An Improved NSGA-III Procedure for Evolutionary Many-Objective Optimization

Yuan Yuan, Hua Xu, Bo Wang
State Key Laboratory of Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, P.R.China
yyxhdy@gmail.com, xuhua@tsinghua.edu.cn, wbthu@outlook.com

ABSTRACT

Many-objective (four or more objectives) optimization problems pose a great challenge to the classical Pareto-dominance based multi-objective evolutionary algorithms (MOEAs), such as NSGA-II and SPEA2. This is mainly due to the fact that the selection pressure based on Pareto-dominance degrades severely with the number of objectives increasing. Very recently, a reference-point based NSGA-II, referred as NSGA-III, is suggested to deal with many-objective problems, where the maintenance of diversity among population members is aided by supplying and adaptively updating a number of well-spread reference points. However, NSGA-III still relies on Pareto-dominance to push the population towards Pareto front (PF), leaving room for the improvement of its convergence ability. In this paper, an improved NSGA-III procedure, called θ-NSGA-III, is proposed, aiming to better tradeoff the convergence and diversity in many-objective optimization. In θ-NSGA-III, the non-dominated sorting scheme based on the proposed θ-dominance is employed to rank solutions in the environmental selection phase, which ensures both convergence and diversity. Computational experiments have shown that θ-NSGA-III is significantly better than the original NSGA-III and MOEA/D on most instances no matter in convergence and overall performance.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*; G.1.6 [**Numerical Analysis**]: Optimization

General Terms

Algorithms, Performance

Keywords

Many-objective optimization, NSGA-III, θ-NSGA-III, θ-dominance, non-dominated sorting

GECCO'14, July 12–16, 2014, Vancouver, BC, Canada.

http://dx.doi.org/10.1145/2576768.2598342 .

1. INTRODUCTION

Recently, many-objective optimization, typically referring to the optimization of problems having four or more objectives, has attracted increasing attention in evolutionary multi-objective optimization (EMO) community. The boom of the research on evolutionary many-objective optimization is mainly inspired from two aspects. On the one hand, the optimization problems involving a high number of objectives indeed appear widely in many real-world applications, e.g., control system design [17], industrial scheduling [27], software engineering [26]. Hence, the practitioners are in need of effective optimizer to solve these problems at hand. On the other hand, the popular Pareto-dominance based multi-objective evolutionary algorithms (MOEAs), such as NSGA-II [9] and SPEA2 [31], have encountered great difficulties in many-objective optimization, although they have shown excellent performance on problems with two or three objectives. The primary reason is that almost all the solutions in each population become non-dominated with the number of objectives increasing, which would lead to the severe loss of Pareto-based selection pressure towards the Pareto front (PF).

To overcome the drawback of Pareto-dominance based MOEAs, some efforts have been made in the literature. In summary, the developing techniques can be roughly classified into the following two types:

1) Adoption of new preference relations: Since the Pareto-dominance relation scales poorly in many-objective optimization, it is natural to use other preference relations, including modified Pareto-dominance and different ranking schemes, so as to produce fine selection pressure towards PF. Up to now, many alternative preference relations have been proposed, such as favour relation [13], ϵ-dominance [22], fuzzy Pareto-dominance [15] and so on [21, 12, 25].

2) Adoption of new diversity promotion mechanisms: In many-objective optimization, Pareto-dominance could not provide sufficient selection pressure to make progress in a given population, so the diversity criterion begins to play a key role in such cases. However, the existing diversity-preservation criteria, such as crowding distance [9], are not suitable for many-objective problems [19, 20]. Thus, a new mechanism promoting the diversity is needed. In the work [1], two mechanisms for managing diversity were proposed to investigate their impact on overall convergence in many-objective optimization. The recently proposed NSGA-III [8] replaced the crowding distance operator in NSGA-II with a clustering operator which is aided by a set of well-distributed reference points.

It is worthy noting that, unlike Pareto-dominance based MOEAs, another two class of approaches, decomposition-based and indicator-based approaches, have been found very promising in many-objective optimization. The former decomposes a problem with multiple objectives into a set of single-objective subproblems through scalarizing functions, and then solves these subproblems simultaneously by evolving a population of solutions. MOEA/D [29, 23] is the most typical implementation of this class. The indicator-based approach employs a single performance indicator, such as hypervolume [30, 3], to optimize a desired property of evolutionary population. Regarding the exact calculation of hypervolume is especially computational expensive in high-dimensional objective space, the fast algorithm [2] that uses Monte Carlo simulation to approximate the exact hypervolume values has been developed.

This paper proposes an improved NSGA-III procedure, called θ-NSGA-III, for evolutionary many-objective optimization. θ-NSGA-III inherits the characteristic ideas in the original NSGA-III, including the adaptive normalization and diversity-preservation aided by well-spread reference points. However, quite different from NSGA-III, a new preference relation, referred as θ-dominance, is used instead of Pareto-dominance in the proposed θ-NSGA-III. Our motivation is based on the following consideration. NSGA-III emphasizes population members which are Pareto non-dominated yet close to the reference line of each reference point. Nevertheless, when the number of objectives is high, Pareto-dominance lacks enough selection pressure to pull the population towards PF, so NSGA-III indeed emphasizes diversity more than convergence in such cases. We want to improve the convergence of NSGA-III for many-objective problems but without much deterioration of its good diversity maintenance ability. The proposed θ-dominance could well serve this goal. In the θ-dominance, the solutions are allocated into different clusters represented by well-distributed reference points. Only the solutions within the same cluster have the competitive relationship, where a fitness function similar to penalty-based boundary intersection (PBI) function [29] is defined. When conducting the environmental selection, the non-dominated sorting scheme [16] based on θ-dominance not only prefers solutions with better fitness values in each cluster, but also ensures that the solutions distribute as evenly as possible between these clusters. In the experimental studies, the impact of the parameter θ on the performance of θ-NSGA-III is first investigated. Further, θ-NSGA-III is compared with NSGA-III and MOEA/D on well-known test problems, the results indicate that θ-NSGA-III significantly outperforms the other two algorithms in most cases.

The remainder of this paper is organized as follows. Section 2 presents the background knowledge of this paper. The proposed θ-NSGA-III is described in detail in Section 3. Experimental studies on well known test problems are carried out in Section 4. Finally, our conclusions are drawn in Section 5.

2. BACKGROUND

In this section, some basic definitions in multi-objective optimization are first given. Then, we will briefly introduce the original NSGA-III.

2.1 Basic Definitions

The multi-objective optimization problem (MOP) can be mathematically defined as follows:

$$\begin{aligned} \min \mathbf{f}(\mathbf{x}) &= (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}))^{\mathrm{T}} \\ \text{subject to } \mathbf{x} &\in \Omega \subseteq \mathbb{R}^n \end{aligned} \tag{1}$$

where $\mathbf{x} = (x_1, x_2, ..., x_n)^{\mathrm{T}}$ is a n-dimensional decision variable vector from the decision space Ω; $\mathbf{f} : \Omega \rightarrow \Theta \subseteq \mathbb{R}^m$ consists a set of m objective functions, and is a mapping from n-dimensional decision space Ω to m-dimensional objective space Θ.

DEFINITION 1. *Given two decision vectors $\mathbf{x}, \mathbf{y} \in \Omega$, $\mathbf{x}$ is said to Pareto dominate $\mathbf{y}$, denoted by $\mathbf{x} \prec \mathbf{y}$, if $f_i(\mathbf{x}) \leq f_i(\mathbf{y})$, for every $i \in \{1, 2, \ldots, m\}$, and $f_j(\mathbf{x}) < f_j(\mathbf{y})$, for at least one index $j \in \{1, 2, \ldots, m\}$.*

DEFINITION 2. *A decision vector $\mathbf{x}^* \in \Omega$ is Pareto optimal if there is no $\mathbf{x} \in \Omega$ such that $\mathbf{x} \prec \mathbf{x}^*$.*

DEFINITION 3. *The Pareto set, PS, is defined as:*

$$PS = \{\mathbf{x} \in \Omega | \mathbf{x} \text{ is Pareto optimal}\} \tag{2}$$

DEFINITION 4. *The Pareto front, PF, is defined as:*

$$PF = \{\mathbf{f}(\mathbf{x}) \in \mathbb{R}^m | \mathbf{x} \in PS\} \tag{3}$$

DEFINITION 5. *The ideal point $\mathbf{z}^*$ is a vector $\mathbf{z}^* = (z_1^*, z_2^*, \ldots, z_m^*)^{\mathrm{T}}$, where z_i^* is the infimum of f_i for every $i \in \{1, 2, \ldots, m\}$.*

The goal of MOEAs is to move the non-dominated objective vectors towards PF (convergence), and also generate a good distribution of these vectors over the PF (diversity).

2.2 Original NSGA-III

The basic framework of NSGA-III remains similar to the established NSGA-II [9] with significant changes in its selection mechanism. The main procedure of NSGA-III can be briefly described below.

NSGA-III starts with the definition of a set of reference points. Then an initial population with N members is randomly generated, where N is the population size. The next steps are iterated until the termination criterion is satisfied. At the t-th generation, the current parent population P_t is used to produce an offspring population Q_t by using randomly selection, simulated binary crossover (SBX) operator and polynomial mutation [7]. The size of P_t and Q_t are both N. Thereafter, the two populations P_t and Q_t are merged together to form a new population $R_t = P_t \cup Q_t$ (of size $2N$). To choose the best N members from R_t for the next generation, the non-dominated sorting based on usual domination principle [4] is first used, which classifies R_t into different non-domination levels (F_1, F_2, and so on). Then, a new population S_t is constructed by filling members of different non-domination levels one at a time, starting from F_1, until the size of S_t equals to N or for the first time becomes greater than N. Let us suppose that the last level included is the l-th level. Hence, the solutions from the level $l + 1$ onwards are simply rejected. Members in $S_t \setminus F_l$ are already chosen for P_{t+1}, and the remaining population slots are chosen from F_l such that a desired diversity is maintained in the population. In the original NSGA-II, the solutions in

F_l with the largest crowding distance values are selected. However, the crowding distance measure does not perform well for many-objective problems [20]. Thus, the selection mechanism in NSGA-III is modified by conducting a more systematic analysis of members in S_t with respect to the supplied reference points.

To achieve this, objective values and supplied reference points are first normalized so that they have an identical range. After normalization, the ideal point of the set S_t is the zero vector. Thereafter, the perpendicular distance between a member in S_t and each of the reference lines (joining the ideal point with a reference point) is calculated. Each member in S_t is then associated with a reference point having the minimum perpendicular distance. Next, the niche count ρ_j for the j-th reference point, defined as the number of members in $S_t \setminus F_l$ that are associated with the j-th reference point, can be obtained based on the above process. Further, a niche-preservation operation is executed to select members from F_l, and it works as follows. First, the reference point set $J_{\min} = \{j : \text{argmin}_j \rho_j\}$ having the minimum ρ_j value is identified. In case of $|J_{\min}| > 1$, one $\bar{j} \in J_{\min}$ is randomly chosen. If $\rho_{\bar{j}} = 0$, we choose the one having the shortest perpendicular distance to the j-th reference line among members associated with the j-th reference point in F_l, and add it to P_{t+1}. The count of $\rho_{\bar{j}}$ is then increased by one. In the event $\rho_j \geq 1$, a randomly chosen member from front F_l that is associated with the j-th reference point is added to P_{t+1}, and the count of $\rho_{\bar{j}}$ also needs increasing by one. In both of the two cases, once there exists no such member to be selected, the j-th reference point is excluded from further consideration for the current generation. After niche counts are updated, the above niche operation is repeated for a total of $K = N - |S_t \setminus F_l|$ times to fill the remaining population slots of P_{t+1}. For more details of NSGA-III, please refer to [8].

3. PROPOSED ALGORITHM

3.1 Overview

The framework of the proposed θ-NSGA-III can be described in Algorithm 1. First, a set of K reference points is generated, which is denoted as $\mathbf{\Lambda} = \{\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_K\}$. For an m-objective problem, $\boldsymbol{\lambda}_j$ ($j \in \{1, 2, \ldots, K\}$) is an m-dimensional vector represented by $\boldsymbol{\lambda}_j = (\lambda_{j,1}, \lambda_{j,2}, \ldots, \lambda_{j,m})^{\mathrm{T}}$, where $\lambda_{j,k} \geq 0, k = 1, 2, \ldots, m$ and $\sum_{k=1}^{m} \lambda_{j,k} = 1$. Next, the initial population with N members is randomly produced. For the ideal point $\mathbf{z}^*$, it is often very time-consuming to compute exact z_i^*, so it is indeed estimated by the minimum value found so far for objective f_i, and is updated during the search. Steps 5-21 are iterated until the termination criterion is satisfied. In Step 6, the offspring population Q_t is produced by using the same genetic operators with those in NSGA-III. And like wise, Q_t is combined with the current population P_t, and form a new population R_t. Thereafter, R_t is normalized using the ideal point $\mathbf{z}^*$. After normalization, the clustering operator is used to split the members in R_t into a set of K clusters $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$, where the cluster C_j is represented by the reference point $\boldsymbol{\lambda}_j$. Then, the non-dominated sorting based on θ-dominance (not Pareto-dominance) is employed to classify R_t into different θ-non-domination levels (F_1, F_2, and so on). θ-dominance, which is the key concept in θ-NSGA-III, would be introduced later. Once θ-non-dominated-sorting has been finished, the remaining steps fill the population slots in P_{t+1} using one level at a time, starting from F_1. Unlike both NSGA-II and NSGA-III, we just randomly select solutions in the last accepted level F_l in θ-NSGA-III, because θ-dominance has stressed both convergence and diversity. Surely, some strategies to enhance the diversity could also be used as well in Step 18.

Algorithm 1 Framework of the proposed θ-NSGA-III

```
1: Λ ← GenerateReferencePoints()
2: P_0 ← InitializePopulation()
3: z* ← InitializeIdealPoint()
4: t ← 0
5: while the termination criterion is not met do
6:    Q_t ← MakeOffspringPopulation(P_t)
7:    UpdateIdealPoint(Q_t)
8:    R_t ← P_t ∪ Q_t
9:    Normalize(R_t, z*)
10:   C ← Clustering(R_t, Λ)
11:   {F_1, F_2, ...} ← θ-Non-dominated-sort(R_t, C)
12:   P_{t+1} ← ∅
13:   i ← 1
14:   while |P_{t+1}| + |F_i| ≤ N do
15:      P_{t+1} ← P_{t+1} ∪ F_i
16:      i ← i + 1
17:   end while
18:   RandomSort(F_i)
19:   P_{t+1} ← P_{t+1} ∪ F_i[1 : (N − |P_{t+1}|)]
20:   t ← t + 1
21: end while
```

Although θ-NSGA-III borrows several ideas of NSGA-III such as adaptive normalization and association operation, they differ a lot in some aspects. First, we adopt a simpler method that could generate any number of reference points, further resulting in an arbitrary population size. Second, the normalization and association (clustering) procedures occur in different occasions that can be seen in Algorithm 1, and the implementation details are also a little modified. Above all, we replace the original Pareto dominance used in NSGA-III with the proposed θ-dominance, which could well balance the convergence and diversity in many-objective optimization. In the following subsections, the important procedures of θ-NSGA-III are to be described in detail, and the motivation behind them will also be mentioned.

3.2 Reference Points Generation

In the original NSGA-III, the Das and Dennis's systematic approach [6] is adopted to generate structured reference points. However, in this approach, the number of reference points K would be controlled by a parameter H, where $K = C_{H+m-1}^{m-1}$. NSGA-III uses a population size N that is set as the smallest even number greater than or equal to K, so N indirectly depends on H. When the number of objectives m is high, N would become very large even if there exist only one intermediate reference point [8].

In the proposed θ-NSGA-III, we adopt a simpler approach [18] but with more flexibility to produce K uniformly distributed points, which can be depicted in Algorithm 2, where $rand(0, 1)$ is a random function returning a real number between 0 and 1 with uniform distribution.

3.3 Adaptive Normalization

In the original NSGA-III, the authors presented a normalization procedure, which is claimed to lead to more appropriate normalization values. However, the procedure is

Algorithm 2 GenerateReferencePoints()

```
for j ← 1 to K do
    s ← 0
    for k ← 1 to m do
        if k < m then
            r ← rand(0, 1)
            λ_{j,k} ← (1 − s)(1 − r^{1/(m−k)})
            s ← s + λ_{j,k}
        else
            λ_{j,k} ← 1 − s
        end if
    end for
end for
```

a little computational expensive since it would involve the solving of linear system of equations. Moreover, although the procedure could work well on problems with not too many objectives, we find that it sometimes results in abnormal normalization results on problems having a high number of objectives.

In θ-NSGA-III, we just use an usual way to normalize the population. Suppose $\mathbf{z}^{\max} = (z_1^{\max}, z_2^{\max}, \ldots, z_m^{\max})^{\mathrm{T}}$, where $z_i^{\max}$ is the maximum value for objective f_i in the population to be normalized. Then, for a solution $\mathbf{x}$, the normalized objective value $\tilde{f}_i(\mathbf{x})$ can be computed as $\tilde{f}_i(\mathbf{x}) = (f_i(\mathbf{x}) - z_i^*)/(z_i^{\max} - z_i^*)$. The overall procedure is summarized in Algorithm 3.

Algorithm 3 Normalize(R_t, $\mathbf{z}^*$)

```
(z_1^max, z_2^max, ..., z_m^max)^T ← (−∞, −∞, ..., −∞)^T
for each solution x in R_t do
    for i ← 1 to m do
        if f_i(x) > z_i^max then
            z_i^max ← f_i(x)
        end if
    end for
end for
for each solution x in R_t do
    for i ← 1 to m do
        f̃_i(x) ← (f_i(x) − z_i^*)/(z_i^max − z_i^*)
    end for
end for
```

3.4 Clustering Operator

In the proposed θ-NSGA-III, the clustering operator is similar to that in NSGA-III, but it is run on different populations. In the original NSGA-III, the operator is done to S_t, while in θ-NSGA-III, is to R_t. The clustering works in the normalized objective space, where the ideal point is the origin. In Figure 1, $\tilde{\mathbf{f}}(\mathbf{x}) = (\tilde{f}_1(\mathbf{x}), \tilde{f}_2(\mathbf{x}), \ldots, \tilde{f}_m(\mathbf{x}))^{\mathrm{T}}$ is the normalized objective vector for the solution $\mathbf{x}$, L is a line passing through the origin with direction $\boldsymbol{\lambda}_j$, and $\mathbf{u}$ is the projection of $\tilde{\mathbf{f}}(\mathbf{x})$ on L. Let $d_{j,1}(\mathbf{x})$ be the distance between the origin and $\mathbf{u}$, and $d_{j,2}(\mathbf{x})$ be the perpendicular distance between $\tilde{\mathbf{f}}(\mathbf{x})$ and L. They can be computed respectively as follows

$$d_{j,1}(\mathbf{x}) = \|\tilde{\mathbf{f}}(\mathbf{x})^{\mathrm{T}}\boldsymbol{\lambda}_j\|/\|\boldsymbol{\lambda}_j\| \tag{4}$$

$$d_{j,2}(\mathbf{x}) = \left\|\tilde{\mathbf{f}}(\mathbf{x}) - d_{j,1}(\mathbf{x})(\boldsymbol{\lambda}_j/\|\boldsymbol{\lambda}_j\|)\right\| \tag{5}$$

For the clustering operator, only $d_{j,2}$ will be concerned, $d_{j,1}$ will be involved later in the definition of θ-dominance.

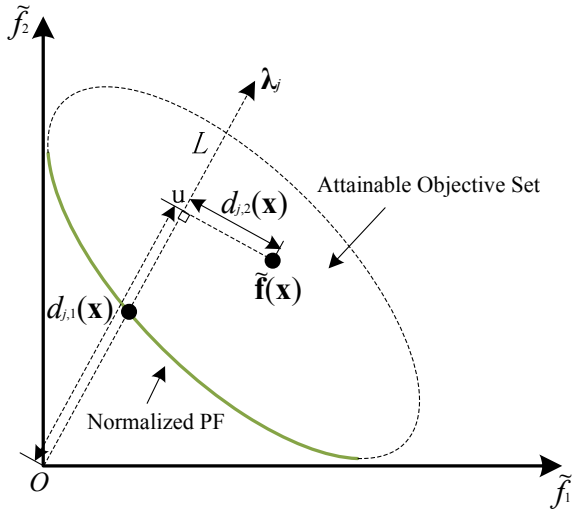


Figure 1: Illustration of distances $d_{j,1}$ and $d_{j,2}$.

We assign a solution $\mathbf{x}$ to the cluster C_j with the minimum $d_{j,2}(\mathbf{x})$ value. The detail process of clustering is shown in Algorithm 4.

Algorithm 4 Clustering(R_t, $\boldsymbol{\Lambda}$)

```
{C_1, C_2, ..., C_K} ← {∅, ∅, ..., ∅}
for each solution x in R_t do
    k ← 1
    min ← d_{1,2}(x)
    for j ← 2 to K do
        if d_{j,2}(x) < min then
            min ← d_{j,2}(x)
            k ← j
        end if
    end for
    C_k ← C_k ∪ {x}
end for
```

3.5 θ-dominance

Let $\mathcal{F}_j(\mathbf{x}) = d_{j,1}(\mathbf{x}) + \theta d_{j,2}(\mathbf{x})$, $j \in \{1, 2, \ldots, K\}$, where θ is a predefined penalty parameter. The form of $\mathcal{F}_j(x)$ is the same with that of PBI function [29]. But here, the distances $d_{j,1}$ and $d_{j,2}$ are both computed in the normalized objective space. It is clear that $d_{j,2}(\mathbf{x}) = 0$ ensures that $\tilde{\mathbf{f}}(\mathbf{x})$ is always in L resulting in perfect diversity, while smaller $d_{j,1}$ value means better convergence. With the definition of $\mathcal{F}_j$, several concepts related with θ-dominance can be defined as follows

DEFINITION 6. *Given two solutions $\mathbf{x}, \mathbf{y} \in \Omega$, $\mathbf{x}$ is said to θ-dominate $\mathbf{y}$, denoted by $\mathbf{x} \prec_\theta \mathbf{y}$, if $\mathbf{x} \in C_j$, $\mathbf{y} \in C_j$, and $\mathcal{F}_j(\mathbf{x}) < \mathcal{F}_j(\mathbf{y})$, where $j \in \{1, 2, \ldots, K\}$.*

DEFINITION 7. *A solution $\mathbf{x}^* \in \Omega$ is θ-optimal if there is no other solution $\mathbf{x} \in \Omega$ such that $\mathbf{x} \prec_\theta \mathbf{x}^*$*

DEFINITION 8. *All feasible solutions that are θ-optimal form the θ-optimal set (θ-OS), and the corresponding mappings of θ-OS in the objective space Θ is called θ-optimal front (θ-OF).*

Based on the definition of θ-dominance, we have the following two properties, which respectively illustrate that the relation $\prec_\theta$ is asymmetric and transitive.

PROPERTY 1. *If two solutions* $\mathbf{x}, \mathbf{y} \in \Omega$ *satisfy* $\mathbf{x} \prec_\theta \mathbf{y}$, *then* $\mathbf{y} \not\prec_\theta \mathbf{x}$

PROOF. Suppose $\mathbf{y} \prec_\theta \mathbf{x}$, then $\exists j \in \{1, 2, \ldots, K\}$, $\mathbf{y} \in C_j$, $\mathbf{x} \in C_j$, and $\mathcal{F}_j(\mathbf{y}) < \mathcal{F}_j(\mathbf{x})$. However, according to $\mathbf{x} \prec_\theta \mathbf{y}$, $\mathcal{F}_j(\mathbf{x}) < F_j(\mathbf{y})$. So, the supposition is invalid, and the proposition is true. □

PROPERTY 2. *If three solutions* $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \Omega$ *satisfy* $\mathbf{x} \prec_\theta \mathbf{y}$ *and* $\mathbf{y} \prec_\theta \mathbf{z}$, *then* $\mathbf{x} \prec_\theta \mathbf{z}$

PROOF. By $\mathbf{x} \prec_\theta \mathbf{y}$, we have that $\exists j \in \{1, 2, \ldots, K\}$, $\mathbf{x} \in C_j$, $\mathbf{y} \in C_j$, and $\mathcal{F}_j(\mathbf{x}) < \mathcal{F}_j(\mathbf{y})$. And according to $\mathbf{y} \prec_\theta \mathbf{z}$, then $\mathbf{z} \in C_j$, $\mathcal{F}_j(\mathbf{y}) < \mathcal{F}_j(\mathbf{z})$. Overall, $\mathbf{x} \in C_j$, $\mathbf{z} \in C_j$ and $\mathcal{F}_j(\mathbf{x}) < \mathcal{F}_j(\mathbf{z})$. Thus, $\mathbf{x} \prec_\theta \mathbf{z}$ □

Due to Property 1 and 2, the fast non-dominated sorting approach [9] can be directly used in θ-dominance sense, and the population R_t would be partitioned into different θ-non-domination levels.

4. EXPERIMENTAL STUDIES

4.1 Experimental Setup

The proposed θ-NSGA-III is implemented in the jMetal framework [14]. The well-known DTLZ1-4 problems [10] are used as test functions, where the number of objectives is varied by us between 4 to 20. In Figure 2, the PFs of DTLZ1 and DTLZ2 for three objectives are shown. The PFs of DTLZ3 and DTLZ4 are the same with DTLZ2, which only differ in the mapping from the decision space to the objective space. In order to evaluate the performance of the concerned algorithms, the generational distance (GD)[28] and inverted generational distance (IGD) [5] would be involved as performance metrics in our experiments, which can be explained as the following. Let P^* be a set of uniformly distributed points along the PF. Let A be an approximation to the PF. The metrics GD and IGD of the set A are respectively defined as:

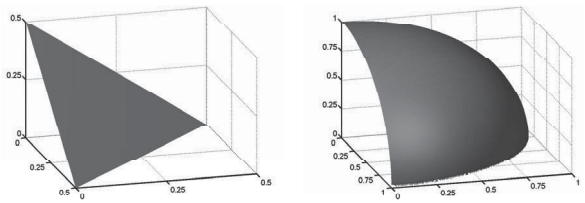
$$\mathrm{GD}(A, P^*) = \frac{1}{|A|}\sqrt{\sum_{i=1}^{|A|} d_i^2} \tag{6}$$

$$\mathrm{IGD}(A, P^*) = \frac{1}{|P^*|}\sqrt{\sum_{i=1}^{|P^*|} \tilde{d}_i^2} \tag{7}$$

where d_i ($\tilde{d}_i$) is the Euclidean distance between the i-th member in the set A (P^*) and its nearest member in the set P^* (A). GD can only reflect the convergence of an algorithm, while IGD could measure both convergence and diversity in a sense. For both of two metrics, smaller value means better quality.

To verify the effectiveness of proposed θ-NSGA-III, we compare its performance to the original NSGA-III [8] and MOEA/D [23]. In both NSGA-III and MOEA/D, the population size N is controlled by a parameter H ($N = C_{H+m-1}^{m-1}$), while in θ-NSGA-III, the population size can be arbitrary. But for ensuring a fair comparison, we adopt the uniform population size and the maximum number of fitness evaluations in all the three considered algorithms, which are listed in Table 1.

The other parameters of θ-NSGA-III are shown in Table 2. Here, we haven't decided the parameter θ, which would be further investigated in Section 4.2. As for NSGA-III and MOEA/D, their other parameters are just set according to the original papers. Each algorithm is run 30 times independently for each case, and average results are recoded.



(a) DTLZ1. (b) DTLZ2.

Figure 2: Visualization of PFs of DTLZ1 and DTLZ2 for three objectives.

Table 1: The population size and the maximum number of fitness evaluations.

m	H	N	Fitness Evaluations
4	9	220	100,100
5	6	210	99,960
6	5	252	100,044
8	4	330	99,990
10	3	220	100,100
12	3	364	100,100
14	3	560	100,240
16	3	816	100,368
18	3	1140	100,320
20	2	210	99,960

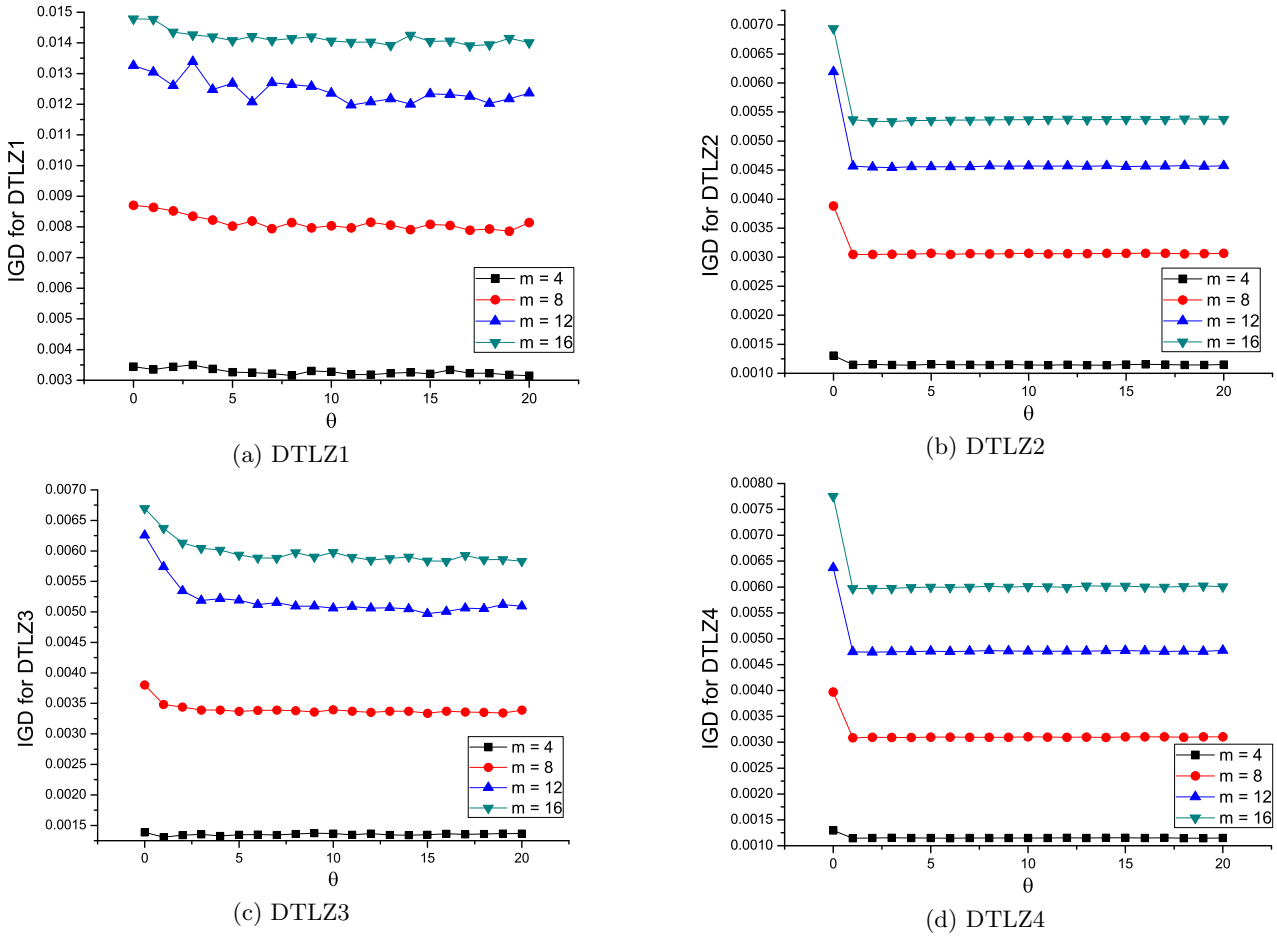
Table 2: The other parameters of θ-NSGA-III

Parameter	Value
Number of reference points (K)	N
Crossover probability (p_c)	0.9
Mutation probability (p_m)	$1/n$
Distribution index for crossover (η_c)	30
Distribution index for mutation (η_m)	20

4.2 Influence of Parameter θ

Compared to the original NSGA-III, θ-NSGA-III has an additional parameter θ. In this subsection, we will examine the influence of θ on the performance of θ-NSGA-III. IGD metric is used here, since it can reflect overall performance including both convergence and diversity. We consider the situation $m \in \{4, 8, 12, 16\}$ and $0 \leq \theta \leq 20$. Figure 3 shows that how the performance of θ-NSGA-III varies over the change of θ.

From Figure 3, it can be seen that, for DTLZ1, the IGD value presents slight fluctuations except for the 12-objective instance, where the IGD change seems more instable roughly between $\theta = 2$ and $\theta = 7$. For both DTLZ2 and DTLZ4, the variation trend of IGD is similar. That is, $\theta = 0$ would result in a relative poor performance, and when $\theta > 0$, IGD value is maintained nearly at the same level all through. As for DTLZ3, the IGD for 12-objective and 16-objective instances decreases smoothly at the beginning (approximately from $\theta = 0$ to $\theta = 5$), then the value become stable; the IGD for 4-objective instance has almost no variation all the while; the IGD for 8-objective instance is the worst when $\theta = 0$,



(a) DTLZ1 (b) DTLZ2 (c) DTLZ3 (d) DTLZ4

Figure 3: Investigation of the influence of θ on the performance of θ-NSGA-III for DTLZ 1-4 problems with varying number of objectives m. The figures show the average of 30 independent runs each.

and stays almost unchanged at the remaining θ values. To sum up, $\theta = 0$ is not recommended in our θ-NSGA-III since it nearly always leads to the worst IGD result, and the performance of θ-NSGA-III is robust over a wide range of θ values which is beneficial to the practical use of θ-NSGA-III. In the following experiments, θ is just set to be 5.

4.3 Comparison Results

In this subsection, we will present the comparison results between the proposed θ-NSGA-III, the original NSGA-III and MOEA/D. Table 3 provides the average GD and IGD of each compared algorithm for DTLZ1-4 problems having 4 to 20 objectives. In order to draw statistically sound conclusions, the Wilcoxon signed-rank test [11] at a 5% significance level is further carried out on the three average GD or IGD results obtained by three algorithms respectively for each instance, and the one that is significantly better than the other two is marked in bold face.

From Table 3, regarding GD metric, θ-NSGA-III performs significant better than NSGA-III and MOEA/D on 9 out of 10 DTLZ1 instances except for the 4-objective instance, where NSGA-III has the smallest GD value but there exists no statistical significance. For DTLZ2 and DTLZ4 problems, θ-NSGA-III exhibits obvious superiority since it is always significant better than NSGA-III and MOEA/D on all the concerned instances. As for DTLZ3, θ-NSGA-III significantly outperforms NSGA-III and MOEA/D on 6 out of 10 instances; NSGA-III performs best significantly only on 4-objective and 5-objective instances; on the remaining two instances, θ-NSGA-III still obtains the best GD values although the results are without significance. Since GD metric purely measures the convergence of an algorithm, it can be safely concluded that θ-NSGA-III generally shows the best convergence ability for many-objective problems among all the three algorithms. Specially, the fact θ-NSGA-III outperforms NSGA-III on GD metric illustrates that the proposed θ-dominance indeed improves the convergence of original NSGA-III in replace of the Pareto-dominance, which partly confirms the motivation for this study.

When it comes to IGD metric, θ-NSGA-III is significant better than NSGA-III and MOEA/D in most instances, including 7 DTLZ1 instances, 9 DTLZ2 instances, 8 DTLZ3 instances, and 9 DTLZ4 instances. θ-NSGA-III is significantly outperformed by NSGA-III on all 4-objective instances and the 5-objective DTLZ1 instance, but the gap between them seems not so clear, and θ-NSGA-III still performs better than MOEA/D on these instances. Because IGD metric combines the information of convergence and diversity into a whole, the results indicate that θ-NSGA-III has the best overall performance and is very promising in

Table 3: Comparison between θ-NSGA-III, NSGA-III, and MOEA/D

Problem	m	GD			IGD		
		θ-NSGA-III	NSGA-III	MOEA/D	θ-NSGA-III	NSGA-III	MOEA/D
DTLZ1	4	7.147367E-02	1.194571E-02	3.030940E-01	3.233821E-03	**2.622568E-03**	6.569777E-03
	5	**7.953799E-03**	8.694054E-03	1.421262E+00	4.897804E-03	**4.326618E-03**	1.731600E-02
	6	**9.282109E-03**	1.175294E-02	1.331502E+00	5.795215E-03	6.000549E-03	2.018833E-02
	8	**1.530516E-02**	3.414592E-02	5.305738E-01	**8.056204E-03**	1.260810E-02	2.805876E-02
	10	**5.432319E-02**	9.316793E-02	4.252602E-01	**1.170874E-02**	1.841571E-02	3.272573E-02
	12	**1.819230E-02**	1.195657E-01	4.218465E-01	**1.199624E-02**	2.745254E-02	2.854125E-02
	14	**1.721140E-02**	1.834626E-01	6.101847E-01	**1.307193E-02**	4.125902E-02	6.028564E-02
	16	**1.757701E-02**	1.215636E+00	1.194802E+00	**1.415559E-02**	7.177616E-02	1.097786E-01
	18	**2.242161E-02**	2.744731E+00	2.045014E+00	**1.542844E-02**	1.039266E-01	1.159088E-01
	20	**4.587515E-02**	8.450395E-01	1.285424E-01	**2.396450E-02**	6.738313E-02	3.351446E-02
DTLZ2	4	**2.046303E-03**	2.538939E-03	3.967537E-03	1.149039E-03	**1.100917E-03**	1.742903E-03
	5	**4.396954E-03**	6.125664E-03	7.795486E-03	**1.780862E-03**	2.012950E-03	3.415192E-03
	6	**6.410115E-03**	9.267768E-03	1.033922E-02	**2.256722E-03**	2.806871E-03	3.747893E-03
	8	**9.823255E-03**	1.478273E-02	1.363422E-02	**3.060328E-03**	4.321556E-03	4.677150E-03
	10	**1.693581E-02**	2.632225E-02	2.101903E-02	**4.075165E-03**	6.385276E-03	5.588852E-03
	12	**1.686319E-02**	2.612782E-02	2.019314E-02	**4.564802E-03**	7.338915E-03	6.308985E-03
	14	**1.598765E-02**	2.466134E-02	1.895783E-02	**4.929844E-03**	8.454116E-03	6.580670E-03
	16	**1.536926E-02**	2.353691E-02	1.818666E-02	**5.356541E-03**	9.205518E-03	6.930945E-03
	18	**1.451685E-02**	2.206333E-02	1.773535E-02	**5.704462E-03**	9.864807E-03	7.981181E-03
	20	**3.816497E-02**	5.874783E-02	4.465366E-02	**7.239501E-03**	1.277330E-02	1.010432E-02
DTLZ3	4	6.660327E-03	**3.028080E-03**	8.312969E-03	1.343317E-03	**1.145557E-03**	1.751199E-03
	5	9.058163E-03	**6.806226E-03**	1.063779E+00	2.042259E-03	2.096937E-03	6.577953E-03
	6	**9.159316E-03**	9.180469E-03	9.678572E-01	**2.533024E-03**	2.804483E-03	7.353213E-03
	8	**1.331580E-02**	1.707303E-02	3.425063E-01	**3.424340E-03**	5.141099E-03	8.537354E-03
	10	3.429092E-02	6.045586E-02	1.783019E-01	**4.742726E-03**	7.866186E-03	8.268878E-03
	12	**2.711708E-02**	1.241488E-01	5.711991E-02	**5.219557E-03**	1.000520E-02	9.794224E-03
	14	**2.158932E-02**	3.559196E-01	1.556527E-01	**5.538358E-03**	1.336140E-02	1.270166E-02
	16	**2.040819E-02**	1.209573E+00	5.214353E-01	**5.954946E-03**	2.165244E-02	2.516194E-02
	18	**3.284361E-02**	1.958558E+00	1.287473E+00	**6.616016E-03**	3.365187E-02	3.213193E-02
	20	7.879964E-02	2.509615E-01	1.026432E-01	**1.109914E-02**	1.566994E-02	1.168683E-02
DTLZ4	4	**2.037898E-03**	2.542045E-03	4.283030E-03	1.147480E-03	**1.100780E-03**	2.527278E-03
	5	**4.415283E-03**	6.123178E-03	7.945216E-03	**1.789928E-03**	2.014448E-03	4.075787E-03
	6	**6.433575E-03**	9.272363E-03	1.086709E-02	**2.270834E-03**	2.809236E-03	4.605452E-03
	8	**1.008925E-02**	1.475719E-02	1.450697E-02	**3.096551E-03**	4.316074E-03	5.951227E-03
	10	**1.740289E-02**	2.641742E-02	2.395872E-02	**4.139658E-03**	6.162423E-03	7.316202E-03
	12	**1.795626E-02**	2.614657E-02	2.346515E-02	**4.752917E-03**	7.073280E-03	8.515172E-03
	14	**1.782045E-02**	2.504145E-02	2.249369E-02	**5.313945E-03**	7.828505E-03	9.323785E-03
	16	**1.785509E-02**	2.457113E-02	2.277746E-02	**5.990658E-03**	8.725350E-03	1.029241E-02
	18	**1.756051E-02**	2.358901E-02	2.209823E-02	**6.667148E-03**	9.509864E-03	1.090675E-02
	20	**4.253799E-02**	6.011912E-02	5.007374E-02	**7.821877E-03**	1.133464E-02	1.152812E-02

many-objective optimization compared with state-of-the-art algorithms.

