308(2)	0.5778±0.0643(4)	**0.2444±0.0212(1)**	0.9097±0.1298(6)	0.7880±0.0540(5)	0.0368
3	0.3409±0.0224(3)	0.2840±0.0193(2)	0.7787±0.1079(4)	**0.2563±0.0110(1)**	1.4697±0.2889(6)	0.9826±0.0230(5)	0.0000
4	0.4197±0.0191(3)	0.3393±0.0152(2)	0.9909±0.0171(4)	**0.2874±0.0123(1)**	2.8600±0.3790(6)	1.3322±0.0200(5)	0.0000
5	0.4572±0.0270(3)	0.3989±0.0230(2)	1.0241±0.0490(4)	**0.3135±0.0072(1)**	4.2693±0.3625(6)	1.9481±0.0476(5)	0.0000
6	0.4866±0.0223(3)	0.4175±0.0143(2)	1.0817±0.0098(4)	**0.3452±0.0029(1)**	5.6254±0.2595(6)	2.6593±0.0741(5)	0.0000
7	0.5427±0.0250(3)	0.4495±0.0128(2)	1.0892±0.0577(4)	**0.4047±0.0056(1)**	7.3276±0.3906(6)	3.6971±0.1540(5)	0.0000
8	0.5577±0.0141(3)	0.4578±0.0074(2)	1.1338±0.0105(4)	**0.4209±0.0016(1)**	8.9882±0.5267(6)	4.2131±0.2347(5)	0.0000
9	0.5953±0.0135(3)	**0.4975±0.0083(1)**	1.1561±0.0120(4)	0.5140±0.0052(2)	10.6843±0.2953(6)	5.1798±0.2180(5)	0.0000
10	0.6349±0.0112(3)	**0.5300±0.0070(1)**	1.1759±0.0122(4)	0.5410±0.0050(2)	12.2246±0.2342(6)	6.4616±0.6112(5)	0.0000

and Two_Arch follow, IBEA, AGE-II, and MOEA/D are in ranks 4, 5, and 6 correspondingly.

Comparing with the DTLZ problems, the WFG problems have more transformation functions. Diversity is more difficult to be maintained. IBEA cannot perform well on the WFG problems by the same reason as on the DTLZ problems. The ranges of the objectives of the WFG problems are different, but AGE-II still divides all the objectives into same ε-grids, which leads to its unbalanced search on different objectives. Therefore, AGE-II cannot perform well on the WFG problems. Furthermore, most WFG problems are deceptive, even the decomposition method in MOEA/D cannot solve. Two_Arch performs better than IBEA, AGE-II, and MOEA/D, because of its strict selection principle of CA to guarantee the convergence.

Generally, the IGD values of two well-performing algorithms (Two_Arch2 and NSGA-III) on the WFG problems with high-dimensional objectives are very close, because their convergence abilities and extents are at the same level. For the separable problems WFG1 and WFG7 (the easiest problems among the WFG problems), NSGA-III keeps their extents (extreme points) better than Two_Arch2. Hence, NSGA-III has better IGD than Two_Arch2 on WFG1 and WFG7. $I_{\varepsilon+}$ is very effective for deceptive problems, which results in good convergence but poor diversity. That is the reason why Two_Arch2 outperforms NSGA-III on deceptive WFG5. Similarly, due to the effectiveness of $I_{\varepsilon+}$, Two_Arch2 performs better than NSGA-III on the multimodal problem WFG4. However, NSGA-III performs better than Two_Arch2 on the less-deceptive problems WFG8 and WFG9 according to the IGD values. IGD evaluates both convergence and diversity. WFG8 and WFG9 have one more transformation function t^3 than WFG5, which makes the diversity on WFG8 and WFG9 harder to achieve than that of WFG5. Due to the disadvantage of $I_{\varepsilon+}$ on diversity, Two_Arch2 cannot outperform NSGA-III on WFG8 and WFG9, where NSGA-III can obtain a wider spread. For the nonseparable problem WFG6 and the disconnected problem WFG2, the differences between the performance of Two_Arch2 and NSGA-III are insignificant, though Two_Arch2 has slightly better IGD values than NSGA-III on the high-dimensional WFG2 and WFG6.

3) Discussion: From the results in Sections IV-F1 and IV-F2, we can gain some insight into the different behaviors of compared algorithms on ManyOPs and the reason why they behave so.

Because of the effectiveness of $I_{\varepsilon+}$, IBEA has good performance on convergence. However, IBEA focuses on convergence too much, it obtains a solution set in a small region close to the true PF. Its diversity on ManyOPs is far from satisfactory.

