An Improved Two Archive Algorithm for Many-Objective Optimization

Bingdong Li[*], Jinlong Li[*], Ke Tang[*], Xin Yao[†*]

[*]School of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, China
[†]School of Computer Science, the University of Birmingham, Birmingham B15 2TT, U.K.
Emails: lbingd17@mail.ustc.edu.cn, jlli@ustc.edu.cn, ketang@ustc.edu.cn, x.yao@cs.bham.ac.uk

Abstract—**Multi-Objective Evolutionary Algorithms have been deeply studied in the research community and widely used in the real-world applications. However, the performance of traditional Pareto-based MOEAs, such as NSGA-II and SPEA2, may deteriorate when tackling Many-Objective Problems, which refer to the problems with at least four objectives. The main cause for the degradation lies in that the high-proportional non-dominated solutions severely weaken the differentiation ability of Pareto-dominance. This may lead to stagnation. The Two Archive Algorithm (TAA) uses two archives, namely Convergence Archive (CA) and Diversity Archive (DA) as non-dominated solution repositories, focusing on convergence and diversity respectively. However, as the objective dimension increases, the size of CA increases enormously, leaving little space for DA. Besides, the update rate of CA is quite low, which causes severe problems for TAA to drive forth. Moreover, since TAA prefers DA members that are far away from CA, DA might drag the population backwards. In order to deal with these weaknesses, this paper proposes an improved version of TAA, namely ITAA. Compared to TAA, ITAA incorporates a ranking mechanism for updating CA which enables truncating CA while CA overflows. Besides, a shifted density estimation technique is embedded to replace the old ranking method in DA. The efficiency of ITAA is demonstrated by the experimental studies on benchmark problems with up to 20 objectives.**

Keywords—*Many-objective, Multi-objective, evolutionary algorithm, archive method.*

I. INTRODUCTION

In the real world applications, we are often faced with Multi-Objective Problems (MOPs). MOPs with at least four objectives are referred to as Many-Objective Problems (ManyOPs) [1]. ManyOPs appear widely in many real-world applications, such as engineering design, air traffic controlling, nurse rostering, and water supply management. Recently, a series of algorithms have been proposed to solve ManyOPs in the evolutionary multi-objective optimization (EMO) community.

A Multi-Objective Problem (MOP) can be stated as follows:

$$minimize\ \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}))^T$$
$$subject\ to\ \mathbf{x} \in \Omega \qquad (1)$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ is the decision vector, Ω is the nonempty *decision space*, the objective function vector is denoted as $\mathbf{F}: \Omega \rightarrow \Lambda$ with m ($m \geq 2$) objectives, and Λ is the *objective space* [2]. Without loss of generality, we consider minimization problems in this paper, where all the objectives are supposed to be minimized. The dominance preference relation is a critical concept in Multi-Objective Evolutionary Algorithms (MOEAs). Given two objective vectors $\mathbf{F}(\mathbf{x})$, $\mathbf{F}(\mathbf{y}) \in R^m$, $\mathbf{F}(\mathbf{x})$ is said to *dominate* $\mathbf{F}(\mathbf{y})$ if and only if $\forall i \in \{1, 2, ..., m\}, f_i(\mathbf{x}) \leq f_i(\mathbf{y})$ and $\exists j \in \{1, 2, ..., m\}, f_j(\mathbf{x}) < f_j(\mathbf{y})$ [3]. A decision vector $\mathbf{x}^* \in \Omega$ is *Pareto optimal* if there exist no other feasible solutions that dominate it [3]. The set of Pareto optimal solutions is called *Pareto Set* (PS) and the corresponding objective vector set is the *Pareto Front* (PF).

Due to their population-based property and robust search power for complex problems, MOEAs have been recognized to be well suitable for tackling MOPs [4]. Ever since the 1990s, plenty of MOEAs have been proposed which show promising performance when tackling bi-objective or three-objective problems. However, it has been shown that traditional Pareto-based approaches, such as NSGA-II [5] and SPEA2 [6], may deteriorate significantly for ManyOPs [7]. The main reason for this phenomenon is that the proportion of globally non-dominated solutions in a randomly generated population rises enormously as the dimension of objective space increases. Thus the dominance level based primary selection criterion fails to discriminate solutions in most cases and the secondary selection criterion aiming at improving diversity determines the environmental selection, which is referred to as active diversity promotion [8]. As a result, the convergence of the Pareto-based MOEAs on ManyOPs is severely deteriorated and the final solution set might not even converge to PF but stagnate far away from it [9].

Intuitively, there are two avenues to improve the performance of Pareto-based MOEAs on ManyOPs, i.e. modifying the dominance-based primary criterion to enhance the selection pressure towards the PF and improving the diversity-oriented secondary criterion to weaken the detrimental effect of active diversity promotion [10]. On one hand, a series of relaxed dominance definitions have been proposed to improve the selection pressure, such as CADS [11], and grid-dominance [4]. These domination forms are more tolerant, thus the solutions are less likely to be non-dominated. As a result, the population are pushed forward instead of stagnating far away from the PF. On the other hand, modifying the diversity management mechanism seems to be another way to deal with this issue. In [12], Adra et al. introduced a diversity management mechanism to NSGA-II which named DM1. DM1 determines whether the algorithm activates diversity promotion according to the spread of the population. Yang et al. proposed a Grid-based Evolutionary Algorithm (GrEA) to solve ManyOPs [4]. In

GrEA, based on the adaptive constructed grids, the density of solutions are measured by the grid crowding distance and grid coordinate point distance. In [13], Deb et al. replaced the crowding distance with a reference set based mechanism to measure the density of solutions, which showed quite promising performance.

Aggregation-based algorithms are competitive alternatives non-Pareto-based MOEAs for tackling ManyOPs. One of the most popular aggregation-based algorithms is MOEA/D proposed by Zhang and Li [3]. When dealing with a ManyOP, they optimize a series of single-objective sub-problems instead of directly working on the original problem. Due to its problem decomposition strategy and solution sharing mechanism for neighboring sub-problems, MOEA/D shows competitive, if not better, convergence and diversity performance compared with other state-of-the-art algorithms.

The indicator-based methods, on the other hand, use the values of metrics, such as hypervolume and approximation indicator, to guide the search process [14], [15], [16]. In [14], Emmerich et al. proposed a steady-state algorithm named SMS-EMOA based on the gradient of the 1-D hypervolume metric. However, the computational cost of the hypervolume indicator is so expensive that the applications of SMS-EMOA on problems with more than 10 objectives is limited [16]. Bader and Zitzler applied Monte Carlo simulation to approximate the exact hypervolume values, allowing a tradeoff between the accuracy of the estimates and the available computing resources [15]. In [17], Bringmann et al. defined an additive approximation indicator and proposed an algorithm based on the indicator. Compared to the hypervolume indicator, it costs much less computational time.

