Adaptive Reference Point Generation for Many-Objective Optimization using NSGA-III

Atiya Masood, Gang Chen, Yi Mei, and Mengjie Zhang

Victoria University of Wellington, Wellington, New Zealand
{masood.atiy, aaron.chen, yi.mei, mengjie.zhang}@ecs.vuw.ac.nz

Abstract. In NSGA-III, the diversity of solutions is guided by a set of uniformly distributed reference points in the objective space. However, uniformly distributed reference points may not be efficient for problems with disconnected and non-uniform Pareto-fronts. These kinds of problems may have some reference points that are never associated with any of the Pareto-optimal solutions and will become useless reference points during evaluation. The existence of these useless reference points in NSGA-III significantly affects its performance. To address this issue, a new reference points adaptation mechanism is proposed that generates reference points according to the distribution of the candidate solutions. The use of this proposed adaptation method improves the performance of evolutionary search and promotes population diversity for better exploration.The proposed approach is evaluated on a number of unconstrained benchmark problems and is compared with NSGA-III and other reference point adaptation approaches. Experiment results on several benchmark problems clearly show a prominent improvement in the performance by using the proposed reference point adaptation mechanism in NSGA-III.

Keywords: Many-objective optimization. Genetic programming. Reference points. Evolutionary computation.

1 Introduction

NSGA-III [4] is one of the prominent and effective algorithms in the field of many-objective optimization. It is an extension of NSGA-II [5] which uses the widely distributed reference points for preserving diversity. Therefore, the obtained Pareto-optimal solutions are also likely to be widely distributed on the Pareto-optimal front. Previous studies have shown [4], [9] that NSGA-III performs better on 3 to 15 objectives of constrained and unconstrained optimization problems.

Even though NSGA-III has successfully solved various practical many-objective optimization problems, it still has challenges when applying the algorithm on real-world problems such as engineering problem. These real-world problems usually have non-uniform and irregular Pareto-fronts and the adoption of uniformly distributed reference points affect the performance of NSGA-III adversely [8], [9]. This is because many of these reference points are never associated with any of the optimal solutions and become useless reference points. Evidently, useless reference points will also notably affect the performance of NSGA-III [8], [9].

Particularly, in problems with irregular, non-uniform and disconnected Pareto-fronts, useful reference points are associated with more than one optimal solutions in their closest proximity. Selecting some of these popular reference points with a number of solutions may not help to span all solutions uniformly over the entire Pareto-fronts [9]. This may reduce the solution diversity of current and future population evolved by NSGA-III.

To address this key issue of useless reference points in NSGA-III, the main goal of this study is to develop a new *effective mechanism for reference point generation*. This mechanism will improve the association between reference points and the Pareto-fronts during evaluation. Further, a *proposed algorithm* will discover well-distributed solutions on the Pareto-optimal fronts. Guided by this goal, we will develop an adaptation mechanism by using a modelling technique and accurately approximates the Pareto-fronts based on evolved solutions. In particular, we introduce a *density-based model* that estimates the density of solutions from each defined sub-location in a whole objective space. Using distribution density information, we can further identify the distribution of candidate solutions in each generation and generate reference points in more promising regions. Furthermore, reference points in each partition are generated uniformly at that specific location. Therefore, associated solutions of these reference points are also well-distributed over the Pareto-fronts. Consequently, the proposed algorithm will decrease the existence of useless reference points for the close match between reference points and the evolved Pareto-front. Moreover, well distributed solutions over the entire Pareto-fronts will enhance the solution's diversity.

Driven by the goal of reducing the useless reference points and promoting the solution diversity, this paper is organized as follow. Section 2 presents the problem definition and related works in the literature for adaptive reference points approaches. Section 3 provides the technical description of the proposed algorithm. Section 4 outlines the experimental design and parameter setting. Section 5 analyses the experimental studies on very well known many-objective test problems and finally our conclusion in section 6.

2 Research Background

This section briefly introduces many-objective optimization problems and then discusses in more detail s several adaptive reference points approaches that have been proposed previously in the literature [8],[9].