Another popular non-Pareto-based MOEA for ManyOPs is MOEA/D. It assigns a number of weight vectors in the entire objective space in advance, then decomposes the original MOP into a series of single-objective optimization problems. Thus, MOEA/D never faces the less-effective Pareto-based selection. It also has good diversity because of the weight vectors spread uniformly over the weight space. However, the performance of MOEA/D decreases when the number of objectives increases. Only one solution kept for each sub-problem cannot guarantee the diversity in the high-dimensional objective space. Therefore, MOEA/D can obtain solutions for ManyOPs with better diversity than IBEA, but its performance is still not satisfactory.

AGE-II is a Pareto-based MOEA with ε-grid to approximate the objective space. As we know, the Pareto dominance hardly contributes to ManyOPs, because most solutions are nondominated. The ε-grid method can reduce that effect to some extent by lowering the conflicting degree among objectives. In addition, its removal step for extra individuals is based on a sensitive indicator, which compensates the disadvantage from the Pareto dominance. However, it uses the same ε for all the objectives with different scales. It is the reason why AGE-II cannot work well on the WFG problems.

Two_Arch is another Pareto-based MOEA with CA and DA to balance convergence and diversity. Two_Arch gives CA a higher priority than DA to achieve better convergence than other Pareto-based MOEAs, it is still a Pareto-based MOEA, its convergence on ManyOPs is not satisfactory.

Comparing with Two_Arch, another Pareto-based MOEA, NSGA-III, takes more consideration of the diversity maintenance on ManyOPs. Given a set of uniform reference points in advance, NSGA-III forces the population to distribute across the reference set, which leads to good results on both convergence (guaranteed by nondominated sort) and diversity (guaranteed by reference points), which is a break-through for Pareto-based MOEAs on ManyOPs. However, these reference

TABLE XIV
IGD VALUES OF TWO_ARCH2, TWO_ARCH, IBEA, NSGA-III, MOEA/D, AND AGE-II ON WFG9 ANALYZED BY WILCOXON SIGNED-RANK TEST. THE SIGNIFICANT RESULTS ARE IN BOLD FACE (SIGNIFICANT LEVEL = 0.05). THE NUMBERS IN BRACKETS ARE THE RANKING NUMBERS

Obj #	Two_Arch2	Two_Arch	IBEA	NSGA-III	MOEA/D	AGE-II	p-value of algorithms of ranks 1 and 2
2	0.0889±0.0001(2)	0.0907±0.0022(3)	0.1366±0.0860(4)	0.0888±0.0001(1)	0.3829±0.2674(5)	0.5150±0.0903(6)	0.2989
3	0.1038±0.0008(2)	0.1088±0.0104(3)	0.3625±0.2378(4)	**0.0986±0.0007(1)**	0.7579±0.2545(6)	0.6370±0.0261(5)	0.0000
4	0.1506±0.0024(2)	0.1751±0.0108(3)	0.5781±0.2672(4)	**0.1480±0.0007(1)**	1.8983±0.4096(6)	1.0623±0.1013(5)	0.0000
5	0.2255±0.0018(2)	0.2582±0.0179(3)	0.7809±0.2371(4)	**0.2198±0.0013(1)**	2.6714±0.6072(6)	1.9045±0.1929(5)	0.0000
6	0.2834±0.0108(2)	0.3266±0.0251(3)	0.8624±0.1692(4)	**0.2794±0.0021(1)**	4.7838±2.1735(6)	2.7943±0.4853(5)	0.0010
7	0.3588±0.0202(2)	0.4053±0.0307(3)	0.9108±0.1529(4)	**0.3393±0.0036(1)**	5.5983±1.7514(6)	3.7582±0.4205(5)	0.0000
8	0.4249±0.0627(2)	0.5341±0.1038(3)	1.0030±0.1317(4)	**0.3991±0.0555(1)**	8.9925±2.8092(6)	6.5720±1.4155(5)	0.0001
9	0.4727±0.0261(2)	0.5572±0.0913(3)	1.0303±0.1183(4)	**0.4491±0.0048(1)**	11.5753±3.5448(6)	8.4170±1.4263(5)	0.0003
10	0.5331±0.0500(2)	0.5975±0.0810(3)	1.0653±0.1071(4)	**0.4803±0.0037(1)**	14.2644±3.3317(6)	11.1821±2.2768(5)	0.0000

points also have some side-effect that leads to a kind of diversity loss (shown by Fig. 10). In other words, NSGA-III transfers the traditional diversity maintenance task to the approximation to several reference points, which results in uniformity but also diversity loss to certain extent (due to concentration around the reference points). When the number of objectives increases, the number of divisions on each objective for generating reference points decreases because of the limited size of the population. That is the reason why the diversity loss problem becomes more severe with an increasing number of objectives.

