Learning to Decompose: A Paradigm for Decomposition-Based Multi-Objective Optimization

Mengyuan Wu, Ke Li, *Memeber, IEEE*, Sam Kwong, *Fellow, IEEE* and Qingfu Zhang, *Fellow, IEEE*

***Abstract*—Decomposition-based evolutionary multi-objective optimization algorithms decompose a multi-objective optimization problem into subproblems using a set of predefined reference points. The convergence is guaranteed by optimizing the single-objective or simplified multi-objective subproblems while the diversity is handled by the evenly distributed reference points. Nevertheless, studies have shown that the performance of decomposition-based algorithms is strongly dependent on the Pareto front shapes due to unadaptable reference points and subproblem formulation. In this paper, we investigate the causes from three aspects and propose a learning-to-decompose paradigm consisting of a learning module and an optimization module to address these issues. Specifically, given the current non-dominated solutions from the optimization module, which can be any decomposition-based multi-objective optimizer, the learning module learns an analytical model that characterizes the estimated PF. Thereafter, useful information are extracted from the learned model to guide the decomposition in the optimization module. In particular, we utilize the learned model to sample reference points compliant to the PF and formulate subproblems with appropriate contours and search directions according to the current status. We integrate the learning-to-decompose paradigm with two most popular decomposition-based evolutionary optimizers, i.e., MOEA/D and NSGA-III, and compare them with several state-of-the-art adaptive methods. The comprehensive experiments validate the effectiveness and robustness of the proposed paradigm on 14 test problems with various Pareto front shapes.**

I. Introduction

This paper considers a multi-objective optimization problem (MOP) formulated as:

$$\begin{array}{ll}\text{minimize} & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_m(\mathbf{x}))^T \\ \text{subject to} & \mathbf{x} \in \Omega \end{array}, \qquad (1)$$

where $\mathbf{x} = (x_1, \cdots, x_n)^T$ is an n-dimensional decision vector and $\mathbf{F}(\mathbf{x})$ is an m-dimensional objective vector. Ω is the feasible set in the decision space $\mathbb{R}^n$ and $\mathbf{F} : \Omega \rightarrow \mathbb{R}^m$ is the corresponding attainable set in the objective space $\mathbb{R}^m$. Given two solutions $\mathbf{x}^1, \mathbf{x}^2 \in \Omega$, $\mathbf{x}^1$ is said to dominate $\mathbf{x}^2$ if and only if $f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2)$ for all $i \in \{1, \cdots, m\}$ and $\mathbf{F}(\mathbf{x}^1) \neq \mathbf{F}(\mathbf{x}^2)$. A solution $\mathbf{x} \in \Omega$ is said to be Pareto-optimal if and only if there is no solution $\mathbf{x}' \in \Omega$ that dominates it. The set of all Pareto-optimal solutions is called the Pareto-optimal set (PS) and their corresponding objective vectors form the Pareto front (PF). Accordingly, the ideal point is defined as $\mathbf{z}^{id} = (z_1^{id}, \cdots, z_m^{id})^T$, where $z_i^{id} = \min_{\mathbf{x} \in PS} f_i(\mathbf{x})$ for all $i \in \{1, \cdots, m\}$, and the nadir point is defined as $\mathbf{z}^{nd} = (z_1^{nd}, \cdots, z_m^{nd})^T$, where $z_i^{nd} = \max_{\mathbf{x} \in PS} f_i(\mathbf{x})$.

Evolutionary algorithm (EA), which is able to approximate the whole PF/PS in a single run due to its population-based property, has been widely accepted as a major approach for multi-objective optimization. It is well-known that the balance between convergence and diversity is the cornerstone of evolutionary multi-objective optimization (EMO) [1]. According to the ways of achieving this balance, the current EMO algorithms are generally classified into three major categories, i.e., Pareto- [2]–[4], indicator- [5]–[7] and decomposition-based algorithms [8]–[10]. In particular, the decomposition-based algorithm, especially since the development of multi-objective EA based on decomposition (MOEA/D) [8], has become an increasingly popular choice for posterior multi-objective optimization. Generally speaking, the basic idea of MOEA/D is to decompose the original MOP into a set of subproblems, either in the form of single-objective optimization problem [8] or simlified MOP [11], [12] and optimize them in a collaborative manner. It provides a natural way to achieve the balance between convergence and diversity, where the convergence is guaranteed by the optimization of each subproblem while the diversity is implicitly controlled by a predefined set of evenly distributed reference points[1]. The reference points and the subproblem formulation constitute two key components of the decomposition.

As reported in a recent study [13], the performance of decomposition-based EMO algorithms strongly depends on the PF shapes. In particular, MOEA/D and its variants work well on problems with regular PF shapes, especially when they are in line with the unit simplex from which the the evenly distributed reference points are sampled; otherwise it performs poorly, e.g., PFs with disparate scales, discontinuous segments or other complex shapes. This can be generally attributed to an inappropriate decomposition, e.g., the distribution of reference points is not compliant with the PF shape, or the search direction and contours induced by the adopted subproblem formulation are not adaptable to various problem landscapes. In recent years, researchers have proposed methods to achieve appropriate decomposition mainly in two aspects:

- A natural idea to achieve better decomposition is to adapt the distribution of reference points to be compliant with the PF shape. For example, Jiang *et al.* [14] proposed to

M. Wu, S. Kwong and Q. Zhang are with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong SAR (e-mail: mengyuan.wu@my.cityu.edu.hk, cssamk@cityu.edu.hk, qingfu.zhang@cityu.edu.hk).

K. Li is with the Department of Computer Science, University of Exeter, Exeter, EX4 4QF, UK (e-mail: k.li@exeter.ac.uk).

[1]In this paper, we use the term reference point without loss of generality, although some other literatures, e.g., the original MOEA/D [8], also use the term weight vector interchangeably.

use non-dominated solutions stored in an external archive to fit an estimated PF in the form of $\sum_{i=1}^{m} f_i^p = 1$. Thereafter, reference points are sampled from the estimated PF so that the Hypervolume (HV) [15] is expected to be maximized. Unfortunately, this method fails to estimate complex PFs; and using the HV as the selection criterion is highly sensitive to the choice of the worst point [16]. Gu *et al.* [17] proposed to use an equidistant interpolation to estimate the PF. The reference points are periodically updated according to the mean of several interpolation points. However, since the piecewise linear interpolation may fail to estimate highly nonlinear PFs and can easily cause overfitting, the estimation will be largely impaired by some outliers, which is not uncommon at the early stage of the optimization. Recently, Gu and Cheung [18] have developed a reference point generation method based on self-organizing map (SOM) [19]. It uses the objective vectors of recent solutions to train a SOM network periodically. The reference points are directly set as the weights of the neurons. However, this method is computationally expensive since the training of a SOM network requires to maintain a large external archive (data intensive). In [20], Qi *et al.* proposed to dynamically adjust the reference points at the late stage of the optimization. Specifically, it maintains an external population to estimate the density of solutions with respect to each reference point. Reference points in the most crowded regions are periodically removed while new reference points are generated in the most sparse regions by using objective vectors chosen from the external population. In [21], Wang *et al.* developed a co-evolutionary framework which co-evolves the population and the reference points simultaneously. Although this method improves the population diversity to a certain extent, it can hardly maintain evenly distributed solutions.

