A Decomposition-Based Multi-Objective Evolutionary Algorithm with Pareto Front Grid

Ying Xu, Huan Zhang, Lei Huang, Rong Qu, *Senior Member*, *IEEE*, Yusuke Nojima, *Member*, *IEEE*

Abstract—**Current widely used decomposition methods in MOEA/D (multi-objective evolutionary algorithms based on decomposition) showed to be lack of diversity or sensitive to the shape of the Pareto Front during the search. This research investigates the recently proposed grid-based decomposition methods, which reflect the inherent characteristics of the neighborhood structure in the solution to address the issues of diversity and sensitiveness. The performance of the grid-based decomposition method, however, depends on the size of its grid segmentation, and its time complexity increases with the number of grids. In order to improve the computational efficiency, we propose a new concept of Pareto Front grid to guide the search in MOEA/D. Based on the idea of knee point, a novel grid-based knee point selection method is proposed. In addition, a new nadir point selection strategy is also proposed based on statistical analysis estimating population with samples. Finally, a decomposition-based multi-objective evolutionary algorithm with Pareto Front Grid (PFG-MOEA) is proposed. Extensive experimental analysis demonstrates the effectiveness of the proposed PFG-MOEA against state-of-the-art multi-objective evolution algorithms. As the extension of the CDG-MOEA algorithm in the literature, PFG-MOEA is effective by consuming much less computing time.**

Index Terms—**Evolutionary Multi-objective Optimization, Pareto Front, MOEA/D.**

I. INTRODUCTION

Many real world optimization problems involve multiple conflicting objectives, such as in the airship deployment system [1] and resource scheduling [2], etc. This type of optimization problem is called multi-objective optimization problem (MOP) [3-4], its mathematical expression can be defined as shown in (1):

$$\begin{aligned} \text{minimize} \quad & F(x) = \big(f_1(x), f_2(x), \dots, f_m(x)\big) \\ \text{subject to} \quad & x \in X \end{aligned} \quad (1)$$

where X is the decision space, $x = (x_1, \dots, x_D)$ represents the decision vector, D is the dimension of the decision variable, $f_i(x)$ represents the optimization objective function for the i-th objective, $i \in \{1, \dots, m\}$, and m is the number of objectives. Due to the conflicting objectives, there is usually no single solution that optimizes all objectives at the same time. Instead a set of non-dominated solutions can be obtained as a trade-off between different objectives. For a minimization problem, assuming x_A and x_B are two solutions of MOP given in (1), x_A dominates x_B (expressed as $x_A \prec x_B$), if and only if $f_i(x_A) \le f_i(x_B)$ for $i \in \{1, \dots, m\}$, and there is at least one objective which satisfies $f_j(x_A) < f_j(x_B)$, for $j \in \{1, \dots, m\}$. The set of all non-dominated Pareto optimal solutions is called the Pareto optimal Set (PS), and the projection of PS in the objective space is called Pareto optimal Front (PF) [5-6].

In order to obtain an approximation of the Pareto optimal Set, evolutionary algorithms (EAs) have been widely used to solve various multi-objective optimization problems. A lot of multi-objective evolutionary algorithms (MOEAs) have been proposed, for example the Pareto dominance-based MOEAs [7-9], the decomposition-based MOEAs [10-12] and the indicator-based MOEAs [13-14].

Zhang et al. [10] proposed a decomposition-based multi-objective optimization algorithm (MOEA/D), which is one of the most popular MOEA frameworks in recent years. MOEA/D decomposes MOP into a series of single-objective sub-problems, and uses the evolutionary algorithm to optimize these sub-problems simultaneously. Commonly used decomposition methods in MOEA/D include the Weighted Sum approach (WS) [5], the Tchebycheff decomposition approach (TCH) [5] and the Penalty-based Boundary Intersection approach (PBI) [10]. However, these three common decomposition methods are very sensitive to the shape of Pareto Front for a MOP with irregular distributed Pareto Front. Another issue of these well-known decomposition methods is that one solution may be associated with several different sub-problems, leading to the loss of diversity of the Pareto non-dominated solution set.

Numerous works have been carried out to overcome these problems in MOEA/D. For example, Zhang et al. [15] proposed a new decomposition method combining a boundary crossover with the Tchebycheff method, which has achieved good performances for solving MOPs, especially those with two objectives. However, when the number of objectives increases, the solution qualities of the decomposition-based MOEA become not satisfactory. Sato et al. [16] proposed an inverse PBI decomposition method to effectively solve MOP with
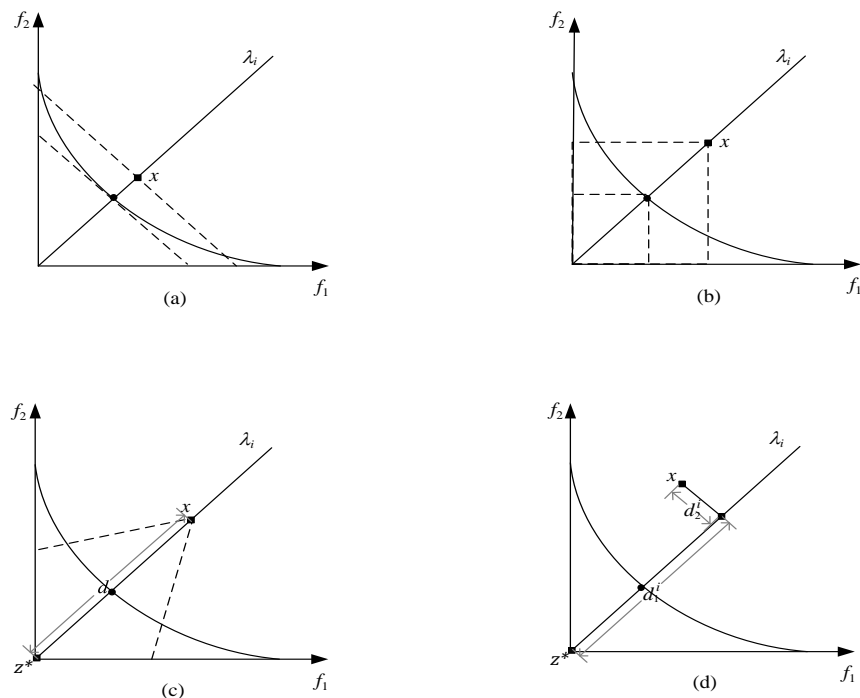
Fig. 1. Four typical decomposition methods in MOEA/D.

convex Pareto front. Asafudoula et al. [17] used an adaptive contrast method to balance the convergence and diversity of the population. In the algorithm proposed by Gee et al. [19], an on-line geometric measurement method is proposed to improve the diversity of MOEA/D. Wang et al. [20] used an angle parameter to limit the update range of each sub-problem in MOEA/D, so that the diversity of the algorithm is optimized due to the enhanced replacement of the sub-problem. However, in different stages of the search process, the setting of the angle parameter is a challenging task.

Since the decomposition-based algorithm is sensitive to the reference vector, Ge et al. [21] proposed a potential active reference point re-allocation mechanism based on a SVM classifier to reduce the number of useless reference vectors. Wang et al. [35] proposed an integrated algorithm framework Two-Arch2 with two archives (a convergence archive and a diversity archive) with different selection criteria. Combining the advantages of indicators and Pareto-based criteria, a diversity preservation mechanism was designed to solve the MOP. The 1by1EA algorithm in [36] uses an effective convergence index to select individuals one by one in order to increase the selection pressure towards the front of the Pareto optimal solution. When an individual is selected, its neighbors ensure diversity through a niche method. Other proposed algorithms include hpaEA [37] which uses reference vectors to improve the population diversity and DEA-GNG [38] which addresses the irregular PF issue by using a growth neural network.

In addition to improving the traditional decomposition method of MOEA/D, some new decomposition methods have also been proposed recently. Cai et al. [22] proposed a Constrained Decomposition approach with Grids (CDG). With the inherent characteristics of the grid reflecting neighborhood structure information of the solution, CDG can obtain satisfactory results when dealing with particularly convex or concave problems, or when there is a large gap between different objective numerical ranges. It means CDG is robust to the Pareto front of the optimization problem. However, the effect of CDG depends on the size of grid segmentation. In their work, the number of grid segmentation is set to 180, which means $180 \times 180 = 32400$ grids have been generated in the two-dimensional decision space, requiring a large storage space and consuming a long computing time. The time and space complexity of the algorithm are thus very high.

In order to improve the search efficiency, we propose a multi-objective evolutionary algorithm with Pareto Front Grid based on decomposition (PFG-MOEA), where Pareto Front Grid (PFG) is a new concept defined in this paper. It has been proven in many works [24,35] that the offspring of the more advanced individuals in the population will perform better than the offspring of other individuals. In order to improve the evolutionary pressure of the algorithm, only leading individuals in the Pareto Front Grid are stored during the evolution to guide the search. The main contributions of this paper include:

1) The idea of Pareto Front Grid (PFG) is defined, where only the individuals in PFG stored during the evolution process are used to guide the search.