Two_Arch2, a hybrid MOEA, takes the advantages of both $I_{\varepsilon+}$ and the $L_{1/m}$-norm-based distance, which leads to its both good convergence and diversity on ManyOPs. Because of $I_{\varepsilon+}$, Two_Arch2 outperforms NSGA-III on some deceptive and multimodal problems, i.e., DTLZ1, DTLZ3, WFG5, and WFG4. In comparison with NSGA-III, the $L_{1/m}$-norm-based distance diversity maintenance in Two_Arch2 is a more natural way to maintain diversity. That is the reason why it adds more diversity for ManyOPs than NSGA-III. However, Two_Arch2 cannot perform better than NSGA-III on the problems with more transformation functions as WFG1, WFG8, and WFG9. Obviously, the diversity of such problems are hard to be maintained, there is no strategy to extend the spread of PF in Two_Arch2 as that in NSGA-III. That is the reason why NSGA-III outperforms Two_Arch2 on such problems. In summary, Two_Arch2 is good at handling the problems with deceptiveness for convergence and few transformation functions for diversity.

In order to analyze the performance of all the compared algorithms over generations on ManyOPs with high-dimensional objectives, we show the IGD values of Two_Arch2, Two_Arch, IBEA, NSGA-III, MOEA/D, and AGE-II over generations on DTLZ1 with ten objectives in Fig. 11. IBEA and Two_Arch have the worst IGD values due to poor diversity and convergence correspondingly. The poor diversity of IBEA comes from $I_{\varepsilon+}$, and the poor convergence of Two_Arch comes from the Pareto dominance. In the first 2000 function evaluations, AGE-II and MOEA/D converge faster than Two_Arch2 and NSGA-III, because the ε-grid method and aggregation functions are effective in the initial stage of optimization of ManyOPs. However, in the next 3000 function evaluations, the situation changes, Two_Arch2 decreases the IGD value faster than MOEA/D, and the IGD value of MOEA/D closes to that of NSGA-III, because both MOEA/D and NSGA-III maintain their diversity by reference points or vectors, they cannot add more diversity to ManyOPs than Two_Arch2. In the last 4000 function evaluations, both Two_Arch2 and NSGA-III continue to decrease their IGD values, but MOEA/D and AGE-II cannot decrease their IGD values any more. Although the ε-grid method improves the performance of AGE-II on ManyOPs, AGE-II as a Pareto-based MOEA cannot outperform Two_Arch2. Finally, Two_Arch2 obtains a smaller IGD value than NSGA-III.

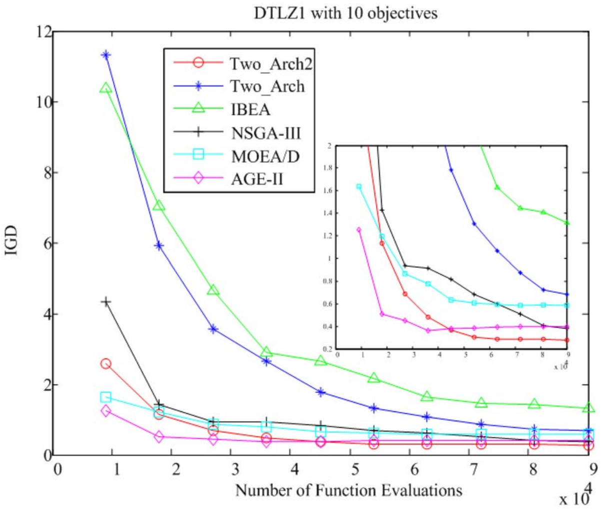

Fig. 11. Average IGD values of Two_Arch2, Two_Arch, IBEA, NSGA-III, MOEA/D, and AGE-II over generations on DTLZ1 with ten objectives.

G. Experiments on Scalability

As Section IV-F shows, Two_Arch2 and NSGA-III are two excellent MOEAs on ManyOPs, we compare them on the DTLZ problems with 15 and 20 objectives to test their scalability. All the experimental settings have been shown in Section IV-A.

The results are evaluated by IGD in Table XV. Although the increasing number of objectives decreases the performance of both algorithms, Two_Arch2 outperforms NSGA-III on almost all the test problems, except for DTLZ2 with 15 objectives. As discussed in Section IV-F3, NSGA-III suffers from more diversity loss when the number of objectives increases, which is clearly shown by parallel coordinate plots of the best solution