- From another aspect, some researchers proposed to improve the decomposition by adapting the existing subproblem formulations or developing new ones that are suitable to the underlying problem landscapes. For example, Wang *et al.* [22] analyzed the properties of a family of frequently used subproblem formulations, i.e., L_p method, and proposed a Pareto adaptive scalarizing approximation to approximate the optimal p value adaptively. In [23], Yang *et al.* investigated the influence of the penalty factor θ of the penalty-based boundary intersection (PBI), a popular subproblem formulation in MOEA/D. They suggested two adaptive penalty schemes to enhance its search ability. Note that even though these parameter adaptation methods seem to be intuitive, they may not always generate appropriate contours. In addition, the search direction is restricted by the corresponding subproblem formulation. More recently, Jiang *et al.* [24] developed two new subproblem formulations that are self-adaptive according to the dynamics of the optimization process. Although these newly developed subproblem formulations are able to generate more controllable contours to a certain extent, they do not take the PF shape into consideration.

In this paper, we analyze the issues of inappropriate decomposition from three aspects and propose a learning-to-decompose (LTD) paradigm for to address these issues. The LTD paradigm consists of two modules, i.e., the optimization module that can be any decomposition-based EMO algorithm and the learning module that periodically learns an analytical module characterizing the estimated PF. More specifically, by using the current non-dominated solutions from the optimization module as the training data, the learning module employs Gaussian process (GP) regression [25] to learn the model of the estimated PF, from which, useful information are extracted to: 1) sample effective reference points compliant to the estimated PF shape; and 2) formulate subproblems, single-objective in particular, by which the population can be guided toward to the true PF. Note that the reference points and the subproblem formulation, derived from the LTD paradigm, can be readily used in any decomposition-based EMO algorithm. Comprehensive experiments on a series of benchmark problems with various PF shapes fully demonstrate the effectiveness of our proposed LTD paradigm.

The remainder of this paper is organized as follows. Section II and Section III describes the motivation and technical details of our proposed LTD paradigm step by step. Section IV provides the setup of the experiments, and the performance of our proposed method is then validated and discussed in Section V. Finally, Section VI concludes the paper and discusses the future work.

II. Motivations

This section develops our motivations of the LTD paradigm by analyzing difficulties encountered by MOEA/D, a representative decomposition-based EMO algorithm, when solving problems with various characteristics.

In the past decade, we have witnessed the significantly growing interests in the development of MOEA/Ds, given their promising results on a variety of problems, e.g., problems with complicated PS [26] and many objectives [9], combinatorial optimization problems [27], and applications like antenna design [28], electrical power production [29] and community detection in networks [30]. However, more and more researchers have noticed that MOEA/D becomes less effective or even completely fails when solving problems with complex PF shapes [13]. Generally speaking, this can be attributed to three major reasons.

- Firstly, the reference points, which are evenly sampled on a unit simplex by default, play an essential role in MOEA/D. As shown in Fig. 1(a), each reference point corresponds to a reference vector originated from the ideal point. The optimum of each subproblem (i.e., a Pareto-optimal solution), with respect to a reference point, is expected to be located at the intersection between the corresponding reference vector and the PF. Note that the evenly sampled reference points work well when the PF shape is perfectly in line with the unit simplex like PF1 shown in Fig. 1(a); otherwise, the distribution of the obtained solutions might become unsatisfactory. For example, we can observe an undesirable concentration of

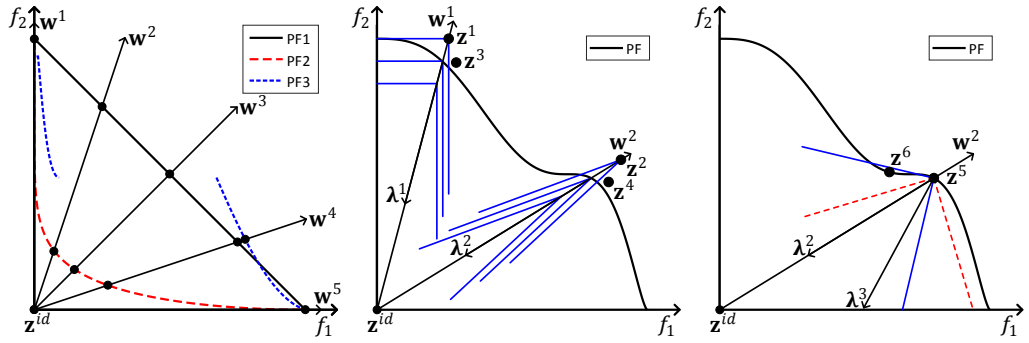


Fig. 1: (a) Intersections between the reference vectors and PFs of different shapes; (b) Contours of TCH and PBI with $\theta = 5$; (c) Contours of PBI with $\theta = 1$ and its variant.

the expected Pareto-optimal solutions in the knee region of PF2, which has a sharp convex shape. As for PF3, which consists of two disconnected segments, some of the reference points have no intersection between the corresponding reference vectors and the PF, thus might not end up with desirable Pareto-optimal solutions.