Other techniques, such as user preference incorporation and dimensionality reduction, are also adopted to MOEAs to deal with ManyOPs [18], [19]. Dimensionality reduction algorithms try to reduce less necessary objectives to reduce computational load, while preference incorporation methods use the decision makers' preference information to bias the population towards a sub-region of the PF.

The Two Archive Algorithm (TAA) uses quite different mechanism for environmental selection [20]. In TAA, non-dominated solutions of each generation are selected and separated into two archives namely Convergence Archive (CA) and Diversity Archive (DA) focusing on convergence and diversity respectively. However, as the objective dimension increases, the size of CA increases enormously, leaving little space for DA. Besides, the update rate of CA is quite low, which might slow down the convergence of TAA. Moreover, since TAA prefers the DA members which far away from CA, DA might drag the population backwards.

In this paper, we proposed an improved version of TAA, named ITAA, where a Penalty-based Boundary Intersection (PBI) [3] method is incorporated into CA truncation. PBI method enables the algorithm to truncate CA when the size of CA is too big. Besides, ITAA replaces the old ranking mechanism of DA with a recent proposed Shift-based Density Estimation (SDE) [10] method to improve the convergence of DA and prevent the DA from dragging the population backwards from the PF. The experimental results on benchmark functions with up to 20 objectives demonstrated that ITAA outperformed TAA in terms of both convergence and diversity in most cases.

The remaining part of the paper is organized as follows. In section II, we will give a detailed introduction of the two archive mechanism. Section III is devoted to discuss the Improved Two Archive Algorithm (ITAA). The empirical evaluation of ITAA, with a detailed description of experimental settings and results included, are presented in Section IV. The last section summarizes the paper and discusses some future directions for further investigation.

II. TWO ARCHIVE MECHANISM

Generally speaking, there are two goals for MOEAs: minimizing the distance of the solutions to the true PF and improving the distribution quality of the solutions, aka convergence and diversity. However, it is often the case that these two goals are contradictory and finding solutions with both good convergence and good diversity might be hard. Besides, when tackling ManyOPs, MOEAs are faced with a large proportion of non-dominated solution.

Praditwong and Yao [20] proposed an effective way to provide a coarse-grained discrimination for the non-dominated solutions by maintaining two archives: the Convergence Archive (CA) and the Diversity Archive (DA). During the searching process, these two archives focus on convergence and diversity respectively. Only the non-dominated population members which are not dominated by any archive members are qualified to enter the archives. In order to determine whether they are admitted into CA or DA, the non-dominated solutions are tested if they can dominate archive members. The solutions which can dominate at least one existing archive member are deemed to show more convergence ability, thus they are added into CA. These solutions are addressed as non-dominated solutions with domination. Meanwhile, the dominated archive members are discarded from the archives. The rest non-dominated solutions, which are called non-dominated solutions without domination, are added into DA instead. If the cardinality of the union of CA and DA (denoted as UA, hereinafter) is over the threshold, DA members with the minimum distance to CA are discarded iteratively, aiming at obtaining a diverse solution set. The experimental results have demonstrated that the Two Archive Algorithm outperformed PESA [21] on DTLZ benchmark functions with up to 8 objectives [20].

Although some promising results are obtained in [20], there do exist some problems in TAA. First, as the number of objective increases, the proportion of non-dominated solutions increases enormously. However, CA members won't be updated (replaced) unless dominated by new solutions. In that case, the size of CA accounts for a large proportion of UA (roughly larger than 90% in many cases), leaving little space for DA. Moreover, the update rate of CA will remain quite low, which may slow down the convergence speed of the algorithm. Second, the diversity maintenance strategy of DA is also questionable. Based on the Euclidean distance, the DA truncation mechanism prefers the solutions which are far away from CA, thus DA may fall far behind the population and drag UA backwards.

III. The Improved Two Archive Algorithm

The main loop of ITAA is shown in Algorithm 1. As shown in the pseudo-code, it begins with a randomly initialized population. At each generation, the archives are updated after evaluating the population. There are two phases for the updating: adding non-dominated solutions into the archives and truncating archives (line 5 and 6 in Algorithm 1). In general, ITAA remains the same with TAA except for the truncation phase (line 6 in Algorithm 1).

Algorithm 1 The Improved Two Archive Algorithm (ITAA): Main Loop
---
1: Initialize the population
2: Evaluate initial population
3: Set $CA \leftarrow \varnothing, DA \leftarrow \varnothing, t \leftarrow 0$
4: **while** $t \leq$ MaxGen **do**
5:     Adding non-dominated solutions into the archives
6:     Truncating archives
7:     Mating selection
8:     Apply genetic operators to generate a new population
9:     Evaluate the new generation
10:     $t \leftarrow t + 1$
11: **end while**

In order to deal with the CA size issue, an intuitive way is to set a threshold for CA. When CA overflows, a ranking mechanism is carried out and the redundant members are discarded from CA according to the ranking mechanism. The ranking mechanism should be able to keep a good convergence of CA and provide a high updating rate for CA. In this paper, we used a Penalty-based Boundary Intersection (PBI) [3] to rank the CA members. Besides, We replaces the old ranking mechanism with a recent proposed Shift-based Density Estimation (SDE) [10] method. SDE is better than the original Euclidean distance measure in that it introduces some convergence information into the density estimation mechanism. Thus the convergence of DA will be improved and DA won't drag the population backwards from the PF. Also, it should be noted that modifying DA truncation without changing the maintenance strategy of CA might obtain limited performance improvement since there is a good chance that CA will still occupy a large proportion of UA. So we consider modifying CA updating strategy first.

A. PBI-based CA Truncation

In order to rank CA members, we adopt PBI from literature [3] to evaluate the convergence quality of each solution. If CA overflows, a fitness assignment procedure is conducted for all CA members. The CA member with the worst fitness value is discarded iteratively until CA meets the capacity constraint. First, a reference direction is defined by a unit weighted vector $\mathbf{w}_k = \{w_k^1, w_k^2, \ldots, w_k^m\}, \mathbf{w}_k \in \mathbf{W}, k = 1, 2, \ldots, |\mathbf{W}|$. In order to keep a fair search strength among all directions, a global counter gc is maintained. At each generation, gc is updated by $gc = (gc + 1)mod|\mathbf{W}| + 1$. The PBI fitness of a CA member according to the reference direction of $\mathbf{w}_k$ is the weighted sum of the distance ($d1$) along the reference direction and the perpendicular distance ($d2$) from the reference direction, which is stated as follows:

$$PBI(\mathbf{F}(\mathbf{x}), \mathbf{w}_k) = d1 + \theta d2 \qquad (2)$$

where $d1 = \mathbf{w}_k^T \cdot \mathbf{F}(\mathbf{x})$, $d2 = ||\mathbf{F}(\mathbf{x}) - \mathbf{w}_k^T \cdot \mathbf{F}(\mathbf{x}) \cdot \mathbf{w}_k||$

PBI enables a selection among CA members so that the size of CA can be controlled. Besides, the convergence of MOEA/D-PBI [1] has implied that PBI is able to select solutions which is better in terms of convergence. Thus although the size of CA in ITAA is smaller than TAA in most cases, the convergence of CA in ITAA won't be diminished.