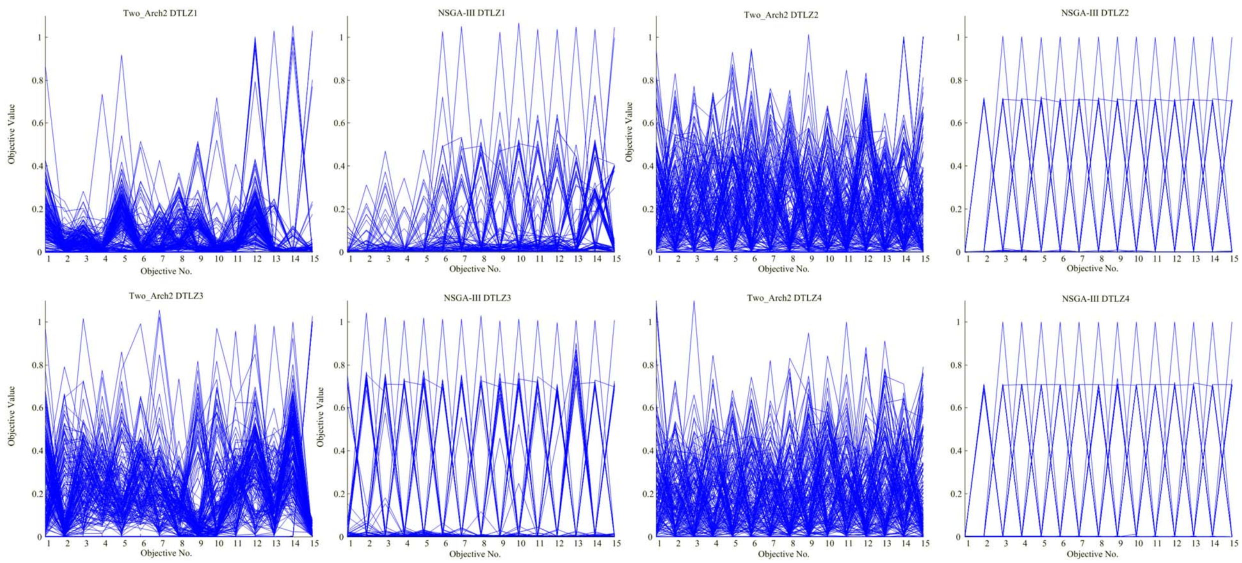

Fig. 12. Parallel coordinate plot of the best solution set of Two_Arch2 and NSGA-III on the DTLZ problems with 15 objectives.

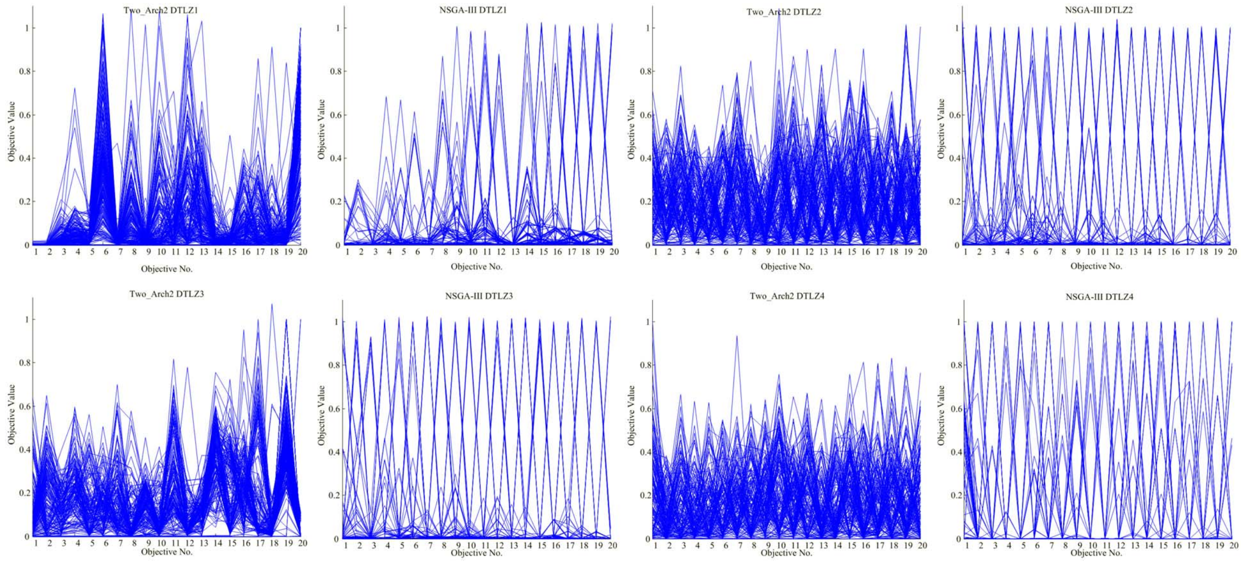

Fig. 13. Parallel coordinate plot of the best solution set of Two_Arch2 and NSGA-III on the DTLZ problems with 20 objectives.