5. CONCLUSION

In this paper, we have presented a new many-objective evolutionary algorithm, referred as θ-NSGA-III, whose selection mechanism is based on θ-dominance. θ-NSGA-III aims to improve the convergence ability of the original NSGA-III, so as to better balance the convergence and diversity in many-objective optimization. To achieve this goal, a new preference relation, θ-dominance, is introduced into the proposed algorithm, which emphasizes both convergence and diversity. Through the experiments, the performance of θ-NSGA-III is found to be insensitive to the parameter θ, which could perform well over a wide range of θ values. We validate the effectiveness of θ-NSGA-III using well-known test problems having 4 to 20 objectives. The comparison results with two established algorithms (NSGA-III and MOEA/D) shows that θ-NSGA-III significantly outperforms NSGA-III and MOEA/D on most instances no matter in convergence and overall performance. In the future, we will incorporate the diversity enhancement strategy into θ-NSGA-III. And also, we would like to apply θ-NSGA-III to more benchmark problems and real-world problems.

Acknowledgment

This work is supported by National Natural Science Foundation of China (Grant No. 61175110), National Basic Research Program of China (973 Program) (Grant No. 2012CB 316305), National S&T Major Projects of China (Grant No. 2011ZX02101-004) and National Banking Information Technology Risk Management Projects of China.

6. REFERENCES

[1] S. F. Adra and P. J. Fleming. Diversity management in evolutionary many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 15(2):183–195, 2011.

[2] J. Bader and E. Zitzler. Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 19(1):45–76, 2011.

[3] N. Beume, B. Naujoks, and M. Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 2007.

[4] V. Chankong and Y. Y. Haimes. *Multiobjective Decision Making Theory and Methodology*. New York: North-Holland, 1983.

[5] C. A. C. Coello and N. C. Cortés. Solving multiobjective optimization problems using an artificial immune system. *Genetic Programming and Evolvable Machines*, 6(2):163–190, 2005.

[6] I. Das and 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):631–657, 1998.

[7] K. Deb and R. B. Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9:1–34, 1994.

[8] K. Deb and J. Himanshu. An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part I: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, in press.

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

[10] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. *Scalable test problems for evolutionary multiobjective optimization*. TIK-Technical Report No. 112, 2001.

[11] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.

[12] F. di Pierro, S.-T. Khu, and D. A. Savic. An investigation on preference order ranking scheme for multiobjective evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 11(1):17–45, 2007.

[13] N. Drechsler, R. Drechsler, and B. Becker. Multi-objective optimisation based on relation favour. In *Evolutionary multi-criterion optimization*, pages 154–166. Springer, 2001.

[14] J. J. Durillo and A. J. Nebro. jmetal: A java framework for multi-objective optimization. *Advances in Engineering Software*, 42(10):760–771, 2011.

[15] M. Farina and P. Amato. A fuzzy definition of "optimality" for many-criteria optimization problems. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 34(3):315–326, 2004.

[16] D. E. Goldberg. *Genetic algorithms in search, optimization, and machine learning*. Addison-wesley Reading Menlo Park, 1989.

[17] J. G. Herrero, A. Berlanga, and J. M. M. Lopez. Effective evolutionary algorithms for many-specifications attainment: application to air traffic control tracking filters. *IEEE Transactions on Evolutionary Computation*, 13(1):151–168, 2009.

[18] A. Jaszkiewicz. Momh multiple-objective metaheuristics. Apr. 2013. [Online]. Available: http://home.gna.org/momh/index.html.

[19] M. Köppen and K. Yoshida. Substitute distance assignments in nsga-ii for handling many-objective optimization problems. In *Evolutionary Multi-Criterion Optimization*, pages 727–741. Springer, 2007.

[20] S. Kukkonen and K. Deb. Improved pruning of non-dominated solutions based on crowding distance for bi-objective optimization problems. In *IEEE Congress on Evolutionary Computation (CEC) 2006*, pages 1179–1186. IEEE, 2006.

[21] S. Kukkonen and J. Lampinen. Ranking-dominance and many-objective optimization. In *IEEE Congress on Evolutionary Computation (CEC) 2007*, pages 3983–3990. IEEE, 2007.

[22] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary computation*, 10(3):263–282, 2002.

[23] H. Li and Q. Zhang. Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation*, 13(2):284–302, 2009.

[24] M. Li, J. Zheng, K. Li, Q. Yuan, and R. Shen. Enhancing diversity for average ranking method in evolutionary many-objective optimization. In *Parallel Problem Solving from Nature, PPSN XI*, pages 647–656. Springer, 2010.

[25] A. López, C. A. C. Coello, A. Oyama, and K. Fujii. An alternative preference relation to deal with many-objective optimization problems. In *Evolutionary Multi-Criterion Optimization*, pages 291–306. Springer, 2013.

[26] W. Mkaouer, M. Kessentini, S. Bechikh, K. Deb, and M. Ó. Cinnéide. High dimensional search-based software engineering: Finding tradeoffs among 15 objectives for automating software refactoring using NSGA-III. COIN Report No. 2013002, 2013.

[27] A. Sülflow, N. Drechsler, and R. Drechsler. Robust multi-objective optimization in high dimensional spaces. In *Evolutionary multi-criterion optimization*, pages 715–726. Springer, 2007.

[28] D. A. Van Veldhuizen. Multiobjective evolutionary algorithms: classifications, analyses, and new innovations. Technical report, DTIC Document, 1999.

[29] Q. Zhang and H. Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007.

[30] E. Zitzler and S. Künzli. Indicator-based selection in multiobjective search. In *Parallel Problem Solving from Nature-PPSN VIII*, pages 832–842. Springer, 2004.

[31] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm. In *Proceedings of Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems*, pages 95–100, 2001.