2.1 Problem Definition

Without losing generality, Many-Objective Optimization Problems (MaOPs) involve four or more objectives [1] which often conflict with each other. In general, an MaOPs can be formulated as follows:

$$min \quad f\overrightarrow{(x)} = \{f_1(\overrightarrow{x}) \dots f_m(\overrightarrow{x})\} : s.t. \quad \overrightarrow{x} \in X \quad f \in Y \tag{1}$$

Given two solutions x_1 and x_2, it is said that x_1 *dominates* x_2 if and only if

$$\forall i, 1 \leq i \leq D, f_i(x_1) \leq f_i(x_2) : where D \geq 4$$

and

$$\exists i, f_i(x_1) < f_i(x_2).$$

Moreover, a solution x^* is said to be Pareto optimal if there does not exist another solution x_1 that dominates it.

2.2 Related Works

Several experimental and analytical studies [7],[11] have shown that Evolutionary Multi-Objective (EMO) algorithms were vulnerable when handling many-objective (four or more objective) problems due to the lack of adequate selection pressure toward the Pareto-fronts.

To cope with many-objective issues, reference points based approach is one of the state-of-the-art approaches that plays an important role for selecting well diversified solutions during evaluation [4],[10],[14]. These points are used to guide the solutions toward targeted locations. Therefore, the reference points based approach has been used in several EMO algorithms for handling many-objective optimization problems.

As an effective reference point based version of NSGA-II [5], NSGA-III [4] is one of the most effective many-objective optimization algorithm which works on uniformly distributed reference points. Although NSGA-III performs better on a number of problems with uniformly distributed Pareto-fronts such as DTLZ1 problem, uniformly distributed reference points NSGA-III has an issue when it is applied on non-uniform and irregular Pareto-front problems such as DTLZ7 problem. This limitation is also highlighted by Deb and Jain [9]. They have witnessed in several many-objective problems that some reference points can never be associated with a well-dispersed Pareto-optimal set while others are associated with more than one candidate solutions. Several adaptive extensions have been proposed [8],[15] in the literature for alleviating an issue of NSGAIII.

Reference Points based Evolutionary Algorithms for Many objective Optimization (REPA)[12] is one of the extension of NSGA-III which adaptively generates a series of reference points. These points are generated by adopting a series of local ideal points. Later individuals are selected by calculating the euclidean distance between the reference points and individuals in the environmental selection process.

ANSGA-III [9] is one of the well-known adaptive extension of NSGAIII. This extension of NSGA-III relocates the reference points adaptively. Further, relocation of the reference points adopt the distribution of candidate solutions on current generation.This relocation of reference points is carried out by two major operations: inclusion and exclusion. In the inclusion procedure, m-objective reference points are added around the j-th reference points in form of $m-1$ dimensional simplex. Moreover, the j-th reference points are kept as a centroid and the side length of the simplex is equal to the distance between two existing closest reference points. Unfortunately, this inclusion procedure requires adding the reference points outside the simplex, if new reference points introduces around the vertices of simplex. Due to this reason, ANSGA-III is not able to fully relocate the reference points and may fail to guide the evolution of a well-distributed set of Pareto-optimal solutions.

One of our earlier work, Density Model based Reference Point Adaptation (NSGA-III-DRA)[13] demonstrates the potential usefulness of the density model. In addition, this algorithm estimates the density of solutions in each sub location. NSGA-III-DRA generates reference points according to the average distance between selected solution and the centroid of all the existing solutions in the location. Random distribution of reference points does not allow to achieve an ideal association, thus the algorithm still has the issue of useless reference points.

Our proposed algorithm overcomes the limitations of NSGA-III and previously proposed adaptive approaches. Our proposed algorithm enables close match between reference points and the Pareto-front. In addition, our algorithm generates reference points that distribute Pareto optimal points uniformly across the entire Pareto front, thus alleviating the issue of randomness in NSGA-III-DRA. Moreover, our approach does not add any extra reference points during evolution and it is easy to implement regardless of the number of optimization objectives under consideration.

3 Proposed Algorithm

Our proposed adaptive algorithm is inspired by a density-based model that estimates the density of solutions at each sub location $\hat{w}$. Building this density-based probabilistic model consists of two steps. First, the whole objective space is decomposed into several sub-locations $\hat{w_1}, \hat{w_2}, \hat{w_3}....., \hat{w_k} \in W$. This decomposition uses Das and Dennis's [3] systematic approach. Then the number of the associated solutions with $\hat{w}$ is recorded in archive $E(\hat{w})$ where $E(\hat{w})$ preserves the index of associated individuals. The association between each solution $\hat{s}$ with $\hat{w}$ is obtained by a perpendicular distance ($\perp$). As a result, a solution is associated with a sub-location where the perpendicular distance between the two reaches the minimum. Lastly, solutions in $E(\hat{w})$ are divided by the total of the non-dominated solutions ($\parallel S \parallel$) so far. Then the algorithm calculates the density of solutions of each sub-simplex locations $\hat{w}$. The density-based probabilistic model is defined as

$$P(D|\hat{w} \in W) = \frac{\parallel \sum(argmin_{s \epsilon S} d^{\perp}(s,w)) \parallel}{\parallel S \parallel} \tag{2}$$