- Secondly, as discussed in [31], the search dynamics of MOEA/D is affected by the shape of the contours induced by the corresponding subproblem formulation. Fig. 1(b) presents the contours of two widely used subproblem formulations [8], i.e., Tchebycheff (TCH) and PBI, with respect to two different reference points. As shown in this figure, a contour divides the objective space into two subspaces, where objective vectors lying in the subspace covering $\mathbf{z}^{id}$ are superior to those in the other subspace in terms of their function values of the corresponding subproblem. Therefore, the shape of the contours determines the superiority between different solutions. In particular, the opening angle of the contours of TCH is $\pi/2$; while for PBI, it is controlled by its penalty factor θ, where we use $\theta = 5$ as recommended in [8]. Since these are fixed beforehand without considering the PF shape, it might lead to an inappropriate comparison between solutions. For example, as shown in Fig. 1(b), objective vectors $\mathbf{z}^3$ and $\mathbf{z}^4$ are closer to the PF than $\mathbf{z}^1$ and $\mathbf{z}^2$, but $\mathbf{z}^3$ is inferior to $\mathbf{z}^1$ in terms of the TCH function value and $\mathbf{z}^4$ is inferior to $\mathbf{z}^2$ in terms of the PBI function value.
- Last but not the least, the subproblem formulation also determines the search direction of the corresponding subproblem. For example, as shown in Fig. 1(b), the search directions of both TCH and PBI are opposite to the corresponding reference vectors, denoted by λ_1 and λ_2. Let us consider another example shown in Fig. 1(c) where the blue solid line represents the contour of PBI functions with $\theta = 1$, which provides a larger opening of the contour. It has been discussed in [32] that a larger opening angle of the contours contributes to better convergence but may be harmful to the population diversity. For the given reference point in Fig. 1(c), it is desirable that the corresponding subproblem finds the Pareto-optimal solution $\mathbf{z}^5$, i.e., the intersecting point between $\mathbf{w}^2$ and the PF. However, according to the contour, we find that $\mathbf{z}^6$, which is away from $\mathbf{w}^2$, has a better PBI function value than $\mathbf{z}^5$. In this case, more than one subproblem may have a risk to end up with the same superior solution, thus damaging the population diversity. But if we change the search direction to be normal to the PF at $\mathbf{z}^5$, denoted by λ^3, while keeping the shape of the contour the same as before (the red dashed line shown in Fig. 1(c)), the optimal solution of this subproblem will still be $\mathbf{z}^5$. In this paper, we argue that the search direction normal to the PF is optimal for subproblem formulation.

According to the above discussions, to make MOEA/D adaptable to problems with various characteristics, a natural idea is to learn the characteristics of the estimated PF progressively during the optimization process. In the next section, we will develop the LTD paradigm, based on which we are expected to have: 1) a set of reference points compliant to the PF shapes; and 2) a subproblem formulation with appropriate contours and search direction normal to the PF.

III. Pareto-Driven Evolution Algorithm

The general framework of our proposed LTD paradigm is given in Fig. 2. It consists of two interdependent modules: optimization and learning. Specifically, the optimization module is a decomposition-based EMO algorithm. The learning module aims to characterize the PF via an analytical model. Depending on the requirements of the optimization module, useful information can be extracted from this analytical model to guide the decomposition. In this paper, we start with MOEA/D as the optimization module, and generalize the applicability of the proposed LTD paradigm to NSGA-III later. In particular, given MOEA/D as the optimization module, we use the learned model to: 1) sample effective reference points that are compliant with the shape of the estimated PF, and 2) formulate subproblems whose contours are suitable for the problem landscapes and search directions are normal to the estimated PF. In the following paragraphs, we will describe each part step by step.

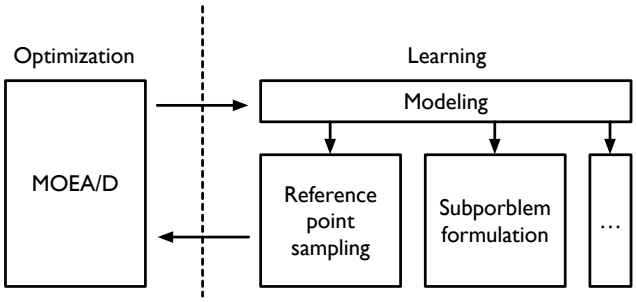


Fig. 2: General framework of our proposed algorithm.

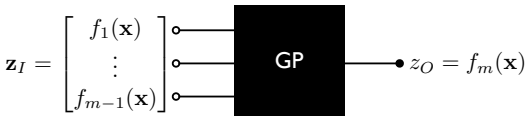


Fig. 3: Illustration of GP regression process.

A. Modeling: PF Learning via Gaussian Process Regression

The major purpose of the learning module is to build a model that characterizes the estimated PF. From this model, useful information, e.g., normal vectors and curvatures of the estimated PF, can be derived and used to guide the optimization process. Specifically, this modeling process is treated as a regression problem where the non-dominated solutions in the current population are used as the training data. Inspired by [33], we choose Gaussian process (GP) [25] to build the regression model. There are two major reasons for choosing GP: 1) it has the ability to quantify the variances of the predicted outputs. In particular, these variances are used to detect the discontinuous regions of the estimated PF; and 2) if the mean and covariance functions are appropriately chosen, we can derive the first and second derivatives of the predicted mean which are finally used for calculating the normal vector and curvature of the estimated PF.

As shown in Fig. 3, for each non-dominated solution $\mathbf{x}$, its $m-1$ objective function values constitute the input vector, denoted as $\mathbf{z}_I = (f_1(\mathbf{x}), \cdots, f_{m-1}(\mathbf{x}))^T$, while the remaining objective function value serves the target, denoted as $z_O = f_m(\mathbf{x})$. Note that the mapping learned from this model is one-to-one, given that there does not exist two non-dominated solutions $\mathbf{x}^1$ and $\mathbf{x}^2$ whose $m-1$ objective function values are identical whereas the remaining ones are not. To keep the input space and output space in the same scale, each element of $\mathbf{z}_I$ and z_O is normalized by the currently estimated ideal and nadir points. Given a set of training data $\mathcal{D} = \{(\mathbf{z}_I^i, z_O^i) | i = 1, \cdots, M\}$, GP regression aims to learn a latent function $g(\mathbf{z}_I)$ by assuming $z_O^i = g(\mathbf{z}_I^i) + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$ is an independently and identically distributed Gaussian noise. For each testing input vector $\mathbf{z}_I^* \in [0,1]^{m-1}$, the mean and variance of the target $g(\mathbf{z}_I^*)$ are predicted as:

$$\begin{aligned} \overline{g}(\mathbf{z}_I^*) &= m(\mathbf{z}_I^*) + \mathbf{k}^{*T}(K + \sigma_n^2 I)^{-1}(\mathbf{z}_O - \mathbf{m}(Z_I)) \\ \mathbb{V}[g(\mathbf{z}_I^*)] &= k(\mathbf{z}_I^*, \mathbf{z}_I^*) - \mathbf{k}^{*T}(K + \sigma_n^2 I)^{-1}\mathbf{k}^* \end{aligned}, \quad (2)$$