TABLE XV
IGD VALUES OF TWO_ARCH2 AND NSGA-III ON THE DTLZ PROBLEMS WITH 15 AND 20 OBJECTIVES ANALYZED BY WILCOXON SIGNED-RANK TEST. THE SIGNIFICANT RESULTS ARE IN BOLD FACE (SIGNIFICANT LEVEL = 0.05)

DTLZ	Obj #	Two_Arch2	NSGA-III	p-value
1	15	**0.3453±0.0184(1)**	0.3940±0.0464(2)	0.0000
	20	**0.3994±0.0154(1)**	0.5352±0.3195(2)	0.0050
2	15	0.6393±0.0072(2)	**0.6288±0.0004(1)**	0.0000
	20	**0.7730±0.0101(1)**	0.8056±0.0046(2)	0.0000
3	15	**0.7142±0.0305(1)**	0.7729±0.4121(2)	0.0039
	20	**0.8274±0.0363(1)**	1.3846±0.5941(2)	0.0001
4	15	**0.6154±0.0058(1)**	0.6284±0.0001(2)	0.0000
	20	**0.7430±0.0096(1)**	0.7965±0.0079(2)	0.0000

set of Two_Arch2 and NSGA-III in Figs. 12 and 13. In contrast, Two_Arch2 can handle ManyOPs very well, even for 20-objective problems.

H. Computational Complexity Analysis

The large number of objectives of ManyOPs increases the computational complexity of MOEAs. Taking an MOP with m objectives and an algorithm with a population of N individuals as an example, the computation complexity is shown in Table XVI. The complexity of IBEA comes from its indicator-based selection only, which is $O(N^2)$. For Two_Arch, the step of assigning individuals to CA and DA costs $O(mN^2)$, and the removal step costs $O(mN^2)$. Thus, the total complexity of Two_Arch is $O(mN^2)$. For MOEA/D, the updating of its external population by nondominated solutions costs $O(mN^2)$, and its diversity maintenance by aggregation functions costs $O(mNT), T \leq N$ (where T is the size of the neighborhood). Thus, the total complexity of MOEA/D approximates to $O(mN^2)$. For AGE-II, its nondominated sort costs $O(Nlog^{m-2}N)$, and its fast diversity

TABLE XVI
COMPUTATIONAL COMPLEXITY OF COMPARED ALGORITHMS (WITH A POPULATION OF N INDIVIDUALS) ON AN MOP WITH m OBJECTIVES

Algorithm	Convergence (Pareto-based)	Diversity Maintenance	Indicator-based Selection	Total
Two_Arch2	$O(Nlog^{m-2}N)$	$O(mN^2)$	$O(N^2)$	$max\{O(Nlog^{m-2}N), O(mN^2)\}$
Two_Arch	$O(mN^2)$	$O(mN^2)$	NA	$O(mN^2)$
IBEA	NA	NA	$O(N^2)$	$O(N^2)$
NSGA-III	$O(Nlog^{m-2}N)$	$O(mN^2)$	NA	$max\{O(Nlog^{m-2}N), O(mN^2)\}$
MOEA/D	$O(mN^2)$	$O(mNT)$	NA	$O(mN^2)$
AGE-II	$O(Nlog^{m-2}N)$	$O(mN)$	NA	$max\{O(Nlog^{m-2}N), O(mN)\}$

maintenance uses $O(mN)$, thus the total complexity of AGE-II is $\max\{O(Nlog^{m-2}N), O(mN)\}$. For NSGA-III, the complexity of nondominated sort is $O(Nlog^{m-2}N)$, and that of diversity maintenance is $O(mNH)$, where H is the size of the reference point set. As $N \approx H$, $O(mNH)$ approximates to $O(mN^2)$. Thus, the total complexity of NSGA-III is $\max\{O(Nlog^{m-2}N), O(mN^2)\}$. For Two_Arch2, the complexity of updating CA, i.e., indicator-based selection, is the same as IBEA, the complexity of updating DA, i.e., Pareto-based selection (convergence and diversity maintenance in Table XVI), is $\max\{O(Nlog^{m-2}N), O(mN^2)\}$. Therefore, the total complexity of Two_Arch2 is at the same level as that of NSGA-III.

V. CONCLUSION

In order to solve ManyOPs with satisfactory convergence, diversity, and efficiency at the same time, we proposed a new MOEA, Two_Arch2. Through extensive comparative experiments on the DTLZ and WFG problems with varied numbers of objectives, Two_Arch2 is shown to be good at coping with ManyOPs in all three aspects. The main novelties of this paper include the following.