Previous efforts on improving the adaptiveness of reference points in NSGA-III focused mainly on adapting uniformly distributed reference points, guided implicitly by the distribution of solutions (i.e. no distribution models are explicitly constructed and utilized to adjust reference point locations). However, in our proposed algorithm, we emphasize clearly on the importance of using modelling techniques to obtain a more accurate approximation of the Pareto-fronts. Accordingly, our algorithm is capable of generating references points that matches closely with the distribution model. Furthermore, with the help of a new technique that generates reference points around the centroids of associated solutions, our algorithm can effectively handle solutions in close proximity to the simplex vertices. Additionally improvements have also been made to ensure even distribution of reference points around any solutions that fall well inside the simplex. Therefore, our algorithm has the ability to improve the diversity of solutions in NSGA-III.

3.1 Reference Point Adaptation

The basic framework of our proposed work is shown in **Algorithm 1**. In this framework, the density model is built first. This formation of the density model is shown in **Algorithm 2**. Next, a new adaptive procedure (see **line 15** of **Algorithm 1**) is introduced into **Algorithm 3**.

Algorithm 1: The framework of NSGA-III-DRAU.

Input : Parent population $\boldsymbol{P_g}$
Output: A set of non-dominated solutions
1 Initialize the population $\boldsymbol{P}_0$;
2 evaluate the population $\boldsymbol{P}_0$;
3 Generate the $\boldsymbol{W}$ that partition the Objective Space into sub-simplex locations;
4 Set $g \leftarrow 0$;
5 **while** $g < g_{\max}$ **do**
6   Generate the offspring population $\boldsymbol{Q}_g$ using the crossover, mutation and reproduction ;
7   **foreach** $Q \in Q_g$ **do** Evaluate Q;
8   $\boldsymbol{R}_g \leftarrow \boldsymbol{P}_g \cup \boldsymbol{Q}_g$;
9   Apply non-dominated sorting on (R_g) and find $(F_1, F_2 \dots)$;
10   Normalize the population members :$\overline{\boldsymbol{S_g}} = \boldsymbol{ObjectiveNormalization(S_g)}$;
11   **foreach** $w \in W$ **do**
12   identify member of $\overline{S_g}$ associated with w ;
13   Assign $\boldsymbol{(E(\hat{w}), D(\hat{w})) = Associatew(\overline{S_g}, W)}$;
14   **end**
15   Assign $\boldsymbol{Z_g^* = Generate(E(\hat{w}), D(\hat{w}), \overline{S}_g, W)}$;
16   Construct the new population P_{g+1} by the $NSGA-III$ association and Niching;
17   $g \leftarrow g + 1$;
18 **end**
19 **return** *The non-dominated individuals* $P^* \subseteq P_{g_{\max}}$;

Algorithm 2: $\boldsymbol{Associatew(\overline{S_g}, W)}$

Input : $\overline{S}_g, W$
Output: $\boldsymbol{E(\hat{w})}$ (individuals at $\hat{w}$) &$\boldsymbol{D(\hat{w})}$ (solution's density at $\hat{w}$)
foreach $w \in W$ **do**
  $E(w) = \phi$;
end
foreach $s \in \overline{S}_g$ **do**
  foreach $w \in W$ **do**
    compute $d^{\perp}(s, w)$; // perpendicular distance of each solution from $\hat{w}$
  end
  Assign $\hat{w} = argmin_{s \in S} d^{\perp}(s, w)$; // associate the solution with the sub-location
  Save s in $E(\hat{w})$;
end
foreach $s \in E(\hat{w})$ **do**
  Calculate the number of associated solutions with $\hat{w}$ and store in $A(\hat{w})$;
end
while $i \leq \| A(\hat{w}) \|$ **do**
  Assign $P(D|\hat{w}) = \| A(\hat{w}) \| \div \| S \|$; // probability of the associated solution
  Assign $D(\hat{w}) = \| P(\hat{w}) \|$*length of reference points; // return solution's density
  set i=i+1 ;
end
return $\boldsymbol{E}(\hat{w})$ &$\boldsymbol{D}(\hat{w})$;