where $Z_I = (\mathbf{z}_I^1, \cdots, \mathbf{z}_I^M)^T$ and $\mathbf{z}_O = (z_O^1, \cdots, z_O^M)^T$. $\mathbf{m}(Z_I)$ is the mean vector of Z_I, $\mathbf{k}^*$ is the covariance vector between Z_I and $\mathbf{z}_I^*$, and K is the covariance matrix of Z_I. The predicted mean $\overline{g}(\mathbf{z}_I^*)$ is directly used as the prediction of z_O^*, and the prediction variance $\mathbb{V}[g^*]$ quantifies the uncertainty. All in all, a sample on the estimated PF is represented as $\mathbf{z}^* = (\mathbf{z}_I^*, \overline{g}(\mathbf{z}_I^*))^T$.

A GP is specified by a mean function and a covariance function. A prior knowledge of the mean function eases the training of the hyperparameters and leads to better regression results. Inspired by [22], we consider using the following general assumption about the PF:

$$\begin{aligned} &\textstyle\sum_{i=1}^{m} c_i \overline{f_i}(\mathbf{x})^{a_i} = 1 \\ &\text{subject to} \quad a_i > 0, c_i > 0, i = 1 \cdots, m \end{aligned}, \quad (3)$$

where $\overline{f_i}(\mathbf{x})$ is the i-th normalized objective function of a Pareto-optimal solution. Accordingly, we set the mean function as:

$$m(z_O) = \left(\frac{1 - \sum_{i=1}^{m-1} c_i z_{I,i}^{a_i}}{c_m}\right)^{\frac{1}{a^m}} . \quad (4)$$

By letting $a = a_i$ and $c = 1/c_i$ where $i = 1, \cdots, m$, (4) is further simplified as:

$$\begin{aligned} &m(z_O) = (c - \textstyle\sum_{i=1}^{m-1} z_{I,i}^a)^{\frac{1}{a}} \\ &\text{subject to} \quad a > 0, c > 0 \end{aligned} . \quad (5)$$

This simplified mean function significantly reduces the number of hyperparameters that need to be learned for GP regression, especially when having many objectives. Note that even though this mean function might fail to accurately represent some PFs with irregular shapes, the covariance function of the GP can make it up. As recommended in [25], we use the popular rational quadratic covariance function in this paper. The hyperparameters are learned by maximizing the log marginal likelihood:

$$\begin{aligned} &\log p(\mathbf{z}_O | Z_I) \\ =& -\frac{1}{2}(\mathbf{z}_O - \mathbf{m}(Z_I))^T (K + \sigma_n^2 I)^{-1}(\mathbf{z}_O - \mathbf{m}(Z_I)) \\ &-\frac{1}{2}\log|K + \sigma_n^2 I| - \frac{M}{2}\log 2\pi. \end{aligned} \quad (6)$$

From the learned model of the estimated PF, we can extract useful information to guide the optimization process. In this paper, we are particularly interested in the first and second derivatives of the predicted mean, which can be used to obtain the normal vector and curvature of a sample $\mathbf{z}^*$ with respect to the manifold of the estimated PF, i.e., $z_O - \overline{g}(\mathbf{z}_I) = 0$. Specifically, the unit normal vector is computed as:

$$\begin{aligned} \mathbf{n}^* = (&-\frac{\frac{\partial \overline{g}(\mathbf{z}_I^*)}{\partial \mathbf{z}_{I,1}}}{\sqrt{1 + \sum_{i=1}^{m-1}(\frac{\partial \overline{g}(\mathbf{z}_I^*)}{\partial \mathbf{z}_{I,i}})^2}}, \cdots, \\ &-\frac{\frac{\partial \overline{g}(\mathbf{z}_I^*)}{\partial \mathbf{z}_{I,m-1}}}{\sqrt{1 + \sum_{i=1}^{m-1}(\frac{\partial \overline{g}(\mathbf{z}_I^*)}{\partial \mathbf{z}_{I,i}})^2}}, \frac{1}{\sqrt{1 + \sum_{i=1}^{m-1}(\frac{\partial \overline{g}(\mathbf{z}_I^*)}{\partial \mathbf{z}_{I,i}})^2}})^T \end{aligned} \quad (7)$$

Note that the computation of the curvature depends on the dimension of the estimated PF. In particular, the estimated PF is a curve when $m = 2$, of which the curvature at $\mathbf{z}^*$ is

computed as:

$$\kappa^* = -\frac{\frac{d^2\overline{g}(z_I^*)}{dz_I^2}}{(1+(\frac{d\overline{g}(z_I^*)}{dz_I})^2)^{\frac{3}{2}}}. \quad (8)$$

When $m > 2$, the estimated PF is a manifold, which has infinite number of curvatures at $\mathbf{z}^*$ in principle. Here we are only interested in the principal curvatures, i.e., the maximum and minimum curvatures. Let $\mathbf{r}(\mathbf{z}_I) = (z_{I,1}, \ldots, z_{I,m-1}, \overline{g}(\mathbf{z}_I))^T$ be a regular parametrization of the manifold. Its second fundamental form is written as:

$$\mathbf{II} = \begin{bmatrix} \frac{\partial^2 \mathbf{r}(\mathbf{z}_I)}{\partial z_{I,1}\partial z_{I,1}} \cdot \mathbf{n} & \cdots & \frac{\partial^2 \mathbf{r}(\mathbf{z}_I)}{\partial z_{I,1}\partial z_{I,m-1}} \cdot \mathbf{n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 \mathbf{r}(\mathbf{z}_I)}{\partial z_{I,m-1}\partial z_{I,1}} \cdot \mathbf{n} & \cdots & \frac{\partial^2 \mathbf{r}(\mathbf{z}_I)}{\partial z_{I,m-1}\partial z_{I,m-1}} \cdot \mathbf{n} \end{bmatrix}. \quad (9)$$

The principal curvatures at $\mathbf{z}^*$ are the eigenvalues of $\mathbf{II}(\mathbf{z}_I^*)$. Details of the computation of the normal vectors and curvatures can be found in [34].

