A Novel Two-archive Strategy for Evolutionary Many-objective Optimization Algorithm Based on Reference Points

Rui Ding[a,b], Hongbin Dong[a,*], Jun He[c], Tao Li[a]

[a]College of Computer Science and Technology, Harbin Engineering University, Harbin, China
[b]College of Computer Science and Information Technology, Mudanjiang Normal University, Mudanjiang, China
[c]School of Science and Technology, Nottingham Trent University, Nottingham, UK

Abstract

Current evolutionary many-objective optimization algorithms face two challenges: one is to ensure population diversity for searching the entire solution space. The other is to ensure quick convergence to the optimal solution set. In this paper, we propose a novel two-archive strategy for evolutionary many-objective optimization algorithm. The uniform archive strategy, based on reference points, is used to keep population diversity in the evolutionary process, and to ensure that an evolutionary algorithm is able to search the entire solution space. The single elite archive strategy is used to ensure that individuals with the best single objective value are able to evolve into the next generation and have more opportunities to generate offspring. This strategy aims to improve the convergence rate. Then this novel two-archive strategy is applied to improving the Non-dominated Sorting Genetic Algorithm (NSGA-III). Simulation experiments are conducted on benchmark test sets and experimental results show that our proposed algorithm with the two-archive strategy has a better performance than other state-of-art algorithms.

Keywords: many-objective optimization, evolutionary algorithms, reference points, two-archive, decomposition

1. Introduction

A many-objective optimization problem (MaOP) refers to a problem consisting of four or more objectives to be optimized [1]. Because the number of solutions might increase exponentially as the number of objectives, it becomes difficult to distinguish advantages and disadvantages of solutions only using the Pareto-dominant selection pressure [2]. Traditional methods of solving multi-objective optimization problems are not effective in dealing with MaOPs [3].

Different methods proposed to solve MaOPs, such as objective reduction, decomposition, preference information, modification of Pareto dominance, reference points, and use of indicator functions. Several of the more popular methods are discussed below.

Methods based on objective reduction. Objective reduction refers to the process of transforming many-objective optimization problems into multi-objective optimization problems by analyzing the relationships between objectives or index functions [4]. Through an analysis of the relationships between objectives, the number of objectives for some MaOPs can be reduced to 2 or 3 [5] and the impact of this reduction can be remarkable [6, 7]. However, for some MaOPs, it is difficult, or impossible, to reduce the number of objectives based on an analysis of relationships [8]. The indicator-based approach uses performance indicators to guide the search process of algorithms [9]. IBEA [10] is the first indicator-based method, but it does not include an indicator of diversity. The fast hypervolume-based evolutionary algorithm (HypE) [11] is

[*]Corresponding author:Hongbin Dong

Email addresses: `mdjdingrui@hrbeu.edu.cn` (Rui Ding), `donghongbin@hrbeu.edu.cn` (Hongbin Dong)

a classic indicator-based method which uses a hypervolume (HV) value to balance both convergence and diversity effectively. The disadvantage of objective reduction methods based on index functions is that the computation time generally increases exponentially with an increasing number of objectives.

Methods based on objectives decomposition. Aggregation-based optimization algorithms use an aggregate function to decompose many objectives. Multiobjective Evolutionary Algorithm based on Decomposition (MOEA/D) [12] is a classic decomposition-based optimization algorithm which decomposes objectives by direction vectors and aggregate functions. It performs evolutionary operations among neighboring individuals. This method can effectively solve discrete optimization problems with a complex Pareto set (PS). Asafuddoula et al. [13] proposed an improved decomposition-based multiobjective optimization algorithm. As a variant of MOEA/D, the algorithm selects offspring into the next generation only when an offspring solution is not dominated by a current solution. Cheng et al. [14] proposed a reference vector-guided evolutionary algorithm named RVEA. As a decomposition-based approach, RVEA divides the objective space into some subspaces by using a set of reference vectors which can guide search process. And an angle-penalized distance is used to balance the convergence and diversity of solutions. Cai et al. [15] proposed a method of adjusting two kinds of weight vectors based on MOEA/D. On one hand, it adjusts the number of weight vectors so that the weight vectors of the boundary converge more quickly to the complete Pareto front (PF). On the other hand, it adjusts the positions of invalid weight vectors, which makes the algorithm suitable for solving MaOPs with irregular PF. Although the decomposition-based multiobjective optimization method may effectively solve a multiobjective optimization problem, the size of direction vectors used in decomposition will increase quickly as the number of objectives. Moreover, it is necessary to consider setting the weight vectors uniformly distributed [16].

Modification of Pareto dominance. As the number of objectives increases, the number of non-dominant solutions will increase dramatically. It is difficult to distinguish the pros and cons of solutions [17, 18]. The traditional dominance relation needs to be modified for ensuring the selection pressure. Many methods are also used to select non-dominant solutions.

Variants of the traditional Pareto dominance Kokolo et al. [19] proposed an α-domination strategy to distinguish the advantages and disadvantages of non-dominated solutions. But the selection of the parameter value is a difficult job. Too large value of α will lead to the deterioration of the diversity of solutions, and too small value of α is not conducive to the convergence of the algorithm. Laumanns et al.[20] proposed a classic ϵ dominant strategy and increased the selection pressure by relaxing Pareto dominance. Yuan et al.[21] proposed a new dominance relation named θ dominance, and developed θ-DEA to tackle MaOPs effectively.

Using technologies in Pareto dominance Bentley and Wakefield [22] proposed a combined ranking method to extract a ranking table for each objective value of each solution. But the method is not ideal when the objective number increases. Yuan et al. [23] proposed an algorithm named EFR which uses an ensemble fitness ranking method to determine the merits of individuals. On the basis of EFR, EFRRR [24] uses the idea of ranking restriction. A solution is only allowed to be ranked on part of the objective functions which are close to their corresponding weight vectors. This method can increase the selection pressure of the solutions efficiently. Yang et al. [25] developed a grid-based evolutionary algorithm(GrEA) in which a grid dominance is introduced to strengthen the selection pressure toward the optimal direction. Zitzler et al. [26] presented a strength Pareto evolutionary algorithm (SPEA) which uses a clustering technique to estimate the crowding degree of an individual. In 2004, SPEA2 [27] is proposed. The improved algorithm combines a strength value of each individual with a k_{th} nearest neighbor method. Li et al. [28] uses a Shift-based Density Estimation (SDE) in SPEA2 to reflect the convergence of individuals in population. The performance of SPEA2SDE significantly outperforms SPEA2. SPEA, SPEA2 and SPEA2SDE evaluate an individual's fitness dependent on the number of external non-dominated points that dominate it. They can be considered as Pareto dominance-based method to some extent.

Preference-based methods. The preference-based approaches focus on the search within a user's preferred solution space. It can be classified from perspectives of goals, weights, reference vectors, preference relation, utility functions, out-ranking, and implicit preferences. [29]. From the view of the balance of convergence and diversity, two-archive strategy can be considered as a preference-based method to some extent. Praditwong et al. [30] proposed a two-archive algorithm named TAA. In the algorithm, convergence archive (CA) and diversity archive (DA) are used to consider the diversity and convergence, respectively. Li et al. [31] proposed an improved version named ITAA which incorporates a ranking mechanism to truncate CA and a shifted density estimation technique to truncate DA. Compared with TAA, ITAA is suitable for many-objective problems. Wang et al. [32] gave a different improved algorithm named Two-Arch2. Considering convergence, diversity, and complexity simultaneously, Two-Arch2 uses two selection principles (indicator-based selection and Pareto-based selection), and uses a new *Lp*-norm-based diversity maintenance scheme for MaOPs. Two-Arch2 can solve MaOPs with satisfactory comprehensive performance.

The goal of evolutionary many-objective optimization algorithms is to quickly find a set of uniformly distributed solutions as close as possible to the PF [33]. Therefore, there are two goals in the design of efficient many-objective optimization algorithms [31]. One is to ensure the diversity of a population for searching the entire solution space. The other is to ensure an algorithm can quickly converge to the optimal solution set [34].

In order to implement the above two goals, we present a novel two-archive strategy based on a recent algorithm, NSGA-III [35]. First, a uniform archive strategy based on reference points is proposed to preserve the diversity of the population. Secondly, a single elite archive is proposed to improve the convergence. Then, an improved algorithm using the two-archive strategy (called NSGA-III-UE) is proposed. The whole research idea is shown in Fig. 1. The main contributions of the article include three points.

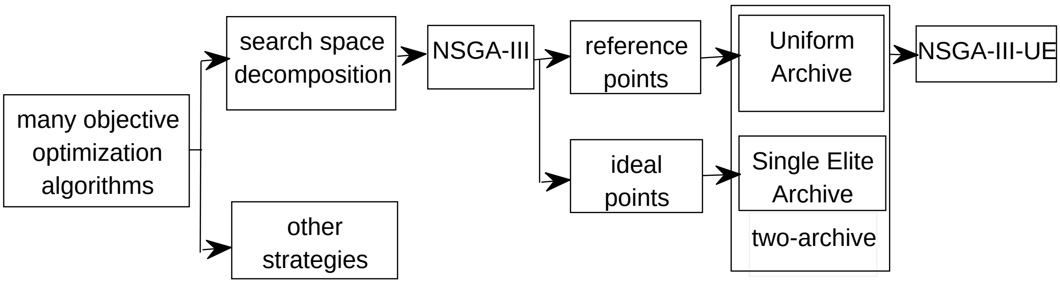


Figure 1: The overall research ideas and innovations

1. In order to maintain the diversity of an algorithm, a uniform archive strategy based on reference points is proposed which ensures the algorithm is able to search the entire solution space during iterative process.
2. In order to improve the convergence of an algorithm, a single elite archive strategy is proposed to keep the best individual in each objective to participate in evolution.
3. A two-archive strategy is presented which combines the uniform archive and single elite strategies together, and an improved algorithm based on NSGA-III is designed.

The remainder of this article is organized as follows. The second section reviews evolutionary many-objective optimization methods based on reference points, and explains our research motivation. The third section explains the innovation of our method in detail and analyzes the time complexity of the proposed algorithm. In the fourth section, experiments are used to illustrate the validity of the proposed strategies. Finally, the fifth section gives the conclusion and the future research direction.

2. Related Work

2.1. Evolutionary Many-Objective Optimization Algorithms Based on Reference Points

Reference points-based evolutionary many-objective optimization algorithms use reference points (or reference directions) to decompose the solution space into multiple subspaces. These algorithms search subspaces in parallel during evolutions. Non-dominated Sorting Genetic Algorithm (NSGA-II [36] and NSGA-III [35]) are two classic evolutionary multiobjective/many-objective optimization algorithms. NSGA-II uses a crowding distance-based diversity preserving mechanism to solve multiobjective optimization problems. However, when the number of objectives is more than three, the performance of NSGA-II degrades rapidly. Based on reference points, NSGA-III improves the method of maintaining diversity in NSGA-II. NSGA-III uses a niche individual selection strategy to replace the crowding distance-based diversity preserving mechanism. NSGA-III achieves better results in solving MaOPs than NSGA-II does. In recent years, researchers have continued to find the method of preserving diversity of populations and improve the speed of convergence.

In recent years, several algorithms are proposed which are based on NSGA-III. MOEA/DD [37] is a combination of MOEA/D [12] and NSGA-III [35]. It uses hybrid mechanism of the Pareto dominance-based fitness evaluation and the framework of MOEA/D . Ibrahim et al. [38] proposed Elite-NSGA-III, in which solutions closest to reference lines are kept as elites to participate in evolution. Khan et al. [39] used reference points to maintain diversity. Their algorithm is combined with the idea of objective decomposition and selects neighbors for evolutionary operations. Yuan et al. [21] used the idea of clustering in many-objective evolutionary algorithm. Current solutions are assigned to different clusters by uniform reference points in each generation. The optimal individual in certain cluster is determined by a fitness function which is similar to the function of penalty-based boundary intersection (PBI). The competition among solutions only occurs in the same cluster, so as to realize the balance of individual diversity and convergence. Instead of selecting individuals into the next generation, Bi et al. [40] proposed an eliminated operator that the worst individuals within the niche area of each reference point will be eliminated until the remaining individuals reach the size of the population. Eliminate operator makes outstanding individuals involve in evolution, and improves the convergence rate.

Although the existing methods have some good performance, they do not consider searching the areas near the reference points to which no individual is attached during an iteration. They also don't make full use of ideal point information.

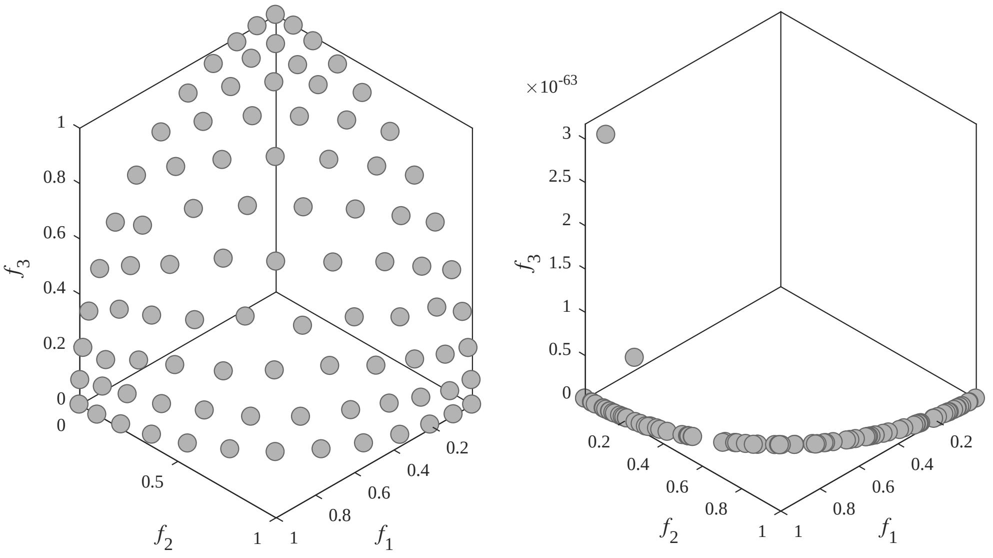
2.2. NSGA-III

Our algorithm is built upon NSGA-III and aims to improve its performance. In this subsection, NSGA-III is introduced and analyzed in detail.

NSGA-III follows the framework of NSGA-II and a reference-point-based selection strategy is added to maintain population diversity. Initially, NSGA-III defines a set of reference points within the solution space and sets an ideal point as zero vector. In each iteration, individuals in a population are divided into several different non-dominated layers $(F_1, F_2, \ldots)$. Staring from the layer F_1, NSGA-III adds solutions of each layer $F_i (i = 1, 2, \cdots)$ into population P_{t+1} successively until the size of P_{t+1} equals to or is greater than the population size N for the first time. If the last acceptable layer is the layer F_l and the number of solutions in layers $\cup_{i=1}^{l-1} F_i$ to the layer F_l already equals to N, then these N solutions are added to P_{t+1}, and solutions at and after the layer F_{l+1} are discarded. If there are more than N solutions at this time, some individuals are selected from the layer F_l according to the niche selection strategy based on reference points so that the number of individuals in P_{t+1} is N. Connecting the ideal point and reference points to form reference lines, the niche selection strategy needs to calculating the distance between individuals in set S_t (the set of individuals currently selected to enter the next generation) and each reference line. An individual will attach to a certain reference point with the smallest vertical distance from it. With the niche selection strategy based on the reference points, NSGA-III selects individuals into the next generation, and then, the diversity of the algorithm is maintained.

The performance of NSGA-III is excellent, but the algorithm still faces two challenges.

Challenge in diversity. In the process of optimizing a many-objective problem, NSGA-III may encounter a situation, that is, some reference point has no attached individual. Individuals are concentrated on certain reference points, while there has no individual in the vicinity of other reference points. Thus the algorithm cannot search for the areas near these reference points without attached individuals. NSGA-III focuses on searching for good individuals in some regions, therefore, individuals selected into next generation are from these regions. The algorithm loses its diversity to a certain degree. In extreme cases, all the individuals in a population are located in some areas, and other areas will not be searched. Let us take the DTLZ4 [41] function as an example. Fig. 2-(a) shows the evenly distribution of the optimal solutions of DTLZ4 when the population size is 100. Fig. 2-(b) shows a distribution of solutions optimized by NSGA-III. It can be seen that most areas of the solution space are not being searched. NSGA-III discards searching some areas and the distribution of the solutions is concentrated in a region like a curve. In order to overcome this challenge, a strategy needs to be considered to preserve the diversity of the population.



(a) An evenly distribution of the solutions of DTLZ4

(b) A non-evenly distribution of solutions optimized by NSGA-III on DTLZ4

Figure 2: The distribution of the optimal solutions on DTLZ4

Challenge in convergence. The ideal point in NSGA-III is composed of the minimum value of each objective currently obtained, but it does not correspond to a real point in the solution space, or an actual individual in a population. However, there exists a point (individual) which corresponds to the minimum value of each objective. Although this individual does not take a minimum value on other objectives, it still indicates a potential good search area. Using an individual with a minimum value on a single objective may speedup the convergence of the algorithm. An individual with a minimum value on one objective belongs to the layer F_1. In theory, all the individuals in this layer should be selected to enter the next generation. However, in many MaOPs, the number of non-dominated individuals increases sharply as the number of objectives increases. If the number of individuals in F_1 exceeds the population size, then a niche selection strategy is used to select individuals for next generation. This selection strategy can not guarantee that an individual with a single objective minimum enters the next generation. Thus it is necessary to retain these individuals in a special way to guide the algorithm to converge quickly.

3. A Novel Two-archive Strategy

In this paper, a novel two-archive strategy is proposed to improve the performance of evolutionary many-objective optimization algorithms based on reference points. The two archives are uniform archive and single elite archive.

3.1. Uniform Archive Strategy

The uniform archive strategy aims to ensure that an algorithm is able to search the entire solution space, and in particular, to search an area in which no individual currently exists. This strategy keeps individuals closest to each reference point into a uniform archive. These individuals will participate in evolutionary operations with the same probability as other individuals in each generation. From the perspective of assuring population diversity, this strategy ensures that an algorithm has the ability of searching the entire solution space.