Algorithm 3: $\boldsymbol{Generate(E(\hat{w}), D(\hat{w}), \overline{S}_g, W)}$

Input : $E(w), P(w), D(w), \overline{S_g}, W$
Output: $\boldsymbol{Z_g^*}$
foreach $\hat{w} \in W$ **do**
  set nref= $\| D(\hat{w}) \|$; `// number of reference points required at location` $\hat{w}$
  Assign $Z^r = \hat{w}$; `// set` $\hat{w}$ `as a first reference point`
  if Z^r *!=Vertex Points* **then**
    Assign $\boldsymbol{Z_g^*} = IntermediatePoints(E(\hat{w}), D(\hat{w}), nref, \overline{S}_g, W, Z^r)$; `// call intermediate points method`
  end
  if *Vertex points* **then**
    Assign $\boldsymbol{Z_g^*} = VertexPoints(E(\hat{w}), D(\hat{w}), nref, \overline{S}_g, W, Z^r)$; `// call vertex points method`
  end
end
return $\boldsymbol{Z_g^*}$;

3.2 Reference Point Generation

Our proposed algorithm is broken into two parts: (1) handling references points on the vertex and (2) dealing with the intermediate points.

References points on the vertex The first method of the proposed algorithm handles the issue of ANSGA-III. This issue relates to the generation of the reference points around the vertices of simplex. In this method, the reference points are generated from the centroid location of the associated solutions and these reference points are always generated inside a simplex location. In this procedure we have used the following steps:

1. Obtain the centre location from existing solutions in the sub-simplex $\hat{w} \in W$, where $\hat{w}$ is one of the vertices of the hyperplane.
2. Calculate the perpendicular distance from the centroid to associated solutions of $\hat{w}$.
3. Select a solution s based on a minimum perpendicular distance.
4. Calculate a mid-point value between the selected solution and the centroid for generating a corresponding reference point. This mid-point of each dimension is considered as one of the reference points around the vertices
5. Repeat steps 1 to 4 until the required number of reference points are generated.

Intermediate Points The generation of reference points at any intermediate location is described in **Algorithm 4**. Consider the situation in $M = 3$ objective case where M points are generated around any of the intermediate locations on the simplex. This example is shown in **Fig**. 1. In this example, $\{Z^1, Z^2, Z^3\}$ reference points are generated using the following two equations:

$$points^i = Z^r - (Interval)/M \quad (3)$$

$$Z^i_{new} = Z^i_{new}/div + points^i \quad (4)$$

where the interval is the difference between two consecutive reference points on the hyperplane and the division(div) is the total number of partitions on the original simplex.

Algorithm 4: $\boldsymbol{IntermediatePoints(E(\hat{w}), D(\hat{w}), nref, \overline{S}_g, W, Z^r)}$

```
Input  : E(ŵ), D(ŵ), nref, S̄_g, W, Z^r
Output: Z*_g
foreach ŵ ∈ W do
    associate(Z^r, s ∈ E(ŵ)) ;    // associate solutions with the reference point
    if ρ(Z^r) = 1 then
        Assign Z_g = Z^r : nref=nref-1;
    end
    while nref ≥ 0 do
        foreach z^r ∈ Z^r do
            if ρ(Z^r) ≥ 2 and Flag(Z^r) = 0 then
                while i ≤ M do
                    Z^r = Z^r − interval ÷ M ;
                    Z_new = Z_new ÷ div + Z_r ;    // generate new reference point
                    i=i+1;
                end
                while i ≤ M do
                    associate(Z^i_new, s ∈ E(ŵ)) ;    // associate the solutions with the
                     new reference point
                    if ρ(Z^i_new) ≠ 0 then
                        set Flag(Z_new)=0;
                        if already − exist(Z^i_new) = FALSE and Z^i_new lie in first quadrant
                         then
                            Assign Z^r = Z^i_new ∪ Z_r;
                        end
                    end
                    i=i+1;
                end
                foreach z^r ∈ Z^r do
                    if ρ(z^r) = 1 then
                        Z_g = z^r : set Flag(z_r)=1 ;
                        nref=nref-1;
                    end
                    if ρ(z^r) = 0 then
                        remove(z^r) ;    // remove reference point
                    end
                end
            end
        end
    end
end
return Z*_g;
```