B. Reference Point Sampling

As discussed in Section II, the originally evenly distributed reference points along a unit simplex may lead to some side effects for MOEA/D, especially when tackling problems with irregular PFs. To address this issue, by using the PF model learned in Section III-A, we develop a reference point sampling method that is able to generate a set of reference points widely distributed on the estimated PF. Specifically, the reference points are sampled according to the following three-step process:

- **Step 1:** Randomly generate $20N$ test input vectors in $[0,1]^{m-1}$; and another $20N$ test input vectors within the neighborhood of the training input vectors Z_I, where N is the population size. All these $40N$ test input vectors together with Z_I constitute Z_I^*. We use (2) to predict their corresponding targets $\mathbf{z}_O^*$. Afterwards, Z_I^* and $\mathbf{z}_O^*$ are combined to form a set of samples Z^* on the estimated PF.
- **Step 2:** Remove dominated samples and those whose prediction variances are higher than a threshold $1.5 \times \max\{\sqrt{\mathbb{V}[\overline{g}(\mathbf{z}_I^*)]} | \mathbf{z}_I^* \in Z_I^*\}$. This helps remove samples in the discontinuous regions or beyond the PF.
- **Step 3:** Trim the remaining samples in Z^* by repeatedly removing the one that has the highest density until the size of Z^* equals N. In particular, the density of a sample $\mathbf{z}^{*,i}$, $i \in \{1, \cdots, |Z^*|\}$, is computed as:

$$density(\mathbf{z}^{*,i}) = \sum_{j=1, j\neq i}^{|Z^*|} \frac{1}{dist(\mathbf{z}^{*,i}, \mathbf{z}^{*,j})}, \quad (10)$$

where $dist(\mathbf{z}^{*,i}, \mathbf{z}^{*,j})$ is the Euclidean distance between $\mathbf{z}^{*,i}$ and $\mathbf{z}^{*,j}$. We choose this density estimation due to its high efficiency to deal with a large set.

C. Subproblem Formulation

Subproblem formulation, which usually aggregates multiple objectives into a scalar value function, is one of the most important ingredients in MOEA/D. It determines the way of fitness assignment for each solution and thus the search direction of the optimization process. By utilizing useful information (normal vector and curvature in particular) extracted from the learned model of the estimated PF, we formulate the subproblem with respect to a reference point $\mathbf{z}^*$ generated in Section III-B as follows:

$$\text{minimize } y(\mathbf{x}|\mathbf{n}^*, \mathbf{z}^*) = h(\overline{\mathbf{F}}(\mathbf{x})|\mathbf{n}^*, \mathbf{z}^*) = d_1 + \theta_1 d_2^2 + \theta_2 d_2^4, \quad (11)$$

where

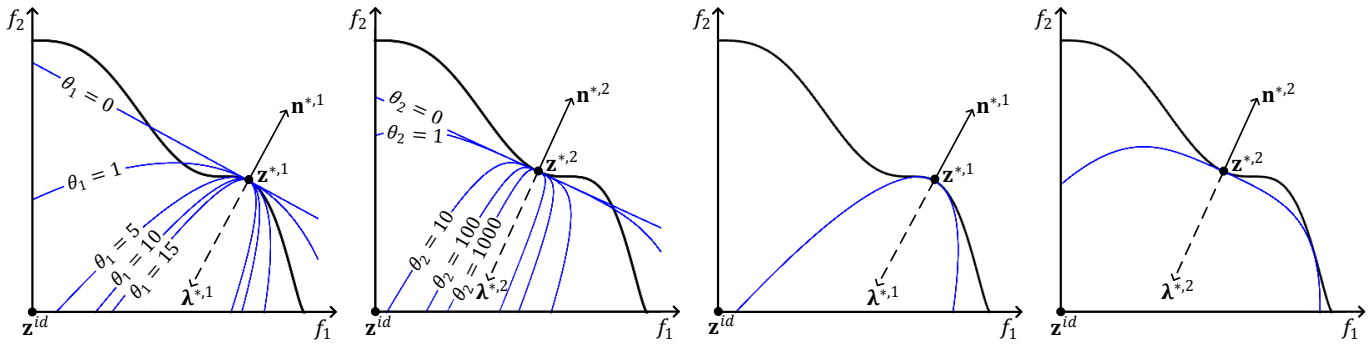
$$\begin{aligned} d_1 &= (\overline{\mathbf{F}}(\mathbf{x}) - \mathbf{z}^*)^T \mathbf{n}^* \\ d_2 &= ||\overline{\mathbf{F}}(\mathbf{x}) - \mathbf{z}^* - d_1 \mathbf{n}^*||. \end{aligned} \quad (12)$$

d_1 is the signed Euclidean distance between $\mathbf{z}^*$ and the projection of a normalized objective vector $\overline{\mathbf{F}}(\mathbf{x})$ on $\mathbf{n}^*$ calculated by (7). d_2 is the Euclidean distance between $\overline{\mathbf{F}}(\mathbf{x})$ and its projection. $\theta_1 > 0$ and $\theta_2 > 0$ are two parameters that control the shape and distribution of the contours of $h(\overline{\mathbf{F}}(\mathbf{x})|\mathbf{n}^*, \mathbf{z}^*)$. The search direction of this subproblem is normal to the estimated PF at $\mathbf{z}^*$, as shown in Fig. 4. Accordingly, we expect that the optimal solution to a subproblem is at the intersection between the corresponding search direction and the PF.

Fig. 4 presents the contours of $h(\overline{\mathbf{F}}(\mathbf{x})|\mathbf{n}^*, \mathbf{z}^*) = 0$ under different settings of θ_1 and θ_2, where the black curve is the estimated PF and λ^* represents the search direction of the subproblem with respect to $\mathbf{z}^*$. From this figure, we can see that a contour is tangent to the estimated PF and divides the objective space into two subspaces. In particular, objective vectors in the subspace toward the search direction λ^* are superior to those in the other subspace in terms of the subproblem formulation function. From Fig. 4(a) and Fig. 4(b), we can observe that both θ_1 and θ_2 control the opening of a contour. In particular, a smaller value of θ_1 or θ_2 will lead to a wider opening. As discussed in Section II, a overly wide opening might be harmful to the population diversity since other solutions may have better values than the Pareto-optimal solution at the intersection in terms of the subproblem formulation function. Even worse, different subproblems can share the same optimal solution. On the other hand, the opening of a contour will be very narrow if θ_1 or θ_2 is set too small. This reduces the search region of the corresponding subproblem and may slow down the convergence progress. Furthermore, from Fig. 4(a), it is worth noting that θ_1 not only controls the opening of the contour, but also the curvature of the contour at the vertex. Thus, θ_1 helps to fine-tune the contour close to the vertex. In contrast, as shown in Fig. 4(b), θ_2 does not have any effect on the curvature of the contour at the vertex.