The process of implementing the strategy is described as follows: First, calculate the distance values between individuals and each reference point. Then the individual which is the closest to each reference point is added into the uniform archive. When more than one individual are at an equal distance from a certain reference point, the individual which is the closest to the ideal point is retained.

Determination of reference points. Das and Dennis's [42] systematic approach is used to determine the set of reference points Z on a normalized hyper-plane. It can be formalized as Eq. 1.

$$\begin{aligned} &Z = \{\lambda^1, \lambda^2, \dots, \lambda^H\}, \\ &H = (\begin{matrix} m+p-1 \\ p \end{matrix}). \end{aligned} \tag{1}$$

Here, H is the number of reference points. λ^j is the j^{th} reference point, and satisfies $\lambda_i^j \in \{\frac{0}{p}, \frac{1}{p}, \dots, \frac{p}{p}\}$ and $\sum_{i=1}^{m} \lambda_i^j = 1$, $j = 1, 2, \dots, H$. m is the number of objectives of a MaOP, and each objective coordinate will be divided into p part. More details about the determination of reference points can be seen in [35].

Normalization. Since the values of sub-objective functions of a many-objective problem cannot be compared directly, the $min - max$ method is used to normalize the function values of each sub-objective. The normalization equation is described as follows:

$$\begin{aligned} &\overline{F_i}(X) = (\overline{f_1^i}(x), \overline{f_2^i}(x), ..., \overline{f_m^i}(x)), \\ &\overline{f_i}(x) = \frac{f_i(x) - Z_i^{min}}{Z_i^{max} - Z_i^{min}}. \end{aligned} \tag{2}$$

In the above Eq. 2, $f_i(x)$ represents the value of the i^{th} objective, where $i = 1, 2, ..., m$, and m is the number of objectives, $\overline{f_i}(x)$ is the value of $f_i(x)$ after normalization, Z_i^{min} is the minimum of the i^{th} objective, and Z_i^{max} is the maximum of the i^{th} objective. $\overline{f_m^i}(x)$ is the m^{th} objective value of the i^{th} individual after normalization. $\overline{F_i}(x)$ is the fitness value of the i^{th} individual after normalization.

Select individuals in the uniform archive. After normalization, the selection of individuals remaining to the uniform archive is formalized as Eq. 3:

$$\begin{aligned} &u_k = \arg \min_{j=1}^{N} ||\overline{F_j}(X) - Z^k||, \\ &U = \{u_1, u_2, ...u_k...u_h\}. \end{aligned} \tag{3}$$

In Eq. (3), u_k represents the elite individual corresponding to the k^{th} reference point, and $\overline{F_j}(x)$ is the fitness value of the j^{th} individual after normalization, where $j = 1, 2, ..., N$, and N is the number of attached individuals of the k^{th} reference point. Z^k is the k^{th} reference point. U is the set of individuals in the uniform archive, h is the number of reference points.

Clearly, if there are k reference points in the solution space, there are corresponding k individuals in the uniform archive. In order to maximize the diversity of a population, if an individual closest to some reference point has been placed in the uniform archive, another individual not in the uniform archive but sub-closest to the reference point will be selected to ensure there has no repetitive individual in the archive.

Algorithm of generating the uniform archive. Algorithm 1 describes the procedure of generating the uniform archive. In Algorithm 1, line 1 is used to calculate the Euclidean distance between each member of P and each reference point of Z, and save the distance values in matrix B. Then sort B on line and save the indexes in matrix $newB$ and the corresponding Euclidean distance values in matrix $Bdist$. A *while* loop is used to determine non-repeated individuals of the uniform-archive in line 4-11. The numbers of rows and columns of $newB$ need to be calculated for the *while* loop. The results of Algorithm 1 are stored in U and Ud, respectively. The members of the first column of matrix $newB$ are the indexes of non-repeated individuals which are the closest to each reference point. They are the individuals of the uniform-archive and stored in U. The distance values between reference points and the corresponding individuals in the uniform-archive are stored in Ud.

Algorithm 1 Generate-uniform-archive

```
Require: $P$: a set of individuals as population.
         $Z$: a set of reference points on the normalized hyper-plane.
Ensure: $U$: a set of individuals in the uniform archive.
        $Ud$: a set of Euclidean distance values between each reference point and the corresponding individual
        in the uniform archive.
 1: $B$ = pdist2($Z, P$) //Calculate the Euclidean distance between each member of $P$ and each reference
    point of $Z$. The distance values are saved in matrix $B$.
 2: [$Bdist, newB$] = sort($B$,2) //Sort $B$ on line and save the index in matrix $newB$ and the corresponding
    distance values in matrix $Bdist$.
 3: [$Bline, Bnum$]=size($newB$)
    //Before the while loop, calculate the numbers of rows and columns of $newB$, and store them in $Bline$
    and $Bnum$, respectively.
 4: while size(unique($newB(:, 1)$), 1) $< Bline$ do
 5:   for $i$ =1 to $Bnum - 1$ do
 6:     if ismember($newB(i+1,1)$,$newB(1:i,1)$) then
 7:         $newB(i + 1, 1 : Bline - 1) = newB(i + 1, 2 : Bline)$
 8:         $Bdist(i + 1, 1 : Bline - 1) = Bdist(i + 1, 2 : Bline)$
          //If there has a duplicate individual, replace it by the next column one.
 9:     end if
10:   end for
11: end while
12: $U = newB(:, 1)$
13: $Ud = Bdist(:,1)$
```

Algorithm of updating the uniform archive. During the iterative process, individuals in the uniform archive are updated constantly and participate in the evolution with the same probability as other individuals. Algorithm 2 shows how to update the uniform archive. In the process of updating the individuals in uniform archive, Algorithm 1 is used in the newly generated individuals because only the Euclidean distances between the newly generated individuals and each reference point need to be calculated. The values of the index and distance are sorted in matrices B and $Bdist$, respectively. Then the Euclidean distance values of Ud_t and $Bdist$ are compared. Consider the nearest individual of each reference point one by one. For the Ud_{t+1} of the i^{th} reference point, if the distance value of the newly generated individual $Bdist(i, 1)$ is less than

Algorithm 2 Update-U-archive

```
Require: $osP$ : a set of individuals of offspring population.
         $Z$: a set of reference points on normalized hyper-plane.
         $U_t$: a set of individuals in the uniform archive in the $t^{th}$ iteration.
         $Ud_t$: a set of Euclidean distance values between each reference point and the corresponding
    individual in the uniform archive in the $t^{th}$ iteration.
Ensure: $U_{t+1}$; $Ud_{t+1}$.
$[Bdist, B]$ = Generate-uniform-archive($osP, Z$)
for $i$ =1 to size($Z$,2) do
    if $Bdist(i, 1) < Ud_t(i)$ then
        $Ud_{t+1}(i) = Bdist(i, 1)$
        $U_{t+1}(i) = B(i, 1)$
    else
        $Ud_{t+1}(i) = Ud_t(i)$
        $U_{t+1}(i) = U_t(i)$
    end if
end for
```

the value of $Ud_t(i)$, replace $Ud_t(i)$ with $Bdist(i, 1)$. $U_{t+1}(i)$ is also replaced by $B(i, 1)$. The individual corresponding to the reference point in the original uniform archive will be updated.

3.2. Single Elite Archive Strategy

In NSGA-III, the ideal point is a virtual point, which is used to guide the evolution direction. Each element value of the ideal point is the smallest single objective value obtained by the current population. For example, $z_i^{\min}$ is the minimum value of the i^{th} objective function obtained by the current population. The ideal point $Z^{ideal} = (z_1^{\min}, z_2^{\min}, \ldots z_m^{\min})$ does not correspond to a real individual in the population. Recalling particle swarm optimization (PSO), the global optimal solution g_{best} and the local optimal solution p_{best} are kept to guiding the evolution. Inspired from PSO, the individual with the minimum value of a single objective is kept in a single elite archive and participate in evolution to generate new individuals. These single elite individuals in the current generation can help the algorithm converge rapidly.

Select individuals in the single elite archive. The single elite archive strategy is described as follows: let $individual_i^{best}$ be the i^{th} single elite individual corresponding to $\arg\min z_i$. Retain such individuals into the single elite archive to participate in evolution, and update them constantly. The elite individuals can help the algorithm converge rapidly. The selection of individuals that remain to the single elite archive is formalized by Eq. (4):

$$\begin{aligned} &individual_i^{best} = \arg\min z_i, \\ &F(individual_i^{best}) = (f_1(\arg\min z_i), f_2(\arg\min z_i), \ldots, f_m(\arg\min z_i)). \end{aligned} \tag{4}$$

In Eq. (4), z_i is the i^{th} element of the ideal point Z^{ideal}, $Z^{ideal} = (z_1^{\min}, z_2^{\min}, \ldots z_m^{\min})$. $individual_i^{best}$ is the i^{th} single elite individual, and $F(individual_i^{best})$ represents its fitness value.

Algorithm of generating the single elite archive. Algorithm 3 describes the procedure of generating the single elite archive. In Algorithm 3, the input population P is composed by individuals whose type are defined as a structure. The objective function value of each individual is stored in $P.objs$. The i^{th} objective minimum value of the current population is saved as a part of ideal point Z^{ideal}, $Z^{ideal} = (z_1^{\min}, z_2^{\min}, \ldots z_m^{\min})$. Line 1 is used to store the minimum value of each objective function of P in Z^{ideal}, and returns the indices of these corresponding individuals in Z_{indi}. Line 2 is used to store these corresponding individuals in the single elite archive.

```
Algorithm 3 Generate-single-elite-archive
Require: $P$ : a population composed of individuals.
Ensure: SEA: a set of individuals, each individual is a single elite.
        $Z^{ideal}$: a vector named ideal point that each element value of $Z$ is the minimum of each objective.
  $[Z^{ideal}, Z_{indi}] = \min(P.objs, [\ \ ], 1)$
  SEA= $P(Z_{indi})$
```

Discussion about single elite individuals. In fact, single elite individuals are Pareto dominant solutions of the current population. They belong to the layer F_1. According to the selection strategy of NSGA-III, they will be the first choice for the next generation population. If the number of individuals in F_1 is less than the population size N, these single elite individuals will surely be chosen to enter the next generation. But in most MaOPs, unfortunately, the number of individuals in the layer F_1 is often larger than the size of population. In this case, a niche technique is applied to selecting N individuals from F_1, that is, a single elite individual may not be selected. Thus, a single elite archive is necessarily to be used to ensure that these individuals get involved in evolutionary operations.

3.3. Proposed Algorithm: NSGA-III-UE

NSGA-III-UE is designed within the framework of NSGA-III to which the uniform archive and single elite archive are added. The uniform archive is used to keep individuals closest to each reference point for maintaining population diversity. The single elite archive is used to keep individuals with the best single objective values for fastening the algorithm convergence. Algorithm 4 describes NSGA-III-UE.

At the beginning of NSGA-III-UE, line 4 is Algorithm 1 which generates a uniform archive. Line 5 is Algorithm 3 which generates a single elite archive. They are used to determine initial individuals in the two archives. During the iteration, line 7 and line 9 allow these individuals in the two archives to participate in evolution with other individuals. Simulated two-point crossover (SBX) [43, 44] operator and polynomial variation [45]are used as same as NSGA-III's operator. Line 10 is the process of updating the uniform archive using Algorithm 2. Only the distance values from the newly generated individuals to each reference point need to be calculated. After that, parents and descendants are merged in line 11, and sorted based on the non-dominated relationship in line 12. Then, the algorithm uses the framework of NSGA-III to select the individuals of next generation in lines 13-24. Line 25 is the updating process of the single elite archive. Single elite individuals are selected from individuals of the $(t+1)^{th}$ generation.

3.4. Complexity Analysis of NSGA-III-UE

NSGA-III-UE is an improved version of the classical NSGA-III. Like NSGA-III, when NSGA-III-UE selects individuals into the next generation, these individuals are attached to corresponding reference points. The complexity of this process is $O(MNH)$. For the process of selecting individuals from the layer F_l by the niche strategy, its computational complexity is $O(H)$. The computational complexity of determining the ideal point is $O(MN)$.

Different from NSGA-III, NSGA-III-UE adds two extra archives. However, none of them needs to be handled in a special way. For the single elite archive, single elite individuals can be directly reserved when NSAG-III determines the $Z_{\min}$, therefore, it does not require additional calculation time. For the uniform archive strategy, individuals closest to reference points are retained no matter whether there has an individual attached to the reference point or not. Calculating the vertical distances from individuals to reference points requires $O(NH)$ steps, and determining the closest individual for each reference point requires $O(N \log H)$ steps. Therefore, in the worst case, NSGA-III-UE increases the computational complexity to $\max\{O(NH), O(N \log H)\}$. Due to $O(N \log H) < O(NH) < O(N2M)$, the overall computational complexity of NSGA-III-UE is still as same as that of NSGA-III, that is, $\max\{O(N2 \log M - 2N), O(N2M)\}$.

In fact, for most of the MaOPs, the number of individuals in the layer F_1 is larger than the size of population N. NSGA-III needs to take some time to select N individuals from F_1 by using the niche technique, while NSGA-III-UE needs less time. Therefore, for some MaOPs, the operating time of NSGA-III-UE is less than that of NSGA-III.

Algorithm 4 NSGA-III-UE

```
Require: $H$: the number of reference points.
         $N$: the size of population.
Ensure: $P_{t+1}$.
t = 0
$P_0$= population initialization
$Z$=Generate-Reference-Points($H$)
$[U_0, Ud_0]$= Generate-uniform-archive($P_0, Z$)
$[Z^{ideal}$, $SEA_0]$=Generate-single-elite-archive($P_0$)
while termination criteria is not met do
    $St = SEA_t$
    $i = 1$
    $Q_t$=Operator($P_t, U_t$)
    $[U_{t+1}, Ud_{t+1}]$=Update-U-archive($Q_t, Z, U_t, Ud_t$)
    $R_t = P_t \cup Q_t$
    $(F_1, F_2, \ldots)$=Non-dominated-sort($R_t$)
    while $|S_t| < N$ do
        $S_t = S_t \cup F_i$
        $i = i + 1$
    end while
    $F_l = F_i$
    if $|S_t| = N$ then
        $P_{t+1} = S_t$
    else
        $P_{t+1} = \cup_{j=1}^{l-1} F_j$
        $P_n$=Niching($N - |S_t|, Z, Z_{\min}, F_l, P_{t+1}$) //Choose $N - |S_t|$ individuals from $F_l$ according to the Niching selection.
        $P_{t+1} = P_{t+1} \cup P_n$
    end if
    $[Z^{ideal}$, $SEA_{t+1}]$= Generate-single-elite-archive($P_{t+1}$) //Update the single elite archive.
    $t = t + 1$
end while
```

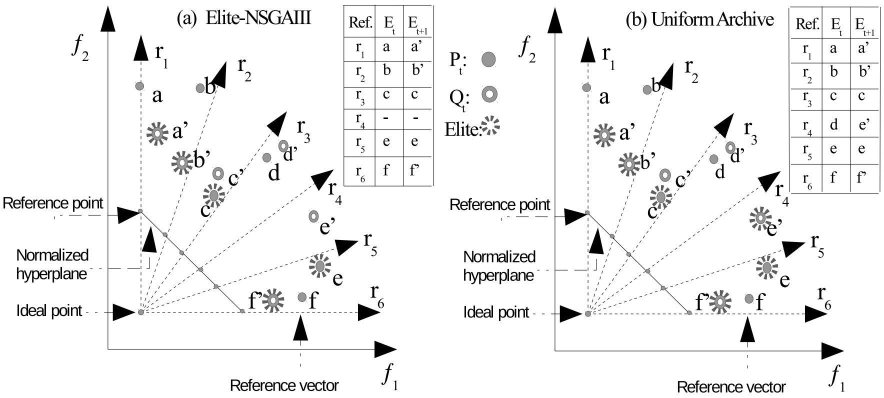
3.5. Differences from Other Algorithms

NSGA-III-UE shares a few common features with existing methods in references, such as Two-Arch2 [32] which uses two-archive strategy to balance diversity and convergence and Elite-NSGA-III [38] which uses a reference point-based strategy to maintain population diversity. However, NSGA-III-UE significantly differs from the existing methods in two aspects.

Two archives. Some algorithms [30, 32, 46, 47] adopt two-archive strategy which divides the non-dominated solution set into two archives. Although the term of two-archive in these algorithms is the same as ours, the context of two-archive is completely different from ours which means a uniform archive and a single elite archive. In [30, 32, 46, 47], one is a convergence archive (CA) which stores non-dominated solutions, while the other is a diversity archive (DA) that stores individuals not dominating any solution. During the evolutions of these algorithms, individuals in two archives act as parents to generate offspring. The offspring that dominate any solution in either CA or DA are added to CA, while the offspring that dominate no solution are added to DA.

Uniform strategy: Elist-NSGA-III [38] also adopts a uniform archive strategy which is similar to ours. Both algorithms use reference points to maintain population diversity. However, the implementations are different. In Elite-NSGA-III, the individuals attached to a reference point have a chance to be taken as an elite individual and participate in evolution. If there is no individual attached to a reference point, the area near that reference point will not be searched. In our algorithm, the uniform archive strategy retains the individuals closest to each reference point, and does not abandon any search area even no individual attached to the reference point.

In addition, Elite NSGA-III selects an individual which is the closest to the origin as elite among all individuals attached to a reference point, whereas our uniform archive selects the individual closest to the reference point as elite. Our uniform archive strategy ensures the population diversity and has the ability of searching the entire solution space. Fig.3 illustrates the difference between the elite pool in Elite-NSGA-III and our proposed uniform archive method.



(a) Elite-NSGAIII

Ref.	E_t	E_{t+1}
r_1	a	a’
r_2	b	b’
r_3	c	c
r_4	-	-
r_5	e	e
r_6	f	f’

(b) Uniform Archive

Ref.	E_t	E_{t+1}
r_1	a	a’
r_2	b	b’
r_3	c	c
r_4	d	e’
r_5	e	e
r_6	f	f’

Figure 3: Difference between Elite-NSGA-III and NSGA-III-UE

In Elite-NSGA-III, no individual covers the direction of vector r_4, and the area around r_4 is hard to be searched. In NSGA-III-UE, according to the uniform archive strategy, the individual d in the t^{th} generation and e' in the $(t+1)^{th}$ generation closest to r_4 are used as reserved individuals of the reference point r_4. So that each reference point has an adjacent individual to be retained, and the diversity of the population is maintained.