These newly generated reference points can be inserted in the reference points archive called Z^r if they satisfy the two main conditions: (i) a reference point must be inside the boundary of entire simplex; (ii) duplication is not allowed and reference points must be unique. Once new reference points are added into archive Z^r, then the association between existing members of Z^r and solutions in $E(\hat{w})$ must be checked. If the i-th reference point from Z^r still has $\rho_i \geq 2$, reference points are generated around i-th reference points but this time a parameter value of interval is set to half of the current value and the division(div) is set to be double its existing value. This process is also shown in **Fig**. 1. **Fig**. 1 demonstrates that the i-th reference point is kept as a centroid location for newly generated reference points and the reference points are generated as a layer approach. These layers are also shows into the **Fig**. 1 with two different colours. Thus, we named this method a centroid layer approach.

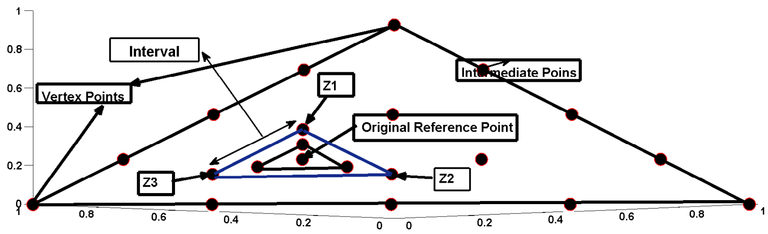


Fig. 1: Generate reference points until $M-1$ times

4 Experimental Setup

4.1 Test Problems

In order to verify the quality of the proposed algorithm, we have compared the performance of NSGA-III-DRAU with NSGA-III, ANSGA-III and NSGAIII-DRA on benchmark problems with three to eight objectives. We selected four many-objective test problems, DTLZ and Inverted DTLZ (IDTLZ), introduced by Deb et al.[6]. The characteristics of DTLZ and IDTLZ problems [2] are mentioned in **Table** 1.

Table 1: The Characteristics of DTLZ Problems

Problems	No. of Obj(m)	n	Characteristics
$IDTLZ1$	3,5,8	m+4	Linear, multi-model, inverted
$IDTLZ2$	3,5,8	m+9	Concave, inverted
$DTLZ5$	3,5,8	m+9	Concave, degenerate
$DTLZ7$	3,5,8	m+19	Mixed, disconnected, multi model

4.2 Parameter Setting

The number of decision variables for DTLZ and inverted DTLZ test problems are set as recommended in [6]. The population size of all compared algorithms are set to 92 for the three-objective, 212 for the five-objective and 156 for eight-objective. The size of reference points are also kept same as the population size. 91 reference are supplied to all compared algorithms for three-objective case, 210 for five-objective case and 156 for eight-objective case. The crossover and the mutation parameters of NSGA-III are kept identical in the proposed algorithm. In order to maintain a consistent and fair comparison the parameter settings of compared algorithms are kept the same in all experiments

4.3 Performance Measures

To evaluate the performance of the all proposed algorithm on DTLZ problems, we used the Inverted Generational Distance (IGD) [16] and Hyper-Volume (HV) [17]. These two indicators have been commonly used to evaluate the performance of EMO algorithms. In this study, the exact Pareto-optimal surface of DTLZ test problems are known. Therefore, we use the true Pareto-fronts for calculating IGD. In the case of HV the nadir point is set as (1, 1, 1,...1). The HV values in this study are normalized to [0,1].

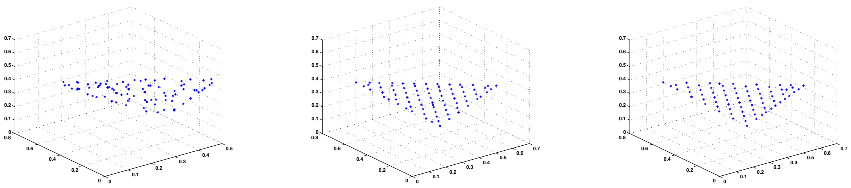
5 Results and Discussions

In the experiment, for each algorithm, 30 independent runs are carried out. Then, the mean and the standard deviation of HV and IGD values are reported. The best value for each problem is marked in boldface.

5.1 Overall Results

Table 2 presents the mean and standard deviation of the four compared algorithms on DTLZ problems. The Wilcoxon rank sum test with the significance level of 0.05 is carried out on both HV and IGD values.