Algorithm 2 Truncating CA in ITAA

1: $gc = (gc + 1)mod|\mathbf{W}| + 1$
2: **if** sizeof(CA) $> limitCA$ **then**
3:     **for all** $i = 1$ to sizeof(CA) **do**
4:         Compute $PBI(CA(i), \mathbf{w}_{gc})$ according to Eqn. (2)
5:     **end for**
6:     **while** sizeof(CA) $> limitCA$ **do**
7:         Remove the CA member with the worst PBI value
8:     **end while**
9: **end if**

B. SDE-based DA Truncation

After truncating CA, if UA still overflows, we will need to truncate DA too. The truncating procedure is shown in Algorithm 3. In order to rank DA members, TAA calculates their distances to CA. The redundant DA members with shortest distances to CA are removed. When measuring the similarity between two points in the objective space $\mathbf{p} = (p_1, p_2, \ldots, p_m)$ and $\mathbf{q} = (q_1, q_2, \ldots, q_m)$, instead of using Euclidean distance $Ed(\mathbf{p}, \mathbf{q})$, we use the Shifted Density Estimation (SDE) method [10] which is defined as follows:

$$SDE(\mathbf{p}, \mathbf{q}) = Ed(\mathbf{p}, \mathbf{q}')) \qquad (3)$$

where the shift version of $\mathbf{q}' = (q_1', q_2', \ldots, q_m')$ is defined as:

$$q_i' = \begin{cases} p_i & \text{if } q_i < p_i \\ q_i & \text{otherwise} \end{cases} \quad i \in \{1, 2, \ldots, m\} \qquad (4)$$

Compared to Euclidean distance, SDE is more suitable for ManyOPs since it considers both diversity and convergence information of the solutions [10]. As a result, the convergence of DA is expected to be improved so that DA won't drag the whole population backwards.

IV. Experimental Studies

A. Experimental Settings

In order to examine the effect of CA truncation and DA modification mentioned above, we implemented three variants of the Two Archive Algorithms under the framework of jMetal [22] [2], as shown in Table I. To be specific, TAA represents the old version, ITAA-a modifies the CA truncation of TAA while ITAA-b modifies both CA and DA maintenance strategies.

TABLE I.   Algorithm Variants

Modifications	TAA	ITAA-a	ITAA-b
CA truncation	-	✓	✓
DA truncation	-	-	✓

---

[1] MOEA/D-PBI is a variant of MOEA/D using Penalty-based Boundary Intersection to aggregate objective values [3].

[2] jMetal(Metaheuristic Algorithms in Java) is a Java-based framework for multi-objective optimization.

Algorithm 3 Truncating DA in ITAA

1: **for all** $i = 1$ to sizeof(DA) **do**
2:    Set DA(i).DistanceToCA $\leftarrow$ maxReal
3:    **for all** $j = 1$ to sizeof(CA) **do**
4:      **if** DA(i).DistanceToCA $>$ Dist(CA(j),DA(i)) **then**
5:        Set DA(i).DistanceToCA $\leftarrow$ Dist(CA(j),DA(i))
6:      **end if**
7:    **end for**
8: **end for**
9: **if** sizeof(CA) + sizeof(DA) $> limitUA$ **then**
10:    **while** sizeof(CA) + sizeof(DA) $> limitUA$ **do**
11:      Delete the member with the minimal DistanceToCA from DA
12:    **end while**
13: **end if**

DTLZ is a well-defined continuous test problem suite which is widely used to evaluate MOEAs [23]. More importantly, it is scalable in terms of objective numbers, which makes DTLZ popular in evaluating many-objective evolutionary algorithms. Problems with various characteristics such as different Pareto front shapes, different number of local Pareto-optimal fronts and different solution density are included in the test function family. In our experiments, we used four DTLZ functions (DTLZ1-DTLZ4) since their Pareto front is known in advance. The number of objectives was set to 5, 10, 15, and 20. As a result, there are in total 16 test instances.

Although the ideal population size will increase exponentially to the objective dimension, we cannot afford this. Thus we used a linear function to set the population size. Specifically, the population size, denoted as popSize, was set to $5 * m$ where m is the number of objectives. The evolving generation MaxGen was set to 500. Each algorithm was executed 20 independent times. As to the genetic operators, we set them according to [5]. In all algorithm variants, the simulated binary crossover (SBX) operator and polynomial mutation were applied to generate offspring solutions. The crossover probability p_c was set to 0.9 while the mutation probability p_m was set to 1/ND (ND is the number of decision variables). Both distribution indexes for crossover and mutation (η_c and η_c) were set to 20. In TAA, the parameter $limitCA$ is tuned to 0.6*popSize since it obtained best overall performance compared with other $limitCA$ values in our preliminary experiments. Other than that, all parameters in these algorithms were kept the same in order to perform a fair comparison. The parameter k of the test problems was set according to [23]. In our experiments, the weight vector set **W** was generated using the sampling method from [24]. The parameter θ in PBI was set to 5 according to [3].

In order to verify the performance of ITAA-b, we compared it with four state-of-the-art many-objective evolutionary algorithms in the literature: ϵ-MOEA [25], MOEA/D-TCH [3], SPEA2+SDE [10], and NSGA-III [13]. ϵ-MOEA uses ϵ-dominance to enhance the selection pressure towards the Pareto front while MOEA/D-TCH uses Tchebycheff functions as aggregation method to decompose a many-objective problem into a series of single-objective sub-problems. In SPEA2+SDE, Shift-based Density Estimation (SDE) are combined into SPEA2 algorithm to incorporate convergence information into

diversity measuring mechanism. NSGA-III takes advantage of reference direction based niching strategy to balance the search power along all the directions. In order to guarantee a fair comparison, the parameters are set as follows: The population size, the evolving generation, and the weight vectors of ITAA-b, MOEA/D-TCH, and NSGA-III were set the same way as described in the previous part. So are the crossover and mutation settings. In order to make the archive size of ϵ-MOEA, which is determined by the value of ϵ, approximately the same with that of the remaining algorithms, we set the value of ϵ according to Table II. Other than that, the parameter settings of the four algorithms are the same with their original paper.

TABLE II.   THE VALUES OF ϵ IN ϵ-MOEA

problem	5d	10d	15d	20d
DTLZ1p	0.05	0.3	0.4	0.5
DTLZ2p	0.15	0.25	0.275	0.3
DTLZ3p	0.1	0.8	0.9	1.0
DTLZ4p	0.1	0.25	0.275	0.3

To measure the convergence quality of the final solution set, we used the Inverted Generational Distance (IGD) [26]. Given an approximation set A and a subset of the real Pareto front $PF' = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_{|PF'|}\}$, IGD is defined as follows:

$$IGD_p = \left(\frac{1}{|PF'|} \sum_{\mathbf{p} \in PF'} (d(\mathbf{p}, A))^p \right)^{\frac{1}{p}} \qquad (5)$$

where $d(\mathbf{p}, A)$ is the Euclidean distance between $\mathbf{p}$ and its nearest member in A. In our experiments, the coefficient p was set to 2.

In order to measure the diversity quality, we also used the Generalized Spread (GSpread) [27] as evaluation indicator. It's defined as follows:

$$GSpread = \frac{\sum_{i=1}^{s} d(\mathbf{e}_i, A) + \sum_{\mathbf{p} \in PF'} |d(\mathbf{p}, A) - \bar{d}|}{\sum_{i=1}^{s} d(\mathbf{e}_i, A) + |PF'|\bar{d}} \qquad (6)$$

where $\{\mathbf{e}_1, \ldots, \mathbf{e}_s\}$ are s extreme solutions in PF', $d(\mathbf{p}, A)$ is the Euclidean distance of $\mathbf{p}$ from its nearest neighbor in A, $\bar{d}$ is the average values of $d(\mathbf{p}, A)$ for all solutions in PF'. If A is well distributed and the extreme solutions in PF' are included in A, its GSpread value will be zero. A smaller GSpread indicates a better population diversity.

B. Experimental Results and Discussions

1) Performance Comparisons of TAA, ITAA-a, and ITAA-b: The IGD and GSpread results of TAA, ITAA-a, and ITAA-b are shown in Tables III and IV respectively. Based on these two tables, we can carry out a Friedman Test on these results. Finally, Table V is obtained. When p = 0.05, degrees of freedom is 2, the critical value is 5.99. Both 16.125 and 18.375 are larger than 5.99. Thus the null hypotheses are rejected and there are statistically significant differences among the IGD and GSpread performances of the three algorithms. Thus we can say that ITAA-b outperforms TAA in terms of both

Instances[1]	TAA	ITAA-a	ITAA-b
DTLZ1p5d	$2.34e-02_{1.3e-02}$	$5.91e-03_{2.0e-03}$	$1.74e-03_{6.5e-05}$
DTLZ2p5d	$2.55e-03_{9.4e-05}$	$2.46e-03_{1.0e-03}$	$3.03e-03_{2.3e-04}$
DTLZ3p5d	$2.37e-01_{6.9e-02}$	$1.34e-01_{4.7e-02}$	$3.17e-03_{3.7e-04}$
DTLZ4p5d	$4.65e-03_{2.1e-03}$	$4.78e-03_{1.7e-03}$	$4.53e-03_{1.7e-03}$
DTLZ1p10d	$3.38e-02_{2.0e-02}$	$1.83e-02_{1.4e-02}$	$3.22e-03_{3.5e-04}$
DTLZ2p10d	$6.18e-03_{1.7e-04}$	$1.13e-02_{4.6e-04}$	$6.51e-03_{2.9e-04}$
DTLZ3p10d	$2.84e-01_{9.3e-02}$	$2.56e-01_{7.2e-02}$	$9.87e-03_{6.3e-03}$
DTLZ4p10d	$8.26e-03_{1.6e-04}$	$1.15e-02_{4.8e-04}$	$7.27e-03_{3.3e-04}$
DTLZ1p15d	$4.38e-02_{1.5e-02}$	$2.04e-02_{1.1e-02}$	$5.31e-03_{7.1e-04}$
DTLZ2p15d	$1.03e-02_{3.8e-04}$	$1.48e-02_{2.8e-04}$	$1.01e-02_{3.5e-04}$
DTLZ3p15d	$3.88e-01_{8.1e-02}$	$2.67e-01_{1.1e-01}$	$1.36e-02_{5.0e-03}$
DTLZ4p15d	$1.62e-02_{7.2e-04}$	$1.68e-02_{3.7e-04}$	$1.09e-02_{4.0e-04}$
DTLZ1p20d	$4.85e-02_{2.5e-02}$	$2.93e-02_{1.7e-02}$	$7.91e-03_{9.5e-04}$
DTLZ2p20d	$1.57e-02_{6.9e-04}$	$1.87e-02_{3.5e-04}$	$1.31e-02_{2.7e-04}$
DTLZ3p20d	$4.70e-01_{1.3e-01}$	$3.14e-01_{7.0e-02}$	$3.86e-02_{8.2e-02}$
DTLZ4p20d	$2.34e-02_{5.8e-04}$	$2.20e-02_{9.7e-04}$	$1.44e-02_{3.7e-04}$

[1] DTLZαpβd mean β-objective DTLZα test function, hereinafter.

Instances	TAA	ITAA-a	ITAA-b
DTLZ1p5d	$1.48e+00_{1.6e-01}$	$1.65e+00_{9.2e-02}$	$8.41e-01_{5.2e-02}$
DTLZ2p5d	$5.37e-01_{7.7e-02}$	$6.64e-01_{9.9e-02}$	$6.17e-01_{9.2e-02}$
DTLZ3p5d	$1.25e+00_{1.0e-01}$	$1.33e+00_{8.5e-02}$	$6.16e-01_{8.4e-02}$
DTLZ4p5d	$4.60e-01_{1.2e-01}$	$5.95e-01_{1.3e-01}$	$6.50e-01_{1.1e-01}$
DTLZ1p10d	$1.53e+00_{1.3e-01}$	$1.58e+00_{1.6e-01}$	$6.58e-01_{9.6e-02}$
DTLZ2p10d	$5.84e-01_{5.4e-02}$	$1.10e+00_{6.8e-02}$	$6.50e-01_{7.9e-02}$
DTLZ3p10d	$1.20e+00_{8.1e-02}$	$1.34e+00_{8.9e-02}$	$7.41e-01_{9.1e-02}$
DTLZ4p10d	$7.94e-01_{7.6e-02}$	$1.17e+00_{6.4e-02}$	$6.24e-01_{6.0e-02}$
DTLZ1p15d	$1.52e+00_{8.0e-02}$	$1.60e+00_{1.2e-01}$	$7.56e-01_{1.3e-01}$
DTLZ2p15d	$7.61e-01_{4.3e-02}$	$9.64e-01_{6.2e-02}$	$6.58e-01_{8.3e-02}$
DTLZ3p15d	$1.31e+00_{4.8e-02}$	$1.35e+00_{6.2e-02}$	$7.35e-01_{1.0e-01}$
DTLZ4p15d	$6.50e-01_{1.6e-01}$	$1.11e+00_{7.8e-02}$	$6.00e-01_{4.9e-02}$
DTLZ1p20d	$1.41e+00_{1.8e-01}$	$1.59e+00_{1.2e-01}$	$9.35e-01_{1.4e-01}$
DTLZ2p20d	$9.59e-01_{3.9e-02}$	$9.39e-01_{3.3e-02}$	$7.06e-01_{5.1e-02}$
DTLZ3p20d	$1.41e+00_{5.1e-02}$	$1.36e+00_{8.5e-02}$	$7.73e-01_{1.0e-01}$
DTLZ4p20d	$3.10e-01_{2.3e-02}$	$7.27e-01_{1.2e-01}$	$5.67e-01_{5.0e-02}$

convergence and diversity on most test cases. ITAA-a ranks a little better than TAA in terms of IGD while is beaten on the GSpread metric. Also, since the experimental results in [20] showed that TAA outperformed PESA, thus we can hint that ITAA also outperforms PESA, although more comparisons are needed to draw a strict conclusion.

However, it is interesting that in Table III, ITAA-a doesn't converge very well on DTLZ2 instances. One possible cause might be that some good members in CA might be eliminated by PBI-based truncation when the Pareto front is spherical since PBI emphasizes a single direction in each generation. Although DTLZ3 instances also have the same Pareto front shape, it is more harder for algorithms to convergence since they introduce many local Pareto-optimal fronts. These local fronts might cause TAA to stagnate. On the other hand, ITAA-a with PBI-embedded is less likely to be attracted to these local Pareto fronts. This may be the reason why ITAA-a achieved better IGD values than TAA on DTLZ3 instances. Although more experiments are needed for some in-depth analysis.

2) Further Experiments and Discussions: In order to further analyze the behavior of the PBI-based CA truncation and SDE-based DA truncation, we carried out some more experiments on DTLZ3 and DTLZ4 instances, since they are designed to investigate an MOEA's ability in terms of convergence and diversity respectively.

First, we tested if the CA truncation mechanism could improve the evolving of CA. Thus the update rates of CA for TAA and ITAA-a (colored black and red respectively) on the DTLZ3 instances with 5-20 objectives are calculated and shown in Fig. 1. In the figure, update rate is the ratio of the number of new CA members to the size of CA. For certain instance and algorithm, the error bar is generated by mean,

Algorithm [3]	IGD Ranking[1]	GSpread Ranking[2]
TAA	2.5	1.875
ITAA-a	2.3125	2.8125
ITAA-b	1.1875	1.3125

[1] Friedman statistic considering reduction performance (distributed according to chi-square with 2 degrees of freedom: 16.125).

[2] Friedman statistic considering reduction performance (distributed according to chi-square with 2 degrees of freedom: 18.375).

[3] When $p = 0.05$, degrees of freedom is 2, the critical value is 5.99. Both 16.125 and 18.375 are larger than 5.99. Thus the null hypotheses are rejected and there are statistically significant differences among the IGD and Spread performances of the three algorithms.

maximum and minimum of its update rate among 20 runs every 50 generations. Similar generating method is used for the remaining figures. From Fig. 1, we can see that ITAA-a achieves a higher update rate of CA compared to TAA during the evolving process on these DTLZ3 instances. This means PBI-based CA truncation is more likely to help CA evolving instead of stagnating.

Second, one critical role of CA is to achieve a good convergence performance so that it can drive the whole population forward. Having this in mind, we recorded the IGD value of CA for TAA and ITAA-a during the evolving process every 50 generations. From Fig. 2, we can say that on these instances, ITAA-a achieves better IGD values for CA. This means that PBI-based CA truncation mechanism helps CA drive forth in the searching process, thus ITAA-a converges better than TAA, as shown in Table V.

Moreover, since the only difference between ITAA-a and ITAA-b is the DA truncation strategy, these two algorithms are compared to examine the effect of SDE-based DA truncation. DTLZ4 instances are studied since it can demonstrate an algorithm's ability to maintain a good distribution of solutions. Here, we tested the online GSpread of UA for ITAA-a and ITAA-b during the evolving process. We didn't use the GSpread of DA because that DA truncation involves not only DA but also CA and the major goal of DA truncation is to maintain the diversity of UA instead of DA. The results are shown in Fig. 3. We can see that ITAA-b (colored blue) is at least comparable to, if not better than ITAA-a (colored red) in terms of the GSpread of UA. This means that SDE is superior to Euclidean distance since it resulted in better spread of UA.

In order to explain the results, we further examined the IGD value of CA and DA for these two algorithms. The results are shown in Fig. 4. As the figure shows, the DA of ITAA-a falls further behind CA, while in ITAA-b, DA are comparable to CA in terms of convergence. Actually this is because that SDE takes into consideration not only distribution but also convergence information. We think the improvement of the IGD of DA might be the main reason why the GSpread of UA for ITAA-a is beaten by ITAA-b. Actually an evidence for the prediction can be inferred from Fig.3(b) and Fig.4(b), where the change of the GSpread of UA in ITAAa (the solid red line) seems to be in accordance with the IGD of DA in ITAAa (the dashed red line).
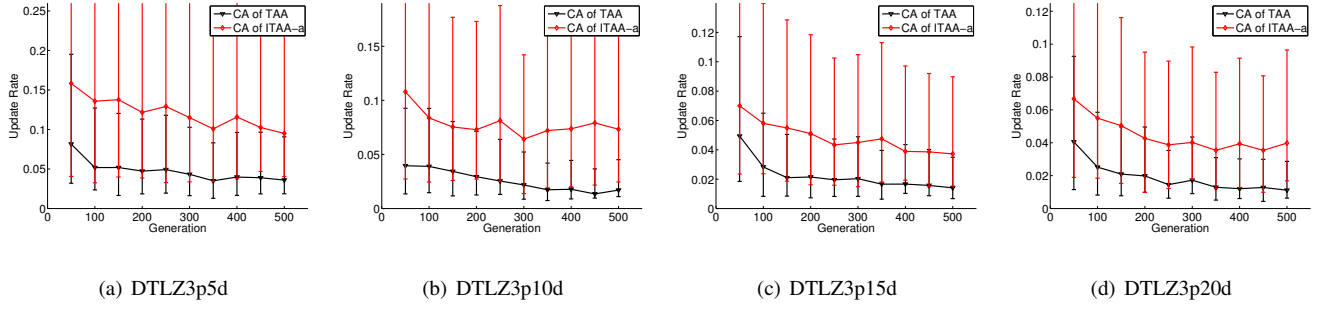
Fig. 1. Update rate of CA in TAA and ITAA-a on DTLZ3 instances. Update rate is the ratio of the number of new CA members to the size of CA. For certain instance and algorithm, the error bar is generated by mean, maximum and minimum of its update rate among 20 runs every 50 generations. Similar generating method is used for the remaining figures.
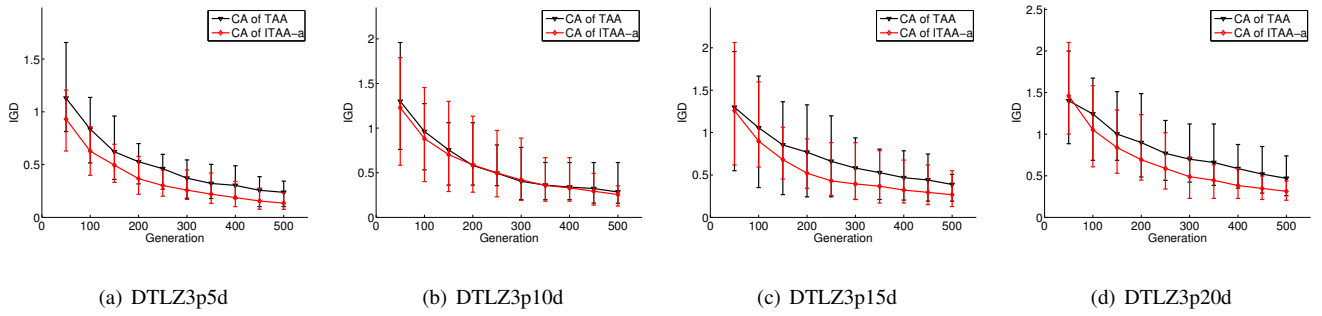

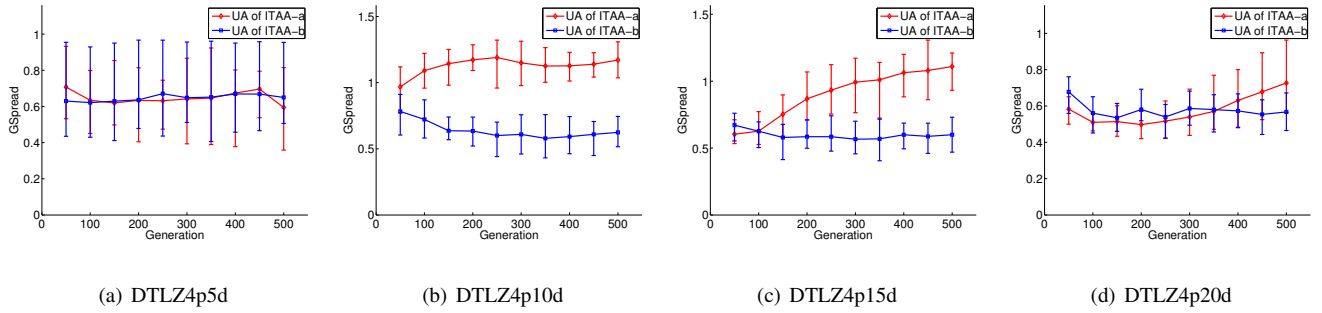Fig. 2. IGD of CA in TAA and ITAA-a on DTLZ3 instances.

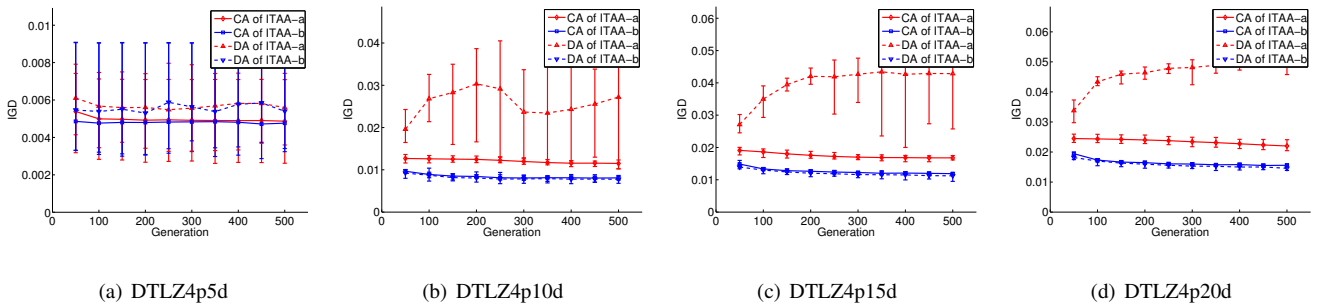Fig. 3. GSpread of UA in ITAA-a and ITAA-b on DTLZ4 instances.

Fig. 4. IGD of CA and DA in ITAA-a and ITAA-b on DTLZ4 instances.

3) Comparison with State-of-the-art Algorithms: The IGD and GSpread results are shown in Tables VI and VII. Dark/light gray entries in the tables indicate the corresponding values rank first/second place among all the values in the same row, hereinafter. The Friedman testing results can be seen from Tables VIII. From these tables, we can see that SPEA2+SDE seems to show the best overall performance in terms of both IGD and GSpread metrics. ITAA-b is competitive with two state-of-the-art many-objective algorithms: ϵ-MOEA and MOEA/D-TCH, especially in terms of convergence. As [10] indicated, SDE will reduce the density estimation accuracy to some extent. Although it works quite well in the SPEA2 framework, it might need some custom modifications when embedded into TAA. We think this might be the reason why the ranking of diversity performance of ITAA-b is not very good. We will study this problem in our future work. It's interesting that NSGA-III didn't perform very well on these instances. One possible reason might be that our settings of population size, maximum generation, and weight vectors might not be the best choice for NSGA-III.

TABLE VIII.   AVERAGE RANKINGS OF THE IGD VALUES OF THE FIVE ALGORITHMS

Algorithm[3]	IGD Ranking[1]	GSpread Ranking[2]
ITAA-b	2.625	3.125
ϵ-MOEA	2.9375	3.5625
MOEA/D-TCH	2.9375	3.3125
SPEA2+SDE	1.625	1.0
NSGA-III	4.875	4.0

[1] Friedman statistic considering reduction performance (distributed according to chi-square with 4 degrees of freedom: 35.55).

[2] Friedman statistic considering reduction performance (distributed according to chi-square with 4 degrees of freedom: 34.75).