2) A new nadir point selection method is proposed based on the statistical analysis estimating the population with samples.

3) In order to improve the convergence of the proposed PFG-MOEA algorithm, a grid-based knee point selection method is proposed based on the idea of knee point. The proposed

evolutionary algorithm with Pareto Front Grid based on decomposition showed a good performance on the GLT and UF test functions

The rest of this paper is organized as follows. Section 2 introduces related works; the third section describes the framework of our proposed PFG-MOEA; Section 4 tests the performance of the algorithm through a large number of experiments; finally, the fifth section summarizes the work.

II. RELATED WORK

In decomposition-based evolutionary algorithms, MOP is generally transformed into a set of single-objective sub-problems through a set of weight vectors and aggregate functions, and then these single-objective sub-problems are optimized separately. In MOEA/D [10], the MOP problem is decomposed into multiple sub-problems and then these sub-problems are optimized simultaneously. Let $\lambda^i = (\lambda_1, \ldots, \lambda_m)^T$ be the reference vector of the i-th sub-problem, where $\lambda_j \geq 0$ ($j = 1, \ldots, m$), and $\sum_{j=1}^m \lambda_j = 1$. Fig. 1 shows four commonly used decomposition methods in MOEA/D, including the weighted sum approach (WS, see Fig.1(a)), the Tchebycheff decomposition approach (TCH, see Fig.1(b)), the ordinary Boundary Intersection approach (BI, see Fig.1(c)), and the Penalty-based Boundary Intersection approach (PBI, see Fig.1(d)).

Taking the MOP problem with minimization objectives for example, for the weighted sum approach (WS), the i-th sub-problem is defined as follows in (2):

$$\begin{aligned} \text{minimize} \quad & g\big(x\big|\lambda^i\big) = \sum_{j=1}^m \lambda_j^i f_j(x) \\ \text{subject to} \quad & x \in X \end{aligned} \tag{2}$$

For the Tchebycheff decomposition approach (TCH), the i-th sub-problem is defined as shown in (3):

$$\begin{aligned} \text{minimize} \quad & g^{te}\big(x\big|\lambda^i, z^*\big) = \max_{1 \leq j \leq m}\left\{\big|f_j(x) - z^*\big|/\lambda_j^i\right\} \\ \text{subject to} \quad & x \in X \end{aligned} \tag{3}$$

where $z^* = (z_1^*, \ldots, z_m^*)^T$ is a m-dimensional reference point vector, which is composed of the optimal value for each objective function, i.e. $z_j^* = min\{f_j(x)\big|x \in X\}$, for $j = 1, \ldots, m$.

The penalty-based boundary intersection approach (PBI) is a variant of the ordinary boundary intersection method (BI). The i-th sub-problem of the BI method can be defined as in (4):

$$\begin{aligned} \text{minimize} \quad & g^{bi}(x|\lambda^i, z^*) = d \\ \text{subject to} \quad & F(x) - z^* = d\lambda^i \\ & x \in X \end{aligned} \tag{4}$$

Where d is the distance of a solution x to the idea point z^* along the reference vector λ_i as shown in Fig.1(c).

Thus, the i-th sub-problem of the PBI approach is given as follows in (5):

$$\begin{aligned} \text{minimize} \quad & g^{pbi}(x|\lambda^i, z^*) = d_1^i + \beta d_2^i \\ & d_1^i = (F(x) - z^*)^T \lambda^i / \|\lambda^i\| \\ & d_2^i = \big\|F(x) - (z^* - d_1^i \lambda^i)\big\| \\ \text{subject to} \quad & x \in X \end{aligned} \tag{5}$$

where $\|\cdot\|$ is the L_2-norm, β is the penalty parameter, d_1^i is the distance from the solution to the idea point along the reference vector and d_2^i represents the vertical distance to the reference vector.

These traditional decomposition methods based on the reference vector are very sensitive to the shape of Pareto Front. When dealing with the MOP with uneven or irregular Pareto Front, traditional algorithms are often lack of diversity in the search. In addition, according to our observation, a solution generated by these traditional decomposition methods may be associated with different sub-problems, which aggravates the diversity in solution sets.

In the recently proposed CDG algorithm [22], for each objective function $f_l(x)$, $l \in \{1, \ldots, m\}$, suppose that the distance between the ideal point z_l^* and the nadir point z_l^{nad} is divided into GK intervals, where the parameter GK is the number of grid segmentations. Then the length of an interval for the l-th objective is $d_l = (z_l^{nad} - z_l^* + 2\sigma)/GK$, where $\sigma > 0$ is a very small random number. $Grid_l(x) = [(f_l(x) - z_l^* + 2\sigma)/d_l]$ represents the location of the solution x within the intervals for the l-th objective. As shown in Fig.2(a), solution x is located in the third grid for the objective function $f_1(x)$, i.e. $Grid_1(x)=3$; and the third grid of the second objective $f_2(x)$, i.e. $Grid_2(x)=4$.

In CDG, the k-th sub-problem on the l-th objective is defined as follows in (6):

$$\begin{aligned} \text{minimize} \quad & f_l(x) \\ \text{subject to} \quad & Grid_j(x) = gk_j, \; j \neq l \\ & gk_j \in \{1, \ldots, GK\} \\ & x \in X \end{aligned} \tag{6}$$

where GK is the number of grids for each objective. So for a multi-objective optimization problem with m objectives, it can be decomposed into $m \times GK^{m-1}$ sub-problems. The solution set for the k-th sub-problem of the l-th objective $S_l(k)$ can be defined as in shown (7):

$$\begin{aligned} S_l(k) = \{x|Grid_1(x) = gk_1, \ldots, Grid_{l-1}(x) = gk_{l-1}, \\ Grid_{l+1}(x) = gk_{l+1}, \ldots, Grid_m(x) = gk_m\} \end{aligned} \tag{7}$$

Note that solution x may belong to solutions of different sub-problems for different objectives, and thus can be included in different solution sets. As shown in Fig. 2(b), x is a solution for sub-problem k_1 for objective function $f_1(x)$ and is also a solution for sub-problem k_2 for objective function $f_2(x)$. So solution sets $S_1(k_1)$ and $S_2(k_2)$ both contain solution x.

Algorithm 1 The Framework of PFG-MOEA

Input: 1) Gen_{max}: the maximum number of generations;   2) N: the population size; 3) GK: the number of the grids in each objective.

OutPut: a solution set Pop.

1: $Pop = Initialization(N)$

2: **while** $Gen < Gen_{max}$ **do**

3:   set Q as an empty set

4:   $z^* = \min_{x \in P} f(x)$, $z^{nad} = UpdateZ(P)$

5:   $(PFG_1^1, \dots, PFG_m^{GK}) = Pareto\_Grid(Pop, GK, z^*, z^{nad})$;   // PFG: Pareto Front Grid

6:   **for** j=1 to m **do**

7:    **for** i=1 to GK **do**

8:      **for each** individual x in $|PFG_j^i|$ **do**

9:        generate a random number r in $(0,1)$;

10:        **if** $r < \delta$ **then**

11:         select x_A and x_B from $PFG_j^i \cup PFG_j^{i+1}$ randomly;

12:        **else**

13:         select x_A and x_B from Pop randomly;

14:        **end if**

15:        generate an offspring v from x, x_A and x_B by the DE operator;

16:        $Q = Q \cup v$;

17:      **end for**

18:    **end for**

19:   **end for**

20:   $Pop = Environmental\_Selection(Pop \cup Q, N)$;

21:   $Gen= Gen+1$;

22: **end while**

III. THE PROPOSED PFG-MOEA

The main framework of our proposed PFG-MOEA, see Algorithm 1, consists of the following four components:

1. ***Initialization***: randomly generate N individuals to form the initial population Pop (line 1);

2. ***Generation of PFG***: obtain the ideal point z^* and the nadir point z^{nad}, generate the Pareto Front Grid (PFG) (see Section 3.2) and all solutions in PFG $\{PFG_j^i\}$ (lines 4-5);

3. ***Evolution Procedure***: for individuals in each PFG_j^i, the mating pool MP is defined as shown in (8):

$$MP = \begin{cases} PFG_j^i \cup PFG_j^{i+1} & if\ rand < \varepsilon \\ Pop & otherwise \end{cases} \quad (8)$$

where PFG_j^{i+1} is a randomly selected Pareto Front Grid adjacent to PFG_j^i, ε is the selection probability. For each solution x in PFG_j^i, two solutions x_A and x_B are randomly selected in the mating pool MP, and a new individual v formed offspring Q (lines 8-17) is generated by using the *DE* operator [12].

The DE operator works as follows. For parent solutions x, x_A, x_B, their child v is generated as in (9):

$$v^i = \begin{cases} x^i + F \times (x_A^i - x_B^i) & with\ probability\ CR \\ x^i & with\ probability\ 1 - CR \end{cases} \quad (9)$$

Here v^i represents the i-th decision variable of the offspring individual v. F is the scaling factor and CR is the cross probability. Given a given random number is less than CR, a new decision variable for the child is generated based on differential evolutionary, otherwise the decision variable remains unchanged.

4. ***Environmental Selection***: N individuals will be selected as the next generation from the joint population $Pop \cup Q$ (line 20).

In the early stage of evolution, the obtained Pareto Front Grid and the solution in the grid will not be too scattered, which provides a better evolutionary pressure for the evolution of future generations. The above procedure repeats until the termination condition is met. The pseudo code of the main framework of PFG-MOEA is given in Algorithm 1.

A. The Statistic Estimation based Nadir Point Selection

In order to obtain the Pareto Front Grid, we need to locate the ideal point z^* and the nadir point z^{nad} in the current population. The value for each objective of the ideal point z^* is the optimal value for each objective of all solutions in the current population. For the nadir point z^{nad}, a commonly used method is to take the worst value of each objective of all the solutions in the current population. However, we observed that if the value of the nadir point z^{nad} is set to a very large value, it can easily produce a large number of dominated solutions which may lead to a slow convergence.
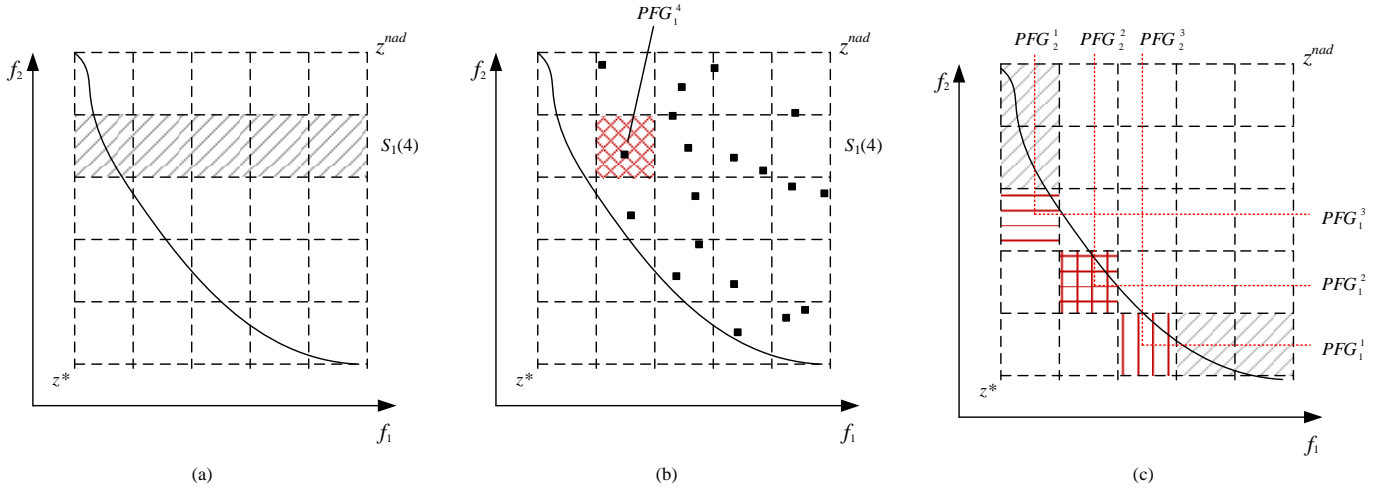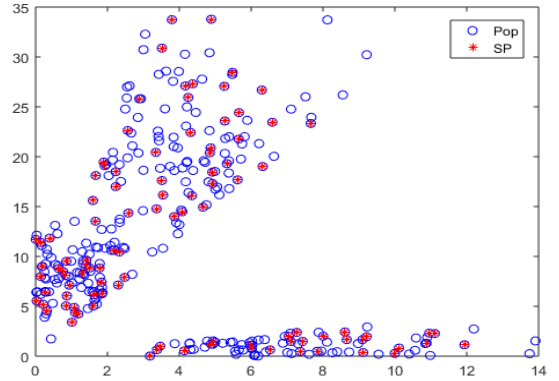
Fig. 4. The definition of the Pareto Front Grid

Algorithm 2 The proposed selection of the nadir point z^{nad}

Input: *Pop*

OutPut: z^{nad}.

1: randomly select $|Pop| / 3$ solutions from *Pop*, *SP* is the sub-population of these selected solutions;

2: $NDS\text{-}SP = Non\text{-}Dominated\text{-}Sort(SP)$
// Function: *Non-Dominated-Sort(SP)* is defined to sort the //SP using the non-dominated sort in //NSGA-II [7]

3: **for** $i = 1$ to m **do**

4: $z^{nad} = \max\limits_{x \in NDS-SP} f_i(x)$

5: **end for**

Fig. 3. Initial population Pop and subpopulation SP of GLT1

In this paper, a new statistic estimation strategy for estimating the nadir point z^{nad} is thus proposed by using a number of sample individuals to estimate the population distribution. The applicability of the statistic estimation strategy was tested on the test problem GLT1 [27] with two objectives. After a number of preliminary experiments, we found that 1/3 of the randomly selected individuals in the initial population can statistically estimate the distribution of the whole population. As shown in Fig. 3, for the initial population *Pop*, 1/3 of the individuals have been randomly selected to construct the sub-population *SP*. The mean value of $f_1(x)$ in the initial population *Pop* is 4.0171 with a standard deviation of 2.9116; and the mean value of $f_2(x)$ is 11.6525 with a standard deviation of 9.1517. For the sub-population *SP*, the mean value of $f_1(x)$ is 4.0574, the standard deviation is 3.0056; the mean on $f_2(x)$ is 11.2871, and the standard deviation is 9.1985. We then performed the one-way statistical analysis of the variance on results

for both objectives. For $f_1(x)$, the corresponding P value is 0.91>0.05, which is greater than the significance level, so there is no significant difference between the population *Pop* and the sub-population *SP* at the 0.05 level. For $f_2(x)$, the corresponding P value is 0.73>0.05, which is greater than the significance level, so there is no significant difference between *Pop* and *SP* at the 0.05 level. This statistic analysis justifies that the sub-population *SP* of 1/3 of the individuals can be used to statistically estimate the initial population *Pop*. This statistic estimation strategy is thus used in our proposed PFG-MOEA, where 1/3 individuals are selected from the whole population to estimate the actual nadir point z^{nad} of the current population and reduce the computing time.

The pseudo code of the proposed nadir point selection is shown in Algorithm 2. Firstly, one third of the individuals is randomly selected from the current population *Pop* to form a sub-population *SP* (line 1). Then the sub-population *SP* is sorted by the non-dominated sorting. The first layer of non-dominated solutions is selected to reconstruct the *NDS-SP* (line 2). The maximum value of each objective of individuals in the *NDS-SP* is set as the value of the nadir point z^{nad} for each objective.

Algorithm 3 The Generation of Pareto Front Grids

Input: *Pop, GK, z^*, z^{nad}*

OutPut: $\{PFG_j^i\}$

1: **for** i = 1 to m **do**

2: 　$d_j = (z_j^{nad} - z_j^* + 2\sigma)/GK$

3: **end for**

4: **for each** x in *Pop* **do**

5: 　**for** j = 1 to m **do**

6: 　　$Grid_j(x) = \lceil (f_j(x) - z_j^* + \sigma)/d_j \rceil$

7: 　**end for**

8: 　$Grid(x) = (Grid_1(x), \ldots, Grid_m(x))$

9: **end for**

10: **for** i = 1 to m **do**

11: 　**for** j = 1 to GK **do**

12: 　　Find the solution set $S_i(j)$

13: 　　$g_{min} = \min\{ Grid(S_i(j))\}$

14: 　　**for each** x in $S_i(j)$ **do**

15: 　　　**if** $Grid_j(x) = g_{min}$ **then**

16: 　　　　$PFG_j^i = PFG_j^i \cup x$

17: 　　　**end if**

18: 　　**end for**

19: 　**end for**

20: **end for**

Algorithm 4 The Environmental Selection

Input: 1) *U*: the combined population from *Pop* and the offspring, 2) *N*: the population size;

OutPut: a new solution set *Pop*.

1: [{*F*}, *l*] = *Non-Dominated-Sort* (*U*)

// Function: *Non-Dominated-Sort(SP)* is defined to sort the //SP using the non-dominated sort in //NSGA-II [7]

2: $Pop = F_1 \cup \cdots \cup F_{l-1}$

3: [*Knee_Point*] = *Find_Knee_Point*(F_l)

// Function: *Find_Knee_Point*(F_l) is defined to find Knee Point as in KnEA [24]

4: $Pop = Pop \cup (F_l \cup Knee\_Point)$

5: **if** $|Pop| > N$ **then**

6: 　delete $|Pop| - N$ solutions from *Pop* which belong to ($F_l \cup Knee\_Point$) and have the maximum distance to the hyper-plane

7: **else**

8: 　add $N - |Pop|$ solutions from $F_l - (F_l \cup Knee\_Point)$ which have the minimum distance to the hyper-plane

9: **end if**

B. The Definition of the Pareto Front Grid

We define the Pareto Front Grid as the grid which contains the optimal solution for each objective function in the solution set of each sub-problem within the current population. As shown in Fig. 4, the solution set $S_1(4)$ of the 4th sub-problem for the 1st objective is shown in the shadow section of Fig. 4(a), and the solutions of population *Pop* are shown in Fig. 4 (b). We can see that the shadow grid contains the optimal solution of solution set $S_1(4)$ for objective function $f_1(x)$, which is defined as the Pareto Front Grid PFG_1^4 corresponding to the 4th sub-problem of the 1st objective function $f_1(x)$.

After locating the ideal point z^* and the nadir point z^{nad}, each objective is divided into GK equal intervals, and the interval length d_j for the j-th objective is shown in (10):

$$d_j = (z_j^{nad} - z_j^* + 2\sigma)/GK \tag{10}$$

For each solution x in population *Pop*, the grid coordinate $Grid_j(x)$ to the j-th objective is calculated as shown in (11):

Fig. 5. The grid-based Knee point

$$Grid_j(x) = \lceil (f_j(x) - z_j^* + \sigma)/d_j \rceil \tag{11}$$

where $\lceil \cdot \rceil$ is an upward rounding function. For m objectives, the grid coordinate $Grid(x)$ of solution x is defined as shown in (12):

$$Grid(x) = (Grid_1(x), \ldots, Grid_m(x)) \tag{12}$$

For each objective function $f_j(x)$, for $j = 1, \ldots, m$, g_{min} represents the minimum value of the grid coordinate $Grid_j(x)$ among all solutions in $S_j(i)$ for the i-th sub-problem. Solutions with the grid coordinate $Grid_j(x) = g_{min}$ in $S_j(i)$ will be selected to form PFG_j^i. So we can obtain all Pareto Front Grid $\{PFG_j^i\}$, i represents the i-th sub-problem, as shown in the shadow part of Fig. 4(c). Note that the PFG of the sub-problem on different objectives may be repetitive, such as PFG_1^3 and PFG_2^1 (red bar shadow grids in Fig. 4(c)), PFG_1^2 and PFG_2^2 (red square shadow grids in Fig. 4(c)), PFG_1^1 and PFG_2^3 (red vertical shadow grids in Fig. 4(c)).

Fig. 6. The comparison of IGD values

TABLE I
CHARACTERISTIC OF THE GLT AND UF TEST FUNCTIONS

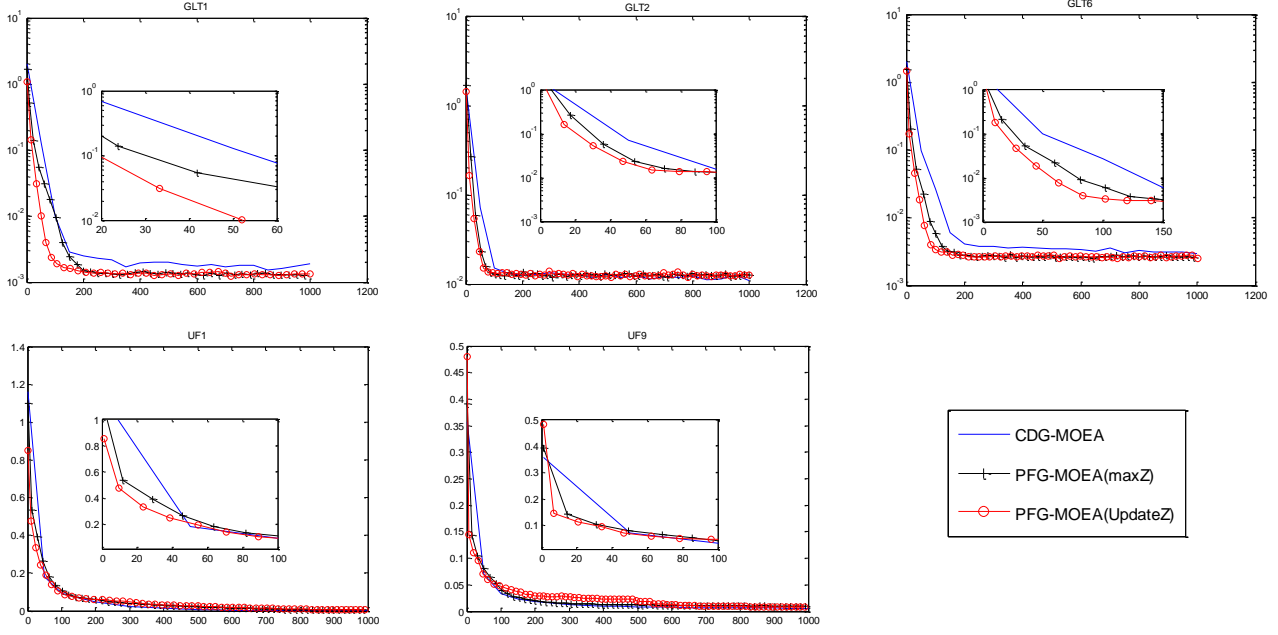| Test Problem	m	N	D	Feature
UF1	2	300	30	Convex
UF2	2	300	30	Convex
UF3	2	300	30	Convex
UF4	2	300	30	Concave
UF5	2	300	30	Linear, Disconnected
UF6	2	300	30	Linear, Disconnected
UF7	2	300	30	Linear
UF8	3	600	30	Concave
UF9	3	600	30	Disconnected
UF10	3	600	30	Concave
GLT1	2	300	10	Linear, Disconnected
GLT2	2	300	10	Convex
GLT3	2	300	10	Convex
GLT4	2	300	10	Convex, Disconnected
GLT5	3	300	10	Convex
GLT6	3	300	10	Convex, Disconnected

C. The Environment Selection based on Knee Point

Based on the characteristics of the grid decomposition, we adopt the concept of knee point [23-25] in each PFG and propose the grid-based knee point selection on offspring population to form the next generation.

As can be seen from Fig. 5, in the process of generating the grid, solutions are divided into different grids, each containing a number of neighboring solutions within the grid (i.e. the black dots represent solutions in the same grid). The dotted blue line is drawn in Fig. 5 between the ideal point with the nadir point, and the vertical blue line represents the Hyper-plane in Fig. 5. The knee point is defined as the solution which is the closest to the Hyper-plane in the grid.

By using the fast non-dominate sorting, the first l layer Pareto fronts $\{F_1, \ldots, F_l\}$ can be obtained, where ($|F_1| + \cdots + |F_l| > N$, $|F_1| + \cdots + |F_{l-1}| < N$), $Pop = F_1 \cup \cdots \cup F_{l-1}$) (lines 1-2). For each solution x in F_l, the distance to the Hyper-plane is calculated. All knee points are added to the population Pop (lines 3-4). If $|Pop| > N$, then $|Pop| - N$ solutions with the largest distance to the Hyper-plane are removed. If $|Pop| < N$, then $N - |Pop|$ solutions which have the smallest distance to the Hyper-plane are added to Pop (lines 5-9).

IV. PERFORMANCE EVALUATION

In this section, we first provide the basic information of the benchmark test instances used in this experiment, then present

TABLE II
PARAMETER SETTING

Parameters	Value
The crossover rate in the DE operator	$CR = 1.0$
The weighting factor in the DE operator	$F = 0.5$
The distribution index	$\eta = 20$
The mutation rate	$p_m = 1/n$
The number of evaluation times	3000000
the neighboring solution size in MOEA/D-DE	20
the probability of selecting neighbor in CDG-MOEA	$\delta = 0.9$
GK for the two-dimensional optimization problem in CDG-MOEA	180
GK for the three-dimensional optimization problem in CDG-MOEA	30
GK in PFG-MOEA	5

TABLE III
COMPARISONS OF IGD ON GLT PROBLEMS

instance		PFG-MOEA	CDG-MOEA	ε-MOEA	GrEA	MOEA/D-DE	MOEA/D	MSOPS-II	NSGA-III
GLT1	mean	**1.242E-03**	1.918E-03	1.409E-01	6.835E-02	1.179E-03	1.422E-01	8.231E-02	4.043E-02
	std	2.300E-05	1.476E-04	5.575E-02	4.997E-02	9.424E-07	7.864E-02	1.897E-02	1.396E-02
GLT2	mean	**9.911E-03**	1.666E-02	5.504E-01	9.140E-02	1.395E-01	1.862E+00	2.465E-01	6.284E-02
	std	2.344E-04	1.897E-03	8.896E-02	1.065E-01	2.500E-02	1.283E-02	1.800E-01	3.977E-02
GLT3	mean	**1.800E-03**	1.890E-03	3.394E-01	4.339E-02	6.094E-03	3.092E-01	1.014E-01	3.294E-02
	std	2.137E-05	3.900E-05	9.738E-02	2.087E-02	7.864E-05	1.083E-03	4.416E-03	1.489E-02
GLT4	mean	**2.530E-03**	2.985E-03	3.410E-01	9.472E-02	3.383E-03	2.255E-01	3.061E-02	8.999E-02
	std	3.922E-02	5.005E-05	1.317E-01	6.179E-02	5.105E-05	2.613E-01	5.857E-03	7.395E-02
GLT5	mean	**1.975E-02**	2.372E-02	1.822E-01	1.077E-01	1.169E-01	1.492E-01	1.189E-01	9.520E-02
	std	3.697E-03	1.735E-03	2.228E-02	3.635E-03	2.885E-04	1.459E-03	6.514E-03	2.531E-03
GLT6	mean	**1.550E-02**	3.178E-02	1.588E-01	1.171E-01	1.190E-01	2.559E-01	1.796E-01	1.167E-01
	std	2.305E-03	5.346E-03	1.048E-02	6.230E-03	1.068E-05	7.422E-02	4.364E-02	1.791E-03
best/all		6/6	1/6	0/6	0/6	0/6	0/6	0/6	0/6
+/−/≈			6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0

TABLE IV
COMPARISONS OF IGD ON UF PROBLEMS

instance		PFG-MOEA	CGD-MOEA	ε-MOEA	GrEA	MOEA/D-DE	MOEA/D	MSOPS-II	NSGA-III
UF1	mean	**2.084E-03**	2.592E-03	1.241E-01	8.100E-02	2.164E-03	2.180E-01	7.208E-02	8.501E-02
	std	4.843E-04	8.238E-05	1.773E-02	6.140E-03	4.535E-04	8.952E-02	1.227E-02	7.699E-03
UF2	mean	**4.969E-03**	9.264E-03	7.234E-02	2.550E-02	1.161E-02	1.267E-01	4.314E-02	2.463E-02
	std	1.654E-03	1.335E-03	4.200E-03	6.498E-03	9.155E-03	5.994E-02	9.458E-03	2.623E-03
UF3	mean	**3.808E-03**	2.835E-02	2.220E-01	1.367E-01	9.245E-03	3.060E-01	2.967E-01	1.604E-01
	std	4.709E-02	1.054E-02	5.687E-02	2.210E-02	6.751E-03	2.817E-02	1.495E-02	3.197E-02
UF4	mean	**4.280E-02**	4.372E-02	7.631E-02	4.319E-02	6.461E-02	5.139E-02	4.018E-02	4.062E-02
	std	4.194E-03	7.395E-04	5.292E-03	8.043E-04	5.114E-03	2.664E-03	4.385E-04	3.581E-04
UF5	mean	2.356E-01	**1.704E-01**	2.681E-01	2.050E-01	3.941E-01	4.896E-01	2.440E-01	2.437E-01
	std	1.336E-01	2.641E-02	6.040E-02	3.224E-02	2.028E-01	7.231E-02	4.193E-02	4.857E-02
UF6	mean	**5.331E-02**	8.604E-02	2.247E-01	1.166E-01	3.907E-01	3.472E-01	1.634E-01	1.263E-01
	std	8.827E-02	1.432E-02	1.290E-01	8.692E-03	2.840E-01	1.376E-01	1.227E-01	4.313E-03
UF7	mean	**2.609E-03**	3.282E-03	1.577E-01	3.719E-02	8.586E-03	4.478E-01	1.647E-01	3.119E-02
	std	6.395E-02	1.481E-04	1.443E-01	5.784E-03	8.009E-03	6.592E-02	1.759E-01	4.742E-03
UF8	mean	1.380E-01	**8.539E-02**	4.814E-01	2.230E-01	7.835E-02	3.468E-01	2.326E-01	5.191E-01
	std	1.512E-01	1.387E-02	1.132E-01	1.276E-01	1.886E-02	2.846E-01	1.455E-02	1.730E-02
UF9	mean	**5.333E-02**	5.814E-02	2.410E-01	1.060E-01	1.060E-01	2.639E-01	2.527E-01	1.869E-01
	std	1.056E-01	7.899E-03	1.317E-01	3.845E-02	7.957E-02	1.875E-02	8.799E-02	4.801E-02
UF10	mean	9.456E-01	1.779E+00	5.578E-01	**3.087E-01**	5.128E-01	7.596E-01	3.087E-01	3.652E-01
	std	4.789E-01	2.790E-01	8.686E-02	3.823E-02	5.615E-02	1.463E-01	6.669E-02	4.288E-02
best/all		7/10	2/10	0/10	1/10	0/10	0/10	0/10	0/10
+/−/≈			8/2/0	10/0/0	8/1/1	10/0/0	10/0/0	10/0/0	9/1/0

TABLE V
COMPARISONS OF HV ON GLT PROBLEMS

instance		PFG-MOEA	CDG-MOEA	ε-MOEA	GrEA	MOEA/D-DE	MOEA/D	MSOPS-II	NSGA-III
GLT1	mean	**8.482E-01**	8.128E-01	5.161E-01	6.365E-01	8.297E-01	5.567E-01	6.474E-01	6.710E-01
	std	1.680E-02	3.679E-03	9.400E-02	7.764E-02	2.151E-02	8.780E-02	4.418E-02	3.672E-02
GLT2	mean	**1.247E+01**	1.223E+01	1.132E+01	1.204E+01	1.217E+01	9.948E+00	1.164E+01	1.209E+01
	std	9.864E-03	2.245E-04	2.875E-02	1.988E-02	1.150E-02	3.962E-03	3.172E-02	1.276E-02
GLT3	mean	**1.418E+00**	1.389E+00	1.320E+00	1.378E+00	1.389E+00	1.350E+00	1.367E+00	1.381E+00
	std	1.340E-02	6.441E-04	8.426E-03	7.225E-03	5.225E-03	5.683E-03	5.595E-03	7.225E-03
GLT4	mean	**1.729E+00**	1.635E+00	1.099E+00	1.537E+00	1.666E+00	1.473E+00	1.558E+00	1.580E+00
	std	1.444E-02	9.133E-05	1.453E-01	1.445E-02	1.834E-02	6.393E-02	7.280E-03	1.350E-02
GLT5	mean	**1.687E+00**	1.676E+00	1.500E+00	1.597E+00	1.617E+00	1.619E+00	1.616E+00	1.625E+00
	std	2.442E-04	2.481E-03	1.917E-02	6.595E-03	2.608E-03	1.079E-02	7.000E-03	8.820E-03
GLT6	mean	1.674E+00	**1.686E+00**	1.531E+00	1.596E+00	1.619E+00	1.349E+00	1.545E+00	1.616E+00
	std	1.166E-03	1.281E-03	9.985E-03	1.532E-02	9.418E-03	1.432E-01	2.712E-02	1.071E-02
best/all		5/6	1/6	0/6	0/6	0/6	0/6	0/6	0/6
+/−/≈			6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0

describe the parameter settings of this experiment. Three groups of experiments have been carried out as follows:

1) The comparison of the performance of the proposed Nadir Point Selection Method

2) The performance comparison between PFG-MOEA and State-of-the-Arts Approaches:
   a) Comparison of IGD values of experimental results of various algorithms
   b) Comparison of HV values of experimental results of various algorithms
   c) Comparison of Pareto front graphs of experimental results of various algorithms

3) The comparison of the computing time.

A. Benchmark Test Instances

The performance of the proposed PFG-MOEA is firstly tested on 16 benchmark functions, including UF1-10 [26] and GLT1-6 [27, 28] shown in Table I. In the table, m is the objective number, N is the population size, and D is the number of decision variables. Among these problems, UF1-7 and GLT1-4 are two-objective optimization problems, UF8-10 and GLT5-6 are three-objective optimization problems. UF series test functions are commonly used benchmark functions, which can be used to demonstrate the convergence and diversity of algorithms for MOPs. GLT test functions have more complex PFs, which can be used to test the robustness of algorithms for solving complex MOPs.

B. Performance Evaluation Metrics

Two commonly used performance evaluation metrics are used to evaluate the performance of our proposed algorithms.

1) *Inverted Generational Distance* (IGD) [29] is to measure the convergence and diversity of the approximate solution set, which calculates the average distance between the approximate solution set and a set of uniformly sampled solutions of the true PF. The smaller the IGD value, the better the performance of the solution set.

Let P^* be a set of uniformly sampled solutions of the true PF

of the MOP, and Pop be a set of approximate non-dominated solutions obtained by the optimization algorithm. Then IGD of Pop can be expressed as follows in (13):

$$\text{IGD}(Pop, P^*) = \frac{1}{|P^*|} \sum_{v \in P^*} dist(v, Pop) \quad (13)$$

Where $dist(v, Pop)$ calculates the Euclidean distance between solution v and the nearest solution in the solution set Pop, and $|P^*|$ is the number of solutions in P^*.

2) *Hypervolume* (HV) [30] is a metric to measure the area that is dominated by a set of solutions for a given reference point as the boundary. The larger the HV value, the better this set of solutions.

Assuming Pop is a set of solutions and $r = (r_1, \dots, r_m)^T$ is the reference point, HV can be calculated as follows in (14):

$$HV(S) = volumn\left(\bigcup_{x \in Pop}[x_1, r_1] \times \cdots \times [x_m, r_m]\right) \quad (14)$$

Both IGD and HV values can measure the convergence and diversity of a solution set. We select these two metrics to compare the quality of solution sets generated by different algorithms.

C. Experimental Settings

The parameter settings in the comparative experiments are shown in Table II. All experiments were conducted via PlatEMO [31] with MATLAB R2016a on Intel Core i7-8700k (4.70GHz). Each algorithm was run independently for 30 times on each test problem, and the average value of each metric and the standard deviation are compared.
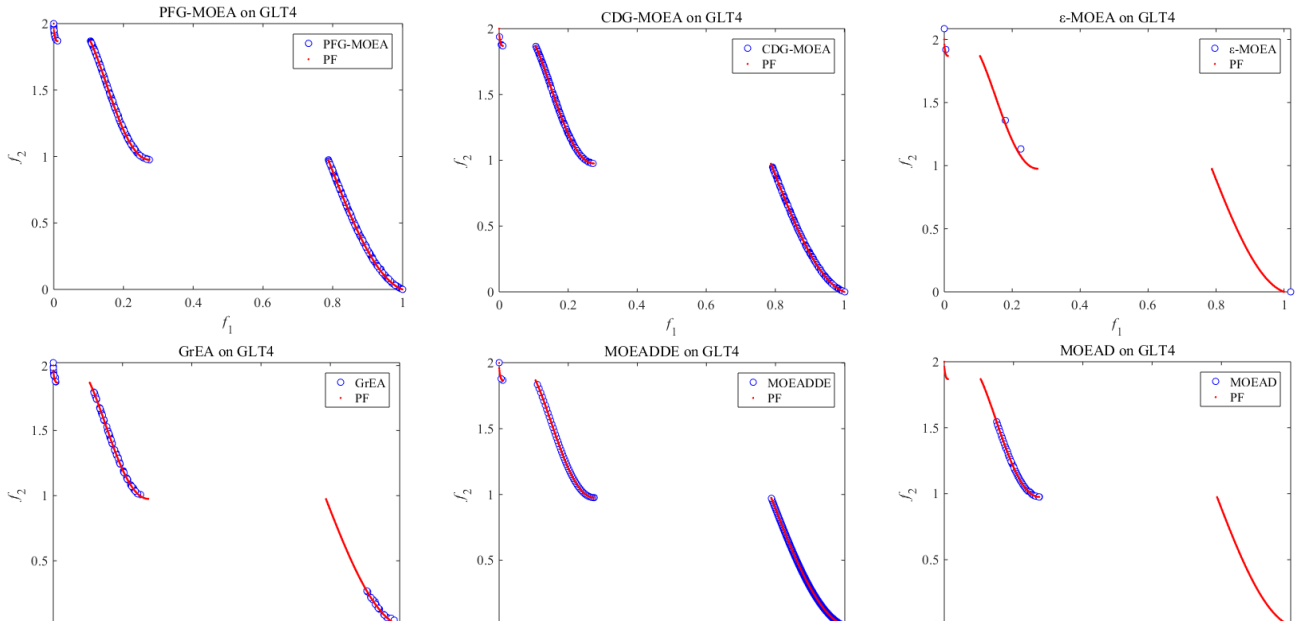
TABLE VI
COMPARISONS OF HV ON UF PROBLEMS

instance		PFG-MOEA	CGD-MOEA	ε-MOEA	GrEA	MOEA/D-DE	MOEA/D	MSOPS-II	NSGA-III
UF1	mean	**3.724E+00**	2.836E+00	1.635E+00	1.702E+00	1.816E+00	1.469E+00	1.652E+00	1.711E+00
	std	7.851E-03	9.105E-03	9.633E-03	3.928E-02	1.493E-02	9.422E-02	1.643E-02	2.628E-02
UF2	mean	**3.708E+00**	2.824E+00	1.718E+00	1.799E+00	1.807E+00	1.678E+00	1.740E+00	1.797E+00
	std	1.078E-02	1.011E-02	1.871E-02	1.932E-02	1.695E-02	2.772E-02	2.270E-02	5.727E-03
UF3	mean	**3.676E+00**	2.938E+00	1.368E+00	1.526E+00	1.830E+00	1.286E+00	1.309E+00	1.460E+00
	std	5.886E-02	1.180E-02	4.924E-02	1.681E-02	1.776E-02	2.030E-02	5.459E-03	3.797E-02
UF4	mean	**3.320E+00**	3.141E+00	1.577E+00	1.604E+00	1.594E+00	1.538E+00	1.610E+00	1.611E+00
	std	1.500E-02	1.366E-02	1.417E-02	1.190E-02	1.011E-02	1.441E-02	1.055E-02	1.033E-02
UF5	mean	2.648E+00	**3.157E+00**	1.275E+00	1.474E+00	1.151E+00	1.019E+00	1.272E+00	1.304E+00
	std	5.729E-02	2.484E-02	4.579E-02	6.433E-02	6.582E-02	4.583E-02	3.094E-02	3.379E-02
UF6	mean	**3.264E+00**	3.136E+00	1.302E+00	1.391E+00	1.224E+00	1.140E+00	1.376E+00	1.444E+00
	std	1.009E-01	2.099E-02	9.416E-02	7.445E-02	1.236E-01	7.614E-02	4.314E-02	7.117E-02
UF7	mean	**3.588E+00**	2.732E+00	1.558E+00	1.701E+00	1.718E+00	1.142E+00	1.524E+00	1.722E+00
	std	1.511E-02	1.161E-02	9.388E-02	1.062E-02	2.528E-02	1.607E-02	1.343E-01	6.221E-03
UF8	mean	6.139E+00	**7.323E+00**	6.403E+00	6.827E+00	7.243E+00	6.005E+00	6.427E+00	6.400E+00
	std	8.830E-02	1.054E-03	1.778E-02	4.768E-02	9.044E-03	1.531E-01	1.513E-02	5.099E-03
UF9	mean	**7.754E+00**	7.653E+00	6.648E+00	7.350E+00	7.406E+00	6.147E+00	6.058E+00	6.718E+00
	std	1.207E-01	6.992E-04	5.529E-02	3.322E-02	3.251E-02	2.502E-02	4.785E-02	3.841E-02
UF10	mean	4.019E+00	2.185E+00	4.171E+00	5.946E+00	3.514E+00	3.469E+00	6.096E+00	**6.131E+00**
	std	2.920E-04	2.796E-02	1.359E-01	6.685E-02	1.295E-02	5.263E-02	1.986E-02	3.670E-02
best/all		7/10	2/10	0/10	1/10	0/10	0/10	0/10	1/10
+/−/≈			8/2/0	10/0/0	8/1/1	10/0/0	10/0/0	10/0/0	9/0/1