From the above discussions, we see that θ_1 and θ_2 determine the search behaviors of the corresponding subproblem. To avoid a notorious parameter configuration by trial and error, we develop the following method that automatically sets θ_1 and θ_2, which takes the estimated PF shape into consideration:

- Firstly, θ_1 is determined by letting the curvature of the contour $h(\overline{\mathbf{F}}(\mathbf{x})|\mathbf{n}^*, \mathbf{z}^*) = 0$ at $\mathbf{z}^*$ just larger than the curvature (or principal curvatures) of the estimated PF. In such a way, the better region covered by $h(\overline{\mathbf{F}}(\mathbf{x})|\mathbf{n}^*, \mathbf{z}^*) = 0$ is widen without any overlapping



(a) $\theta_2 = 0$ (b) $\theta_1 = 0$ (c) $\theta_1 = 4.6,\ \theta_2 = 0.1$ (d) $\theta_1 = 0.1,\ \theta_2 = 3.5$

Fig. 4: Contours of $h(\overline{\mathbf{F}}(\mathbf{x})|\mathbf{n}^*, \mathbf{z}^*) = 0$ under (a) different settings of θ_1 when $\theta_2 = 0$; (b) different settings of θ_2 when $\theta_1 = 0$; (c) the adaptively determined θ_1 and θ_2 for $\mathbf{z}^{*,1}$; (d) the adaptively determined θ_1 and θ_2 for $\mathbf{z}^{*,2}$.

with the current estimated PF. As the curvature (or principal curvatures) of the contour $\kappa_h^* = 2\theta_1$ at $\mathbf{z}^*$ (see Appendix I), we set $\theta_1 = \max(\frac{\kappa^*}{2}, 0) + 0.1$, where the second term is added for the sake of estimation error.

- Given θ_1, θ_2 is determined by maximizing the opening of the contour $h(\overline{\mathbf{F}}(\mathbf{x})|\mathbf{n}^*, \mathbf{z}^*) = 0$ constrained by no overlapping with the current estimated PF except at $\mathbf{z}^*$. In other words, all other samples on the estimated PF have worse function values than $\mathbf{z}^*$ on $h(\mathbf{z}|\mathbf{n}^*, \mathbf{z}^*)$. Accordingly, we set $\theta_2 = \max(\min\{\theta_2 | h(\mathbf{z}|\mathbf{n}^*, \mathbf{z}^*) > 0, \forall \mathbf{z} \in Z^* \backslash \mathbf{z}^*\}, 0) + 0.1$.

Fig. 4(c) and Fig. 4(d) demonstrate the contours of $h(\overline{\mathbf{F}}(\mathbf{x})|\mathbf{n}^*, \mathbf{z}^*) = 0$ under the adaptively determined θ_1 and θ_2 for $\mathbf{z}^{*,1}$ and $\mathbf{z}^{*,2}$. By doing so, we expect to formulate the most appropriate subproblem according to the current status.

D. Incorporation with MOEA/D

Algorithm 1 presents the pseudo-code of MOEA/D-LTD, which uses MOEA/D as the optimization module in the LTD paradigm. At the beginning (line 1 to 3) of Algorithm 1), we initialize the set of reference points $Z^* = \{\mathbf{z}^{*,1}, \cdots, \mathbf{z}^{*,N}\}$ using the Das and Dennis's method [35] and obtain their neighborhood structure B as described in [8]. Then, the initial population $S = \{\mathbf{x}^1, \cdots, \mathbf{x}^N\}$ are randomly sampled from Ω and assigned to each subproblem with respect to a reference point. During each iteration of the main while loop, we firstly produce N offspring by parents selected either within the neighborhood of each subproblem or from the whole population controlled by a parameter δ, and add all offspring into the set $\overline{S}$ together with and solutions in S (line 6 to 13 of Algorithm 1). Thereafter, in line 19 of Algorithm 1, an environmental selection mechanism is used to select N elite solutions out of $\overline{S}$ and assign them to each subproblem. In this paper, we adopt the stable matching-based selection [10] to select surviving solutions to S. The learning module in LTD paradigm lies between line 14 to line 18 of Algorithm 1. In the learning module, the PF model is learned using all non-dominated objective vectors in $\overline{S}$, from which the reference points Z^* are sampled together with their unit normal vectors $N^* = \{n^{*,1}, \cdots, n^{*,N}\}$ and curvatures $K^* = \{\kappa^{*,1}, \cdots, \kappa^{*,N}\}$. Accordingly, the parameters θ_1 and θ_2 for each subproblem formulation are determined and the neighborhood structure B are recomputed. On the one hand, to ensure there are enough meaningful training solutions, the

Algorithm 1: MOEA/D–LTD

Input: algorithm parameters
Output: final population S

1 $Z^* \leftarrow$ Generate N initial reference points;
2 $B \leftarrow$ Compute the neighborhood structure of Z^*;
3 $S \leftarrow$ Randomly generate N solutions and associate them to each reference point;
4 $generation \leftarrow 0$;
5 **while** $generation < maxGen$ **do**
6 $\quad \overline{S} \leftarrow \emptyset$;
7 $\quad$ **for** *each* $i \in \{1, \cdots, N\}$ **do**
8 $\quad\quad$ **if** $uniform(0,1) < \delta$ **then**
9 $\quad\quad\quad E \leftarrow \{\mathbf{x}^j | j \in B(i)\}$;
10 $\quad\quad$ **else**
11 $\quad\quad\quad E \leftarrow S$;
12 $\quad\quad$ Randomly select mating solutions from E to generate an offspring $\overline{\mathbf{x}}$, $Q \leftarrow \overline{S} \cup \{\overline{\mathbf{x}}\}$;
13 $\quad \overline{S} \leftarrow S \cup \overline{S}$;
14 $\quad$ **if** $\psi_b < generation/maxGen < \psi_e$ **and** $mod(generation, \tau) == 0$ **then**
15 $\quad\quad$ Learn the PF model using all non-dominated objective vectors in S;
16 $\quad\quad$ $(Z^*, N^*, K^*) \leftarrow$Sample N reference points according to Section III-B, compute their unit normal vectors and curvature;
17 $\quad\quad$ Determine θ_1 and θ_2 for each subproblem formulation according to Section III-C;
18 $\quad\quad$ $B \leftarrow$ Compute the neighborhood structure of Z^*;
19 $\quad S \leftarrow$Select elite N solutions out of $\overline{S}$ and associate them to each reference point;
20 $\quad generation$++;
21 **return** S;

learning module is activated after ψ_b of the maximum generations, denoted by $maxGen$. On the other hand, as frequent adjustments of the reference points and search directions may slow down the convergence rate [21], the learning procedure is performed every τ generations and is deactivated after $\psi_e \times maxGen$ generations. Note that the environmental selection in line 19 of Algorithm 1 employs different subproblem formulations before and after the activation of the learning module, i.e., TCH and the subproblem formulation proposed in Section III-C, respectively. The algorithm terminates when the maximum number of generations is met.