1) *A Hybrid MOEA:* Two_Arch2 follows the setting of CA and DA in Two_Arch, but it assigns different dominance relations to the selection of these two archives. We use the $I_{\varepsilon+}$ indicator as the selection principle for CA to improve the convergence on ManyOPs, while the Pareto dominance as the selection principle for DA to promote diversity. As a result, Two_Arch2 is based on both indicator and Pareto dominance, which combines the advantages of indicator and Pareto-based MOEAs. On one hand, Two_Arch2 can have a good convergence ability by the indicator-based CA. On the other hand, Two_Arch2 can maintain satisfactory diversity by the Pareto-based DA.
2) *$L_{1/m}$-Norm-Based Diversity Maintenance:* In this paper, we explained the reason why most Euclidean distance-based diversity maintenance methods cannot work well on ManyOPs. It is because that L_p-norm-based ($p \geq 1$) distances lose their performance in a high-dimensional space, whereas L_p-norm-based ($p < 1$) distances work well in a high-dimensional space. In Two_Arch2, we use a $L_{1/m}$-norm-based maintenance scheme to delete extra solutions from DA.

Although Two_Arch2 performs very well on ManyOPs, it has several disadvantages that needs further improvement in the future. The performance of Two_Arch2 on several WFG problems is not quite satisfactory, because of less-well extents on PFs. Two_Arch2 did not keep extreme points, in contrast to NSGA-III. We can borrow the extreme point maintenance scheme in NSGA-III to improve our Two_Arch2 further.

REFERENCES

[1] E. Hughes, "Radar waveform optimisation as a many-objective application benchmark," in *Evolutionary Multi-Criterion Optimization*. Berlin, Germany: Springer, 2007, pp. 700–714.

[2] P. Fleming, R. Purshouse, and R. Lygoe, "Many-objective optimization: An engineering design perspective," in *Evolutionary Multi-Criterion Optimization*. Berlin, Germany: Springer, 2005, pp. 14–32.

[3] V. Khare, X. Yao, and K. Deb, "Performance scaling of multi-objective evolutionary algorithms," in *Evolutionary Multi-Criterion Optimization* (Lecture Notes in Computer Science), vol. 2632, C. Fonseca, P. Fleming, E. Zitzler, L. Thiele, and K. Deb, Eds. Berlin, Germany: Springer, 2003, pp. 376–390.

[4] K. Praditwong and X. Yao, "How well do multi-objective evolutionary algorithms scale to large problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Singapore, Sep. 2007, pp. 3959–3966.

[5] K. Miettinen, *Nonlinear Multiobjective Optimization*. Berlin, Germany: Springer, 1999.

[6] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, vol. 16. New York, NY, USA: Wiley, 2001.

[7] D. Hadka and P. Reed, "Diagnostic assessment of search controls and failure modes in many-objective evolutionary optimization," *Evol. Comput.*, vol. 20, no. 3, pp. 423–452, 2012.

[8] E. Hughes, "Evolutionary many-objective optimisation: Many once or one many?" in *Proc. IEEE Congr. Evol. Comput. (CEC)*, vol. 1. Edinburgh, U.K., 2005, pp. 222–227.

[9] H. Ishibuchi, N. Akedo, H. Ohyanagi, and Y. Nojima, "Behavior of EMO algorithms on many-objective optimization problems with correlated objectives," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, New Orleans, LA, USA, Jun. 2011, pp. 1465–1472.

[10] T. Wagner, N. Beume, and B. Naujoks, "Pareto-, aggregation-, and indicator-based methods in many-objective optimization," in *Evolutionary Multi-Criterion Optimization*. Berlin, Germany: Springer, 2007, pp. 742–756.

[11] H. Ishibuchi, N. Tsukamoto, Y. Hitotsuyanagi, and Y. Nojima, "Effectiveness of scalability improvement attempts on the performance of NSGA-II for many-objective problems," in *Proc. 10th Annu. Conf. Genet. Evol. Comput. (GECCO)*, 2008, pp. 649–656.

[12] R. C. Purshouse and P. J. Fleming, "On the evolutionary optimization of many conflicting objectives," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 770–784, Dec. 2007.

[13] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Hong Kong, Jun. 2008, pp. 2419–2426.

[14] M. Köppen and K. Yoshida, "Substitute distance assignments in NSGA-II for handling many-objective optimization problems," in *Evolutionary Multi-Criterion Optimization*. Berlin, Germany: Springer, 2007, pp. 727–741.

[15] S. F. Adra and P. J. Fleming, "Diversity management in evolutionary many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 15, no. 2, pp. 183–195, Apr. 2011.

[16] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[17] H. Ishibuchi, Y. Hitotsuyanagi, H. Ohyanagi, and Y. Nojima, "Effects of the existence of highly correlated objectives on the behavior of MOEA/D," in *Evolutionary Multi-Criterion Optimization*. Berlin, Germany: Springer, 2011, pp. 166–181.

[18] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming, "Generalized decomposition and cross entropy methods for many-objective optimization," Dept. Autom. Control Syst. Eng., Univ. Sheffield, Sheffield, U.K., Res. Rep. 1029, 2012.

[19] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Parallel Problem Solving from Nature—PPSN VIII*. Berlin, Germany: Springer, 2004, pp. 832–842.