[3] When $p = 0.05$, degrees of freedom is 4, the critical value is 9.49. Both 35.55 and 34.75 are larger than 9.49. Thus the null hypotheses are rejected and there are statistically significant differences among the IGD and Spread performances of these algorithms.

V.   CONCLUSIONS AND FUTURE WORK

In this paper, the Two Archive Algorithm has been extended to ITAA by modifying CA and DA truncation strategies. In order to deal with the size issue of CA, we set a threshold for CA and adopted the Penalty-based Boundary Intersection (PBI) to remove CA members when CA overflows. Also, the Euclidean distance for diversity measure in DA is replaced by a Shifted Density Estimation (SDE) technique, which is more suitable for many-objective problems. The performance of ITAA is evaluated on 16 DTLZ test instances with 5–20 objectives. As the experimental results showed, ITAA outperformed TAA in terms of the convergence metric IGD and diversity metric GSpread. In addition, further experiments were carried out to investigate the effect of truncation strategies for both archives. Experimental results on DTLZ3 instances showed that PBI-based CA truncation was able to improve the update rate and IGD of CA. Moreover, SDE-based DA truncation is demonstrated to be effective to improve the convergence of DA and maintain the diversity of the union of CA and DA. Comparison results with four state-of-the-art algorithms demonstrate that ITAA is a competitive alternative when tackling many-objective problems.

There are several future directions for this work. First, ITAA is only tested on 16 DTLZ problem instances, further investigation is needed to evaluate its performance on other problems with more objectives. Second, whether there are any other better ways to embed PBI and SDE into TAA is still an open question. Third, the parameter analysis of the algorithm also needs to be studied.

REFERENCES

[1] M. Farina and P. Amato, "On the optimal solution definition for many-criteria optimization problems," in *Fuzzy Information Processing Society, 2002. Proceedings. NAFIPS. 2002 Annual Meeting of the North American*.   IEEE, 2002, pp. 233–238.

[2] K. Miettinen, *Nonlinear multiobjective optimization*.   Springer, 1999, vol. 12.

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

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

[5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 2, pp. 182–197, 2002.

[6] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," Eidgenössische Technische Hochschule Zürich (ETH), Tech. Rep. 103, May 2001.

[7] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*.   IEEE, 2008, pp. 2419–2426.

[8] R. Purshouse and P. Fleming, "On the evolutionary optimization of many conflicting objectives," *Evolutionary Computation, IEEE Transactions on*, vol. 11, no. 6, pp. 770–784, 2007.

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

[10] M. Li, S. Yang, and X. Liu, "Shift-based density estimation for pareto-based algorithms in many-objective optimization," *Evolutionary Computation, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2013.

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

[12] S. Adra and P. Fleming, "Diversity management in evolutionary many-objective optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 15, no. 2, pp. 183–195, 2011.

TABLE VI.   IGD. Mean and standard deviation of The Five Algorithms

	ITAA-b	ϵ-MOEA	MOEA/D-TCH	SPEA2+SDE	NSGA-III
DTLZ1p5d	$1.58e-03_{5.7e-05}$	$9.59e-03_{8.0e-03}$	$1.55e-03_{6.9e-05}$	$1.37e-03_{2.8e-05}$	$1.95e-02_{1.2e-02}$
DTLZ2p5d	$2.79e-03_{1.6e-04}$	$2.04e-03_{8.4e-05}$	$2.50e-03_{2.2e-05}$	$2.44e-03_{4.8e-05}$	$3.35e-03_{6.7e-04}$
DTLZ3p5d	$3.23e-03_{3.4e-04}$	$4.63e-01_{9.2e-02}$	$6.28e-03_{8.8e-03}$	$2.63e-03_{3.4e-04}$	$1.36e-01_{4.5e-02}$
DTLZ4p5d	$5.65e-03_{1.6e-03}$	$3.97e-03_{2.3e-03}$	$4.51e-03_{3.8e-04}$	$2.47e-03_{9.6e-05}$	$5.52e-03_{1.7e-03}$
DTLZ1p10d	$3.20e-03_{1.5e-04}$	$1.70e-02_{1.0e-02}$	$3.28e-03_{2.6e-04}$	$2.76e-03_{3.2e-05}$	$7.36e-02_{4.0e-02}$
DTLZ2p10d	$6.35e-03_{2.6e-04}$	$4.77e-03_{1.6e-04}$	$6.49e-03_{3.2e-04}$	$6.69e-03_{1.9e-04}$	$7.19e-03_{4.6e-04}$
DTLZ3p10d	$8.41e-03_{3.1e-03}$	$1.01e+00_{2.0e-01}$	$1.34e-02_{1.6e-02}$	$6.77e-03_{2.1e-04}$	$1.03e+00_{2.2e-01}$
DTLZ4p10d	$7.35e-03_{4.3e-04}$	$6.97e-03_{2.2e-04}$	$9.76e-03_{2.1e-04}$	$6.81e-03_{2.0e-04}$	$1.37e-02_{1.2e-03}$
DTLZ1p15d	$5.69e-03_{7.3e-04}$	$1.05e-02_{6.2e-03}$	$4.94e-03_{5.8e-04}$	$3.92e-03_{4.9e-05}$	$1.17e-01_{3.9e-02}$
DTLZ2p15d	$1.01e-02_{2.3e-04}$	$6.89e-03_{2.0e-04}$	$1.05e-02_{6.1e-04}$	$1.08e-02_{2.2e-04}$	$1.30e-02_{5.6e-04}$
DTLZ3p15d	$2.02e-01_{1.6e-02}$	$1.24e+00_{2.7e-01}$	$1.45e-02_{8.7e-03}$	$1.10e-02_{2.2e-04}$	$1.86e+00_{4.4e-01}$
DTLZ4p15d	$1.09e-02_{5.8e-04}$	$1.14e-02_{4.3e-04}$	$1.40e-02_{3.0e-04}$	$1.08e-02_{2.2e-04}$	$2.03e-02_{8.0e-04}$
DTLZ1p20d	$8.16e-03_{8.5e-04}$	$1.12e-02_{6.8e-03}$	$6.72e-03_{6.8e-04}$	$4.89e-03_{6.2e-05}$	$1.98e-01_{6.8e-02}$
DTLZ2p20d	$1.32e-02_{2.9e-04}$	$8.74e-03_{5.7e-04}$	$1.51e-02_{1.0e-03}$	$1.46e-02_{3.0e-04}$	$1.85e-02_{4.8e-04}$
DTLZ3p20d	$3.56e-01_{1.8e-02}$	$1.26e+00_{2.3e-01}$	$1.82e-02_{5.9e-03}$	$1.45e-02_{3.0e-04}$	$2.12e+00_{4.9e-01}$
DTLZ4p20d	$1.47e-02_{4.7e-04}$	$1.54e-02_{3.5e-04}$	$1.76e-02_{4.6e-04}$	$1.47e-02_{2.1e-04}$	$2.43e-02_{5.7e-04}$