E. Incorporation with NSGA-III

NSGA-III [11] is another decomposition-based EMO algorithm that formulates a subproblem as a simpler MOP with respect to each reference point. Similar to MOEA/D, each reference point in NSGA-III constructs a reference vector originated from the ideal point, the opposite of which is the search direction of this subproblem. Therefore, NSGA-III also suffers from the first and third issues discussed in Section II. To remedy these issues, we integrate the proposed LTD paradigm with NSGA-III as the optimization module, denoted by NSGA-III-LTD. In this case, the learning module is used for sampling reference points and formulate the (multi-objective) subproblems with corresponding reference vectors that are normal to the estimated PF. Specifically, in the association operation [36] of NSGA-III-LTD, the distance between a solution and a subproblem is calculated as d_2 in (12). Same to MOEA/D-LTD, the learning module of NSGA-III-LTD is performed every τ generations between ψ_b and ψ_e of $maxGen$.

IV. Experimental Settings

In this section, we describe the settings of our experimental studies, including the algorithms in comparison, test problems, parameter settings and performance indicators.

A. Test Algorithms

In the experimental studies, we compare MOEA/D-LTD and NSGA-III-LTD with four state-of-the-art decomposition-based EMO algorithms, including MOEA/D [8], MOEA/D-PaS [22], RVEA* [37] and A-NSGA-III [36]. In particular, MOEA/D-PaS is a variant of MOEA/D with Pareto-adaptive subproblem formulation. RVEA* and A-NSGA-III are the variants of RVEA [37] and NSGA-III with adaptive reference points adjustment. The LTD paradigm is implemented based on the GPLM toolbox [25]. For the test algorithms, we use the published codes of MOEA/D by Zhang *et al.* [8], MOEA/D-PaS by Wang *et al.* [22], RVEA* and A-NSGA-III by Tian *et al.* [38]. All algorithms are implemented in Matlab.

B. Test Problems

To investigate the effectiveness of our proposed LTD paradigm especially on problems with irregular PF shapes, totally 14 test problems with different PF shapes are selected from the WFG4x [21], DTLZ [39], WFG^{-1} [13] test suites, i.e., WFG41 to WFG48, DTLZ5, DTLZ7 and WFG1^{-1} to WFG4^{-1}. Different from [21], the i-th objective of WFG41 to WFG48 is scaled by i like the WFG test problems in [40]. For problems from WFG4 and WFG^{-1} test suites, the number of decision variables $n = k + l$ is set with $k = 2 \times (m - 1)$ and $l = 20$ as suggested in [40], where m is the number of objectives. For DTLZ test problems, $n = m + r - 1$ is set with $r = 10$ for DTLZ5 and $r = 20$ for DTLZ7 as suggested in [39].

TABLE I: Settings of Population Size and Maximum Number of Generations.

Test Problem	m	N	$maxGen$
WFG41 to WFG48	2	100	250
WFG41 to WFG48	3	91	400
WFG41 to WFG48	5	210	750
DTLZ5, DTLZ7	3	91	300
WFG1^{-1} to WFG4^{-1}	3	91	250
WFG1^{-1} to WFG4^{-1}	5	210	400

C. Parameter Settings

Referring to [11] and [21], the population size N and maximum number of generations $maxGen$ of all six algorithms are set according to Table I. The specific parameter settings of MOEA/D-LTD of NSGA-III-LTD are listed as follows:

- Reproduction operators: The simulated binary crossover (SBX) [41] and polynomial mutation [42] are adopted for offspring reproduction. For the SBX operator, we set the crossover probability $p_c = 1$, the distribution index η_c of MOEA/D-LTD and NSGA-III-LTD to be 20 and 30, respectively. For the polynomial mutation, we set the mutation probability $p_m = \frac{1}{n}$ and distribution index $\eta_m = 20$.
- Neighborhood size for MOEA/D-LTD: $T = 20$.
- Probability of mating selection in the neighborhood for MOEA/D-LTD: $\delta = 0.9$.
- The beginning percentage of LTD procedure: $\psi_b = 50\%$;
- The end percentage of LTD procedure: $\psi_e = 80\%$;
- The interval to perform LTD procedure: $\tau = 20$;

For fair comparisons, the common parameters of MOEA/D and MOEA/D-PaS share the same settings with the MOEA/D-LTD. Note that MOEA/D adopts PBI subproblem formulation. Other settings of the algorithms in comparisons are kept the same as in their original papers.

D. Performance Metrics

The inverted generational distance (IGD) [43] and HV metrics are chosen to assess the performance of the algorithms. Both the IGD and HV metrics evaluate the convergence and diversity of a solution set simultaneously. A smaller IGD or a larger HV typically indicates better convergence and diversity. In particular, the IGD metric requires a reference set of points evenly spread on the PF, whereas the HV metric requires to specify a worse point, which is dominated by the

nadir point. On the one hand, the HV metric is shown to be sensitive to the specification of the worse point, especially for irregular PF shapes [16]. On the other hand, as discussed in [44], the IGD metric tends to favor a set of solutions with a similar distribution to the reference set, which gets worse when the number of objectives increases. Therefore, we generate the reference set for each 2- or 3-objective test problem by selecting 1,000 points from 1,000,000 randomly generated samples on the normalized PF using the method introduced in Section III-B. For 5-objective test problems, we set the worse point to be $(1.1, \cdots, 1.1)^T$. Before calculating the IGD and HV, the solution set obtained by each algorithm are normalized by the same scales that normalize the PF into $[0, 1]^m$. In the experimental studies, each algorithm is run 31 times on each test problem. The medians and interquartile ranges (IQRs) of the IGDs and HVs are calculated and shown in the tables, where the results of the best two algorithms are highlighted in bold with dark and light gray backgrounds, respectively. Meanwhile, the Wilcoxon's rank sum test at a significant level of 5% is performed to investigate whether the differences are significant or not.