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

[21] J. Bader and E. Zitzler, "A hypervolume-based optimizer for high-dimensional objective spaces," in *New Developments in Multiple Objective and Goal Programming*. Berlin, Germany: Springer, 2010, pp. 35–54.

[22] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.

[23] K. Bringmann, "Bringing order to special cases of Klee's measure problem," in *Mathematical Foundations of Computer Science*. Berlin, Germany: Springer, 2013, pp. 207–218.

[24] D. Brockhoff and E. Zitzler, "Are all objectives necessary? On dimensionality reduction in evolutionary multiobjective optimization," in *Parallel Problem Solving from Nature—PPSN IX*. Berlin, Germany: Springer, 2006, pp. 533–542.

[25] D. Brockhoff and E. Zitzler, "Objective reduction in evolutionary multiobjective optimization: Theory and applications," *Evol. Comput.*, vol. 17, no. 2, pp. 135–166, 2009.

[26] A. López Jaimes, C. C. Coello, and D. Chakraborty, "Objective reduction using a feature selection technique," in *Proc. 10th Annu. Conf. Genet. Evol. Comput. (GECCO)*, 2008, pp. 673–680.

[27] K. Deb and D. Saxena, "On finding Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems," vol. 11, Indian Institute of Technology, Kharagpur, India, Kangal Rep. 2005011, 2005.

[28] D. Saxena and K. Deb, "Non-linear dimensionality reduction procedures for certain large-dimensional multi-objective optimization problems: Employing correntropy and a novel maximum variance unfolding," in *Evolutionary Multi-Criterion Optimization*. Berlin, Germany: Springer, 2007, pp. 772–787.

[29] D. K. Saxena, J. A. Duro, A. Tiwari, K. Deb, and Q. Zhang, "Objective reduction in many-objective optimization: Linear and nonlinear algorithms," *IEEE Trans. Evol. Comput.*, vol. 17, no. 1, pp. 77–99, Feb. 2013.

[30] H. Singh, A. Isaacs, and T. Ray, "A Pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 15, no. 4, pp. 539–556, Aug. 2011.

[31] L. Thiele, K. Miettinen, P. Korhonen, and J. Molina, "A preference-based evolutionary algorithm for multi-objective optimization," *Evol. Comput.*, vol. 17, no. 3, pp. 411–436, 2009.

[32] D. Cvetkovic and I. C. Parmee, "Preferences and their application in evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 42–57, Feb. 2002.

[33] K. Sindhya, A. B. Ruiz, and K. Miettinen, "A preference based interactive evolutionary algorithm for multi-objective optimization: PIE," in *Evolutionary Multi-Criterion Optimization*. Berlin, Germany: Springer, 2011, pp. 212–225.

[34] L. Ben Said, S. Bechikh, and K. Ghédira, "The r-dominance: A new dominance relation for interactive evolutionary multicriteria decision making," *IEEE Trans. Evol. Comput.*, vol. 14, no. 5, pp. 801–818, Oct. 2010.

[35] M. Koksalan and I. Karahan, "An interactive territory defining evolutionary algorithm: iTDEA," *IEEE Trans. Evol. Comput.*, vol. 14, no. 5, pp. 702–722, Oct. 2010.

[36] R. Wang, R. Purshouse, and P. Fleming, "Preference-inspired coevolutionary algorithms for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 4, pp. 474–494, Aug. 2013.

[37] J.-H. Kim, J.-H. Han, Y.-H. Kim, S.-H. Choi, and E.-S. Kim, "Preference-based solution selection algorithm for evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 1, pp. 20–34, Feb. 2012.

[38] I. Karahan and M. Koksalan, "A territory defining multiobjective evolutionary algorithms and preference incorporation," *IEEE Trans. Evol. Comput.*, vol. 14, no. 4, pp. 636–664, Aug. 2010.

[39] M. Farina and P. Amato, "On the optimal solution definition for many-criteria optimization problems," in *Proc. Annu. Meeting North Amer. Fuzzy Inf. Process. Soc. (NAFIPS)*, 2002, pp. 233–238.

[40] K. Deb, M. Mohan, and S. Mishra, "Evaluating the ε-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions," *Evol. Comput.*, vol. 13, no. 4, pp. 501–525, 2005.

[41] K. Bringmann, T. Friedrich, F. Neumann, and M. Wagner, "Approximation-guided evolutionary multi-objective optimization," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2011, pp. 1198–1203.

[42] M. Wagner and F. Neumann, "A fast approximation-guided evolutionary multi-objective algorithm," in *Proc. 15th Annu. Conf. Genet. Evol. Comput. Conf. (GECCO)*. Amsterdam, The Netherlands, 2013, pp. 687–694.