IDTLZ1 fitness landscape contains a large number of local optima which may require better exploration. Therefore, a higher degree of population diversity plays an important role for more exploration in this multi-model test problem. **Table** 2 shows that adaptively relocating reference points NSGAIII-DRA, NSGAIII-DRAU and ANSGA-III have better HV and IGD values because they can generate higher population diversity than the predefined uniformly distributed reference points in NSGA-III. Furthermore, the result also reveals that NSGAIII-DRAU performs significantly better than NSGA-III and NSGAIII-DRA but is competitive with ANSGA-III. This can also be seen in **Fig**. 2. **Fig**. 2 also demonstrates that the NSGAIII-DRA has random distribution of solutions and some area of the plane do not have any of the solution. Thus, the NSGAIII-DRA plot indicates that reference points are not widely distributed in the objective space. vspace-3mm



(a) $NSGA-III-DRA$ (b) $NSGA-III-DRAU$ (c) $ANSGA-III$

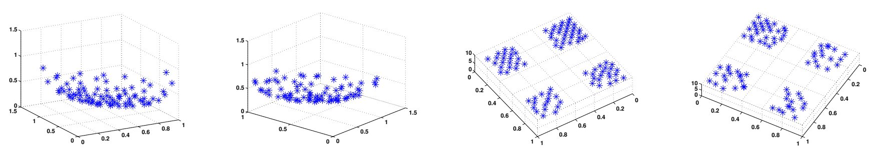
Fig. 2: Approximate Pareto Front for 3-objective Inv-IDTLZ1 problem

For the IDTLZ2 problem, **Table** 2 shows that our proposed algorithm NSGAIII-DRAU significantly outperformed NSGAIII, ANSGA-III and NSGAIII-DRA in terms of HV and IGD. To verify this result, we plotted the Pareto-fronts of our proposed algorithm and ANSGA-III. **Fig**. 3a and 3b show that the proposed algorithm has generated more diversified solutions on the hyperplane than ANSGA-III for this problem.

DTLZ5 has a degenerated Pareto-front,i.e., the Pareto-front is always a curve regardless of the dimensionality of the objective space. For DTLZ5 problem,

Table 2: The mean and standard deviation over the average HV values on M-objectives inverted $DTLZ1$, inverted $DTLZ2$, $DTLZ5$ and $DTLZ7$ problems. The significantly better results are shown in bold.

Function	M	NSGAIII	ANSGA-III	NSGAIII-DRA	NSGAIII-DRAU
HV $Mean(std)$					
Inv-DTLZ1	3	1.07e-1(4.0e-3)	1.30e-1(2.8e-3)	1.19e-1(4.7e-3)	1.31e-1(2.0e-3)
	5	7.91e-4(4.2e-4)	2.05e-3(3.0e-4)	2.75e-3(2.9e-4)	**3.62e-3(2.0e-4)**
	8	2.62e-4(3.8e-5)	**9.10e-3(1.5e-3)**	1.35e-4(4.3e-5)	1.58e-4(4.3e-4)
Inv-DTLZ2	3	4.14e-1(1.7e-2)	4.47e-1(4.0e-3)	4.35e-1(6.74e-3)	**6.74e-1(4.1e-3)**
	5	6.30e-2(2.4e-3)	6.89e-2(5.9e-3)	8.53e-2(3.1e-3)	**2.85e-1(1.1e-2)**
	8	6.62e-3(6.8e-4)	9.37e-3(1.1e-4)	7.92e-3(8.8e-4)	**1.25e-2(7.9e-4)**
DTLZ5	3	8.19e-2(1.7e-2)	**8.54e-2(6.3e-4)**	8.41e-2(1.2e-3)	8.45e-2(2.3e-3)
	5	2.28e-1(2.6e-1)	7.12e-1(4.2e-1)	4.5e-1(4.7e-2)	**7.21e-1(3.5e-2)**
	8	6.03e-1(1.9e-2)	6.92e-1(2.2e-2)	6.44e-1(5.4e-2)	**6.96e-1(2.2e-2)**
DTLZ7	3	3.10e-1(7.9e-3)	3.15e-1(1.4e-2)	2.99e-1(5.5e-3)	**3.19e-1(1.2e-2)**
	5	2.240e-1(3.8e-3)	3.23e-1(6.3e-3)	2.75e-2(2.7e-3)	**3.25-1(4.3e-3)**
	8	3.08e-1(4.8e-3)	3.25e-1(6.3e-3)	2.23e-3(1.7e-2)	**3.28e-1(6.2e-3)**
IGD $Mean(std)$					
Inv-DTLZ1	3	3.22e-2(5.0e-3)	2.37e-2(7.7e-3)	2.82e-2(2.5e-3)	2.34e-2(1.8e-3)
	5	8.62e-2(3.8e-3)	5.53e-2(8.3e-3)	4.47e-2(6.8e-3)	**3.11e-2(3.9e-3)**
	8	9.62e-2(8.8e-3)	7.13e-2(9.2e-3)	6.17e-2(9.9e-3)	**5.49e-2(8.9e-3)**
Inv-DTLZ2	3	6.80e-2(8.3e-3)	6.39e-2(4.6e-3)	6.38e-2(4.2e-3)	**6.19e-2(4.6e-3)**
	5	2.33e-1(1.2e-2)	2.03e-1(1.2e-2)	1.61e-1(1.2e-2)	**1.23e-1(1.4e-2)**
	8	2.62e-1(3.8e-2)	3.53e-1(2.3e-2)	2.78e-1(3.5e-2)	**2.39e-1(3.4e-2)**
DTLZ5	3	2.18e-2(2.5e-3)	**1.35e-2(1.5e-3)**	1.99e-2(3.7e-3)	1.39e-2(2.84e-3)
	5	2.70e-1(5.16e-2)	1.99e-1(5.7e-2)	3.57e-1(8.4e-2)	**1.85e-1(5.4e-2)**
	8	4.04e-1(8.8-e-2)	4.01e-1(7.2e-2)	4.57e-1(8.9e-2)	**3.95e-1(9.14e-2)**
DTLZ7	3	9.6e-2(5.07e-3)	8.60e-2(6.8e-3)	9.57e-2(1.6e-3)	**8.37e-2(6.43e-3)**
	5	4.54e-1(2.2e-2)	**3.44e-1(2.4e-2)**	3.68e-1(2.7e-2)	3.60e-1(2.6e-2)
	8	7.89e-1(3.7e-2)	7.61e-1(2.7e-2)	8.83e-1(5.5e-2)	**7.54e-1(7.9e-2)**



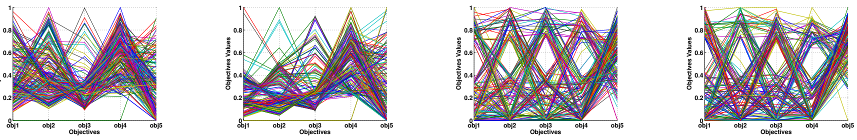
(a) $NSGA-III-DRAU$ on IDTLZ2 (b) $ANSGA-III$ on IDTLZ2 (c) $NSGA-III-DRAU$ on DTLZ7 (d) $ANSGA-III$ on DTLZ7

Fig. 3: Approximate Pareto Front for 3-objective Inv-IDTLZ2 and DTLZ7 test problems

Fig.4a and 4b show that the solutions obtained by NSGAIII-DRAU are well distributed around Pareto-fronts, thus achieving better diversity than ANSGA-III. For the five-and-eight objective test problems, **Table** 2 shows that NSGAIII-DRAU significantly outperforms NSGAIII, ANSGA-III and NSGAIII-DRA.

Similar observations can be made from the results on DTLZ7 with three to eight objectives as well. DTLZ7 has a disconnected Pareto-front. Due to this feature, some algorithms that rely on uniformly distributed reference points cannot perform well on this problem. Hence, the algorithms having adaptive reference points eventually have significantly better performance. **Fig**. 4c and 4d show that NSGAIII-DRAU can converge faster than ANSGA-III. For the three-objective test problem, ANSGA-III apparently failed to cover some location on the Pareto-fronts. This can be seen in **Fig**. 3c and 3d

For the eight-objective test problems, **Table** 2 shows that NSGAIII-DRAU significantly outperforms NSGAIII, ANSGA-III and NSGAIII-DRA on most of the test problems.



(a) $NSGA-III-DRAU$ on 1DTLZ5 (b) $ANSGA-III$ on DTLZ5 (c) $NSGA-III-DRAU$ on DTLZ7 (d) $ANSGA-III$ on DTLZ7

Fig. 4: Parallel coordinate plot for the e fitness values of the population on 5-objective DTLZ5 and DTLZ7 problems.