4. Experiments

4.1. Experimental Design

Our proposed algorithm, NSGA-III-UE, is compared with several classical and state-of-the-art algorithms. Experimental results are analyzed from seven perspectives for investigating the performance of NSGA-III-UE.

The first experiment is to evaluate the performance of these algorithms on WFG functions in terms of IGD, HV and operating time. The second and third experiments are used to show the performance from the perspective of test functions and objective numbers. Furthermore, the fourth experiment is designed to illustrate the efficiency of these algorithms on DTLZ functions. Then, boxplots are used to illustrate the stability of these algorithms. In the sixth experiment, variants of NSGA-III-UE are used to analyze the effect of the uniform archive and the single elite archive, respectively. Finally, the applicability of NSGA-III-UE is analyzed in the seventh experiment.

These experiments were run on Windows (Intel (R) Xeon (R) CPU E5-26030@1.80GHz, 8.00GBRAM) 64-bit operating system. Matlab2016 is used for simulation verification.

4.2. Test Functions

Thanks to the scalable number of objectives, Walking-Fish-Group(WFG) [48] and Deb-Thiele-Laumanns-Zitzler (DTLZ) [41] functions have become two standard test suites for testing evolutionary many-objective optimization algorithms. For DTLZ functions, only DTLZ1, DTLZ2, DTLZ3, DTLZ4, and DTLZ7 are considered because the Pareto fronts of DTLZ5 and DTLZ6 with more than 3 objectives are unknown; CDTLZ2 is a convex problem and IDTLZ2 [49] is an inverse of DTLZ2. Table 1 describes characteristics of these test functions.

Table 1: The characteristics of the test functions

Function	characteristics
DTLZ1	Linear, Multi-modal
DTLZ2	Concave
DTLZ3	Concave, Multi-modal
DTLZ4	Concave, Biased
DTLZ7	Mixed, Multi-modal, Scaled, Disconnected
IDTLZ2	Convex
CDTLZ2	Convex
WFG1	Convex,Mixed,Biased,Scaled
WFG2	Convex,Multi-modal,Non-separable,Disconnected,Scaled
WFG3	Degenerate,linear,Non-separable,Scaled
WFG4	Concave,Multi-modal,Scaled
WFG5	Concave,Deceptive,Scaled
WFG6	Concave,Non-separable,Scaled
WFG7	Concave, Biased,Scaled
WFG8	Concave, Biased,Non-separable,Scaled
WFG9	Concave, ,Multi-modal,Biased,Non-separable, Deceptive,Scaled

4.3. Algorithms under Comparison

The proposed algorithm NSGA-III-UE is compared with eleven state-of-the-art algorithms, listed as below:

1. NSGA-III [35] is a non-dominated sorting genetic algorithm using reference points. It solves many-objective optimization problems based on classical NSGA-II [36].

2. Two-Arch2 [32] uses two archives to focus on convergence and diversity separately, and assigns indicator-based and Pareto-based selection principles to the two archives. It is a low complexity algorithm.
3. I-DBEA [23] is a variant based on MOEA/D which uses reference direction to guide algorithm search and uses priority distance comparison mechanism. It can balance the diversity and convergence speed simultaneously.
4. EFRRR [24] is an aggregation algorithm for MaOPs. EFR uses an integrated ranking method to determine the order of individuals. Using a restriction order based on EFR, solutions of EFRRR are only allowed on sort in the corresponding weight vector which close to the part of function.
5. θ-DEA [21] is an algorithm based on a new dominance relationship that combines the fitness evaluation mechanism of MOEA/D with the diversity of NSGA-III. The ranking pattern can give consideration to both convergence and uniform distribution.
6. Elite-NSGA-III [38] is an improved NSGA-III which is similar to the algorithm that proposed in this paper. The details of the difference between Elite-NSGA-III and the proposed algorithm NSGA-III-UE can be found in Section 3.5.
7. MOEA/DD [37] is a combination of MOEA/D [12] and NSGA-III [35]. It uses hybrid mechanism of the Pareto dominance-based fitness evaluation and the MOEA/D framework. MOEA/DD can be seen as an improved version of MOEA/D which is a classical decomposition-based multiobjective optimization algorithm.
8. dMOPSO [50] combines the idea of Particle Swarm Optimization (PSO) with MOEA/D. dMOPSO uses global optimal particle to guide the search direction and re-initialized memory to ensure the diversity of the algorithm.
9. HypE [11] is a classic metric-based MOEAs. For a given MaOPs, maximizing hypervolume metric is equivalent to finding the Pareto front.
10. SPEA2SDE [28] integrates a Shift-based Density Estimation(SDE) into SPEA2 [27] in the fitness assignment and archive truncation procedures. SPEA2SDE has very competitive on the problem of DTLZs [41].
11. RVEA [14] is a reference vector-guided evolutionary algorithm that can be largely categorized into the decomposition-based approach. In RVEA, reference vectors are used to guide search process, and an angle-penalized distance is used to balance the convergence and diversity of solutions. RVEA has high efficiency in dealing with MaOPs where the functions are not well normalized.

The source code of these algorithms, except for Elite-NSGA-III, comes from Plat EMO [51].

4.4. Parameter Settings

Although parameters involved in each algorithm are not exactly the same and the complexities of the tested functions are different, the identical setting of common parameters is necessary for comparing the advantages and disadvantages of these algorithms.

The number of objectives in test functions is set to $m = \{5, 8, 10, 12, 15\}$. The corresponding number of decision variables is set to $n = \{14, 17, 19, 21, 24\}$, where $n = m + k - 1$ and k is set to 10 except IDTLZ1 in which $n = \{9, 12, 14, 16, 19\}$ and DTLZ7 in which $n = \{24, 27, 29, 31, 34\}$, respectively. The identical setting of population size is 100, and the maximum times of iteration is 10000. Simulated two-point crossover (SBX) [43, 44] operator and polynomial variation [45] are used in NSGA-III, Elite-NSGA-III and θ-DEA which are based on NSGA-III, with the distribution index setting to 20. θ is set to 5 in θ-DEA. MOEA/DD and I-DBEA use the aggregation function of type PBI, and the neighborhood size T is set to 10. The parameter k in EFRRR is set to 2, and the threshold T_a and weight w in dMOPSO is set to 2 and 0.4, respectively. α is set to 2 and fr is set to 0.1 in RVEA. Each algorithm runs independently for 30 times on each test function.

4.5. Evaluation Criterion

The performance evaluation of evolutionary many-objective optimization algorithms generally includes two aspects. One is convergence, which can be measured by calculating the distances between the solutions found by an algorithm and the true Pareto solutions. The other is diversity, that is, whether the solutions found by an algorithm are uniformly distributed along the PF. In this article, IGD[7, 52], HV[53, 54], Spread[55] and Operating time are used to indicate the performance of the algorithms.

Inverted Generational Distance(IGD) is a commonly used comprehensive indicator that can evaluate convergence and diversity simultaneously. The smaller the IGD value, the better the algorithm. The formula can be described as Eq.5:

$$IGD(A, P^*) = \frac{\sum\limits_{x \in P^*} \min\limits_{y \in A} d(x, y)}{|P^*|}. \tag{5}$$

where A is an approximate solution set of the PF obtained by an algorithm, P^* is a set of uniformly distributed sampling points of the real PF, $d(x, y)$ is the Euclidean distance between the individual x in P^* and the individual y in A.

Generalized Spread(I_{GS}) is a diversity indicator. The smaller the spread, the better the distribution diversity of the algorithm. The related formulas are described as Eq. 6:

$$\begin{aligned} I_{GS} &= \frac{\sum\limits_{i=1}^{m} d(e_i, A) + \sum\limits_{x \in A} |d(x, A) - \overline{d}|}{\sum\limits_{i=1}^{m} d(e_i, A) + |P^*|\overline{d}}, \\ d(x, A) &= \min_{x, y \in A, y \neq x} ||F(x) - F(y)||^2, \\ \overline{d} &= \frac{1}{|A|} \sum_{x \in A} d(x, A). \end{aligned} \tag{6}$$

where $d(x, A)$ is the Euclidean distance, the value is the Euclidean distance between x and the point closest to x in A. $\overline{d}$ is the average value of $d(x, A)$, and $e_1, e_2, ...e_m$ are m extreme solutions in P^*.

Hypervolume(HV) is a comprehensive performance metric which is usually considered in MaOPs whose PF have not yet been known. HV calculates the volume of objective space between the obtained solution set and a reference point. The larger the HV value, the better the algorithm. In this paper, we use 10^6 sampling points to ensure the accuracy. The formula of HV [56] can be described as Eq. 7:

$$\begin{aligned} &HV(f^{ref}, A) = \Lambda(\cup[f_1(y), f_1^{ref}] \times \cdots \times [f_m(y), f_m^{ref}]), \\ &f_m^{ref} = max\langle A_m \cup f_m^{ref} \rangle. \end{aligned} \tag{7}$$

where A is an approximate solution set obtained by an algorithm, y is an individual in A, A_m refers to the value of the objective m in current population. f^{ref} is a chosen reference point, and f_m^{ref} refers to the maximum value of the objective m in previous generations. $\Lambda(.)$ is the Lebesuge measure[57].

Operating Time is used to evaluate the computational complexity of an algorithm. The less the operating time, the better the algorithm.

4.6. Experimental Results and Analysis

The proposed algorithm, NSGA-III-UE, is compared with several classical and state-of-the-art algorithms in terms of IGD, HV, Spread, and Operating Time. The performance of each algorithm on WFGs and DTLZs with different objective numbers are illustrated in the following tables. The following tables show the average value of each algorithm running 30 times. In Tab. 2 and Tab. 6, the superscript of each datum is the rank for each algorithm, and the last lines show the number of times that each algorithm obtains the best value. The table-head numbers correspond to the algorithms: NSGA-III-UE (1), NSGA-III (2), Two-Arch2 (3), I-DBEA (4), EFRRR (5), θ-DEA (6), Elite-NSGA-III (7), dMOPSO (8), MOEA/DD (9), HypE (10), RVEA (11), and SPEA2SDE (12), respectively. The optimal value of each line is highlighted by gray shading.

Table 2: Average IGD values of each algorithm running 30 times on WFGs with different objective numbers

Problem	M	1	2	3	4	5	6	7	8	9	10	11	12
WFG1	5	1.89e+0^{9}	1.78e+0^{7}	1.83e+0^{8}	1.74e+0^{6}	1.39e+0^{3}	1.39e+0^{2}	1.65e+0^{4}	2.16e+0^{12}	2.01e+0^{10}	2.13e+0^{11}	1.66e+0^{5}	1.37e+0^{1}
	8	2.75e+0^{5}	2.71e+0^{4}	2.80e+0^{7}	2.75e+0^{6}	3.16e+0^{10}	2.97e+0^{9}	2.52e+0^{2}	3.26e+0^{12}	2.56e+0^{3}	2.93e+0^{8}	3.20e+0^{11}	2.29e+0^{1}
	10	3.32e+0^{7}	3.16e+0^{4}	3.22e+0^{5}	5.72e+0^{12}	3.52e+0^{10}	3.42e+0^{9}	3.24e+0^{6}	3.91e+0^{11}	3.08e+0^{3}	3.35e+0^{8}	2.79e+0^{1}	2.85e+0^{2}
	12	4.04e+0^{10}	3.77e+0^{5}	3.77e+0^{6}	4.09e+0^{11}	3.83e+0^{8}	3.81e+0^{7}	3.70e+0^{4}	4.41e+0^{12}	3.69e+0^{3}	3.87e+0^{9}	3.23e+0^{1}	3.31e+0^{2}
	15	4.87e+0^{6}	4.81e+0^{3}	4.89e+0^{7}	1.42e+1^{12}	4.87e+0^{5}	4.92e+0^{8}	4.77e+0^{2}	5.61e+0^{10}	5.05e+0^{9}	4.84e+0^{4}	5.77e+0^{11}	4.30e+0^{1}
WFG2	5	7.69e-1^{1}	8.24e-1^{2}	8.57e-1^{5}	8.59e-1^{6}	8.25e-1^{3}	8.29e-1^{4}	8.66e-1^{7}	4.36e+0^{12}	4.17e+0^{11}	8.90e-1^{8}	1.40e+0^{9}	2.07e+0^{10}
	8	2.36e+0^{1}	2.84e+0^{4}	2.57e+0^{3}	6.82e+0^{10}	2.56e+0^{2}	2.94e+0^{5}	3.03e+0^{6}	7.98e+0^{11}	8.89e+0^{12}	4.12e+0^{8}	3.25e+0^{7}	5.07e+0^{9}
	10	5.77e+0^{3}	5.73e+0^{2}	5.06e+0^{1}	9.42e+0^{10}	6.63e+0^{6}	6.51e+0^{5}	7.06e+0^{7}	1.62e+1^{11}	1.63e+1^{12}	7.81e+0^{8}	5.87e+0^{4}	8.71e+0^{9}
	12	7.60e+0^{6}	7.64e+0^{7}	7.56e+0^{5}	9.85e+0^{9}	8.17e+0^{8}	7.35e+0^{4}	6.59e+0^{3}	1.94e+1^{11}	2.04e+1^{12}	6.39e+0^{2}	5.22e+0^{1}	1.12e+1^{10}
	15	8.93e+0^{5}	8.98e+0^{6}	8.06e+0^{4}	1.88e+1^{10}	7.68e+0^{2}	7.03e+0^{1}	7.85e+0^{3}	2.72e+1^{11}	2.77e+1^{12}	9.57e+0^{7}	1.00e+1^{8}	1.81e+1^{9}
WFG3	5	8.14e-1^{8}	7.67e-1^{7}	5.08e-1^{2}	8.21e-1^{9}	7.19e-1^{5}	6.63e-1^{4}	7.64e-1^{6}	9.07e-1^{11}	9.27e-1^{12}	7.41e-2^{1}	8.66e-1^{10}	5.96e-1^{3}
	8	1.18e+0^{3}	1.24e+0^{5}	1.16e+0^{2}	8.84e+0^{12}	1.56e+0^{7}	1.68e+0^{8}	1.24e+0^{4}	3.00e+0^{10}	3.01e+0^{11}	1.12e-1^{1}	2.99e+0^{9}	1.42e+0^{6}
	10	1.68e+0^{3}	1.99e+0^{8}	1.54e+0^{2}	1.12e+1^{12}	1.98e+0^{7}	1.90e+0^{5}	1.77e+0^{4}	5.45e+0^{11}	4.07e+0^{10}	1.61e-1^{1}	3.71e+0^{9}	1.96e+0^{6}
	12	2.01e+0^{4}	2.17e+0^{6}	1.91e+0^{2}	1.34e+1^{12}	2.44e+0^{7}	1.96e+0^{3}	2.09e+0^{5}	7.77e+0^{11}	7.66e+0^{10}	2.05e-1^{1}	4.63e+0^{9}	2.75e+0^{8}
	15	5.35e+0^{5}	5.50e+0^{6}	2.58e+0^{2}	1.70e+1^{12}	5.98e+0^{8}	5.33e+0^{4}	5.67e+0^{7}	1.21e+1^{11}	1.12e+1^{10}	2.50e-1^{1}	7.90e+0^{9}	4.52e+0^{3}
WFG4	5	1.35e+0^{2}	1.36e+0^{3}	1.35e+0^{1}	1.40e+0^{7}	1.49e+0^{10}	1.36e+0^{4}	1.37e+0^{5}	2.36e+0^{12}	1.47e+0^{9}	1.51e+0^{11}	1.39e+0^{6}	1.40e+0^{8}
	8	3.82e+0^{1}	3.92e+0^{5}	3.85e+0^{2}	1.41e+1^{12}	4.38e+0^{7}	4.27e+0^{6}	3.91e+0^{4}	8.50e+0^{11}	4.68e+0^{9}	7.08e+0^{10}	4.43e+0^{8}	3.85e+0^{3}
	10	5.63e+0^{4}	5.75e+0^{5}	5.29e+0^{2}	1.91e+1^{12}	5.62e+0^{3}	5.82e+0^{7}	5.80e+0^{6}	1.13e+1^{11}	6.26e+0^{9}	1.05e+1^{10}	6.03e+0^{8}	5.22e+0^{1}
	12	6.94e+0^{1}	7.06e+0^{6}	7.03e+0^{5}	2.31e+1^{12}	6.97e+0^{2}	7.17e+0^{7}	7.00e+0^{4}	1.37e+1^{11}	7.47e+0^{9}	1.26e+1^{10}	7.38e+0^{8}	6.99e+0^{3}
	15	1.18e+1^{2}	1.21e+1^{4}	1.18e+1^{3}	3.12e+1^{12}	1.55e+1^{7}	1.30e+1^{6}	1.21e+1^{5}	1.70e+1^{9}	1.61e+1^{8}	2.17e+1^{11}	1.93e+1^{10}	9.15e+0^{1}
WFG5	5	1.33e+0^{3}	1.33e+0^{2}	1.20e+0^{1}	1.37e+0^{6}	1.50e+0^{10}	1.34e+0^{5}	1.33e+0^{4}	1.65e+0^{12}	1.45e+0^{9}	1.51e+0^{11}	1.37e+0^{7}	1.40e+0^{8}
	8	3.87e+0^{1}	3.95e+0^{4}	3.94e+0^{3}	1.19e+1^{12}	4.65e+0^{9}	4.23e+0^{8}	3.99e+0^{5}	6.34e+0^{11}	5.31e+0^{10}	4.09e+0^{6}	4.16e+0^{7}	3.93e+0^{2}
	10	5.67e+0^{2}	5.75e+0^{5}	5.70e+0^{3}	1.93e+1^{12}	5.79e+0^{6}	5.80e+0^{7}	5.74e+0^{4}	9.33e+0^{11}	7.77e+0^{10}	7.64e+0^{9}	6.02e+0^{8}	5.17e+0^{1}
	12	7.11e+0^{2}	7.16e+0^{4}	7.28e+0^{7}	2.39e+1^{12}	6.96e+0^{1}	7.20e+0^{5}	7.16e+0^{3}	1.06e+1^{11}	9.29e+0^{9}	1.03e+1^{10}	7.54e+0^{8}	7.21e+0^{6}
	15	1.22e+1^{2}	1.23e+1^{4}	1.22e+1^{3}	3.10e+1^{12}	1.48e+1^{8}	1.30e+1^{6}	1.23e+1^{5}	1.50e+1^{10}	1.58e+1^{11}	1.48e+1^{9}	1.47e+1^{7}	9.72e+0^{1}
WFG6	5	1.36e+0^{1}	1.36e+0^{3}	1.36e+0^{2}	1.39e+0^{7}	1.51e+0^{10}	1.36e+0^{4}	1.36e+0^{5}	2.87e+0^{12}	1.45e+0^{9}	1.53e+0^{11}	1.38e+0^{6}	1.45e+0^{8}
	8	4.01e+0^{3}	4.14e+0^{6}	4.13e+0^{5}	1.27e+1^{12}	4.49e+0^{9}	4.28e+0^{8}	4.12e+0^{4}	8.98e+0^{11}	4.61e+0^{10}	3.84e+0^{1}	4.21e+0^{7}	3.86e+0^{2}
	10	5.94e+0^{4}	6.23e+0^{6}	6.03e+0^{5}	1.91e+1^{12}	5.80e+0^{2}	5.93e+0^{3}	6.34e+0^{7}	1.13e+1^{11}	7.02e+0^{9}	8.10e+0^{10}	6.54e+0^{8}	5.16e+0^{1}
	12	7.27e+0^{3}	7.43e+0^{7}	7.36e+0^{4}	2.33e+1^{12}	7.13e+0^{2}	7.38e+0^{5}	7.40e+0^{6}	1.36e+1^{11}	7.70e+0^{9}	9.34e+0^{10}	7.69e+0^{8}	6.90e+0^{1}
	15	1.27e+1^{3}	1.28e+1^{4}	1.06e+1^{2}	3.06e+1^{12}	1.62e+1^{10}	1.31e+1^{6}	1.30e+1^{5}	1.73e+1^{11}	1.52e+1^{8}	1.45e+1^{7}	1.53e+1^{9}	9.15e+0^{1}
WFG7	5	1.37e+0^{3}	1.37e+0^{2}	1.20e+0^{1}	1.40e+0^{7}	1.53e+0^{11}	1.38e+0^{5}	1.37e+0^{4}	2.08e+0^{12}	1.48e+0^{9}	1.50e+0^{10}	1.39e+0^{6}	1.43e+0^{8}
	8	3.90e+0^{1}	4.00e+0^{3}	4.06e+0^{5}	1.17e+1^{12}	4.45e+0^{8}	4.30e+0^{7}	4.03e+0^{4}	7.04e+0^{11}	4.57e+0^{9}	4.80e+0^{10}	4.26e+0^{6}	3.95e+0^{2}
	10	5.60e+0^{2}	5.73e+0^{6}	5.81e+0^{7}	1.85e+1^{12}	5.67e+0^{3}	5.81e+0^{8}	5.73e+0^{5}	9.10e+0^{11}	5.93e+0^{9}	7.40e+0^{10}	5.73e+0^{4}	5.10e+0^{1}
	12	7.08e+0^{2}	7.14e+0^{3}	7.55e+0^{9}	2.27e+1^{12}	7.40e+0^{8}	7.24e+0^{6}	7.18e+0^{4}	1.14e+1^{11}	7.25e+0^{7}	1.08e+1^{10}	7.24e+0^{5}	6.75e+0^{1}
	15	1.19e+1^{4}	1.20e+1^{5}	1.11e+1^{2}	3.09e+1^{12}	1.68e+1^{10}	1.29e+1^{6}	1.19e+1^{3}	1.59e+1^{8}	1.54e+1^{7}	1.80e+1^{11}	1.65e+1^{9}	8.85e+0^{1}
WFG8	5	1.37e+0^{4}	1.35e+0^{2}	1.38e+0^{5}	1.38e+0^{6}	1.47e+0^{11}	1.35e+0^{1}	1.36e+0^{3}	2.01e+0^{12}	1.46e+0^{10}	1.44e+0^{9}	1.40e+0^{7}	1.41e+0^{8}
	8	3.89e+0^{1}	3.95e+0^{2}	3.99e+0^{4}	1.38e+1^{12}	4.38e+0^{9}	4.06e+0^{6}	4.00e+0^{5}	8.49e+0^{11}	4.26e+0^{7}	5.02e+0^{10}	4.35e+0^{8}	3.95e+0^{3}
	10	5.83e+0^{2}	6.20e+0^{9}	6.06e+0^{6}	1.86e+1^{12}	6.00e+0^{4}	5.70e+0^{1}	6.13e+0^{7}	1.09e+1^{11}	6.05e+0^{5}	7.39e+0^{10}	6.18e+0^{8}	5.91e+0^{3}
	12	7.49e+0^{2}	7.49e+0^{3}	8.21e+0^{8}	2.33e+1^{12}	8.21e+0^{9}	7.50e+0^{4}	7.56e+0^{7}	1.31e+1^{11}	7.53e+0^{6}	1.11e+1^{10}	7.52e+0^{5}	6.80e+0^{1}
	15	1.24e+1^{2}	1.27e+1^{4}	1.26e+1^{3}	3.06e+1^{12}	1.57e+1^{8}	1.29e+1^{6}	1.28e+1^{5}	1.65e+1^{9}	1.67e+1^{10}	1.69e+1^{11}	1.56e+1^{7}	9.06e+0^{1}
WFG9	5	1.28e+0^{3}	1.27e+0^{1}	1.30e+0^{5}	1.32e+0^{6}	1.42e+0^{9}	1.28e+0^{2}	1.28e+0^{4}	2.73e+0^{12}	1.43e+0^{10}	1.44e+0^{11}	1.34e+0^{8}	1.33e+0^{7}
	8	3.62e+0^{1}	3.66e+0^{2}	3.69e+0^{4}	1.18e+1^{12}	4.11e+0^{8}	3.97e+0^{6}	3.67e+0^{3}	7.85e+0^{11}	5.00e+0^{9}	5.86e+0^{10}	4.09e+0^{7}	3.77e+0^{5}
	10	5.42e+0^{2}	5.50e+0^{5}	5.61e+0^{7}	1.93e+1^{12}	5.49e+0^{4}	5.49e+0^{3}	5.50e+0^{6}	1.01e+1^{11}	6.79e+0^{10}	6.67e+0^{9}	5.92e+0^{8}	5.15e+0^{1}
	12	7.09e+0^{4}	7.11e+0^{5}	7.76e+0^{8}	2.39e+1^{12}	7.22e+0^{6}	6.94e+0^{2}	7.07e+0^{3}	1.13e+1^{10}	8.17e+0^{9}	1.18e+1^{11}	7.40e+0^{7}	6.90e+0^{1}
	15	1.17e+1^{2}	1.17e+1^{4}	1.17e+1^{5}	3.10e+1^{12}	1.47e+1^{8}	1.23e+1^{6}	1.17e+1^{3}	1.50e+1^{9}	1.47e+1^{7}	1.66e+1^{11}	1.61e+1^{10}	9.23e+0^{1}
count		9	1	4	0	1	3	0	0	0	6	3	18

4.6.1. The Overall Performance on WFG Functions

The first experiment demonstrates the overall performance of twelve algorithms on WFG functions. Table 2 shows the result of IGD on 45 tests of 9 functions. In table 2, NSGA-III-UE obtains the optimal value 9 times, accounting for 20.0%. The results demonstrate NSGA-III-UE is better than most of the other algorithms, except for SPEA2SDE. HypE has the best performance on WFG3 and records 6 times for the best IGD values. Two-Arch2 produces the best results 4 times, θ-DEA and RVEA 3 times. NSGA-III and EFRRR win just once. I-DBEA, Elite-NSGA-III, dMOPSO and MOEA/DD do not return any optimal value on WFGs. As shown in Tab. 2, SPEA2SDE has 18 occurrences of the best values; therefore, more analysis should be done to explain the performance of NSGA-III-UE. Tab. 3 shows the total ranks and average ranks of all test functions of each algorithm. Form these values, we can observe the overall performance of each algorithm. The total rank of NSGA-III-UE is 145, while SPEA2SDE is 171. The overall sorting of SPEA2SDE is lower than the proposed algorithm which has an overall sorting of 1. In summary, although NSGA-III-UE does not achieve the most occurrences of optimal value on WFGs, the overall performance on IGD metric is still better than the other algorithms.

As an overall indicator, HV is used to further illustrate the performance of NSGA-III-UE. Fig. 4 shows the HV ranks of each algorithm on WFGs. The reference vector of HV is generated by taking the maximum objective values from the union of approximation sets. We can see that most of the HV ranks of NSGA-III-UE are better than, or similar to, the other algorithms. That means the performance of NSGA-III-UE on HV is comparative to, and in some cases, better than, other algorithms. Tab. 4 gives the specific values of

Table 3: IGD analysis of various algorithms on WFGs

	1	2	3	4	5	6	7	8	9	10	11	12
times of the best	9	1	4	0	1	3	0	0	0	6	3	18
the rate of optimal values	20.00%	2.22%	8.89%	0.00%	2.22%	6.67%	0.00%	0.00%	0.00%	13.33%	6.67%	40.00%
total of ranks	145	200	188	476	303	234	208	402	492	367	324	171
average rank	3.2	4.4	4.2	10.6	6.7	5.2	4.6	8.9	10.9	8.2	7.2	3.8
overall sorting	1	4	3	11	7	6	5	10	12	9	8	2

Table 4: HV analysis of various algorithms on WFGs

	1	2	3	4	5	6	7	8	9	10	11	12
times of the best	4	0	0	1	7	14	2	0	0	9	0	8
the ratio of optimal values	8.89%	0.00%	0.00%	2.22%	15.56%	31.11%	4.44%	0.00%	0.00%	20.00%	0.00%	17.78%
total of ranks	252	208	375	397	219	117	148	486	328	256	375	162
average rank	5.6	4.62	8.33	8.82	4.87	2.60	3.29	10.80	7.29	5.69	8.33	3.60
overall sorting	6	4	9	11	5	1	2	12	8	7	9	3

HV rankings. The HV values of HypE must be good. Compared to the rest of the algorithms, NSGA-III-UE is better than seven algorithms and worse than SPEA2SDE, EFRRR and θ-DEA. SPEA uses a clustering technique to estimate the density of an individual, and a shift-based density estimation (SDE) can reflect the convergence of the other individuals with regard to the individual. The result of SPEA2SDE is good on both diversity and convergence but the optimization process takes more time. EFRRR uses an integrated ranking method and a restriction order to determine the order of individuals. θ-DEA uses an environmental selection mechanism named θ dominance, and combines the benefits of NSGA-III and MOEA/D. Utilizing the aggregation function-based fitness in MOEA/D, θ-DEA results in a better compromise between diversity and convergence in MaOPs. Similar to SPEA2SDE, both θ-DEA and EFRRR have no advantage in terms of optimization time, and as Tab. 2 shows, they are not good on IGD metric.

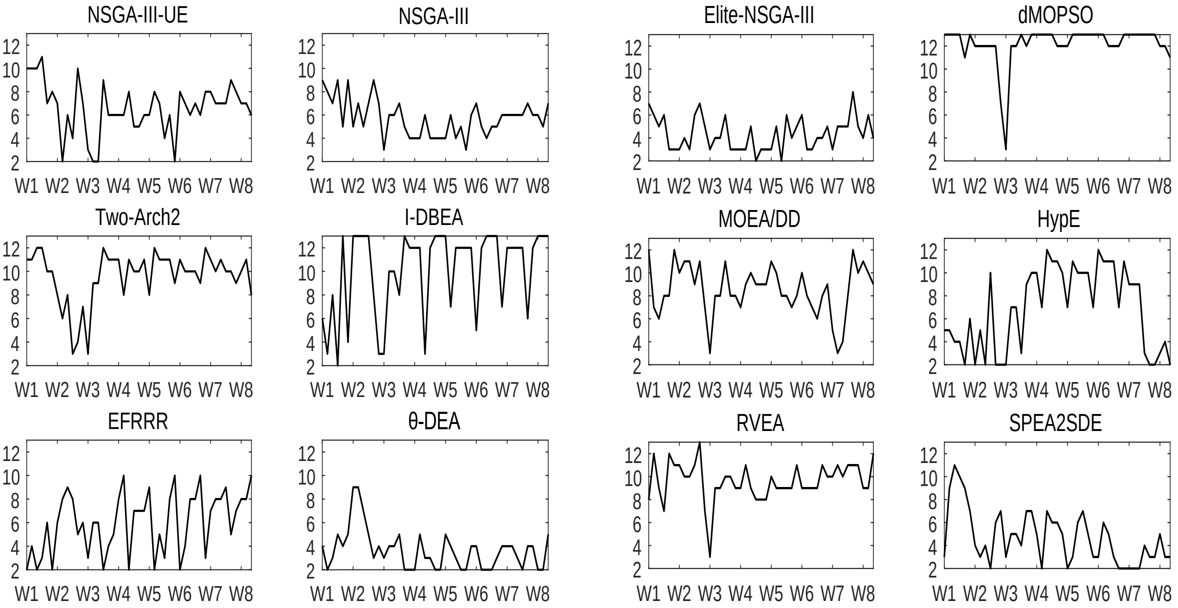


Figure 4: The rank of HV values of each algorithm on each function with different objective numbers

Tab. 5 gives the average operating time values of NSGA-III-UE, Elite-NSGA-III, NSGA-III, EFRRR, θ-DEA, SPEA2SDE and HypE which have advantage on HV or IGD. The data in parentheses after average values are standard deviations. Consistent with our analysis above, NSGA-III-UE is much better than

Table 5: Average operating time values of some algorithms running 30 times on WFGs with different objective numbers

Problem	M	NSGA-III-UE	Elite-NSGA-III	NSGA-III	EFRRR	θ-DEA	SPEA2SDE	HypE
WFG1	5	1.0350(8.52e-2)	1.2025 (3.88e-2)	1.6386 (1.31e-1)	1.1029 (3.88e-2)	1.0836 (1.13e-1)	24.408(5.56e-1)	50.586(1.17e+2)
	8	1.1449 (1.74e-1)	1.7035 (4.41e-1)	1.9423 (2.50e-1)	1.3264 (8.14e-2)	1.4548 (3.53e-1)	27.801 (6.11e-1)	71.918(1.63e+2)
	10	1.2704 (2.28e-1)	2.8580 (8.69e-1)	2.7558 (7.24e-1)	1.3819 (1.25e-1)	1.5328 (3.28e-1)	28.888 (5.68e-1)	87.741(2.04e+2)
	12	1.3035 (1.74e-1)	1.8287 (4.24e-1)	2.2502 (9.62e-1)	1.2637 (4.94e-2)	1.4722 (3.40e-1)	29.869 (7.90e-1)	103.09(2.46e+2)
	15	2.1405 (5.54e-1)	5.6715 (7.06e-1)	4.4708 (3.36e-1)	3.7613 (3.02e-1)	3.8211 (4.73e-1)	30.783 (6.75e-1)	162.73 (4.17e+2)
WFG2	5	1.2200 (1.07e-0)	9.6806e-1 (3.56e-2)	1.5999 (2.09e-1)	1.0936 (3.34e-2)	1.0610 (9.93e-2)	26.001 (4.64e-1)	112.62 (2.64e+2)
	8	1.2167 (1.09e-1)	1.9854 (3.79e-1)	2.1588 (2.70e-1)	1.2879 (4.03e-2)	1.5041 (2.02e-1)	29.556 (5.79e-1)	167.7 (3.40e+2)
	10	1.5166 (2.28e-1)	2.9580 (2.17e-1)	3.0647 (4.69e-1)	1.8402 (1.34e-1)	2.2011 (3.82e-1)	30.787 (6.56e-1)	157.50e (3.64e+2)
	12	1.5939 (1.68e-1)	2.2162 (8.19e-2)	2.8714 (4.00e-1)	1.8147 (1.20e-1)	2.2546 (3.38e-1)	31.696 (7.66e-1)	154.06e (3.47e+2)
	15	2.4495 (7.70e-1)	5.6483 (1.80e-1)	4.4563 (3.30e-1)	3.9520 (2.33e-2)	4.0694 (4.04e-1)	32.471 (8.58e-1)	232.25 (5.80e+2)
WFG3	5	1.5841 (1.66e-0)	1.5705 (2.17e-1)	1.9777 (2.62e-1)	1.1040 (3.15e-2)	1.5333 (2.53e-1)	31.619 (8.42e-1)	74.349 (1.45e+2)
	8	1.6552 (2.42e-1)	3.3741 (2.35e-1)	3.3919 (1.38e-0)	2.0660 (3.52e-2)	2.8250 (6.47e-1)	32.843 (8.07e-1)	88.626 (1.78e+2)
	10	1.7424 (2.45e-1)	3.5850 (2.72e-1)	3.5624 (6.45e-1)	2.2520 (2.51e-2)	2.9402 (5.28e-1)	33.272 (8.14e-1)	95.556 (1.90e+2)
	12	1.7573 (1.46e-1)	2.6392 (3.90e-2)	3.0575 (5.05e-1)	1.9315 (4.77e-2)	2.5539 (4.23e-1)	33.717 (8.93e-1)	101.91 (2.04e+2)
	15	2.3157 (3.19e-1)	5.4811 (7.51e-2)	4.5271 (2.88e-1)	3.9271 (2.19e-2)	4.1926 (3.21e-1)	34.110 (8.89e-1)	175.15(4.14e+2)
WFG4	5	1.0352 (9.24e-2)	9.9099e-1 (5.96e-2)	1.6354 (2.96e-1)	1.0877 (3.41e-2)	1.1121 (1.46e-1)	29.044 (7.49e-1)	443.5(2.93e+1)
	8	1.4613 (1.43e-0)	1.5826 (1.40e-1)	2.0419 (2.34e-1)	1.2204 (2.76e-2)	1.3246 (1.54e-1)	31.144 (8.12e-1)	570.09 (3.39e+1)
	10	1.4141 (1.47e-1)	2.1024 (3.30e-1)	2.6329 (4.63e-1)	1.3424 (6.00e-2)	1.5232 (1.62e-1)	31.712 (7.58e-1)	620.98 (3.22e+1)
	12	1.5484 (1.50e-1)	1.9193 (1.45e-1)	2.5486 (2.67e-1)	1.3786 (4.09e-2)	1.6358 (2.00e-1)	32.331 (8.65e-1)	674.24 (5.45e+1)
	15	2.2643 (3.14e-1)	5.4074 (6.82e-2)	4.5224 (3.13e-1)	3.0513 (4.86e-1)	2.8994 (4.14e-1)	32.850 (8.83e-1)	1410.9(9.75e+1)
WFG5	5	1.0058 (8.26e-2)	9.2283e-1 (2.43e-2)	1.5546 (1.48e-1)	1.0757 (2.65e-2)	1.0946 (1.45e-1)	28.657 (6.25e-1)	65.163(1.47e+2)
	8	1.1388 (9.85e-2)	1.4003 (6.80e-2)	1.9199 (2.58e-1)	1.1890 (2.43e-2)	1.2528 (1.49e-1)	31.319 (7.31e-1)	93.750 (2.43e+2)
	10	1.2509 (1.27e-1)	1.5899 (1.62e-1)	2.2707 (3.01e-1)	1.2842 (2.75e-2)	1.4285 (2.72e-1)	32.036 (7.72e-1)	96.550 (2.40e+2)
	12	1.4965 (2.67e-1)	1.5930 (1.10e-1)	2.2857 (3.39e-1)	1.3031 (5.32e-2)	1.4576 (1.77e-1)	32.647 (7.69e-1)	113.71 (3.03e+2)
	15	1.9246 (3.14e-1)	4.4984 (8.45e-1)	4.0050 (5.38e-1)	2.5733 (5.23e-1)	2.4352 (2.88e-1)	33.284 (7.77e-1)	167.96 (4.84e+2)
WFG6	5	1.0021 (1.27e-1)	9.0701e-1 (2.42e-2)	1.5552 (1.43e-1)	1.0759 (3.27e-2)	1.0714 (1.46e-1)	26.051 (5.19e-1)	45.314 (1.03e+2)
	8	1.1022 (6.97e-2)	1.3873 (8.94e-2)	2.2748 (1.37e-0)	1.1856 (2.19e-2)	1.2441 (1.34e-1)	28.888 (7.04e-1)	60.972(1.45e+2)
	10	1.2480 (1.21e-1)	1.8853 (4.54e-1)	2.3265 (3.40e-1)	1.2805 (3.16e-2)	1.3894 (1.50e-1)	29.420 (6.60e-1)	65.016 (1.50e+2)
	12	1.4435 (1.53e-1)	1.5778 (1.63e-1)	2.2858 (2.69e-1)	1.3353 (5.40e-2)	1.4157 (1.27e-1)	29.776 (5.80e-1)	75.830 (1.81e+2)
	15	2.0639 (3.26e-1)	5.2170 (6.24e-1)	4.2594 (4.91e-1)	2.4231 (6.16e-1)	2.4365 (2.80e-1)	30.414 (6.92e-1)	136.02 (4.07e+2)
WFG7	5	1.0406 (7.63e-2)	9.5735e-1 (3.22e-2)	1.6111 (1.46e-1)	1.0963 (3.66e-2)	1.1069 (1.09e-1)	30.434 (7.05e-1)	70.475 (1.69e+2)
	8	1.1981 (1.00e-1)	1.5005 (1.17e-1)	2.0078 (2.15e-1)	1.2298 (4.04e-2)	1.3598 (1.90e-1)	31.618 (7.22e-1)	81.358 (1.90e+2)
	10	1.3697 (1.49e-1)	1.9856 (3.38e-1)	2.6583 (5.33e-1)	1.3850 (3.72e-2)	1.5165 (1.70e-1)	31.713 (7.76e-1)	90.024 (2.11e+2)
	12	1.8726 (1.25e-0)	2.0769 (1.35e-1)	2.8359 (3.58e-1)	1.3915 (9.31e-2)	1.6190 (2.61e-1)	31.995 (7.03e-1)	94.807 (2.22e+2)
	15	2.4348 (3.18e-1)	5.4583 (9.63e-2)	4.5948 (2.73e-1)	3.3893 (7.03e-1)	3.0894 (4.75e-1)	32.421 (7.30e-1)	171.88 (5.03e+2)
WFG8	5	1.0179 (9.92e-2)	9.3239e-1 (6.22e-2)	1.6048 (2.12e-1)	1.0945 (3.47e-2)	1.0752 (1.36e-1)	25.117 (4.99e-1)	43.010 (9.39e+1)
	8	1.2036 (2.00e-1)	1.6679 (4.49e-1)	2.1347 (5.34e-1)	1.3299 (1.77e-1)	1.3069 (1.88e-1)	27.165 (5.23e-1)	57.981 (1.43e+2)
	10	1.4148 (2.36e-1)	2.4788 (5.32e-1)	3.0074 (6.89e-1)	1.4852 (2.66e-1)	1.6779 (5.03e-1)	27.958 (5.55e-1)	65.979 (1.62e+2)
	12	1.5802 (2.09e-1)	2.0142 (2.63e-1)	2.7024 (5.06e-1)	1.6310 (2.66e-1)	2.1299 (5.79e-1)	28.421 (6.09e-1)	77.617 (1.99e+2)
	15	2.3701 (4.69e-1)	5.3885 (2.78e-1)	4.5610 (3.03e-1)	3.9253 (1.20e-1)	3.3503 (6.15e-1)	29.150 (6.25e-1)	141.12(4.23e+2)
WFG9	5	1.0741 (1.46e-1)	9.8078e-1 (2.73e-2)	1.6260 (1.54e-1)	1.1076 (3.19e-2)	1.1353 (1.05e-1)	30.972 (1.05e+1)	599.39 (1.08e+2)
	8	1.2657 (1.59e-1)	1.6426 (1.71e-1)	2.1684 (3.27e-1)	1.2877 (3.85e-2)	1.3895 (2.55e-1)	32.683 (7.71e-1)	712.85 (1.02e+2)
	10	1.4568 (2.08e-1)	2.0313 (3.14e-1)	2.7478 (4.74e-1)	1.3862 (5.88e-2)	1.5750 (2.87e-1)	33.134 (8.13e-1)	797.74 (9.65e+1)
	12	1.6494 (2.47e-1)	2.0247 (1.09e-1)	2.8474 (4.49e-1)	1.4026 (5.92e-2)	1.6248 (3.01e-1)	33.624 (7.43e-1)	910.99 (1.29e+2)
	15	2.3916 (3.59e-1)	5.3259 (2.07e-1)	4.6353 (3.81e-1)	3.0002 (6.35e-1)	2.5791 (4.01e-1)	34.071 (9.15e-1)	1599.8 (1.86e+2)
count		28	7	0	10	0	0	0

SPEA2SDE, EFRRR and θ-DEA on the operating time metric. When the number of objectives m is more than 5, NSGA-III-UE needs less time than NSGA-III and Elite-NSGA-III. The reason is, for most of the high-dimensional objective problems, NSGA-III-UE does not need to spend time on the niche selection. More details can be seen in Section 3.4.

Take WFG9 with 8 objectives as an example. Fig. 5 shows the parallel coordinates plots [58] of the Pareto set obtained by the compared algorithms. The visualization results of the solution sets [59] obtained by the algorithms can further illustrate the differences of the performance .

Reviewing the results of these algorithms on IGD, HV and Operating time, it can be concluded that NSGA-III-UE has a considerable advantage over other methods on WFGs.

4.6.2. The Perspective of Test Functions

Fig. 6 analyzes the performance of eight algorithms from the perspective of test functions. Because I-DBEA, Elite-NSGA-III, dMOPSO and MOEA/DD do not get any optimal value on WFG functions with $m = \{5, 8, 10, 12, 15\}$, these four algorithms are excluded in the comparison. The histogram in Fig. 6 shows the statistical result of the remaining eight algorithms on the times of obtaining the optimal IGD values. Among these algorithms, NSGA-III-UE obtains the optimal value with 2 times on WFG2 and WFG4, corresponding to 33.33%, and one times on WFG5, WFG6, WFG7, WFG8 and WFG9. It is better than NSGA-III, Two-Arch2, EFRRR, HypE, and RVEA which only obtain once or none. NSGA-III-UE is worse than SPEA2SDE and RVEA on WFG1. On WFG3, HypE is the best algorithm which gets all the five times of the optimal value. Reviewing the above results, we can draw a conclusion that NSGA-III-UE is better than most of the algorithms on most WFG functions except WFG1 and WFG3. In Subsection 4.6.7, an analysis is provided to explain why NSGA-III-UE performs poorly on functions like WFG1 and WFG3.

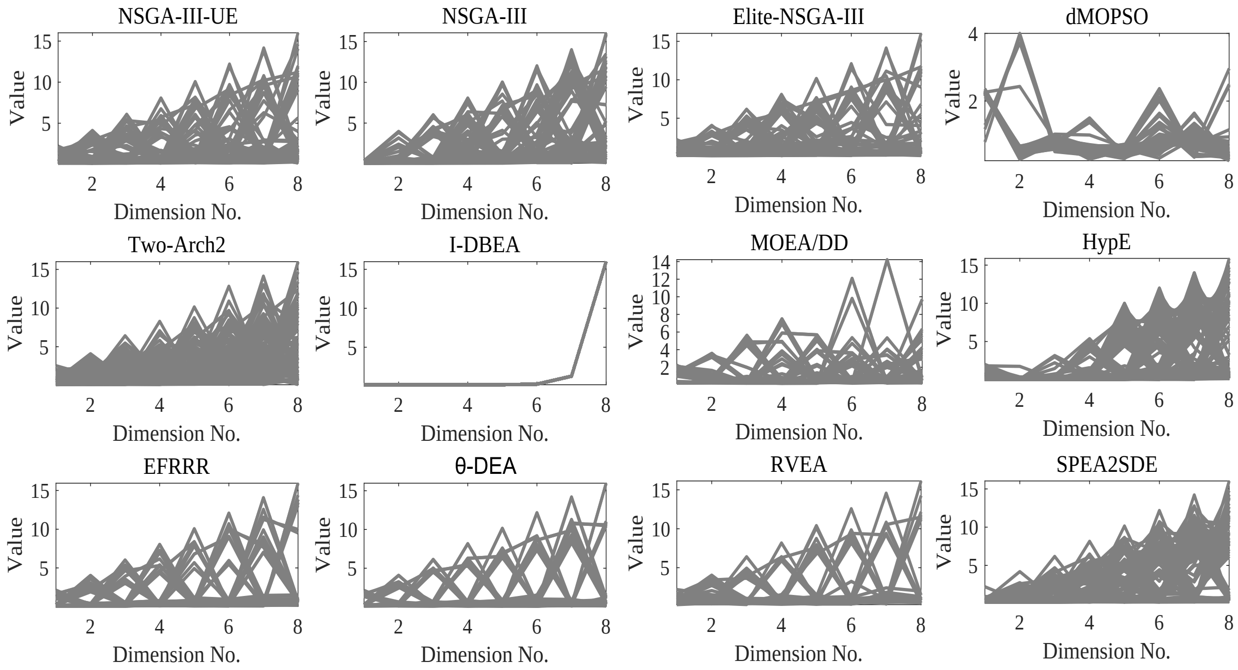


Figure 5: The parallel coordinates plots of the Pareto set obtained by each algorithm on WFG9 with 8 objectives

4.6.3. The Perspective of Objective Numbers

In this subsection, we analyze the performance of NSGA-III-UE from the view of objective numbers. Fig. 7 compares the ratios of optimal IGD values for all the algorithms on WFGs when $m = \{3, 5, 8, 10, 12, 15\}$. As previously mentioned, I-DBEA, dMOPSO and MOEA/DD do not return any good values on WFGs and therefore are again excluded in this comparison. From Fig. 7, it can be seen that NSGA-III-UE has a good performance when the objective number is 8, and the ratio of obtained optimal IGD value is 66.67%. Furthermore, Fig. 8 shows the IGD ranks of all the algorithms with different objective number in more detail. The numbers of the abscissa axis are the objective numbers, and the positions between each two numbers of abscissa represent the functions of WFGs. From this figure, we can observe that the ranks of our proposed algorithm are better than most of the other algorithms. Only when the objective number is 3, which is a multi-objective problem, does NSGA-III-UE fail to get the optimal value. In summary, our proposed algorithm demonstrates a significant advantage on WFGs with different objective numbers in terms of IGD, especially when the number of objectives is 8.

4.6.4. The Overall Performance on DTLZ Functions

Tab. 6 shows experimental results of eleven algorithms except SPEA2SDE which is powerful on DTLZ functions [28]. In 40 groups test of 8 functions, NSGA-III-UE gets the best average IGD value for 5 times. It is better than NSGA-III, I-DBEA, EFRRR, θ-DEA , Elite-NSGA-III, dMOPSO and RVEA, but worse than Two-Arch2, MOEA/DD and HypE. MOEA/DD has the best performance with 11 times of the best IGD value, especially on DTLZ1 and DTLZ3. HypE wins for 8 times. However, both of them do not perform well on other functions. Two-arch2 has 6 times of the best IGD value. Therefore, more analyses should be done to illustrate the performance of NSGA-III-UE.

Tab. 7 shows that the overall IGD rankings of NSGA-III-UE is 3,while Two-Arch2, MOEA/DD and HypE are 5, 8 and 6, respectively; worse than the proposed algorithm. The average ranks of θ-DEA and Elite-NSGA-III are better than NSGA-III-UE, however, the times for obtaining the best IGD values is 2. This is lower than our algorithm which obtains 5.

Taking the analysis further, the average operating time values of each algorithm on DTLZs are shown in

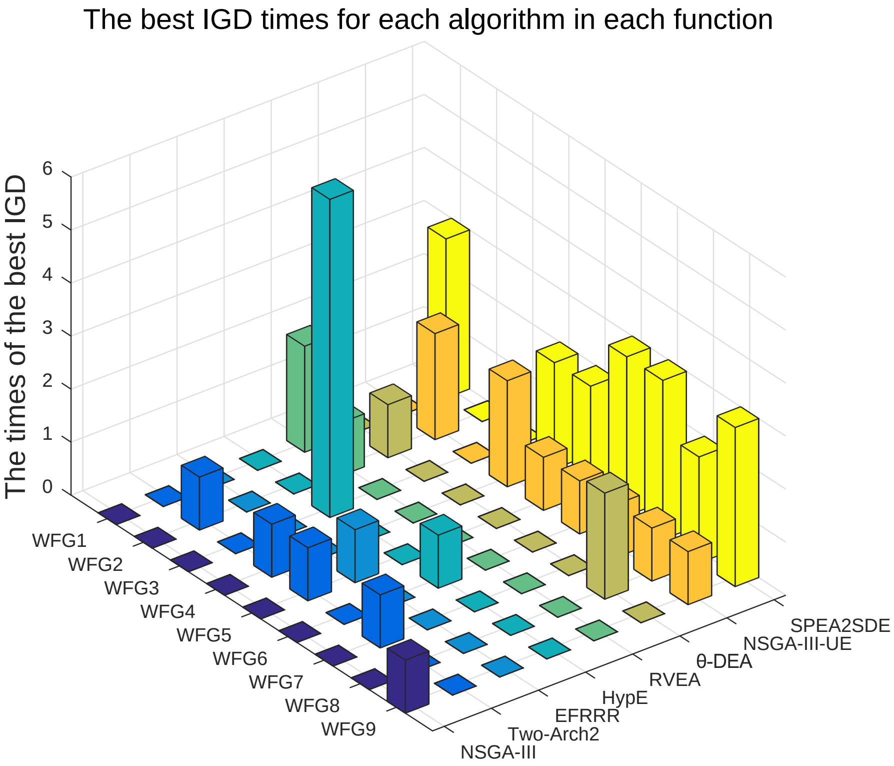


Figure 6: The times of the best IGD value of each algorithm on each function

Tab. 8. The optimization times for RVEA are consistently the smallest; however, RVEA has not achieved the optimal value on IGD. Excluding RVEA, NSGA-III-UE has the shortest optimization time in most of the tests. The operating time of NSGA-III-UE is significantly better than Two-Arch2, I-DBEA, MOEA/DD, HpyE and SPEA2SDE. Tab. 8 shows the advantage of our proposed algorithm in terms of time efficiency.

Considering the results of these algorithms on IGD and operating time, the overall performance of NSGA-III-UE is better than the other algorithms.

4.6.5. The Perspective of Stability

Boxplots Fig. 9 is used to illustrate the stability of our algorithm. The performance of NSGA-III-UE is analyzed on WFG9 with different objective numbers: $m = \{5, 8, 10, 12, 15, 20\}$. The compared algorithms run 30 times on each objective number. The IGD values are plotted on boxplots. The abscissa values correspond to these algorithms: NSGA-III-UE(1), NSGA-III(2), Two-Arch2(3), I-DBEA(4), EFRRR(5), θ-DEA(6), Elite-NSGA-III(7), dMOPSO(8), MOEA/DD(9), HypE(10), RVEA(11) and SPEA2SDE(12), respectively.

It can be seen from Fig. 9 that for WFG9, the overall IGD value distribution of NSGA-III-UE is more evenly than other algorithms. Especially, when $m = \{5, 8, 10\}$, the IGD values obtained by NSGA-III-UE have no outliers. When the number of objectives is 8, the datum is distributed evenly and the value is the minimum. The boxplots indicate that the performance of NSGA-III-UE is stability.

Table 6: Average IGD values of algorithms running 30 times on DTLZs with different objective numbers

Problem	M	1	2	3	4	5	6	7	8	9	10	11
DTLZ1	5	6.75e-1^{6}	6.57e-1^{5}	8.99e-1^{8}	9.03e-1^{9}	2.21e+0^{10}	4.03e-1^{2}	6.80e-1^{7}	4.97e+0^{11}	3.07e-1^{1}	5.30e-1^{3}	6.51e-1^{4}
	8	1.92e+0^{9}	1.42e+0^{7}	1.25e+0^{5}	1.64e+0^{8}	4.07e-1^{2}	7.31e-1^{4}	1.27e+0^{6}	4.32e+0^{11}	2.87e-1^{1}	6.50e-1^{3}	2.22e+0^{10}
	10	1.89e+0^{9}	1.28e+0^{7}	1.11e+0^{6}	3.36e+0^{10}	3.73e-1^{2}	5.49e-1^{3}	1.31e+0^{8}	7.43e+0^{11}	3.02e-1^{1}	6.41e-1^{5}	5.97e-1^{4}
	12	1.81e+0^{9}	1.48e+0^{7}	2.91e+0^{10}	1.50e+0^{8}	4.46e-1^{2}	6.78e-1^{4}	1.19e+0^{6}	9.59e+0^{11}	3.15e-1^{1}	7.26e-1^{5}	4.81e-1^{3}
	15	1.53e+0^{6}	1.59e+0^{7}	2.33e+0^{8}	1.07e+1^{10}	1.17e+0^{4}	8.64e-1^{2}	1.38e+0^{5}	1.27e+1^{11}	4.73e-1^{1}	1.13e+0^{3}	9.39e+0^{9}
DTLZ2	5	2.36e-1^{5}	2.34e-1^{2}	2.14e-1^{1}	2.40e-1^{7}	2.39e-1^{6}	2.34e-1^{3}	2.35e-1^{4}	3.47e-1^{10}	2.43e-1^{9}	3.65e-1^{11}	2.42e-1^{8}
	8	4.82e-1^{4}	4.87e-1^{5}	4.43e-1^{1}	9.26e-1^{11}	4.80e-1^{3}	4.78e-1^{2}	4.96e-1^{6}	6.94e-1^{10}	5.04e-1^{7}	6.04e-1^{9}	5.22e-1^{8}
	10	5.58e-1^{4}	5.99e-1^{6}	6.04e-1^{7}	1.26e+0^{11}	5.18e-1^{1}	5.22e-1^{3}	5.83e-1^{5}	8.56e-1^{10}	6.05e-1^{8}	8.23e-1^{9}	5.20e-1^{2}
	12	6.21e-1^{4}	6.47e-1^{6}	7.25e-1^{8}	1.28e+0^{11}	5.55e-1^{1}	5.66e-1^{2}	6.36e-1^{5}	9.55e-1^{10}	6.87e-1^{7}	8.37e-1^{9}	5.71e-1^{3}
	15	7.86e-1^{2}	8.05e-1^{3}	8.68e-1^{6}	1.32e+0^{10}	8.54e-1^{5}	7.76e-1^{1}	8.12e-1^{4}	1.18e+0^{9}	1.23e+4^{11}	9.21e-1^{7}	9.55e-1^{8}
DTLZ3	5	3.24e+1^{9}	2.22e+1^{8}	1.19e+1^{2}	2.01e+1^{7}	4.61e+1^{10}	1.44e+1^{3}	1.98e+1^{5}	1.51e+2^{11}	1.51e+1^{4}	1.16e+1^{1}	1.98e+1^{6}
	8	6.83e+1^{10}	4.50e+1^{7}	5.91e+1^{8}	1.97e+1^{2}	3.30e+1^{5}	2.07e+1^{3}	4.47e+1^{6}	1.72e+2^{11}	1.05e+1^{1}	2.08e+1^{4}	6.09e+1^{9}
	10	7.58e+1^{9}	7.03e+1^{8}	1.14e+2^{10}	1.75e+1^{3}	2.17e+1^{6}	1.96e+1^{5}	5.19e+1^{7}	1.36e+2^{11}	6.87e+0^{1}	1.83e+1^{4}	1.30e+1^{2}
	12	7.47e+1^{9}	6.41e+1^{8}	1.23e+2^{10}	2.52e+1^{5}	2.90e+1^{6}	2.22e+1^{4}	5.80e+1^{7}	1.95e+2^{11}	6.99e+0^{1}	2.19e+1^{3}	1.60e+1^{2}
	15	4.87e+1^{8}	2.79e+1^{5}	1.34e+2^{10}	3.67e+1^{7}	2.19e+1^{3}	2.51e+1^{4}	2.87e+1^{6}	1.70e+2^{11}	1.07e+1^{1}	2.00e+1^{2}	7.45e+1^{9}
DTLZ4	5	2.68e-1^{4}	3.18e-1^{7}	2.26e-1^{1}	5.28e-1^{8}	2.46e-1^{2}	2.67e-1^{3}	2.84e-1^{6}	5.40e-1^{9}	6.11e-1^{10}	6.21e-1^{11}	2.76e-1^{5}
	8	4.92e-1^{3}	5.46e-1^{6}	4.53e-1^{1}	7.13e-1^{9}	4.86e-1^{2}	5.06e-1^{4}	5.33e-1^{5}	7.00e-1^{8}	9.15e-1^{11}	8.09e-1^{10}	5.82e-1^{7}
	10	5.81e-1^{3}	6.29e-1^{7}	5.82e-1^{4}	8.87e-1^{10}	5.21e-1^{1}	5.25e-1^{2}	6.12e-1^{6}	7.33e-1^{8}	1.03e+0^{11}	8.56e-1^{9}	5.84e-1^{5}
	12	6.47e-1^{6}	6.76e-1^{7}	6.28e-1^{4}	8.08e-1^{9}	5.70e-1^{1}	5.74e-1^{2}	6.32e-1^{5}	7.62e-1^{8}	1.02e+0^{11}	8.93e-1^{10}	6.01e-1^{3}
	15	8.03e-1^{4}	8.08e-1^{6}	7.57e-1^{1}	1.41e+0^{11}	7.72e-1^{2}	7.75e-1^{3}	8.03e-1^{5}	9.00e-1^{7}	1.20e+0^{10}	9.70e-1^{8}	9.92e-1^{9}
DTLZ7	5	5.56e-1^{3}	5.64e-1^{5}	6.44e-1^{6}	1.07e+0^{8}	5.59e-1^{4}	5.44e-1^{2}	5.18e-1^{1}	7.54e-1^{7}	1.58e+0^{10}	2.22e+0^{11}	1.11e+0^{9}
	8	3.41e+0^{8}	1.91e+0^{4}	3.22e+0^{7}	4.72e+0^{9}	1.80e+0^{3}	1.69e+0^{2}	1.95e+0^{6}	1.93e+0^{5}	1.65e+0^{1}	5.02e+0^{10}	7.50e+0^{11}
	10	6.49e+0^{10}	4.67e+0^{7}	5.42e+0^{8}	8.92e+0^{11}	3.17e+0^{3}	2.64e+0^{1}	4.03e+0^{6}	3.01e+0^{2}	3.51e+0^{5}	6.45e+0^{9}	3.22e+0^{4}
	12	1.01e+1^{9}	7.51e+0^{7}	1.31e+1^{10}	1.33e+1^{11}	6.84e+0^{6}	5.84e+0^{4}	6.71e+0^{5}	3.23e+0^{1}	4.45e+0^{2}	8.10e+0^{8}	5.73e+0^{3}
	15	1.82e+1^{8}	9.28e+0^{5}	2.77e+1^{9}	5.84e+1^{11}	2.96e+1^{10}	1.03e+1^{6}	8.84e+0^{3}	9.06e+0^{4}	5.06e+0^{1}	1.29e+1^{7}	6.49e+0^{2}
IDTLZ1	5	5.93e-1^{1}	8.61e-1^{7}	1.94e+0^{9}	1.46e+0^{8}	2.16e+0^{10}	6.42e-1^{4}	7.86e-1^{6}	6.52e+1^{11}	5.97e-1^{2}	6.28e-1^{3}	7.23e-1^{5}
	8	7.05e-1^{4}	7.84e-1^{6}	5.84e+0^{9}	2.64e+0^{7}	4.75e+0^{8}	5.69e-1^{2}	5.18e-1^{1}	9.35e+1^{11}	7.36e-1^{5}	6.56e-1^{3}	9.61e+0^{10}
	10	1.18e+0^{5}	7.09e-1^{2}	7.13e+0^{10}	3.72e+0^{8}	2.26e+0^{7}	9.22e-1^{3}	1.03e+0^{4}	1.05e+2^{11}	1.69e+0^{6}	4.93e-1^{1}	6.18e+0^{9}
	12	2.19e+0^{6}	6.87e-1^{1}	3.37e+0^{7}	4.72e+0^{8}	6.28e+0^{9}	7.48e-1^{2}	1.16e+0^{4}	1.11e+2^{11}	1.54e+0^{5}	9.60e-1^{3}	7.38e+0^{10}
	15	3.11e+0^{6}	2.41e+0^{4}	6.48e+0^{9}	4.69e+0^{8}	4.39e+0^{7}	1.77e+0^{2}	2.01e+0^{3}	1.27e+2^{11}	2.76e+0^{5}	1.51e+0^{1}	1.03e+2^{10}
IDTLZ2	5	2.79e-1^{3}	2.84e-1^{5}	2.16e-1^{2}	3.09e-1^{7}	3.46e-1^{9}	4.29e-1^{11}	2.81e-1^{4}	3.62e-1^{10}	2.96e-1^{6}	2.00e-1^{1}	3.37e-1^{8}
	8	5.25e-1^{3}	5.61e-1^{4}	4.24e-1^{2}	6.56e-1^{6}	6.93e-1^{9}	6.81e-1^{8}	5.65e-1^{5}	7.79e-1^{11}	7.13e-1^{10}	3.69e-1^{1}	6.67e-1^{7}
	10	6.70e-1^{3}	7.00e-1^{4}	5.29e-1^{2}	7.95e-1^{7}	8.81e-1^{10}	7.92e-1^{6}	7.15e-1^{5}	9.24e-1^{11}	8.61e-1^{9}	4.67e-1^{1}	8.03e-1^{8}
	12	7.81e-1^{3}	8.26e-1^{5}	6.15e-1^{2}	8.66e-1^{7}	9.63e-1^{10}	8.60e-1^{6}	8.20e-1^{4}	9.89e-1^{11}	9.62e-1^{9}	5.47e-1^{1}	8.93e-1^{8}
	15	9.52e-1^{3}	9.57e-1^{4}	7.11e-1^{2}	1.01e+0^{7}	1.13e+0^{9}	9.94e-1^{6}	9.64e-1^{5}	1.15e+0^{10}	1.12e+0^{8}	6.60e-1^{1}	1.53e+0^{11}
CDTLZ2	5	9.61e-2^{1}	9.65e-2^{2}	9.87e-2^{3}	1.02e-1^{5}	1.04e-1^{6}	1.15e-1^{8}	9.93e-2^{4}	1.56e-1^{10}	1.36e-1^{9}	1.81e-1^{11}	1.09e-1^{7}
	8	1.56e-1^{1}	1.62e-1^{3}	1.61e-1^{2}	8.96e-1^{11}	2.32e-1^{10}	1.70e-1^{5}	1.70e-1^{6}	2.10e-1^{9}	1.99e-1^{8}	1.66e-1^{4}	1.72e-1^{7}
	10	1.51e-1^{1}	1.78e-1^{6}	1.57e-1^{2}	9.64e-1^{11}	1.91e-1^{7}	1.76e-1^{5}	1.71e-1^{4}	2.25e-1^{10}	2.19e-1^{9}	2.19e-1^{8}	1.69e-1^{3}
	12	1.42e-1^{1}	1.55e-1^{4}	1.69e-1^{6}	9.03e-1^{11}	1.66e-1^{5}	1.77e-1^{7}	1.43e-1^{2}	2.22e-1^{9}	2.21e-1^{8}	2.52e-1^{10}	1.45e-1^{3}
	15	2.37e-1^{5}	2.22e-1^{2}	1.88e-1^{1}	9.99e-1^{11}	2.50e-1^{8}	3.88e-1^{10}	2.24e-1^{3}	2.50e-1^{7}	2.41e-1^{6}	2.35e-1^{4}	3.83e-1^{9}
count		5	1	6	0	4	2	2	1	11	8	0

Table 7: IGD analysis of various algorithms on DTLZs

	1	2	3	4	5	6	7	8	9	10	11
times of the best	5	1	6	0	4	2	2	1	11	8	0
the rate of optimal values	12.50%	2.50%	15.00%	0.00%	10.00%	5.00%	5.00%	2.50%	27.50%	20.00%	0.00%
total of ranks	207	211	219	338	224	159	195	371	232	224	260
average rank	5.18	5.28	5.48	8.45	5.60	3.98	4.88	9.28	5.80	5.60	6.50
overall sorting	3	4	5	10	6	1	2	11	8	6	9

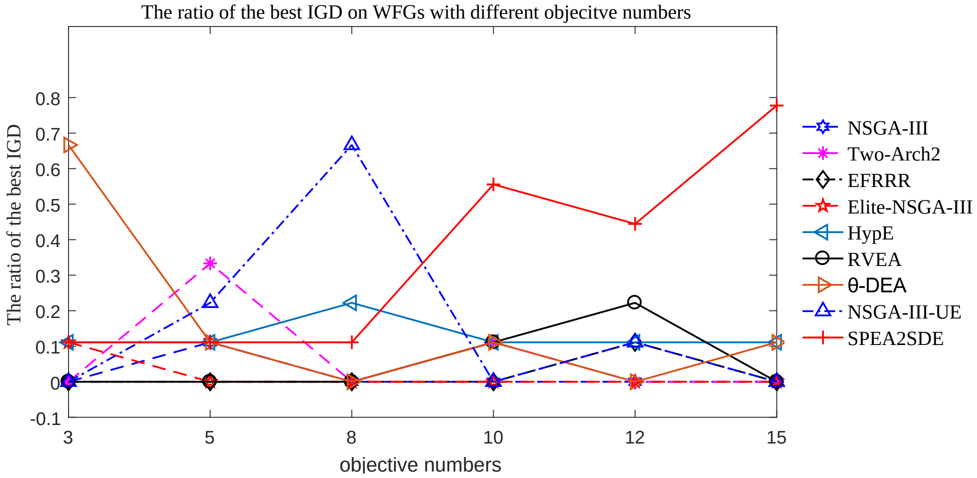


Figure 7: The ratios of optimal IGD values of each algorithm on WFGs with different objective numbers

4.6.6. Verify the Efficiency of the Two-archive Strategy

The variants of NSGA-III-UE are used as compared algorithms to analyze the influence of the uniform archive, the single elite archive and the two-archive strategy, respectively . Based on the frame of NSGA-III, NSGA-III-U is an algorithm which only uses the uniform archive, and NSGA-III-E is an algorithm which only uses the single elite archive. Since the PF of each function with 3 objectives can be visualized and observed intuitively, let us take DTLZ4 with 3 objective numbers as an example to illustrate the effect of the uniform archive on diversity and the single elite archive on convergence. The multi-objective traveling salesman problem(MOTSP) with un-regular PF is also used to illustrate the effectiveness of the two-archive strategy.

Fig.10 is used to illustrate the effect of the uniform archive. The evolution process of the PF obtained by NSGA-III-UE is shown in Fig.10. The sub-figures in Fig.10 are the PFs of the corresponding generations. From the sub-figures, it can be seen that in the initial stage of optimization, solutions are concentrated in a region of the entire space. With the increase of iterations, the search area is gradually expanded to the entire search space under the effect of the uniform archive. Fig.11(a) shows the values of IGD metric to exhibit the overall performance of each algorithm in its evolutionary process. The algorithm NSGA-III-E converges more quickly than NSGA-III, but it is apt to fall into premature convergence. The uniform archive in NSGA-III-U can maintain diversity and prevent algorithm from premature convergence to some extent. Fig.11(b) shows the values of spread indicator. It can be seen that the population diversity of the algorithm using the two-archive strategy is better than the other algorithms. From the above figures, the efficiency of the proposed two-archive strategy is demonstrated. We can learn that the proposed two-archive strategy can help the algorithm converging quickly and having better diversity.

The multi-objective traveling salesman problem(MOTSP) is used as a test problem with an un-regular PF to evaluate the effectiveness of the uniform archive and the single elite archive further. Table 9 gives the average values of NSGA-III, NSGA-III-U, NSGA-III-E and NSGA-III-UE in terms of Spread, HV and IGD. The data in parentheses after average values are standard deviations. Table 9 also shows the population diversity of the algorithm using the two-archive strategy is the best, and the overall performance of NSGA-III-UE is better than NSGA-III, NSGA-III-U and NSGA-III-E.

4.6.7. Discussion of NSGA-III-UE

Although NSGA-III-UE has a good performance on most of the test functions, it still does not work well on some functions. Now we discuss why NSGA-III-UE is not good at these functions.