[43] H. Sato, H. E. Aguirre, and K. Tanaka, "Controlling dominance area of solutions and its impact on the performance of MOEAs," in *Evolutionary Multi-Criterion Optimization*. Berlin, Germany: Springer, 2007, pp. 5–20.

[44] S. Kukkonen and J. Lampinen, "Ranking-dominance and many-objective optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Singapore, 2007, pp. 3983–3990.

[45] M. Köppen, R. Vicente-Garcia, and B. Nickolay, "Fuzzy-Pareto-dominance and its application in evolutionary multi-objective optimization," in *Evolutionary Multi-Criterion Optimization*. Berlin, Germany: Springer, 2005, pp. 399–412.

[46] K. Deb and S. Tiwari, "Omni-optimizer: A procedure for single and multi-objective optimization," in *Evolutionary Multi-Criterion Optimization*. Berlin, Germany: Springer, 2005, pp. 47–61.

[47] H. Wang and X. Yao, "Corner sort for Pareto-based many-objective optimization," *IEEE Trans. Cybern.*, vol. 44, no. 1, pp. 92–102, Jan. 2014.

[48] K. McClymont and E. Keedwell, "Deductive sort and climbing sort: New methods for non-dominated sorting," *Evol. Comput.*, vol. 20, no. 1, pp. 1–26, 2012.

[49] 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 Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2014.

[50] K. Praditwong and X. Yao, "A new multi-objective evolutionary optimization algorithm: The two-archive algorithm," in *Proc. Int. Conf. Comput. Intell. Security*, vol. 1. Guangzhou, China, 2006, pp. 286–291.

[51] G. Rudolph, H. Trautmann, S. Sengupta, and O. Schütze, "Evenly spaced Pareto front approximations for tricriteria problems based on triangulation," in *Evolutionary Multi-Criterion Optimization*. Berlin, Germany: Springer, 2013, pp. 443–458.

[52] Z. Wang, K. Tang, and X. Yao, "Multi-objective approaches to optimal testing resource allocation in modular software systems," *IEEE Trans. Rel.*, vol. 59, no. 3, pp. 563–575, Sep. 2010.

[53] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, *On the Surprising Behavior of Distance Metrics in High Dimensional Space*. New York, NY, USA: Springer, 2001.

[54] R. Morgan and M. Gallagher, "Sampling techniques and distance metrics in high dimensional continuous landscape analysis: Limitations and improvements," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 456–461, Jun. 2014.

[55] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Honolulu, HI, USA, 2002, pp. 825–830.

[56] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 477–506, Oct. 2006.

[57] Q. Zhang *et al.*, "Multiobjective optimization test instances for the CEC 2009 special session and competition," School Comput. Sci. Elect. Eng., Univ. Essex, Tech. Rep. CES-887, 2008, pp. 1–30.

[58] M. Hollander and D. Wolfe, *Nonparametric Statistical Methods*. New York, NY, USA: Wiley, 1999.

Handing Wang (S'10) received the B.Eng. degree in intelligence science and technology from Xidian University, Xi'an, China, in 2010, where she is currently working toward the Ph.D. degree in circuit and system.

She is a member of the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, Xidian University. Her research interests include nature-inspired computation and multiobjective optimization.

Licheng Jiao (SM'89) received the B.S. degree from Shanghai Jiaotong University, Shanghai, China, in 1982 and the M.S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 1984 and 1990, respectively.

Since 1992 he has been a Professor with the School of Electronic Engineering, Xidian University, Xi'an, where he is currently the Director of the Key Laboratory of Intelligent Perception and Image Understanding of the Ministry of Education of China. His research interests include image processing, natural computation, machine learning, and intelligent information processing.

Dr. Jiao is a member of the IEEE Xi'an Section Execution Committee, the President of Computational Intelligence Chapter, IEEE Xi'an Section and IET Xi'an Network, the Chairman of the Awards and Recognition Committee, the Vice Board Chairperson of the Chinese Association of Artificial Intelligence, a Councilor of the Chinese Institute of Electronics, a Committee Member of the Chinese Committee of Neural Networks, and an Expert of the Academic Degrees Committee of the State Council.

Xin Yao (M'91–SM'96–F'03) received the B.Sc. degree from University of Science and Technology of China (USTC), Hefei, China, in 1982; the M.Sc. degree from North China Institute of Computing Technology, Beijing, China, in 1985; and the Ph.D. degree from the USTC.

Since 1999 he has been a Professor (Chair) of Computer Science with University of Birmingham, Birmingham, U.K., where he is currently the Director of the Centre of Excellence for Research in Computational Intelligence and Applications.

Mr. Yao received the 2001 IEEE Donald G. Fink Prize Paper Award, the 2010 IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Outstanding Paper Award, the 2011 IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding Paper Award, and several other best paper awards. He is an IEEE CIS Distinguished Lecturer and a former Editor-in-Chief of IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION.