Dynamic Normalization in MOEA/D for Multiobjective Optimization

Linjun He*†, Hisao Ishibuchi*, Anupam Trivedi† and Dipti Srinivasan†
*Shenzhen Key Laboratory of Computational Intelligence,
University Key Laboratory of Evolving Intelligent Systems of Guangdong Province,
Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China
†Department of Electrical and Computer Engineering, National University of Singapore, Singapore
Email: this.helj@gmail.com, hisao@sustech.edu.cn, eleatr@nus.edu.sg, dipti@nus.edu.sg

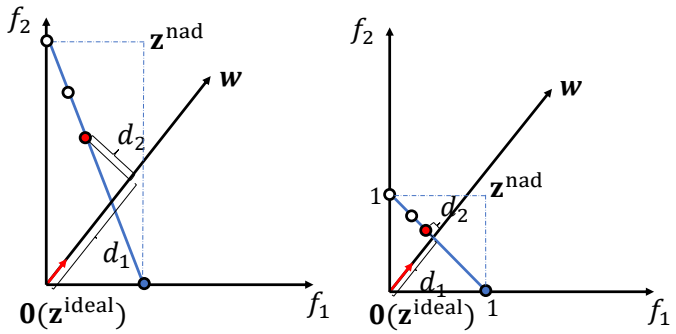
***Abstract*—Objective space normalization is important since a real-world multiobjective problem usually has differently scaled objective functions. Recently, bad effects of the commonly used simple normalization method have been reported for the popular decomposition-based algorithm MOEA/D. However, the effects of recently proposed sophisticated normalization methods have not been investigated. In this paper, we examine the effectiveness of these normalization methods in MOEA/D. We find that these normalization methods can cause performance deterioration. We also find that the sophisticated normalization methods are not necessarily better than the simple one. Although the negative effects of inaccurate estimation of the nadir point are well recognized in the literature, no solution has been proposed. In order to address this issue, we propose two dynamic normalization strategies which dynamically adjust the extent of normalization during the evolutionary process. Experimental results clearly show the necessity of considering the extent of normalization.**

***Index Terms*—Objective space normalization; decomposition-based algorithms; MOEA/D; evolutionary multiobjective optimization**

I. Introduction

A real-world optimization problem usually involves multiple mutually conflicting objective functions, known as a multiobjective optimization problem (MOP) [1]. No single solution can minimize all objective functions at the same time; by contrast, a set of Pareto optimal solutions is obtained owing to the trade-offs among different objectives. In the past two decades, evolutionary algorithms have shown its superiority in solving such problems. A number of multiobjective evolutionary algorithms have been developed to handle various MOPs. These algorithms mainly fall into three classes: dominance-based, indicator-based, and decomposition-based approaches [1], [2].

MOEA/D [3] is one of the most popular decomposition-based EMO algorithms, which solves the MOP by decomposing it into several scalar optimization problems with the help of uniformly distributed weight vectors. For each weight vector, one scalar optimization problem is generated using a scalarizing function. The penalty-based boundary intersection (PBI) function [3] is the most commonly used scalarizing function due to its effectiveness in solving multiobjective problems with more than three objectives [4], [5]. Given a solution $\mathbf{x}$ and its corresponding weight vector $\mathbf{w}$, the PBI function returns $d_1 + \theta d_2$, where d_1 is the projected distance of the solution $\mathbf{x}$ along the weight vector $\mathbf{w}$, and d_2 is the perpendicular distance of the solution $\mathbf{x}$ toward the weight vector $\mathbf{w}$, as shown in Fig 1 (a). Here, θ is a user-defined penalty parameter (we use $\theta = 5$ as in [3]).



(a) Original Objective Space. (b) Normalized Objective Space.

Fig. 1: Illustration of the effect of the objective space normalization when the PBI function is used for fitness evaluation. The solid blue line is the Pareto front.

One important property, disparately-scaled objectives, of a real-world MOP has drawn researchers' attention recently [6], [7]. The multiple objective functions are usually measured in different units, resulting in differently-scaled objectives. In order to handle optimization problems with differently-scaled objectives, objective space normalization is always encouraged and is even sometimes mandatory in an evolutionary multiobjective algorithm [6]. A simple normalization method mentioned in [3] always gains researcher's favor and is frequently used in the literature [8], [9].

Although objective space normalization plays a non-neglectable role in solving MOPs, its effects have rarely been

This work was supported by National Natural Science Foundation of China (Grant No. 61876075), the Program for Guangdong Introducing Innovative and Enterpreneurial Teams (Grant No. 2017ZT07X386), Shenzhen Peacock Plan (Grant No. KQTD2016112514355531), the Science and Technology Innovation Committee Foundation of Shenzhen (Grant No. ZDSYS201703031748284), the Program for University Key Laboratory of Guangdong Province (Grant No. 2017KSYS008). (*Corresponding author: Hisao Ishibuchi*.)

studied [8], [10]. Since the fitness evaluation in MOEA/D is completely based on the distance calculation, the ranking of two solutions can be heavily dependent on the scaling of objective functions [8], [9].

An example is shown in Fig. 1. The red point is worse than the blue point in the original objective space in Fig. 1 (a), while it is better in the normalized objective space in Fig. 1 (b). Thus, it is important to examine the effects of different normalization methods in MOEA/D. In [8], the authors examine the effects of the simple normalization method in MOEA/D, where both positive and negative effects are reported. In [9], four different implementations of the simple normalization method are presented and examined. Experimental results show that all of these implementations lead to certain performance deterioration of MOEA/D on some test problems. Some sophisticated normalization methods [7], [11], [12] are proposed recently. Extreme point identification and hyperplane construction are needed before deriving the estimated nadir point, which makes them different from the simple normalization method.

In this paper, the simple normalization method and two sophisticated normalization methods are compared in the framework of MOEA/D. Our results clearly show that these sophisticated normalization methods are even worse than the simple normalization method. In order to address the performance deterioration, two dynamic normalization strategies are proposed by considering the extent of normalization. Experimental results clearly show that our proposed strategies help remedy the performance deterioration of different normalization methods.

The rest of the paper is organized as follows. In Section II, we present an overview of objective space normalization. Our proposed normalization strategies are presented in Section III and their effectiveness is experimentally validated in Section IV. In Section V, we conclude the paper.