Table 2 has shown that NSGA-III-UE does not achieve the best IGD values on WFG1 and WFG3, but

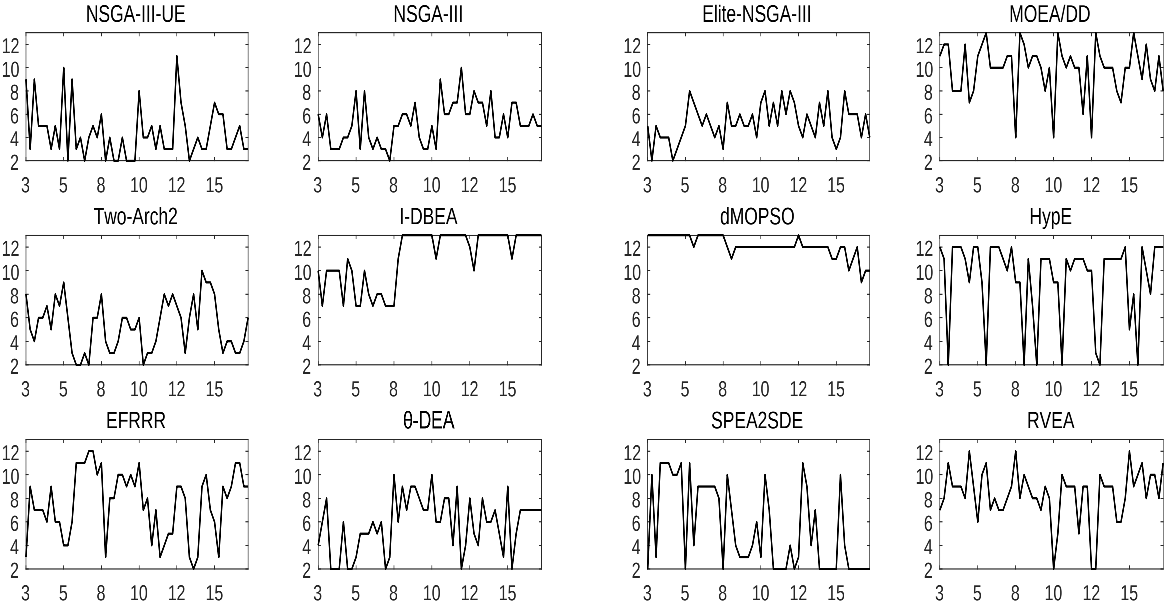


Figure 8: The ranks of IGD values of each algorithm on WFGs with different objective numbers

Table 8: Average operating time values of algorithms running 30 times on DTLZs with different objective numbers

Problem	M	1	2	3	4	5	6	7	8	9	10	11	12
DTLZ1	5	0.7731	1.2662	8.2154	8.2000	1.1019	0.7855	0.8950	2.7382	15.4620	70.5580	0.4829	10.0280
	8	0.9123	1.7895	10.9330	7.2400	1.3229	1.1299	1.6651	2.6875	14.3290	106.2300	0.4617	11.9340
	10	1.0633	2.0447	12.3330	6.9952	1.7147	1.6413	2.4705	2.8301	14.4290	139.4700	0.5103	13.1140
	12	1.1594	1.9842	13.2560	13.2690	1.7203	1.5357	2.0668	2.6121	16.2980	139.4100	0.4763	13.7380
	15	1.5608	3.9826	14.6310	13.1690	3.4271	3.7390	5.7295	3.2514	12.8550	111.4800	0.7053	15.2140
DTLZ2	5	0.9836	1.4726	12.9150	10.3140	1.1029	0.9565	0.8341	2.5327	14.7210	588.6800	0.4871	14.8260
	8	1.0428	2.0366	15.6990	9.1811	1.1215	1.2061	1.5191	2.7211	14.0220	847.6500	0.4823	16.4320
	10	1.1319	2.4222	17.3040	18.3120	1.3241	1.3736	2.1515	2.7277	13.9140	953.9000	0.5478	17.1240
	12	1.3346	2.5420	18.4210	16.5430	1.2988	1.4094	1.9538	2.5329	15.0620	1001.9000	0.5070	17.5690
	15	1.9229	4.1972	19.9420	15.2840	2.6921	2.6220	5.5189	3.2096	12.7720	1911.6000	0.7678	18.1580
DTLZ3	5	0.7729	1.3056	7.9449	12.1200	1.1087	0.7826	0.9225	2.6847	17.0360	19.6620	0.5051	8.3836
	8	0.8960	1.5533	11.3790	7.4470	1.2138	1.0036	1.8724	2.8167	15.4880	57.6020	0.5029	11.8660
	10	1.0670	2.2697	13.1650	15.7970	1.7349	1.2513	2.8703	2.7684	15.3570	69.6880	0.5458	12.8850
	12	1.1948	2.0847	15.1200	14.6370	1.6922	1.4070	2.2337	2.6715	16.8240	85.7350	0.5109	14.3630
	15	1.6408	4.0086	16.4910	13.9990	3.4734	3.7570	5.3828	3.2875	12.9160	104.9200	0.7485	15.8530
DTLZ4	5	0.8375	1.6851	13.6050	55.9350	1.1265	1.1016	1.0553	2.6495	14.6380	64.2510	0.4975	14.5270
	8	1.0250	2.1402	16.4960	18.9540	1.2673	1.5128	2.0524	2.8555	14.0610	170.8800	0.4936	16.5520
	10	0.9764	2.2631	17.5850	21.5670	1.7216	1.0445	1.6217	2.8680	13.8640	210.9100	0.5681	17.5730
	12	1.1560	2.0315	18.9240	20.4050	1.2754	1.2894	1.8157	2.7914	15.3700	237.4100	0.5431	17.7870
	15	1.5800	4.0320	20.0020	17.2040	2.5842	2.6477	5.4288	3.2294	13.0580	268.9600	0.7580	18.5410
DTLZ7	5	1.1157	2.0371	11.0750	8.8137	1.5821	1.1645	2.2682	2.5083	14.7020	35.7550	0.4225	14.3050
	8	1.1895	2.3298	16.0120	9.5644	1.5269	1.5889	2.7731	2.7432	14.6859	133.2400	0.4307	16.8460
	10	1.2399	2.4573	17.2940	19.9660	2.0106	1.8391	2.7100	2.8564	15.0530	192.9100	0.5656	17.5810
	12	1.2356	2.1384	20.3100	19.3640	1.8539	1.6822	2.0515	2.5812	18.4450	230.8700	0.4582	18.0260
	15	1.6733	4.0110	20.9530	17.3340	3.4757	3.6327	5.1920	3.1272	14.4700	263.4000	0.7368	18.2310
IDTLZ1	5	0.9967	1.7894	6.0122	7.6748	2.0151	1.4944	2.6657	2.6653	15.8440	24.4010	0.3864	7.7407
	8	1.0793	2.1578	6.7936	8.2595	1.7358	1.8066	3.4487	2.7144	16.0870	47.3860	0.4970	8.6558
	10	1.1178	2.3238	6.8870	15.8630	2.2566	1.9825	3.6278	2.8390	15.9840	49.1610	0.5287	9.2196
	12	1.1036	2.0303	7.4876	13.3990	1.9399	1.6777	2.8869	2.6573	17.9690	65.9880	0.5152	9.7994
	15	1.5683	3.9946	7.4300	14.9860	3.4211	3.7397	6.8642	3.3134	14.7960	78.5650	0.8053	9.7519
IDTLZ2	5	1.0300	1.8755	11.4370	19.2960	1.3978	1.8277	1.7752	2.4839	18.0250	917.3900	0.4327	15.3160
	8	1.2848	2.6885	13.1250	18.6740	1.6016	2.0612	2.9627	2.8011	16.8300	861.8200	0.5113	16.7670
	10	1.4793	3.0534	14.1820	20.2500	2.2520	2.5577	3.2854	2.7099	17.7850	934.1100	0.5600	17.1940
	12	1.4779	2.6601	14.8290	19.4040	1.9771	2.1362	2.1268	2.5529	19.6500	1039.6000	0.5232	17.5530
	15	1.9155	4.2760	15.7360	17.1170	3.4607	3.7705	5.8756	3.2408	16.1960	2205.5000	0.7923	18.0300
CDTLZ2	5	0.9549	1.5975	12.8990	13.3060	1.2010	0.9714	0.8547	2.6308	18.2980	651.7300	0.5429	15.1000
	8	1.0179	1.8698	15.0470	10.8640	1.3557	1.2526	1.2595	2.6694	16.0230	5975.0000	0.5284	16.9860
	10	1.2929	2.5059	16.6870	17.3930	1.4760	1.8476	2.0719	2.7028	15.9420	1052.1000	0.5820	17.7810
	12	1.4611	2.4484	18.5480	17.1690	1.4992	1.7442	1.8158	2.5588	17.8390	1690.1000	0.5958	17.7920
	15	1.8140	4.1129	20.2690	14.2700	2.1009	2.8508	5.3936	3.2931	15.4160	1897.6000	0.8566	18.1180

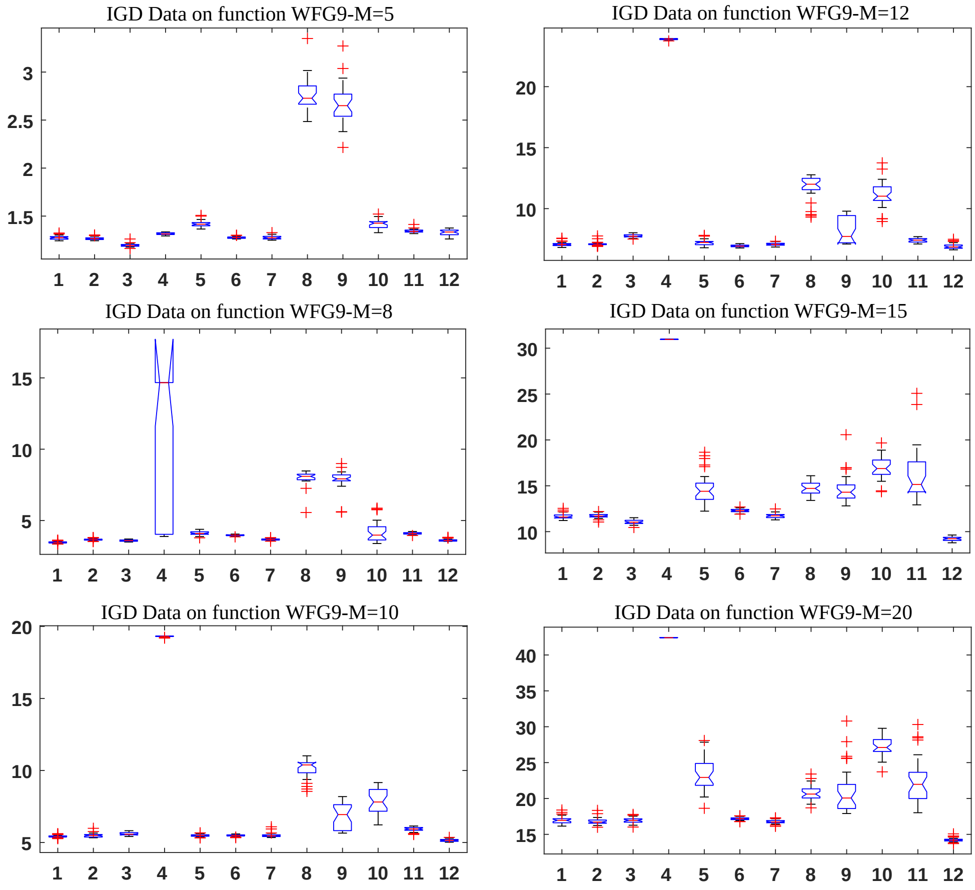


Figure 9: Boxplots of IGD values of algorithms running 30 times on IDTZL2 with m = {5, 8, 10, 12, 15, 20}

performs well on other functions. Let us consider the characteristics of these functions and the performance of NSGA-III-UE together. As previously mentioned, the PF of each function with 3 objectives can be observed intuitively, and therefore are again used as an example. Fig. 12 displays the true PF of WFG functions when the number of objectives is 3. Fig. 13 shows the approximating Pareto Front [60] of each function with 3 objectives optimized by NSGA-III-UE. Let us observe the characteristics of functions through Fig. 12. For WFG1,WFG2 and WFG3, the real PFs are more concentrated than the others, especially for WFG3, which is concentrated in a very small rectangular area. Using the two-archive strategy, NSGA-III-UE aims to search the entire solution space evenly. It may not have an advantage for solving a problem which solution set is concentrated.

Considering above tables and figures together, we can conclude that NSGA-III-UE brings more considerable benefits for solving a problem whose Pareto solution set is distributed wildly in the solution space, not concentrated in a small area. However, the real Pareto solution sets of most real-world optimization problems are unknown. Therefore, it is necessary to search for optimal solutions in entire solution space.

5. Conclusions

After analyzing challenges of NSGA-III for solving MaOPs, a novel two-archive strategy is presented in this paper. The uniform archive strategy addresses the issue of NSGA-III unable to search the area in which no dominant solution is near a reference point. This strategy aims to maintain population diversity and

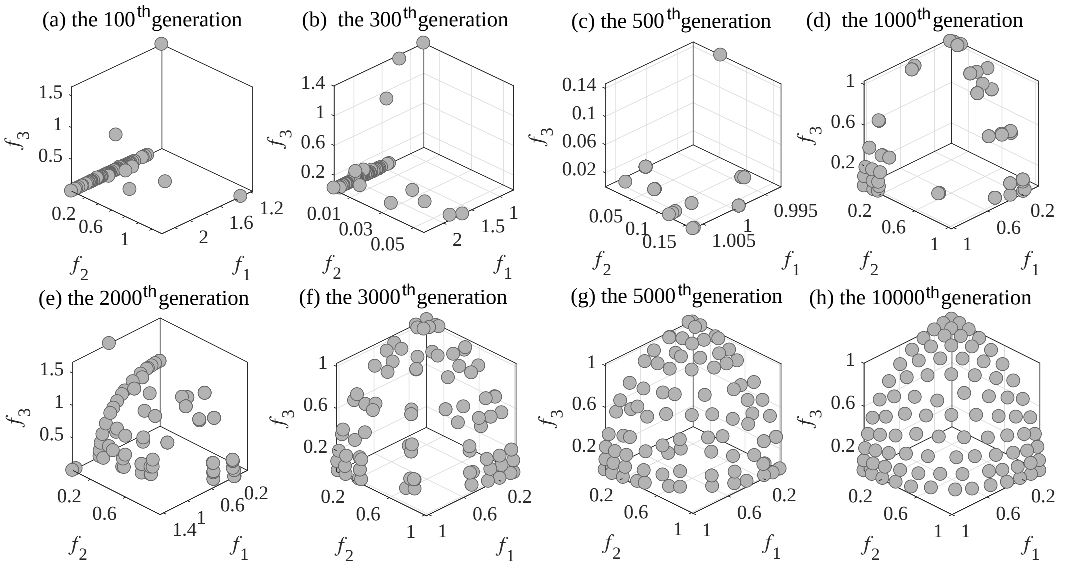


Figure 10: The evolution processes of NSGA-III-UE on DTLZ4 with 3 objectives

ensures that the algorithm is able to search the entire solution space. Inspired from PSO, the single elite archive strategy is proposed to guide search direction. It solves the problem that NSGA-III cannot ensure single elite individuals entering the next generation when the number of non-dominated solutions is larger than the population size. The single elite archive preserves individuals with the best single objective value of the current generation and allows them to participate in evolution. This improves the convergence of the algorithm. Based on the two-archive strategy, an improved NSGA-III algorithm, called NSGA-III-UE, is designed for solving MaOPs. NSGA-III-UE is compared with other eleven state-of-the-art algorithms in terms of IGD, HV, Spread and Operating time. Experimental results show that the proposed NSGA-III-UE performs well on most of the test problems.

However, the effect of the uniform archive depends on the reference points. In this paper, our reference points are generated from Das and Denis's approach which the generated points may not uniformly distribute corresponding to the PF. In the future, we will consider other approach to generate better reference points.

Table 9: Spread, HV and IGD values of variants algorithms on MOTSP

spread				
M	NSGA-III	NSGA-III-U	NSGA-III-E	NSGA-III-UE
5	7.2581e-1 (2.76e-2)	7.1433e-1 (2.17e-2)	7.1403e-1 (2.66e-2)	7.1276e-1 (2.19e-2)
8	7.5001e-1 (1.36e-2)	7.5186e-1 (1.97e-2)	7.5165e-1 (1.91e-2)	7.4777e-1 (1.97e-2)
10	7.8412e-1 (1.65e-2)	7.8325e-1 (1.41e-2)	7.8607e-1 (1.23e-2)	7.8076e-1 (1.72e-2)
12	7.5578e-1 (1.49e-2)	7.4363e-1 (1.39e-2)	7.4619e-1 (1.23e-2)	7.4257e-1 (1.71e-2)
15	9.5909e-1 (1.06e-2)	9.5581e-1 (8.54e-3)	9.5064e-1 (6.49e-3)	9.5025e-1 (1.09e-2)
20	9.5398e-1 (6.72e-3)	9.5461e-1 (8.96e-3)	9.4820e-1 (9.42e-3)	9.4526e-1 (8.60e-3)
HV				
5	7.6394e+6 (5.07e+5)	7.6112e+6 (4.42e+5)	7.4928e+6 (4.06e+5)	7.6605e+6 (3.81e+5)
8	5.8154e+10 (5.90e+9)	6.2936e+10 (6.27e+9)	5.6974e+10 (6.01e+9)	5.8814e+10 (5.69e+9)
10	2.0532e+13 (1.69e+12)	2.1048e+13 (1.92e+12)	2.0971e+13 (2.65e+12)	2.2023e+13 (2.27e+12)
12	9.3812e+15 (1.03e+15)	9.4097e+15 (1.41e+15)	9.1856e+15 (8.75e+14)	8.9927e+15 (1.31e+15)
15	5.0115e+19 (9.16e+18)	5.3000e+19 (9.22e+18)	5.0767e+19 (7.79e+18)	5.7274e+19 (8.66e+18)
20	1.3819e+26 (1.94e+25)	1.3531e+26 (1.78e+25)	1.3166e+26 (1.92e+25)	1.4858e+26 (3.27e+25)
IGD				
5	3.7281e+1 (7.56e-1)	3.6820e+1 (9.61e-1)	3.6986e+1 (8.24e-1)	3.7083e+1 (7.14e-1)
8	4.3564e+1 (8.76e-1)	4.4318e+1 (8.08e-1)	4.3551e+1 (8.27e-1)	4.3541e+1 (9.80e-1)
10	4.7394e+1 (6.82e-1)	4.7525e+1 (9.39e-1)	4.7198e+1 (1.04e+0)	4.7637e+1 (7.22e-1)
12	5.1692e+1 (9.44e-1)	5.1742e+1 (8.38e-1)	5.1479e+1 (7.55e-1)	5.1462e+1 (7.94e-1)
15	5.7167e+1 (1.23e+0)	5.7396e+1 (1.42e+0)	5.6877e+1 (9.06e-1)	5.7805e+1 (8.39e-1)
20	6.5582e+1 (1.10e+0)	6.5664e+1 (1.06e+0)	6.5579e+1 (8.33e-1)	6.5562e+1 (1.03e+0)

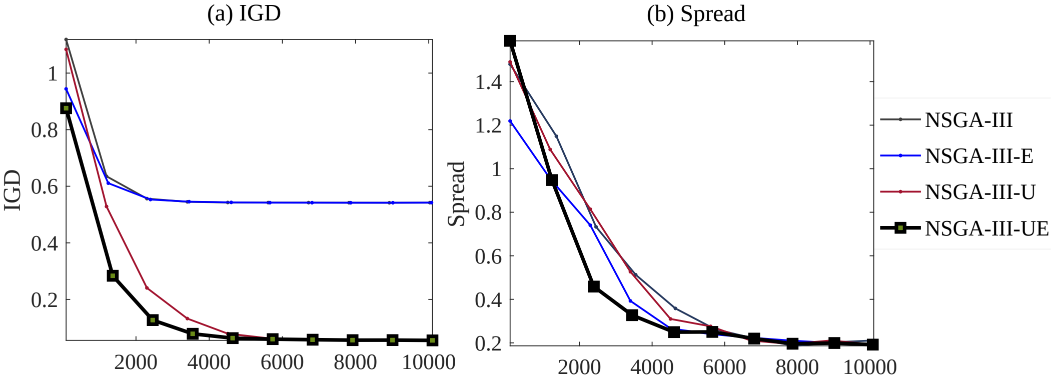


Figure 11: Comparison of IGD and Spread values of each algorithm on DTLZ4

Moreover, the application of the two-archive strategy will also be considered in the other algorithms for MaOPs.