TABLE VII.   GSPREAD. Mean and standard deviation of The Five Algorithms

	ITAA-b	ϵ-MOEA	MOEA/D-TCH	SPEA2+SDE	NSGA-III
DTLZ1p5d	$6.16e-01_{1.1e-01}$	$1.50e+00_{2.8e-01}$	$3.81e-01_{3.9e-02}$	$1.76e-01_{2.3e-02}$	$1.11e+00_{3.3e-01}$
DTLZ2p5d	$6.51e-01_{8.0e-02}$	$9.51e-01_{2.1e-02}$	$4.69e-01_{3.9e-02}$	$2.56e-01_{2.1e-02}$	$7.86e-01_{4.7e-02}$
DTLZ3p5d	$6.92e-01_{9.8e-02}$	$9.85e-01_{8.5e-02}$	$6.33e-01_{1.8e-01}$	$2.55e-01_{1.8e-02}$	$1.36e+00_{1.3e-01}$
DTLZ4p5d	$6.82e-01_{1.2e-01}$	$8.75e-01_{1.7e-01}$	$6.34e-01_{8.2e-02}$	$2.46e-01_{1.5e-02}$	$8.57e-01_{1.4e-01}$
DTLZ1p10d	$6.44e-01_{1.0e-01}$	$9.56e-01_{1.6e-01}$	$4.66e-01_{3.9e-02}$	$1.46e-01_{6.5e-03}$	$1.02e+00_{1.3e-01}$
DTLZ2p10d	$6.71e-01_{8.5e-02}$	$7.79e-01_{4.1e-02}$	$7.17e-01_{2.9e-02}$	$2.02e-01_{1.4e-02}$	$6.11e-01_{4.5e-02}$
DTLZ3p10d	$7.22e-01_{7.7e-02}$	$5.57e-01_{1.9e-02}$	$7.82e-01_{1.0e-01}$	$2.07e-01_{1.4e-02}$	$7.70e-01_{7.4e-02}$
DTLZ4p10d	$7.25e-01_{9.5e-02}$	$1.11e+00_{6.8e-02}$	$8.22e-01_{3.6e-02}$	$1.93e-01_{1.4e-02}$	$1.05e+00_{6.0e-02}$
DTLZ1p15d	$8.66e-01_{1.3e-01}$	$7.22e-01_{2.8e-01}$	$4.40e-01_{1.8e-02}$	$1.40e-01_{5.6e-03}$	$9.98e-01_{1.1e-01}$
DTLZ2p15d	$7.00e-01_{5.9e-02}$	$5.96e-01_{2.9e-02}$	$7.56e-01_{3.1e-02}$	$1.85e-01_{1.2e-02}$	$6.13e-01_{3.6e-02}$
DTLZ3p15d	$7.53e-01_{8.2e-02}$	$5.99e-01_{2.6e-02}$	$9.00e-01_{2.8e-01}$	$1.84e-01_{8.9e-03}$	$7.55e-01_{4.5e-02}$
DTLZ4p15d	$6.19e-01_{6.0e-02}$	$7.53e-01_{6.2e-02}$	$8.04e-01_{1.7e-02}$	$1.84e-01_{7.7e-03}$	$1.08e+00_{5.3e-02}$
DTLZ1p20d	$9.45e-01_{1.0e-01}$	$6.28e-01_{1.9e-01}$	$5.96e-01_{2.8e-01}$	$1.43e-01_{4.8e-03}$	$9.26e-01_{7.5e-02}$
DTLZ2p20d	$7.20e-01_{5.3e-02}$	$4.98e-01_{7.0e-02}$	$7.79e-01_{2.9e-02}$	$1.77e-01_{8.6e-03}$	$6.46e-01_{3.8e-02}$
DTLZ3p20d	$7.39e-01_{1.2e-01}$	$5.87e-01_{1.9e-02}$	$8.31e-01_{1.4e-01}$	$1.81e-01_{9.5e-03}$	$7.71e-01_{4.9e-02}$
DTLZ4p20d	$6.01e-01_{5.7e-02}$	$1.09e+00_{5.1e-02}$	$8.30e-01_{2.1e-02}$	$1.83e-01_{7.4e-03}$	$1.04e+00_{4.2e-02}$

[13] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part i: Solving problems with box constraints," *Evolutionary Computation, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2013.

[14] M. Emmerich, N. Beume, and B. Naujoks, "An emo algorithm using the hypervolume measure as selection criterion," in *Evolutionary Multi-Criterion Optimization*. Springer, 2005, pp. 62–76.

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

[16] M. Wagner and F. Neumann, "A fast approximation guided evolutionary multi-objective algorithm," in *Proc. 15th Annual Conference on Genetic and Evolutionary Computation (GECCO13). ACM Press*, 2013.

[17] K. Bringmann, T. Friedrich, F. Neumann, and M. Wagner, "Approximation-guided evolutionary multi-objective optimization," in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*. AAAI Press, 2011, pp. 1198–1203.

[18] R. Wang, R. Purshouse, and P. Fleming, "Preference-inspired co-evolutionary algorithm using weights for many-objective optimization," in *Proceeding of the fifteenth annual conference companion on Genetic and evolutionary computation conference companion*. ACM, 2013, pp. 101–102.

[19] D. Saxena, J. Duro, A. Tiwari, K. Deb, and Q. Zhang, "Objective reduction in many-objective optimization: Linear and nonlinear algorithms," *Evolutionary Computation, IEEE Transactions on*, vol. 17, no. 1, pp. 77–99, 2013.

[20] K. Praditwong and X. Yao, "A new multi-objective evolutionary optimisation algorithm: the two-archive algorithm," in *Computational Intelligence and Security, 2006 International Conference on*, vol. 1. IEEE, 2006, pp. 286–291.

[21] D. Corne, J. Knowles, and M. Oates, "The pareto envelope-based selection algorithm for multiobjective optimization," in *Parallel Problem Solving from Nature PPSN VI*. Springer, 2000, pp. 839–848.

[22] J. Durillo and A. Nebro, "jmetal: A java framework for multi-objective optimization," *Advances in Engineering Software*, vol. 42, no. 10, pp. 760–771, 2011.

[23] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proceedings of the Congress on Evolutionary Computation (CEC-2002),(Honolulu, USA)*. IEEE, 2002, pp. 825–830.

[24] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," in *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*. IEEE, 2009, pp. 203–208.

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

[26] D. Van Veldhuizen and G. Lamont, "Multiobjective evolutionary algorithm research: A history and analysis," Citeseer, Tech. Rep., 1998.

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