6 Conclusion

In this paper, we proposed a new adaptive generation strategy NSGA-III-DRAU for reference points in the recently proposed many-objective NSGA-III. NSGA-III-DRAU addresses a key research issue of using uniformly distributed reference points NSGA-III on many-objective irregular and disconnected Pareto front problems and attempted to alleviates the limitations of recently proposed adaptive approaches. The proposed algorithm is applied to a number of unconstrained three to eight-objective optimization problems. We compared our proposed algorithm with NSGA-III and previously proposed reference points adaptive approaches. Experimental results on the benchmark problems show that NSGA-III-DRAU reduces the useless reference points and provides a better distribution of Pareto optimal solutions on the entire Pareto fronts. Further, a better distribution of reference points also helps improve the diversity of the solutions that can be observed visually and in terms of HV and IGD. This finding leads us believe that our algorithm NSGA-III-DRAU is capable of handling many-objective problems with non-uniformly distributed Pareto front effectively.

This study opens up a substantial research direction for many-objective optimization problems. It is still in exploration phase and more studies are required in future. Thus, we have a plan to do more analytical and experimental studies to know in detail about the behavior of the solutions in term of non-uniform and irregular Pareto fronts.

References

1. Chand, S., Wagner, M.: Evolutionary many-objective optimization: A quick-start guide. Surveys in Operations Research and Management Science 20(2), 35–42 (2015)
2. Cheng, R., Li, M., Tian, Y., Zhang, X., Yang, S., Jin, Y., Yao, X.: A benchmark test suite for evolutionary many-objective optimization. Complex & Intelligent Systems 3(1), 67–81 (2017)

3. Das, I., Dennis, J.E.: Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. SIAM Journal on Optimization 8(3), 631–657 (1998)
4. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. Evolutionary Computation, IEEE Transactions on 18(4), 577–601 (2014)
5. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6, 182–197 (2002)
6. Deb, K., Thiele, L., Zitzler, E.: Scalable multi-objectove optimization test problems. In: IEEE Congress on Evolutionary Computation. pp. 825–830. IEEE (2002)
7. Jaimes, A.L., Coello, C.A.C.: Many-Objective Problems: Challenges and Methods. In: Kacprzyk, J., Pedrycz, W. (eds.) Handbook of Computational Intelligence, pp. 1033–1046. Springer (2015)
8. Jain, H., Deb, K.: An Improved Adaptive Approach for Elitist Nondominated Sorting Genetic Algorithm for Many-Objective Optimization. In: Purshouse, R.C., Fleming, P.J., Fonseca, C.M., Greco, S., Shaw, J. (eds.) EMO. Lecture Notes in Computer Science, vol. 7811, pp. 307–321. Springer (2013)
9. Jain, H., Deb, K.: 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), 602–622 (2014)
10. Jiang, S., Yang, S.: A strength Pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization. IEEE Transactions on Evolutionary Computation 21(3), 329–346 (2017)
11. Li, B., Li, J., Tang, K., Yao, X.: Many-objective evolutionary algorithms: A survey. ACM Computing Surveys (CSUR) 48(1), 13 (2015)
12. Liu, Y., Gong, D., Sun, X., Zhang, Y.: A reference points-based evolutionary algorithm for many-objective optimization. In: Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation. pp. 1053–1056. ACM (2014)
13. Masood, A., Chen, G., Mei, Y., Zhang, M.: Reference Point Adaption Method for Genetic Programming Hyper-Heuristic in Many-Objective Job Shop Scheduling. Springer (2018)
14. Masood, A., Mei, Y., Chen, G., Zhang, M.: Many-Objective Genetic Programming for Job-Shop Scheduling. IEEE WCCI 2016 conference proceedings, IEEE (2016)
15. Masood, A., Mei, Y., Chen, G., Zhang, M.: A PSO-Based Reference Point Adaption Method for Genetic Programming Hyper-Heuristic in Many-Objective Job Shop Scheduling. In: Wagner, M., Li, X., Hendtlass, T. (eds.) ACALCI. Lecture Notes in Computer Science, vol. 10142, pp. 326–338 (2017)
16. Zhang, Q., Zhou, A., Zhao, S., Suganthan, P.N., Liu, W., Tiwari, S.: Multiobjective optimization test instances for the CEC 2009 special session and competition. University of Essex, Colchester, UK and Nanyang technological University, Singapore, special session on performance assessment of multi-objective optimization algorithms, technical report pp. 1–30 (2008)
17. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Da Fonseca, V.G.: Performance assessment of multiobjective optimizers: an analysis and review. Evolutionary Computation, IEEE Transactions on 7(2), 117–132 (2003)