D. The Effectiveness of the Proposed Nadir Point Selection Method

To verify the effectiveness of the statistic strategy based nadir point selection method, two variants of PFG-MOEA have been designed, namely PFG-MOEA (UpdateZ) with the new nadir point selection method, and PFG-MOEA (maxZ) with the classical strategy, where the nadir point is set as the maximum value for each objective in the population. We compare PFG-MOEA (UpdateZ) and PFG-MOEA (maxZ) with the recent CDG-MOEA with respect to IGD values. PFG-MOEA (UpdateZ) converged the fastest on 17 out of 23 test problems, and has the same convergence speed as the PFG-MOEA (maxZ), while converged slower on 3 test functions. The results of the IGD convergence curve of three algorithms on some selected test problems are shown in Fig. 6. We can see that PFG-MOEA (UpdateZ) with the new nadir point selection method can obtain better IGD values than the other two algorithms and PFG-MOEA (UpdateZ) also has the best the convergence speed of IGD.
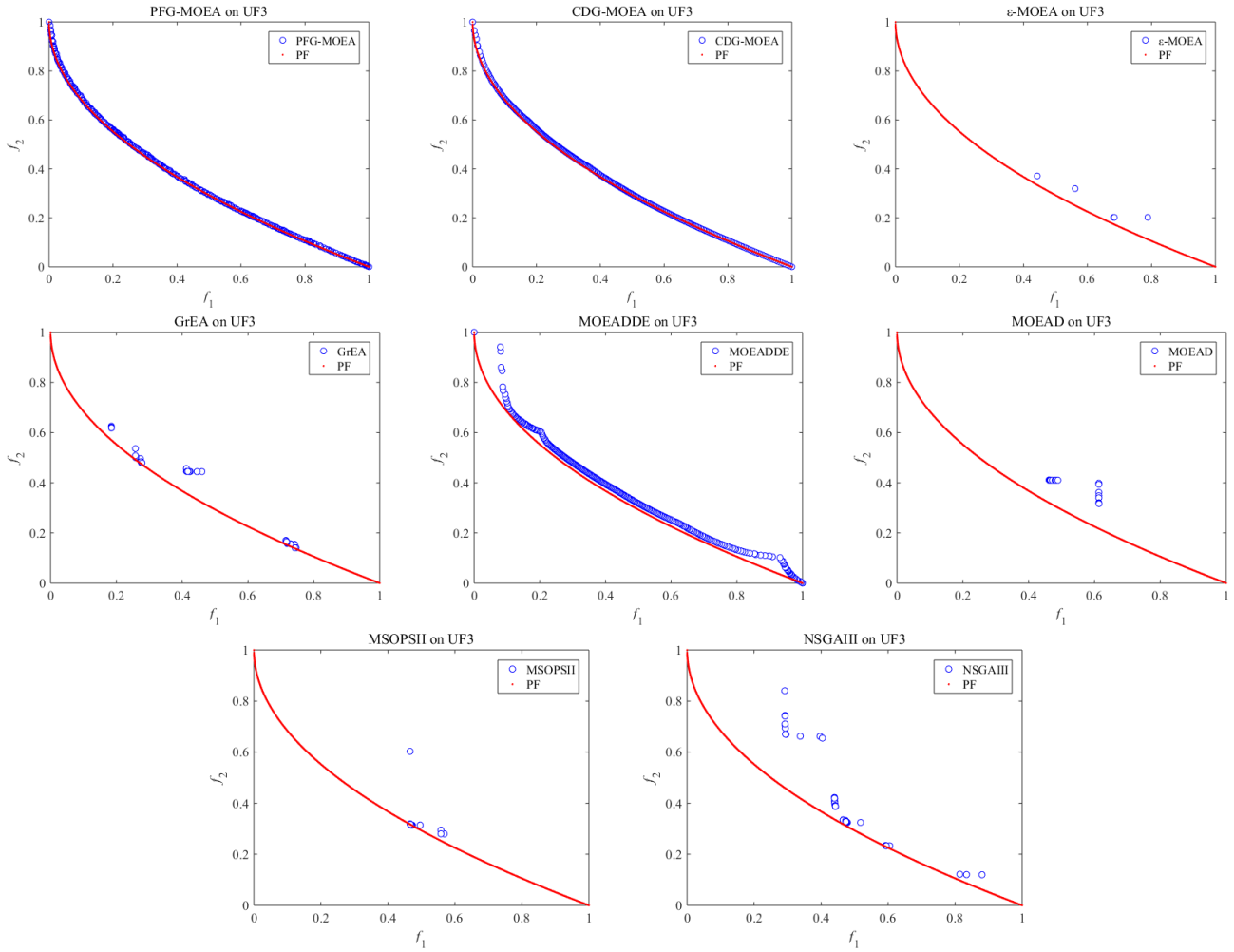
Fig. 8. The comparison of Pareto Fronts on the UF3 Problem

E. The Comparison of PFG-MOEA with State-of-the-Arts Approaches

In order to test the performance of PFG-MOEA, we compare PFG-MOEA with seven state-of-the-art algorithms, including three grid-based algorithms (CDG-MOEA [22], ε-MOEA [32], GrEA [33]), three decomposition-based algorithms (MOEA/D-DE [12], MOEA/D [10] and MSOPS-II [34]), and one domination based algorithm (NSGA-III [8]). All algorithms were performed 30 independent runs on the 16 test GLT and UF functions, and the mean value of the results and standard deviation at a significance level of 0.05 are reported in Table III.

1) The Comparison of IGD

Table III and Table IV show the experimental results of the IGD values of PFG-MOEA compared with the other seven algorithms. The best average IGD values have been marked in bold in grayed table cells. In the tables, the symbol "+", "−" and "≈" indicates that PFG-MOEA is significantly better than, worse than, and of no significant difference to the other algorithms, respectively. These statistical results summarized in the last row of each table indicate that for the GLT series functions, PFG-MOEA outperforms other algorithms on all six
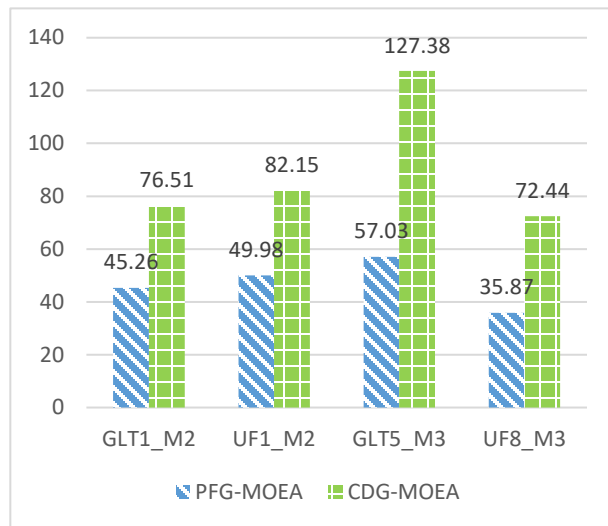
GLT problems in terms of IGD value. For the UF test functions, PFG-MOEA outperforms other algorithms on 7 out of 10 problems. For the UF5 and UF8 functions, CDG-MOEA performed the best, and GrEA outperformed other algorithms on the UF10 function. PFG-MOEA has the overall best results among the eight algorithms.

2) The Comparison of HV

Table V and Table VI compare the HV values of PFG-MOEA with those of other seven algorithms. The best average results have been marked in bold with the gray shadow. From the experimental results, we can see that for the GLT functions, PFG-MOEA has the best performance on 5 out of 6 GLT problems. For UF functions, PFG-MOEA outperformed 7 out of 10 test problems. CDG-MOEA obtained the best results on the UF5 and UF8 functions, and NSGA-III performed the best on the UF10 function. The above results show that PFG-MOEA has the overall best performance for most cases.
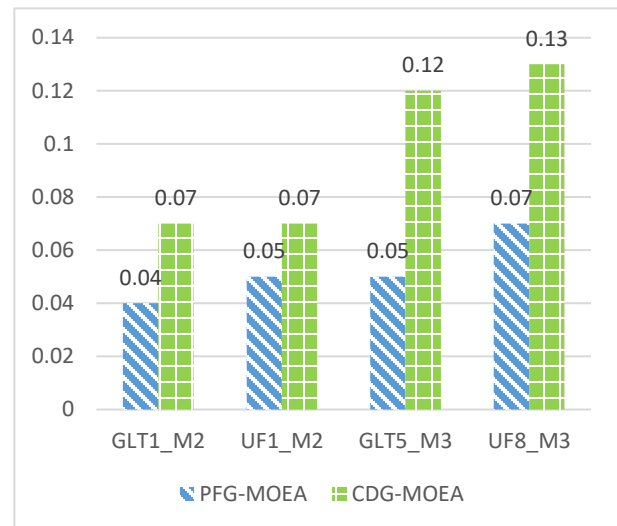
3) The Comparison of Pareto Fronts

Fig. 7 shows Pareto fronts obtained by different algorithms on the GLT4 function. It can be seen that PFG-MOEA obtained well distributed Pareto Front along the true PF, which is better than most of the other algorithms. Although CDG-MOEA and

(a) The average computation time

(b) The average computing time in one generation

Fig. 9. The comparison of computing time of PFG-MOEA and CDG-MOEA

MOEA/D-DE perform slightly better on some regions of the PF, but some solutions are missing in the first segment of the PF.

Fig. 8 shows the Pareto fronts generated by the eight algorithms on the UF3 function. Due to the complexity of the problem, it can be seen that the grid-based algorithms PFG-MOEA and CDG-MOEA perform better than the other six algorithms. PFG-MOEA performed better than CDG-MOEA in the first segment of the PF by finding more Pareto solutions.

Experimental results in Fig. 7 and Fig. 8 demonstrate that PFG-MOEA obtained the best Pareto fronts in terms of the diversity for problems with both regular and irregular Pareto Fronts compared with other seven algorithms.

4) The Comparison of the Computing Time

In this group of experiments, we compare the computing time of PFG-MOEA and CDG-MOEA on four representative test instances GLT1, UF1, GLT5, and UF8. We set the population size to 300 on instances GLT1-6 and UF1-7, while the population size is set to 500 on UF8-10. For a fair comparison, all algorithms are set the same parameters. The average computing time is shown in Fig.9(a). In order to compare the time required for the algorithm in each generation, the average evolution time of each generation is shown in Fig.9(b). It can be seen that for all four instances, the computing time of PFG-MOEA is much less than that of CDG_MOEA. We also observed that the computation time of PFG-MOEA grows relatively slower than CDG-MOEA, with the increasing of the number of objectives.