6. Acknowledgment

We would like to acknowledge the support from the National Natural Science Foundation of China (61472095). Research Foundation of Education Department of Heilongjiang (1352MSYYB016,1352MSYYB005) and Research Foundation of Mudanjiang Normal University (GP2018003).

References

[1] A. Trivedi, D. Srinivasan, K. Sanyal, A. Ghosh, A survey of multiobjective evolutionary algorithms based on decomposition, IEEE Transactions on Evolutionary Computation 21 (3) (2017) 440–462.

[2] T. Wagner, N. Beume, B. Naujoks, Pareto-, aggregation-, and indicator-based methods in many-objective optimization, in: International Conference on Evolutionary Multi-Criterion Optimization, 2006, pp. 742–756.

[3] R. K. Ursem, P. D. Justesen, Multi-objective distinct candidates optimization: Locating a few highly different solutions in a circuit component sizing problem, Applied Soft Computing Journal 12 (1) (2012) 255–265.

[4] Y. Yuan, Y. S. Ong, A. Gupta, H. Xu, Objective reduction in many-objective optimization: Evolutionary multiobjective approaches and comprehensive analysis, IEEE Transactions on Evolutionary Computation 22 (2) (2018) 189–210.

[5] H. Wang, X. Yao, Objective reduction based on nonlinear correlation information entropy, Soft Computing 20 (6) (2016) 2393–2407.

[6] K. Narukawa, Y. Tanigaki, H. Ishibuchi, Evolutionary many-objective optimization using preference on hyperplane, in: International Conference on Genetic and Evolutionary Computation, 2014, pp. 91–92.

[7] Y. Sun, G. G. Yen, Z. Yi, Igd indicator-based evolutionary algorithm for many-objective optimization problems, IEEE Transactions on Evolutionary Computation 99 (1) (2018) 1–54.

[8] R. C. Purshouse, P. J. Fleming, On the evolutionary optimization of many conflicting objectives, IEEE Transactions on Evolutionary Computation 11 (6) (2007) 770–784.

[9] F. Li, R. Cheng, J. Liu, Y. Jin, A two-stage r2 indicator based evolutionary algorithm for many-objective optimization, Applied Soft Computing 67 (C) (2018) 245–260.

[10] E. Zitzler, S. Kunzli, Indicator-based selection in multiobjective search, in: International Conference on Parallel Problem Solving from Nature, 2004, pp. 832–842.

[11] J. Bader, E. Zitzler, Hype: An algorithm for fast hypervolume-based many-objective optimization, Evolutionary Computation 19 (1) (2014) 45–76.

[12] Q. Zhang, H. Li, Moea/d: A multiobjective evolutionary algorithm based on decomposition, IEEE Transactions on Evolutionary Computation 11 (6) (2007) 712–731.

[13] M. Asafuddoula, T. Ray, R. Sarker, A decomposition-based evolutionary algorithm for many objective optimization, IEEE Transactions on Evolutionary Computation 19 (3) (2015) 445–460.

[14] R. Cheng, Y. Jin, M. Olhofer, B. Sendhoff, A reference vector guided evolutionary algorithm for many-objective optimization, IEEE Trans. Evolutionary Computation 20 (5) (2016) 773–791.

[15] X. Cai, Z. Yang, Z. Fan, Q. Zhang, Decomposition-based-sorting and angle-based-selection for evolutionary multiobjective and many-objective optimization, IEEE Transactions on Cybernetics 47 (9) (2017) 2824–2837.

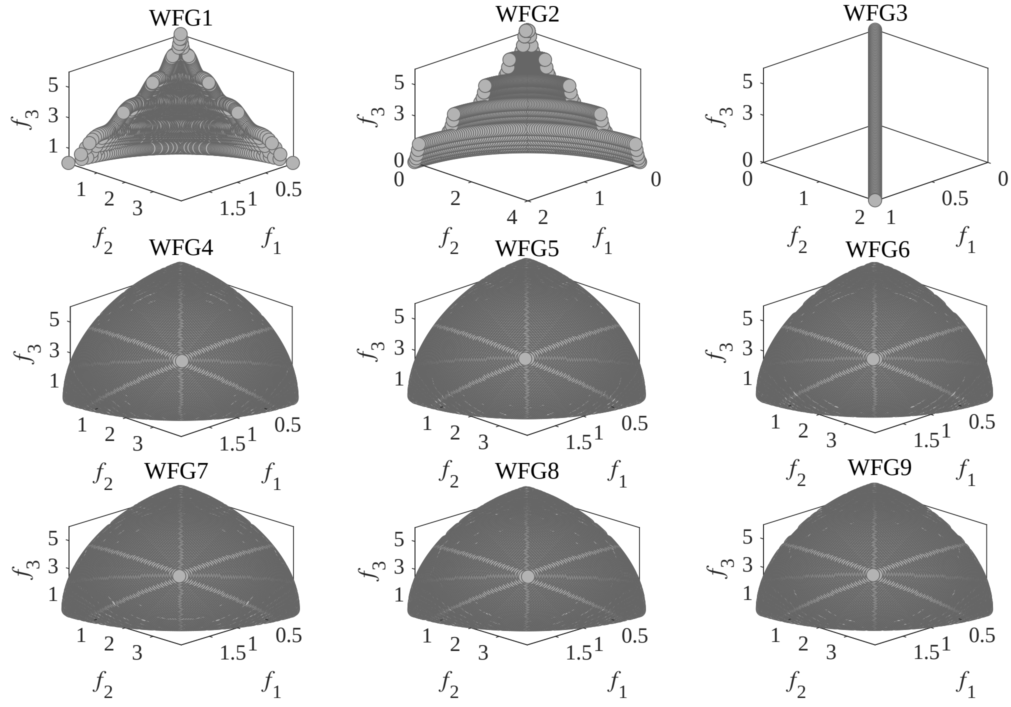


Figure 12: The real Pareto fronts of WFGs

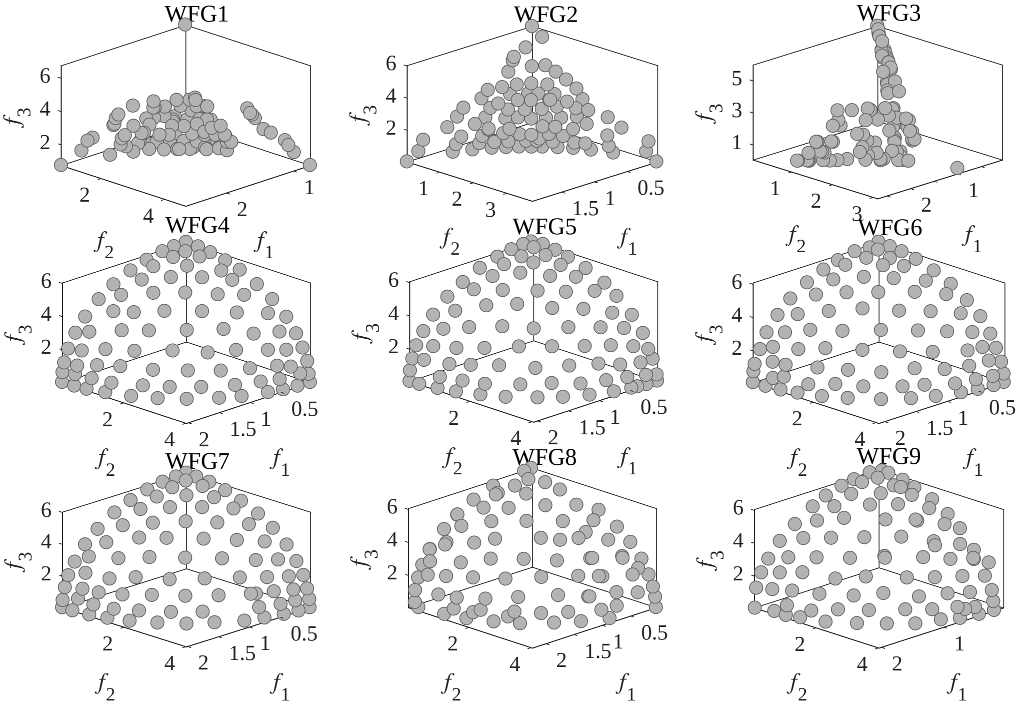


Figure 13: The approximating Pareto fronts of WFGs obtained by NSGA-III-UE

[16] C. Dai, X. Lei, A new evolutionary algorithm based on decomposition for multi-objective optimization problems, in: International Conference on Computational Intelligence and Security, CIS, 2016, pp. 33–38.

[17] L. Cai, S. Qu, Y. Yuan, X. Yao, A clustering-ranking method for many-objective optimization, Applied Soft Computing 35 (C) (2015) 681–694.

[18] H. L. Liu, L. Chen, Q. Zhang, K. Deb, Adaptively allocating search effort in challenging many-objective optimization problems, IEEE Transactions on Evolutionary Computation 22 (3) (2018) 433–448.

[19] K. Ikeda, H. Kita, S. Kobayashi, Failure of pareto-based moeas: does non-dominated really mean near to optimal, in: IEEE Congress on Evolutionary Computation, CEC, 2001, pp. 957–962.

[20] M. Laumanns, L. Thiele, K. Deb, E. Zitzler, Combining convergence and diversity in evolutionary multiobjective optimization, Evolutionary Computation 10 (3) (2014) 263–282.

[21] Y. Yuan, H. Xu, B. Wang, X. Yao, A new dominance relation-based evolutionary algorithm for many-objective optimization, IEEE Transactions on Evolutionary Computation 20 (1) (2016) 16–37.

[22] P. J. Bentley, J. P. Wakefield, Finding Acceptable Solutions in the Pareto-Optimal Range using Multiobjective Genetic Algorithms, Springer, London, 1998.

[23] Y. Yuan, H. Xu, B. Wang, Evolutionary many-objective optimization using ensemble fitness ranking, in: Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation, ACM, 2014, pp. 669–676.

[24] Y. Yuan, H. Xu, B. Wang, B. Zhang, X. Yao, Balancing convergence and diversity in decomposition-based many-objective optimizers, IEEE Transactions on Evolutionary Computation 20 (2) (2016) 180–198.

[25] S. Yang, M. Li, X. Liu, J. Zheng, A grid-based evolutionary algorithm for many-objective optimization, IEEE Transactions on Evolutionary Computation 17 (5) (2013) 721–736.

[26] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach, IEEE Transactions on Evolutionary Computation 3 (4) (1999) 257–271.

[27] E. Zitzler, M. Laumanns, L. Thiele, Spea2: Improving the performance of the strength pareto evolutionary algorithm, Parallel Problem Solving from Nature-PPSN VIII 3242 (4) (2004) 742–751.

[28] M. Li, S. Yang, X. Liu, Shift-based density estimation for pareto-based algorithms in many-objective optimization, IEEE Transactions on Evolutionary Computation 18 (3) (2014) 348–365.

[29] H. Wang, M. Olhofer, Y. Jin, A mini-review on preference modeling and articulation in multi-objective optimization: current status and challenges, Complex Intelligent Systems 3 (4) (2017) 233–245.

[30] Praditwong, Yao, Xin, A new multi-objective evolutionary optimisation algorithm: The two-archive algorithm, in: International Conference on Computational Intelligence and Security, 2006, pp. 286–291.

[31] B. Li, J. Li, K. Tang, X. Yao, An improved two archive algorithm for many-objective optimization, in: IEEE Congress on Evolutionary Computation, CEC, 2014, pp. 2869–2876.

[32] H. Wang, L. Jiao, X. Yao, Two-arch2: An improved two-archive algorithm for many-objective optimization, IEEE Transactions on Evolutionary Computation 19 (4) (2015) 524–541.

[33] R. Cheng, Y. C. Jin, M. Olhofer, B. Sendhoff, A reference vector guided evolutionary algorithm for many-objective optimization, IEEE Transactions on Evolutionary Computation 20 (5) (2016) 773–791.

[34] M. Zhang, H. Li, A reference direction and entropy based evolutionary algorithm for many-objective optimization, Applied Soft Computing 70 (1) (2018) 108–130.

[35] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints, IEEE Trans. Evolutionary Computation 18 (4) (2014) 577–601.

[36] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197.

[37] K. Li, K. Deb, Q. Zhang, S. Kwong, An evolutionary many-objective optimization algorithm based on dominance and decomposition, IEEE Transactions on Evolutionary Computation 19 (5) (2015) 694–716.

[38] A. Ibrahim, S. Rahnamayan, M. V. Martin, K. Deb, Elitensga-iii: An improved evolutionary many-objective optimization algorithm, in: International Conference on Evolutionary Computation, CEC, IEEE, 2016, pp. 973–982.

[39] B. Khan, M. Johnstone, S. Hanoun, C. P. Lim, D. Creighton, S. Nahavandi, Improved nsga-iii using neighborhood information and scalarization, in: IEEE International Conference on Systems, Man, and Cybernetics, SMC, 2017, pp. 3033–3038.

[40] X. Bi, C. Wang, A niche-elimination operation based nsga-iii algorithm for many-objective optimization, Applied Intelligence 48 (1) (2017) 118–141.

[41] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable test problems for evolutionary multiobjective optimization, Vol. 1, Springer, London, 2005.

[42] I. Das, J. E. Dennis, Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems, Siam Journal on Optimization 8 (3) (1998) 631–657.

[43] J. Zou, Y. Zhang, S. Yang, Y. Liu, J. Zheng, Adaptive neighborhood selection for many-objective optimization problems, Applied Soft Computing 64 (1) (2018) 186–198.

[44] W. Khan, Q. Zhang, Moea/d-dra with two crossover operators, in: International Conference on Computational Intelligence, UKCI, IEEE, 2010, pp. 1–6.

[45] S. Tsutsui, Multi-parent recombination in genetic algorithms with search space boundary extension by mirroring, in: International Conference on Parallel Problem Solving From Nature, Springer-Verlag, 1998, pp. 428–437.

[46] L. Cai, S. Qu, G. Cheng, Two-archive method for aggregation-based many-objective optimization, Information Sciences 422 (1) (2018) 305–317.

[47] C. Dai, Y. Wang, M. Ye, A new evolutionary algorithm based on contraction method for many-objective optimization problems, Applied Mathematics and Computation 245 (1) (2014) 191–205.

[48] S. Huband, P. Hingston, L. Barone, L. While, A review of multiobjective test problems and a scalable test problem toolkit, IEEE Transactions on Evolutionary Computation 10 (5) (2006) 477–506.

[49] H. Jain, K. Deb, An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: Handling constraints and extending to an adaptive approach, IEEE Trans. Evolutionary Computation 18 (4) (2014) 602–622.

[50] P. Hu, L. Rong, C. Liang-lin, L. Li-xian, Multiple swarms multi-objective particle swarm optimization based on decomposition, Procedia Engineering 15 (2) (2011) 3371–3375.

[51] Y. Tian, R. Cheng, X. Zhang, Y. Jin, Platemo: a matlab platform for evolutionary multi-objective optimization, IEEE Computational Intelligence Magazine 12 (4) (2017) 73–87.

[52] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, V. G. Da Fonseca, Performance assessment of multiobjective optimizers: An analysis and review, IEEE Transactions on Evolutionary Computation 7 (2) (2003) 117–132.

[53] A. Auger, J. Bader, D. Brockhoff, E. Zitzler, Theory of the hypervolume indicator: optimal u-distributions and the choice of the reference point, in: FOGA '09: Proceedings of the tenth ACM SIGEVO workshop on Foundations of genetic algorithms, 2009, pp. 87–102.

[54] E. Zitzler, L. Thiele, An evolutionary algorithm for multiobjective optimization: The strength pareto approach, Evolutionary Computation 3 (4) (1998) 257–271.

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

[56] S. Rostami, F. Neri, A fast hypervolume driven selection mechanism for many-objective optimization problems, Swarm Evolutionary Computation 34 (1) (2017) 50–67.

[57] M. Benedicks, On fourier transforms of functions supported on sets of finite lebesgue measure, Journal of Mathematical Analysis and Applications 106 (1) (1985) 180–183.

[58] L. Zhen, M. Li, C. Ran, D. Peng, Y. Xin, Adjusting parallel coordinates for investigating multi-objective search, in: Asia-Pacific Conference on Simulated Evolution and Learning SEAL 2017: Simulated Evolution and Learning, 2017, pp. 224–235.

[59] M. Li, L. Zhen, Y. Xin, How to read many-objective solution sets in parallel coordinates, IEEE Computational Intelligence Magazine 12 (4) (2017) 88–100.

[60] J. D. Knowles, D. W. Corne, Approximating the nondominated front using the pareto archived evolution strategy, Evolutionary Computation 8 (2) (2000) 149–172.