II. Preliminaries

A. How Objective Space Normalization is Performed?

Objective space normalization is usually performed by using the ideal point $\mathbf{z}^{\text{ideal}} = (z_1^{\text{ideal}}, z_2^{\text{ideal}}, ..., z_m^{\text{ideal}})^T$ and the nadir point $\mathbf{z}^{\text{nad}} = (z_1^{\text{nad}}, z_2^{\text{nad}}, ..., z_m^{\text{nad}})^T$ for an m-objective optimization problem. As illustrated in Fig. 1 (a), these two points contain the information on the range of each objective in the true Pareto front. Given the decision variable space Ω and the Pareto optimal set PS, the ideal point gives the lower bound of the Pareto front, i.e., $z_i^{\text{ideal}} = \min_{\mathbf{x}\in\Omega} f_i(\mathbf{x}),\ i = 1, 2, \ldots, m$, and the nadir point gives the upper bound of the Pareto front, i.e., $z_i^{\text{nad}} = \min_{\mathbf{x}\in\text{PS}} f_i(\mathbf{x}),\ i = 1, 2, \ldots, m$.

With the ideal and nadir points, objective space normalization can be done by the following two steps:

1) *Objective function translation*: Each objective function is translated as

$$f_i'(\mathbf{x}) = f_i(\mathbf{x}) - z_i^{\text{ideal}}, i = 1, 2, \ldots, m. \quad (1)$$

After objective function translation, the ideal point of the Pareto front is translated to the origin of the coordinate system [3], [7]. This method is used in the standard MOEA/D [3], in which the ideal point is translated to $(0, 0, ..., 0)$, so that the weight vectors start from the origin. It is worth noting that the translated objectives still have differently scaled objective values.

2) *Objective function scaling*: This step relies on both the ideal and nadir points. The translated objective functions are normalized as follows [8], [13]:

$$\tilde{f}_i(\mathbf{x}) = \frac{f_i'(\mathbf{x})}{z_i^{\text{nad}} - z_i^{\text{ideal}}}, i = 1, 2, \ldots, m. \quad (2)$$

After normalization, objective values are commensurable and of approximately the same order of magnitude.

However, the ideal and nadir points are not always known a priori in real-world MOPs [14]. The ideal point can be obtained by minimizing each objective function separately while the nadir point needs the information of the Pareto front, which makes the calculation of the nadir point a difficult task [14]. For nadir point offline estimation methods, one can refer to [14]–[16]. In practice, these two points are usually estimated during the evolutionary process in an online manner.

Objective space normalization using estimated ideal and nadir points is often referred to as adaptive normalization [17] because the ideal and nadir points are estimated adaptively at every generation. We denote the estimated ideal and nadir points as $\mathbf{z}^{\text{min}} = (z_1^{\text{min}}, z_2^{\text{min}}, \ldots, z_m^{\text{min}})^T$ and $\mathbf{z}^{\text{max}} = (z_1^{\text{max}}, z_2^{\text{max}}, \ldots, z_m^{\text{max}})^T$, respectively.

The ideal point can be estimated by taking the minimum value of each objective function from a candidate solution set. As reported in [9], [18], the estimated ideal point can approach to the true ideal point very fast, especially when all solutions examined so far are selected as the candidate solution set. Usually, the ideal point is estimated from the best objective value found so far for each objective [19]–[23]. We use this setting to estimate the ideal point in this paper. However, there are different methods to estimate the nadir point due to its difficulty. The main difference between different normalization methods lies in the way of estimating the nadir point.

B. Simple Normalization Method

In the simple normalization method, the nadir point is estimated as follows:

$$z_i^{\text{max}} = \max_{\mathbf{x}\in S} f_i(\mathbf{x}),\ i \in \{1, 2, \ldots, m\}, \quad (3)$$

where S is a candidate solution set. The current solution set can be either the current population or its non-dominated solutions. Due to its simplicity, this is the most commonly used normalization method.

In this paper, we use non-dominated solutions in the current population as the candidate solution set according to [9]. For other implementations, one can refer to [9].

C. Hyperplane-based Normalization Methods

In the hyperplane-based normalization method, the nadir point estimation involves extreme point identification and hyperplane construction, which makes it more sophisticated

than the commonly used simple normalization method. To be more specific, for an m-objective optimization problem, given a candidate solution set S, the objective function translation is firstly performed. Then, m extreme points are identified and a hyperplane is constructed by using the m extreme points. Finally, the estimated nadir point $\mathbf{z}^{\max}$ is derived from the intercept of the hyperplane with each objective axis. For details of hyperplane construction, one can refer to [11], [22]. The main difference between different hyperplane-based normalization methods lies in their extreme point identification methods. Two extreme point identification methods, which are used in NSGA-III [7] and θ-DEA [11], are explained below.

1) NSGA-III [7]: The extreme point of each objective is defined as the solution that minimizes the following achievement scalarizing function:

$$\begin{aligned} ASF(\mathbf{x},\mathbf{w}) &= \max_{i=1}^{m} f_i'(\mathbf{x})/w_i, \text{ for } \mathbf{x}\in S, \\ w_i &= \begin{cases} 1 & i=k \\ 10^{-6} & i\neq k \end{cases}, \ i\in\{1,2,\ldots,m\}, \end{aligned} \tag{4}$$

where $\mathbf{w}$ is a weight vector with the axis direction corresponding to one objective. For the k-th objective, we have $w_k = 1$ and $w_l = 0$ for $l \neq k$. Notice that $w_l = 0$ here is often replaced by a small number $w_l = 10^{-6}$.

In other words, this extreme point identification method is to identify the closest solution to each objective axis by calculating the Tchebychev distance. This extreme point identification method is used in many recently-proposed algorithms, such as DBEA-Eps [24], DBEA [25], DoD [26], LEAF [27], DECAL [21].

2) θ-DEA [11]: A slightly different extreme point identification method is used in θ-DEA [11]. The extreme points are identified by minimizing the following function:

$$\mathrm{ASF}(\mathbf{x},\mathbf{w}) = \max_{i=1}^{m}\left\{\left|\frac{f_i'(\mathbf{x})}{\dot{z}_i^{\max}-\dot{z}_i^{\min}}\right|/w_i\right\}, \text{ for } \mathbf{x}\in S. \tag{5}$$

This extreme point identification method is also used by many recently-proposed algorithms, such as MOEA/D-DU [28], EFR-RR [28], MOEA/D-AU [29], and ASEA [20].

Similar to NSGA-III [7], the extreme point identification method in θ-DEA [11] also identifies the closest solution to each objective axis by calculating the Tchebychev distance. The only difference is that this method is performed in the normalized objective space. The objective space is normalized before the extreme point identification using the estimated ideal and nadir points in the previous generation. This might help to more accurately identify the extreme points because the Tchebychev distance is a distance metric (which is scaling-dependent). However, if the estimated ideal and nadir points are not accurate, the use of the normalized space may deteriorate the identification process.

Let us denote the candidate solution set S at the t-th generation by S_t. A frequently used method for specifying S_t is to choose the best fronts (based on the non-dominated sorting) containing at least N solutions from the current population [7], [30], where N is the population size. In MOEA/D, the candidate solution set S_t is almost the same as the current population, since only one offspring is generated at each generation. To make a fair comparison, we also consider the non-dominated solution set in the current population, which is used in the simple normalization method in this paper.

It is worth noting that the hyperplane-based nadir point estimation method might fail to obtain reasonable intercepts. The hyperplane can degenerate when the identified extreme points are less than m. Even when the hyperplane is constructed, there exist no intercept when the hyperplane is parallel to certain axes. The intercept can be also negative when the hyperplane intersects an axis below the origin. In NSGA-III [7], any countermeasure against these cases is not presented. Thus, we use the method in θ-DEA [11]. When these degenerated cases happen, the nadir point is estimated by the simple normalization method.

III. Comparisons of Three Normalization Methods

In this section, the three normalization methods (i.e., the simple normalization method and the two hyperplane-based normalization methods) described in Section II are compared. MOEA/D without normalization is also compared in this section. We denote MOEA/D with each normalization method as MOEA/D-N, MOEA/D-N$_N$, and MOEA/D-N$_\theta$, respectively. All the algorithms used in this paper are implemented in PlatEMO [31].

A. Experimental Settings

Our experiments are conducted on the DTLZ test suite [32] with 3, 5, 8, and 10 objectives. The parameter settings of DTLZ are taken from [32]. The population size N and the maximum number of function evaluations are the same as [11]. The neighborhood size of MOEA/D is set as $\lceil N/10\rceil$, i.e., 10 percent of the population size. When $N/10$ is not an integer, we round it to the smallest integer greater than $N/10$. A small value ϵ is usually used in the denominator of (2) to prevent the denominator from being zero in the case of $z_i^{\max} = z_i^{\min}$. We set $\epsilon = 10^{-6}$ in our experiments. To evaluate the quality of the final solution sets, we use the hypervolume metric [33] since it can evaluate the overall performance of a solution set with respect to both its convergence and diversity. The final solution sets are normalized using the true Pareto front before hypervolume calculation. The reference point for hypervolume calculation is specified as $(1.1, 1.1, ..., 1.1)$ in the normalized objective space.

B. Experimental Results on DTLZ

The mean hypervolume values over 21 independent runs of MOEA/D, MOEA/D-N, MOEA/D-N$_N$, and MOEA/D-N$_\theta$ on the DTLZ problems are shown in Table I. We use the Wilcoxon rank sum test at the confidence level of 95% to show the statistical difference of the results. '$+$' and '$-$' indicate that the corresponding result is significantly better than and worse than that of the compared algorithm, respectively. '$\approx$' means

TABLE I: Hypervolume values obtained by MOEA/D and MOEA/D with each of the three objective space normalization methods on the DTLZ test suite with 3, 5, 8, and 10 objectives. The best value of each row is highlighted with bold typeface, while the worst with gray background.

Problem	M	MOEA/D-N	MOEA/D-N$_N$		MOEA/D-N$_\theta$		MOEA/D
			S_t	Non-dominated	S_t	Non-dominated	
DTLZ1	3	8.3772e-1 $\approx$	**8.3885e-1** $\approx$	8.3814e-1 $\approx$	8.3737e-1 $\approx$	8.3835e-1 $\approx$	8.3861e-1
	5	9.7774e-1 $-$	9.7777e-1 $-$	9.7864e-1 $-$	9.7950e-1 $\approx$	9.7942e-1 $\approx$	**9.7974e-1**
	8	6.5773e-1 $-$	3.9361e-1 $-$	3.3686e-1 $-$	3.6825e-1 $-$	3.7284e-1 $-$	**9.9729e-1**
	10	8.9745e-1 $-$	4.8018e-1 $-$	6.4365e-1 $-$	6.6534e-1 $-$	5.8068e-1 $-$	**9.9965e-1**
DTLZ2	3	5.5918e-1 $\approx$	5.5917e-1 $\approx$	5.5919e-1 $\approx$	5.5919e-1 $\approx$	5.5918e-1 $\approx$	**5.5919e-1**
	5	8.1217e-1 $\approx$	8.1205e-1 $\approx$	8.1217e-1 $\approx$	**8.1231e-1** $\approx$	8.1222e-1 $\approx$	8.1217e-1
	8	8.7090e-1 $\approx$	9.2349e-1 $-$	9.2208e-1 $\approx$	9.2294e-1 $-$	9.2352e-1 $\approx$	**9.2385e-1**
	10	9.6986e-1 $\approx$	9.6977e-1 $\approx$	9.6987e-1 $\approx$	9.6982e-1 $\approx$	**9.6988e-1** $\approx$	9.6977e-1
DTLZ3	3	5.5213e-1 $\approx$	5.5342e-1 $\approx$	5.5357e-1 $\approx$	5.5150e-1 $-$	5.5412e-1 $\approx$	**5.5495e-1**
	5	7.2799e-1 $-$	7.7517e-1 $-$	7.6366e-1 $-$	7.7936e-1 $-$	7.5341e-1 $-$	**8.0985e-1**
	8	1.6655e-1 $-$	2.8039e-1 $-$	2.7657e-1 $-$	3.2681e-1 $-$	2.9359e-1 $-$	**7.8473e-1**
	10	1.2531e-1 $-$	2.8586e-1 $-$	2.7533e-1 $-$	2.5856e-1 $-$	2.5800e-1 $-$	**9.3318e-1**
DTLZ4	3	**4.2693e-1** $\approx$	3.7043e-1 $-$	2.4868e-1 $-$	3.3795e-1 $-$	3.0524e-1 $-$	4.2142e-1
	5	**7.7390e-1** $\approx$	3.5170e-1 $-$	5.4389e-1 $-$	4.5003e-1 $-$	4.4506e-1 $-$	7.6644e-1
	8	**8.9051e-1** $+$	4.7226e-1 $-$	5.2807e-1 $-$	4.3009e-1 $-$	6.2024e-1 $-$	8.4422e-1
	10	**9.5590e-1** $\approx$	5.4502e-1 $-$	5.0243e-1 $-$	4.2324e-1 $-$	5.8981e-1 $-$	9.4275e-1
$+/-/\approx$		1/6/9	0/11/5	0/10/6	0/11/5	0/9/7	

'$+$' and '$-$' mean that the corresponding result is significantly better than and worse than that of MOEA/D, respectively. '$\approx$' means that the corresponding result is statistically similar to that of MOEA/D.

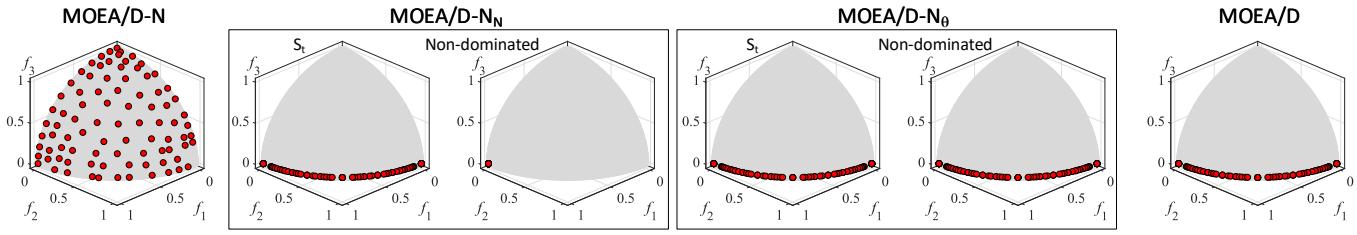


Fig. 2: Obtained solutions by a single run of each algorithm with the median hypervolume value on the 3-objective DTLZ4.

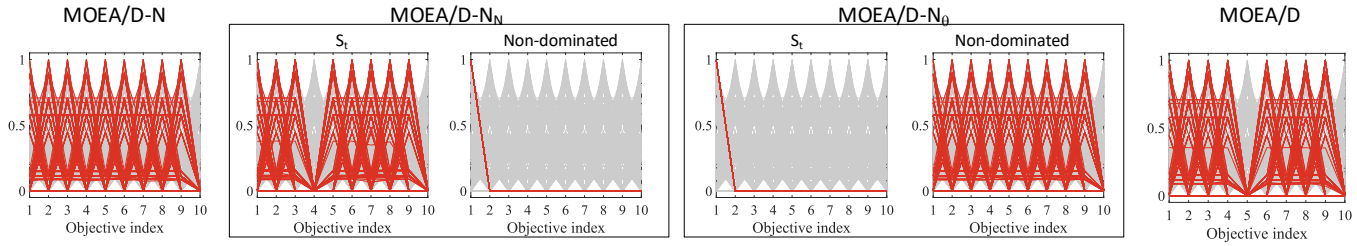


Fig. 3: Obtained solutions by a single run of each algorithm with the median hypervolume value on the 10-objective DTLZ4.

that the corresponding result is statistically similar to that of the compared algorithm.

As can be observed from Table I, MOEA/D without any normalization method obtains the best mean hypervolume values on 9 out of the 16 test problems. From the statistical test results, we can also see that MOEA/D is significantly better than or statistically similar to MOEA/D-N, MOEA/D-N$_N$, and MOEA/D-N$_\theta$ on almost all DTLZ test problems.

For MOEA/D-N, we can observe clear performance deterioration on DTLZ1 and DTLZ3 when the objective normalization method is used, especially on problems with 8 and 10 objectives, while performance improvement is observed on all DTLZ4 problems. Fig. 2 and Fig. 3 show obtained solutions by a single run of each algorithm on the 3- and 10-objecrtive DTLZ4 problems, respectively. As shown in Fig. 2 and Fig. 3, MOEA/D-N can achieve better diversity than MOEA/D on the DTLZ4 problems. It is worth noting that MOEA/D-N still fails to find uniformly distributed solutions over the Pareto fronts of the DTLZ4 problems, as shown in Fig. 2 and Fig. 3, although it achieves the best results among the compared algorithms.

For MOEA/D-N$_N$ and MOEA/D-N$_\theta$, statistically similar results to MOEA/D are obtained on DTLZ2, while perfor-

mance deterioration can be observed on the rest of the DTLZ problems, especially when the number of objectives is larger than 3. The selection of the candidate solution set, S_t or the non-dominated solution set, does not have a large effect on the performance of MOEA/D-N$_N$ and MOEA/D-N$_\theta$. Similar results are obtained by MOEA/D-N$_N$ and MOEA/D-N$_\theta$ when different candidate solution sets are used. Therefore, in the next section, we only consider the candidate solution set S_t.

Comparing MOEA/D-N$_N$ and MOEA/D-N$_\theta$ with MOEA/D-N, we can see that the two sophisticated normalization methods do not show superiority over the simple normalization method. Even worse results are obtained by the two sophisticated normalization methods. As shown in Fig. 2 and Fig. 3, MOEA/D-N can find much more diverse solutions than MOEA/D-N$_N$ and MOEA/D-N$_\theta$ on the DTLZ4 problems.

IV. Proposed Dynamic Normalization Strategies

The negative effects of objective space normalization result from the inaccurate estimation of the nadir point [8], [18]. At early generations, the population is far away from the Pareto front and the estimated nadir point is not able to approximate the upper bound of the Pareto front. With the inaccurately estimated nadir point, the heavily rescaled objective space misleads the search direction, which then makes the nadir point estimation even worse [8]. Although this negative cycle is well recognized in the literature, objective space normalization is still performed at the beginning of the evolutionary process.

In this paper, we propose to control the extent of the objective space normalization. The objective space normalization formula (2) is reformulated as follows:

$$\tilde{f}_i(\mathbf{x}) = \frac{f_i(\mathbf{x}) - z_i^{\min}}{L_i + \epsilon}, i = 1, 2, \ldots, m, \tag{6}$$

$$L_i = \frac{z_i^{\max} - z_i^{\min}}{(1-\alpha)(z_i^{\max} - z_i^{\min} - 1) + 1}, \tag{7}$$

where α is a parameter with a value in the range of $[0, 1]$ that controls the extent of normalization. When $\alpha = 1$, we have $L_i = z_i^{\max} - z_i^{\min}$ (i.e., exactly the same as (2)). When $\alpha = 0$, we have $L_i = 1$, which makes the formula (6) independent of the estimated nadir point. That is, only the objective function translation in (1) happens.

To the best of the authors' knowledge, only these two extreme cases are considered in the literature: 1) when $\alpha = 0$, no normalization is performed; 2) when $\alpha = 1$, normalization is performed. Despite the fact that the population improves with generations (i.e., the estimation of the nadir point and the ideal point improves as the population evolves), no researcher has considered controlling the extent of normalization.

By adjusting the value of α, we can control the extent of normalization (i.e., the reliance on the estimated nadir point). When the nadir point estimation is not reliable, a small value of α can be used to reduce the extent of normalization (i.e., to reduce the influence of the inaccurate range $z_i^{\max} - z_i^{\min}$). As the accuracy of the nadir point improves, a large value of α can be used to increase the extent of normalization.

Based on the assumption that the accuracy of the estimated nadir point improves with generations, we simply increase the value of α linearly from 0 at the initial generation to 1 at the final generation as follows:

$$\alpha(t) = \frac{t-1}{T-1}, \tag{8}$$

where t is the index of the current generation and T is the total number of generations ($t = 1, 2, \ldots, T$). As illustrated in Fig. 4, the value of α changes from 0 to 1. As a result, L_i in (6) changes from $L_i = 1$ to $L_i = z_i^{\max} - z_i^{\min}$.

Instead of the linear function, we can also use the sigmoid function (9) since it places less reliance on the estimated nadir point at early generations and more reliance at late generations.

$$\alpha(t) = \frac{1}{1 + \exp(-8(\frac{t-1}{T-1} - 0.5))}. \tag{9}$$

A normalization method with these two dynamic strategies are named as the linear dynamic normalization (LDN) and the sigmoid dynamic normalization (SDN), respectively.

V. Effectiveness of Proposed Strategies

In this section, we validate the effectiveness of the two proposed dynamic normalization strategies by integrating them into the three normalization methods in Section II. They are denoted by MOEA/D-LDN, MOEA/D-SDN, MOEA/D-LDN$_N$, MOEA/D-SDN$_N$, MOEA/D-LDN$_\theta$, and MOEA/D-SDN$_\theta$, respectively. They are compared to their original versions on the DTLZ [32] and WFG [6] test suites.

The mean hypervolume value over 21 independent runs of each algorithm on each test problem is shown in Table II-IV. We only show the results on the problems with 8 and 10 objectives to make the tables more concise. This is because the performance deterioration mainly happens when the number of objectives is large.

As we can see from Table II, MOEA/D-SDN achieves the best results on 13 out of the 20 test problems in terms

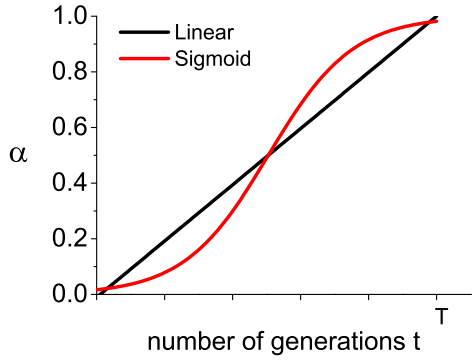


Fig. 4: Illustration of the value of α over the number of generations.

TABLE II: Hypervolume values obtained by MOEA/D-N and MOEA/D-N with each of the two dynamical normalization strategies on the DTLZ and WFG test suites with 8 and 10 objectives. The best value of each row is highlighted with bold typeface, while the worst with gray background.

Problem	M	MOEA/D-SDN	MOEA/D-LDN	MOEA/D-N
DTLZ1	8	**9.9702e-1** +	9.9569e-1 +	6.5773e-1
	10	**9.9960e-1** +	9.9370e-1 +	8.9745e-1
DTLZ2	8	**9.2388e-1** ≈	9.2383e-1 ≈	8.7090e-1
	10	9.6979e-1 ≈	9.6982e-1 ≈	**9.6986e-1**
DTLZ3	8	**9.1472e-1** +	8.9009e-1 +	1.6655e-1
	10	**8.8517e-1** +	7.9464e-1 +	1.2531e-1
DTLZ4	8	9.0702e-1 +	**9.0952e-1** +	8.9051e-1
	10	9.6486e-1 +	**9.6662e-1** +	9.5590e-1
WFG4	8	**9.2017e-1** +	9.1486e-1 +	8.8237e-1
	10	9.6819e-1 ≈	9.6429e-1 −	**9.6909e-1**
WFG5	8	**8.6471e-1** +	8.6349e-1 ≈	8.6240e-1
	10	9.0605e-1 +	**9.0608e-1** ≈	9.0470e-1
WFG6	8	**8.3311e-1** +	8.1992e-1 +	4.4042e-1
	10	8.6567e-1 ≈	**8.6919e-1** ≈	5.3689e-1
WFG7	8	**8.5900e-1** +	8.4500e-1 +	4.3726e-1
	10	**9.2084e-1** +	9.0879e-1 +	5.0426e-1
WFG8	8	**6.1252e-1** +	5.7822e-1 +	0.9584e-1
	10	**7.4313e-1** +	6.1754e-1 +	0.9539e-1
WFG9	8	**7.8041e-1** +	7.0549e-1 +	3.0510e-1
	10	7.7476e-1 +	**8.1486e-1** +	4.3554e-1
$+/-/\approx$		16/0/4	14/1/5	

TABLE III: Hypervolume values obtained by MOEA/D-N$_N$ and MOEA/D-N$_N$ with each of the two dynamical normalization strategies on the DTLZ and WFG test suites with 8 and 10 objectives. The best value of each row is highlighted with bold typeface, while the worst with gray background.

Problem	M	MOEA/D-SDN$_N$	MOEA/D-LDN$_N$	MOEA/D-N$_N$
DTLZ1	8	**9.9731e-1** +	9.9729e-1 +	3.9361e-1
	10	**9.9966e-1** +	9.9966e-1 +	4.8018e-1
DTLZ2	8	9.2384e-1 +	**9.2388e-1** +	9.2349e-1
	10	9.6973e-1 ≈	9.6973e-1 ≈	**9.6977e-1**
DTLZ3	8	9.1922e-1 +	**9.1961e-1** +	2.8039e-1
	10	**9.6969e-1** +	9.6950e-1 +	2.8586e-1
DTLZ4	8	**9.1439e-1** +	9.0941e-1 +	4.7226e-1
	10	**9.6488e-1** +	9.6468e-1 +	5.4502e-1
WFG4	8	9.2018e-1 ≈	9.1303e-1 −	**9.2169e-1**
	10	9.6825e-1 ≈	9.6328e-1 −	**9.6936e-1**
WFG5	8	**8.6412e-1** +	8.6379e-1 +	6.0932e-1
	10	**9.0625e-1** +	9.0541e-1 +	6.6801e-1
WFG6	8	8.1101e-1 +	**8.2921e-1** +	4.9381e-1
	10	**8.6759e-1** +	8.5801e-1 +	8.3429e-1
WFG7	8	**8.6473e-1** ≈	8.4897e-1 ≈	6.6763e-1
	10	**9.2021e-1** +	9.0962e-1 +	8.8204e-1
WFG8	8	**6.4006e-1** +	6.0630e-1 +	1.0722e-1
	10	**6.8628e-1** +	6.2094e-1 +	1.0429e-1
WFG9	8	**7.6096e-1** +	7.2139e-1 +	2.0627e-1
	10	**8.0174e-1** +	7.8019e-1 +	0.9111e-1
$+/-/\approx$		16/0/4	16/2/2	

TABLE IV: Hypervolume values obtained by MOEA/D-N$_\theta$ and MOEA/D-N$_\theta$ with two dynamical normalization strategies on DTLZ and WFG test suites with 8 and 10 objectives. The best value of each row is highlighted with bold typeface, while the worst with gray background.

Problem	M	MOEA/D-SDN$_\theta$	MOEA/D-LDN$_\theta$	MOEA/D-N$_\theta$
DTLZ1	8	9.9725e-1 +	**9.9726e-1** +	3.6825e-1
	10	**9.9966e-1** +	9.9965e-1 +	6.6534e-1
DTLZ2	8	9.2384e-1 +	**9.2388e-1** +	9.2294e-1
	10	9.6977e-1 ≈	9.6978e-1 ≈	**9.6982e-1**
DTLZ3	8	**9.2166e-1** +	9.2038e-1 +	3.2681e-1
	10	9.6949e-1 +	**9.6954e-1** +	2.5856e-1
DTLZ4	8	9.1029e-1 +	**9.1076e-1** +	4.3009e-1
	10	9.6731e-1 +	**9.6782e-1** +	4.2324e-1
WFG4	8	9.1735e-1 ≈	9.0387e-1 −	**9.2138e-1**
	10	9.6585e-1 ≈	9.5436e-1 −	**9.6929e-1**
WFG5	8	**8.6520e-1** +	8.6340e-1 +	5.2636e-1
	10	**9.0728e-1** +	9.0472e-1 ≈	6.4121e-1
WFG6	8	**8.2383e-1** +	8.0197e-1 ≈	5.5548e-1
	10	**8.6525e-1** ≈	8.5522e-1 ≈	8.3044e-1
WFG7	8	**8.6290e-1** ≈	8.4764e-1 ≈	5.7068e-1
	10	**9.3091e-1** +	9.2067e-1 +	8.3696e-1
WFG8	8	**6.1669e-1** +	5.6518e-1 +	1.0506e-1
	10	**7.0245e-1** +	5.7749e-1 +	1.0188e-1
WFG9	8	**7.9890e-1** +	7.7886e-1 +	1.6324e-1
	10	**8.1717e-1** +	7.5840e-1 +	1.3907e-1
$+/-/\approx$		15/0/5	13/2/5	

of the mean hypervolume value. According to the statistical test results, MOEA/D-SDN outperforms MOEA/D-N on 16 test problems and MOEA/D-LDN outperforms MOEA/D-N on 14 test problems. Similarly, as shown in Table III, MOEA/D-SDN$_N$ achieves the best results on 14 test problems in terms of the mean hypervolume value. Both MOEA/D-SDN$_N$ and MOEA/D-LDN$_N$ significantly outperform their original version MOEA/D-N$_N$ on 16 test problems. As shown in Table IV, MOEA/D-SDN$_\theta$ obtains the best results on 12 test problems in terms of the mean hypervolume value. According to the statistical test results, MOEA/D-SDN$_\theta$ significantly outperforms MOEA/D-N$_\theta$ on 15 test problems and MOEA/D-LDN$_\theta$ significantly outperforms MOEA/D-N$_\theta$ on 13 test problems. Overall, from Table II-IV, we can clearly observe the performance improvement when the two dynamic normalization strategies are used.

In Fig. 5 and Fig. 6, we show obtained solutions by a single run of each algorithm with the median hypervolume value on the 3- and 10-objective DTLZ4. As shown in Fig. 5 and Fig. 6, our proposed strategies help all the three normalization methods obtain promising results on the DTLZ4 problems. Compared to Fig. 2 and Fig. 3, uniformly distributed solutions are obtained over the entire Pareto front by each algorithm with the dynamic normalization strategies.

Fig. 7 shows the results obtained by MOEA/D-N, MOEA/D-N$_N$, MOEA/D-N$_\theta$, and their variants with the two proposed strategies on the 10-objective WFG9. We can clearly see that MOEA/D without the proposed adjustment strategies cannot find well-distributed solutions.

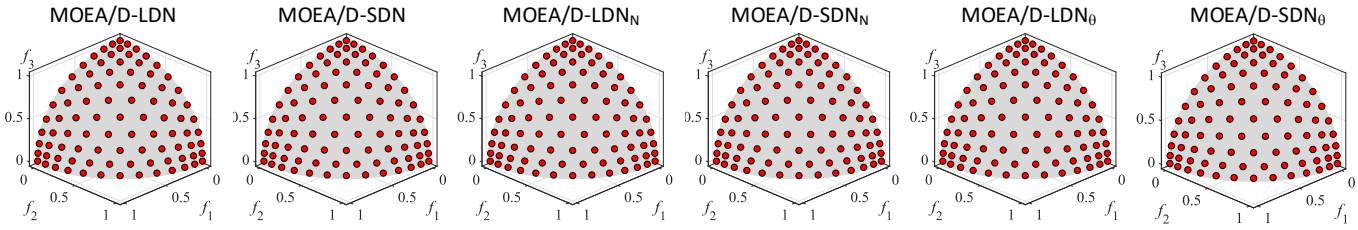


Fig. 5: Obtained solutions by a single run of each algorithm with the median hypervolume value on the 3-objective DTLZ4.

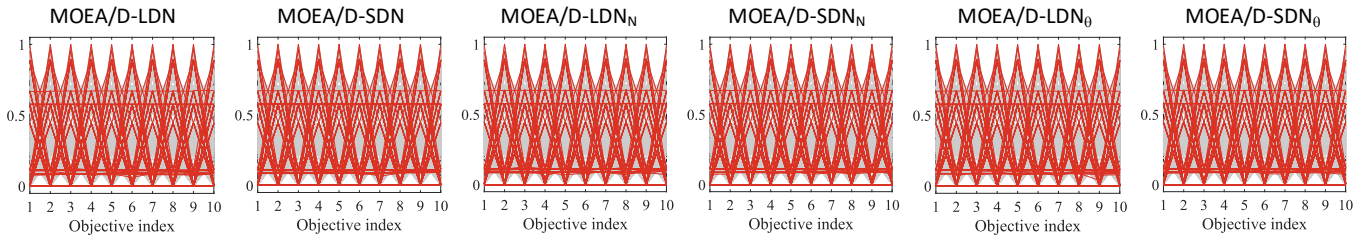


Fig. 6: Obtained solutions by a single run of each algorithm with the median hypervolume value on the 10-objective DTLZ4.

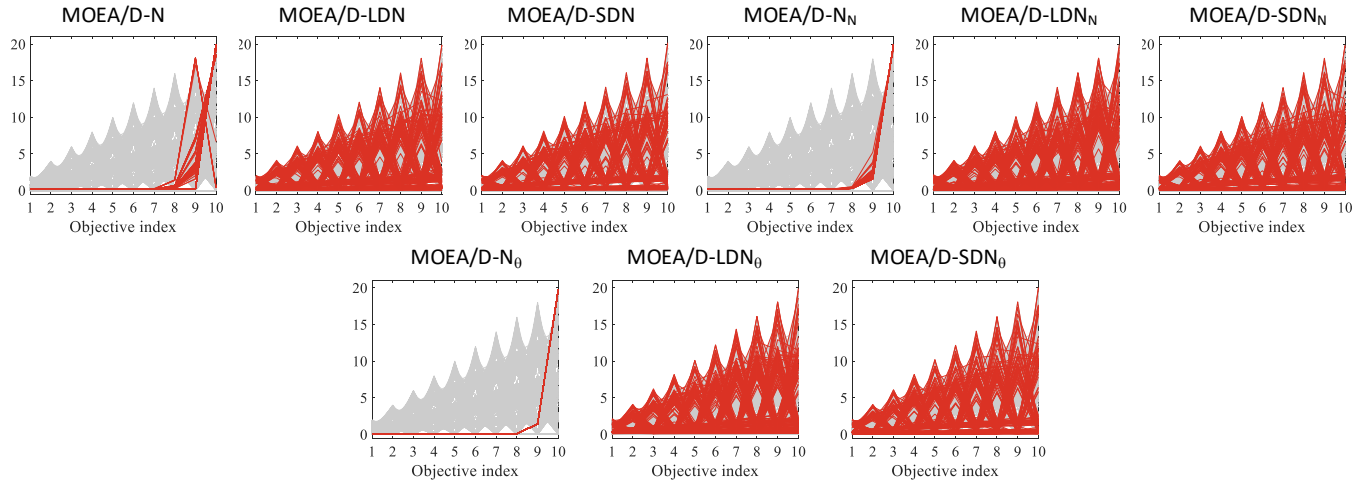


Fig. 7: Obtained solutions by a single run of each algorithm with the median hypervolume value on the 10-objective WFG9.

VI. Conclusion

In this paper, three normalization methods were examined on two commonly used test suites. Through computational experiments, we revealed the limitations of the three normalization methods. All of them lead to performance deterioration. We also demonstrated that the sophisticated normalization methods are not necessarily better than the simple normalization method. These observations clearly showed the negative effects of objective space normalization and the necessity of a robust normalization method.

In order to address the performance deterioration, we proposed to adjust the extent of normalization. Two dynamic normalization strategies were proposed. Our strategies can be easily integrated into any objective space normalization method in a multiobjective evolutionary algorithm. With the proposed strategies, all normalization methods showed significantly improved performance on the DTLZ and WFG problems. Experimental results clearly suggested the necessity of controlling the extent of normalization. In this paper, the extent of normalization was controlled by adjusting the parameter α in a pre-determined manner. In the future, the extent of normalization can be controlled in a self-adaptive manner by monitoring the reliability of the estimation of the ideal and nadir points. For example, we can use the convergence information to adaptively control the parameter α as the constraint handling method in [34]. We may also need to examine the effectiveness of our strategies in other evolutionary multiobjective algorithms on other test problems.

References

[1] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *Proceedings of the IEEE Congress on Evolutionary Computation*, Hong Kong, China, Jun. 2008, pp. 2419–2426.

[2] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Computing Surveys*, vol. 48, no. 1, pp. 1–35, Sep. 2015.

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

[4] H. Ishibuchi, N. Akedo, and Y. Nojima, "A study on the specification of a scalarizing function in MOEA/D for many-objective knapsack problems," in *International Conference on Learning and Intelligent Optimization*, vol. 7997, Catania, Italy, Jan. 2013, pp. 231–246.

[5] ——, "Behavior of multiobjective evolutionary algorithms on many-objective knapsack problems," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 2, pp. 264–283, Apr. 2015.

[6] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, Oct. 2006.

[7] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints." *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, Aug. 2014.

[8] H. Ishibuchi, K. Doi, and Y. Nojima, "On the effect of normalization in MOEA/D for multi-objective and many-objective optimization," *Complex & Intelligent Systems*, vol. 3, no. 4, pp. 279–294, Dec. 2017.

[9] L. He, Y. Nan, K. Shang, and H. Ishibuchi, "A study of the naïve objective space normalization method in MOEA/D," in *IEEE Symposium Series on Computational Intelligence*, Xiamen, China, Dec. 2019.

[10] A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh, "A survey of multiobjective evolutionary algorithms based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 3, pp. 440–462, 2016.

[11] Y. Yuan, H. Xu, B. Wang, and X. Yao, "A new dominance relation-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 16–37, Feb. 2016.

[12] M. Asafuddoula, T. Ray, and R. Sarker, "A decomposition-based evolutionary algorithm for many objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 3, pp. 445–460, 2014.

[13] O. Grodzevich and O. Romanko, "Normalization and other topics in multi-objective optimization," in *Fields-MITACS Industrial Problems Workshop*, 2006.

[14] K. Deb, K. Miettinen, and S. Chaudhuri, "Toward an estimation of nadir objective vector using a hybrid of evolutionary and local search approaches," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 6, pp. 821–841, Dec. 2010.

[15] K. Deb and K. Miettinen, "A review of nadir point estimation procedures using evolutionary approaches: A tale of dimensionality reduction," in *Proceedings of the International Conference on Multiple Criteria Decision Making*, Auckland, New Zealand, Jan. 2009, pp. 1–14.

[16] Handing Wang and Xin Yao, "Corner sort for Pareto-based many-objective optimization," *IEEE Transactions on Cybernetics*, vol. 44, no. 1, pp. 92–102, Jan. 2014.

[17] O. P. Jones, J. E. Oakley, and R. C. Purshouse, "Component-level study of a decomposition-based multi-objective optimizer on a limited evaluation budget," in *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 2018, pp. 689–696.

[18] J. Blank, K. Deb, and P. C. Roy, "Investigating the normalization procedure of NSGA-III," in *Proceedings of Evolutionary Multi-Criterion Optimization*, vol. 11411, East Lansing, MI, USA, Mar. 2019, pp. 229–240.

[19] M. Asafuddoula, T. Ray, and R. Sarker, "A decomposition-based evolutionary algorithm for many objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 3, pp. 445–460, Jun. 2015.

[20] C. Liu, Q. Zhao, B. Yan, S. Elsayed, T. Ray, and R. Sarker, "Adaptive sorting-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 247–257, Apr. 2018.

[21] Y.-H. Zhang, Y.-J. Gong, T.-L. Gu, H.-Q. Yuan, W. Zhang, S. Kwong, and J. Zhang, "DECAL: Decomposition-based coevolutionary algorithm for many-objective optimization," *IEEE Transactions on Cybernetics*, vol. 49, no. 1, pp. 27–41, Jan. 2019.

[22] M. Elarbi, S. Bechikh, C. A. C. Coello, M. Makhlouf, and L. B. Said, "Approximating complex Pareto fronts with pre-defined normal-boundary intersection directions," *IEEE Transactions on Evolutionary Computation*, pp. 1–1, 2019.

[23] R. Hernández Gómez, C. A. Coello Coello, and E. Alba Torres, "A multi-objective evolutionary algorithm based on parallel coordinates," in *Proceedings of the Genetic and Evolutionary Computation Conference*. Denver, Colorado, USA: ACM Press, 2016, pp. 565–572.

[24] M. Asafuddoula, T. Ray, and R. Sarker, "A decomposition based evolutionary algorithm for many objective optimization with systematic sampling and adaptive epsilon control," in *Proceedings of Evolutionary Multi-Criterion Optimization*, Sheffield, United Kingdom, Mar. 2013, pp. 413–427.

[25] S. Jiang and S. Yang, "Convergence versus diversity in multiobjective optimization," in *Proceedings of the International Conference on Parallel Problem Solving from Nature*, Coimbra, Portugal, Sep. 2016, pp. 984–993.

[26] D. Sharma, S. Z. Basha, and S. A. Kumar, "Diversity over dominance approach for many-objective optimization on reference-points-based framework," in *Proceedings of Evolutionary Multi-Criterion Optimization*, East Lansing, MI, USA, Mar. 2019, pp. 278–290.

[27] D. Sharma and P. K. Shukla, "Line-prioritized environmental selection and normalization scheme for many-objective optimization using reference-lines-based framework," *Swarm and Evolutionary Computation*, vol. 51, p. 100592, Dec. 2019.

[28] Y. Yuan, H. Xu, B. Wang, B. Zhang, and X. Yao, "Balancing convergence and diversity in decomposition-based many-objective optimizers," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 2, pp. 180–198, Apr. 2015.

[29] S. Ying, L. Li, Z. Wang, W. Li, and W. Wang, "An improved decomposition-based multiobjective evolutionary algorithm with a better balance of convergence and diversity," *Applied Soft Computing*, vol. 57, pp. 627–641, Aug. 2017.

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

[31] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, Nov. 2017.

[32] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proceedings of the IEEE Congress on Evolutionary Computation*, vol. 1, Honolulu, USA, May 2002, pp. 825–830.

[33] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms—a comparative case study," in *Proceedings of the International Conference on Parallel Problem Solving from Nature*. Springer, 1998, pp. 292–301.

[34] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, and E. Goodman, "Push and pull search for solving constrained multi-objective optimization problems," *Swarm and Evolutionary Computation*, vol. 44, pp. 665–679, Feb. 2019.