V. CONCLUSION

In this work, we propose a new grid-based decomposition multi-objective evolutionary algorithm, namely PFG-MOEA, to make full use of the inherent characteristics of the grid which reflect the information of the neighborhood structure in the solution. The definition of Pareto Front Grid has been proposed for the first time, which showed to guide the search during the evolutionary process. Based on the idea of knee point, a new grid-based knee point selection method has also been proposed. A novel statistical estimation-based nadir point selection strategy has been presented. With these new mechanisms, the PFG-MOEA has demonstrated the overall best performance on most benchmark test problems compared with seven other state-of-the-art multi-objective evolutionary algorithms. As the extension of CDG-MOEA in the literature, PFG-MOEA is effective by consuming much less computing time. Future research includes the extension of PFG-MOEA for solving many-objective optimization problems and the application of PFG-MOEA to real world multi/many-objective problems.

REFERENCES

[1] M. Gong, Z. Wang, Z. Zhu *et al*., "A Similarity-Based Multiobjective Evolutionary Algorithm for Deployment Optimization of Near Space Communication System." *IEEE Trans. Evol. Comput*., vol.21, no. 6, pp. 878-897, 2017.

[2] Y. Xu, K. Li, J. Hu *et al*., "A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues." *Inform. Sci.*, vol.270, pp. 255-287, 2014.

[3] Y. Xu, and R. Qu , "Solving multi-objective multicast routing problems by evolutionary multi-objective simulated annealing algorithms with variable neighbourhoods." *J. Oper. Res. Soc.,* vol.62 , no. 2, pp. 313-325, 2011.

[4] Y. Xu, O. Ding, *et al*., "Hybrid Multi-objective Evolutionary Algorithms based on Decomposition for Wireless Sensor Network Coverage Optimization." *Appl. Soft Comput.*, vol.68, pp. 268-282, 2018.

[5] K. Miettinen, "Nonlinear Multiobjective Optimization." Norwell, MA:Kluwer, 1999.

[6] C.M. Fonseca, P. J . Fleming, "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization.", in *Conf. Genet. Evol. Comput*., 1999, pp. 416–423.

[7] K. Deb, A. Pratap, S. Agarwal *et al*., "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2,pp. 182–197, 2002.

[8] P. Wang, B. Liao,W. Zhu *et al*. "Adaptive region adjustment to improve the balance of convergence and diversity in MOEA/D." *Appl. Soft Comput.,* vol.70, pp. 797-813, 2018

[9] Y. Liu, N. Zhu *et al*. "An angle dominance criterion for evolutionary many-objective optimization." *Inform. Sci.*, to be published. DOI: 10.1016/j.ins.2018.12.078.

[10] 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, 2007.

[11] K. Li, K. Deb, Q. Zhang *et al*., "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 694–716, 2015.

[12] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284-302, 2009.

[13] S. Jiang, J. Zhang, Y. S. Ong *et al*., "A simple and fast hypervolume indicator-based multiobjective evolutionary algorithm," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2202–2213, 2015.

[14] Y. Sun, G. G. Yen, and Z. Yi, "IGD indicator-based evolutionary algorithm for many-objective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 23, no. 2, pp. 1–1, 2018.

[15] Q. Zhang, H. Li, D. Maringer, and E. Tsang. "MOEA/D with NBI-style Tchebycheff approach for portfolio management." in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Barcelona, Spain, 2010, pp. 173 - 187.

[16] H. Sato, "Inverted PBI in MOEA/D and its impact on the search performance on multi and many-objective optimization," in *Proc. Annu. Conf. Genet. Evol. Comput.*, Vancouver, BC, Canada, 2014, pp. 645–652.

[17] T. Heike , T. Wagner , and D. Brockhoff. "R2-EMOA: Focused multiobjective search using R2-indicator-based selection," in *Conf. Lear. and Intel. Opt*. Springer, Berlin, Heidelberg, 2013, pp. 70-74.

[18] M. Asafuddoula , T. Ray , R. Sarker. "A decomposition-based evolutionary algorithm for many objective optimization." *IEEE Trans. Evol. Comput.*, vol. 19, no. 3, pp. 445-460, 2015

[19] S. B. Gee ; K. C. Tan ; V. A. Shim *et al.* "Online diversity assessment in evolutionary multiobjective optimization: A geometrical perspective." *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 542-559, 2015.

[20] Wang L, Zhang Q, Zhou A, et al. "Constrained Subproblems in a Decomposition-Based Multiobjective Evolutionary Algorithm." *IEEE Trans. Evol. Comput.*, vol. 20, no. 3, pp. 475-480, 2016.

[21] H. Ge, M. Zhao, L. Sun *et al*. "A Many-Objective Evolutionary Algorithm with Two Interacting Processes: Cascade Clustering and Reference Point Incremental Learning." *IEEE Trans. Evol. Comput.*, vol. 23, no. 4, pp. 572 - 586, 2019

[22] Xinye Cai, Zhiwei Mei, *et al.* "A Constrained Decomposition Approach With Grids for Evolutionary Multiobjective Optimization." *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 564 - 577, 2018.

[23] J. Branke, K. Deb, H. Dierolf, M. Osswald, "Finding Knees in Multi-Objective Optimization," in *Paral. Prob. Solv. from Nat.*, 2004, pp. 722–731.

[24] X. Zhang, Y. Tian, and Y. Jin, "A knee point-driven evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 6, pp. 761–776, 2015.

[25] J. Zou,C. Ji, S. Yang *et al*., "A knee-point-based evolutionary algorithm using weighted subpopulation for many-objective optimization," *Swar. and Evol. Comput*. to be published. DOI: 10.1016/j.swevo.2019.02.001

[26] Q. Zhang *et al*., "Multiobjective optimization test instances for the CEC 2009 special session and competition," *School Comput. Sci. Electron. Eng.*, Univ. Essex, Colchester, U.K. and School Elect. Electron. Eng., Nanyang Technol. Univ., Singapore, Tech. Rep. CES-487, 2008.

[27] F. Gu, H.-L. Liu, and K. C. Tan, "A multiobjective evolutionary algorithm using dynamic weight design method," *Int. J. Innov. Comput. Inf. Control*, vol. 8, no. 5, pp. 3677–3688, 2012

[28] H. Zhang et al., "A self-organizing multiobjective evolutionary algorithm," *IEEE Trans. Evol. Comput*., vol. 20, no. 5, pp. 792–806, 2016.

[29] C. A. Coello and N. C. Cortes, "Solving multiobjective optimization problems using an artificial immune system," *Genet. Prog and Evol Mach*, vol. 6, no. 2, pp. 163-190, 2005.

[30] 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, 1999.

[31] Y. Tian, R. Cheng, X. Zhang et al., "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization," *IEEE Comput. Intell. M*., vol. 12, no. 4, pp. 73–87, 2017.

[32] K. Deb, M. Mohan, and S. Mishra, "Towards a quick computation of well-spread Pareto-optimal solutions," in *Proc. Intern. Conf. Evol. Mul. Crit. Opt.*, 2003, pp. 222-236.

[33] S. Yang, M. Li, X. Liu, and J. Zheng, "A Grid-Based Evolutionary Algorithm for Many-Objective Optimization." *IEEE Trans. Evol. Comput.*, vol. 17, no. 5, pp. 721-736, 2013.

[34] E. J. Hughes, "Multiple single objective Pareto sampling," in *Proc. Congr. Evol. Comput. (CEC)*, vol. 4. Canberra, Australia, 2003, pp. 2678–2684.

[35] H. Wang, L. Jiao, and X. Yao. Two_Arch2: An improved two-archive algorithm for many-objective optimization. *IEEE Trans. Evol. Comput.*, 19(4):524–541, 2015.

[36] Y. Liu, D. Gong, J. Sun, and Y. Jin. A Many-Objective Evolutionary Algorithm Using A One-by-One Selection Strategy. *IEEE Trans. Cybern.*, 47(9): 2689-2702, 2017.

[37] H. Chen, Y. Tian, W. Pedrycz, G. Wu, R. Wang and L. Wang, "Hyperplane Assisted Evolutionary Algorithm for Many-Objective Optimization Problems," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3367-3380, July 2020, doi: 10.1109/TCYB.2019.2899225.

[38] Y. Liu, H. Ishibuchi, N. Masuyama and Y. Nojima, "Adapting Reference Vectors and Scalarizing Functions by Growing Neural Gas to Handle Irregular Pareto Fronts," *IEEE Trans. Evol. Comput.*, vol. 24, no. 3, pp. 439-453, June 2020, doi: 10.1109/TEVC.2019.2926151.