V. Experimental Studies

A. Performance Comparisons on Multiobjective Test Problems

The medians and IQRs of the IGD results on bi-objective WFG41 to WFG48 test problems are presented in Table II. As shown in the table, the proposed MOEA/D-LTD achieves the best median IGDs on 7 out of 8 test problems, where it also significantly outperforms all other algorithms except for NSGA-III-LTD according to Wilcoxon rank sum test. MOEA/D-LTD is beaten by RVEA* and A-NSGA-III on WFG47 test problem. The IQR of IGDs obtained by MOEA/D-LTD is smaller in 36 out of 40 comparisons, suggesting that the leading performance of MOEA/D-LTD is stable over different test problems. NSGA-III-LTD remains the second best algorithms on 6 test problems and is only beaten by MOEA/D, RVEA* and A-NSGA-III once.

The final solution set with the best IGDs obtained by different algorithms on bi-objective WFG41 to WFG48 test problems are shown in Fig. 5 and Fig. 6. As can be seen from the figures, the PFs of WFG41 to WFG48 test problems have difference characteristics. WFG41 test problem has a classic concave PF shape, which is the most common PF shape in the popular test suites, including DTLZ and WFG test suites. Obviously, MOEA/D-LTD and NSGA-III-LTD obtains the best solution setd in terms of both the convergence and diversity. RVEA* and A-NSGA-III, which also dynamically adjust the reference points, struggle to maintain evenly distributed solutions along the PF. MOEA/D with fixed reference vectors suffers from the different scales of the objectives. In addition, the final solution set obtained by MOEA/D are not as close to the PF as the former mentioned four algorithms due to the small opening of the contours induced by PBI subproblem formulation. The final solution set found by MOEA/D-PaS fails to cover the entire PF. On WFG42 test problem, which has a convex PF, almost all algorithms tends to find solution sets that concentrate on the center part of the PF except for MOEA/D-LTD and NSGA-III-LTD. WFG43 and WFG44 have sharp concave and convex PFs, which lead to further degeneration of the algorithms in caparisons and highlights the strength of the proposed LTD paradigm. The reason why MOEA/D-LTD is still able to maintain solutions widely spread along the PF could be due to the widely spread reference points together with the search directions normal to the estimated PF. In contrast, the non-dominated sorting-based selection of NSGA-III-LTD may weaken its ability to find solutions in difficult regions of the PF [10]. In the case of WFG45 test problem with a PF of mixed shape and WFG46 test problem with a linear PF, the leading performance of the proposed MOEA/D-LTD and NSGA-III-LTD remain in terms of both the convergence and diversity. WFG47 and WFG48 test problems have discontinuous PFs with three segments. Note that MOEA/D-LTD is beaten by RVEA*, A-NSGA-III and NSGA-III-LTD on WFG47 test problem due to the missing part on the third segment. We infer from the final solution set obtained by MOEA/D-LTD that the subproblems with respect to the reference points on the missing part are assigned solutions on the tail of the second segment, which are too close to the search directions. The performance pf MOEA/D-LTD on WFG48 test problem is not affected. It is worth noting that dominated solutions at the discontinuous regions of the PFs are maintained by MOEA/D due to the small opening of the contours.

The IGD results on 3-objective test problems are provided in Table III. On WFG4x test suite, the median IGDs obtained by MOEA/D-LTD and NSGA-III-LTD are the best on all 8 test problems except for WFG43 and WFG47, where RVEA* and A-NSGA-III obtains the second best median IGDs, respectively. The IGD results on DTLZ5, DTLZ7 and WFG1^{-1} to WFG4^{-1} test problems are even better. MOEA/D-LTD and NSGA-III-LTD show the lowest median IGDs and are significantly better than any other algorithm in comparisons except on DTLZ7 test problem, RVEA* obtains comparable results. This strength of MOEA/D-LTD and NSGA-III-LTD is owe to the LTD paradigm that adapts the decomposition to the PF shapes in terms of both the reference points and the subproblem formulation. Even though MOEA/D-PaS adopts the Pareto-adaptive subproblem formulation, RVEA* and A-NSGA-III adjust the reference points dynamically, they seem to be less effective on these test problems. Comparing MOEA/D-LTD and NSGA-III-LTD, we find that the overall performance of MOEA/D-LTD is better than NSGA-III-LTD, where MOEA/D-LTD is significantly better on 7 out of 13 test problems while NSGA-III-LTD wins on 4 test problems.

The final solution sets with the best IGDs obtained by the six algorithms on 3-objective test problems are demonstrated in Fig. 7 to Fig. 10. We can observe from Fig. 7 and Fig. 8 that when the PF of the test problem is convex or has convex parts, e.g., WFG42, WFG44 and WFG48 test problems, the performance of the MOEA/D, MOEA/D-PaS, RVEA* and A-NSGA-III deteriorate significantly. Even on WFG41 and WFG46, which has simpler PFs, these four algorithms struggle to maintain evenly spread and well-converged solution sets. While RVEA* maintains solutions that are moderately evenly distributed upon the PFs of WFG41 and WFG43 test problems,

TABLE II: IGD Results on Bi-Objective Test Problems.

Problem	MOEA/D	MOEA/D-PaS	RVEA*	A-NSGA-III	MOEA/D-LTD	NSGA-III-LTD
WFG41	1.499e-2(1.696e-3)$^{\downarrow}_{-}$	2.137e-2(2.943e-3)$^{\downarrow}_{-}$	6.256e-3(6.420e-4)$^{\downarrow}_{-}$	7.838e-3(1.088e-3)$^{\downarrow}_{-}$	**4.825e-3(2.966e-4)**$_{+}$	**5.647e-3(5.594e-4)**
WFG42	5.326e-2(6.397e-3)$^{\downarrow}_{-}$	2.003e-2(3.455e-3)$^{\downarrow}_{-}$	1.438e-2(1.074e-3)$^{\downarrow}_{-}$	1.330e-2(2.679e-3)$^{\downarrow}_{-}$	**5.153e-3(4.183e-4)**$_{+}$	**6.635e-3(6.860e-4)**
WFG43	**1.605e-2(2.104e-3)**$^{\downarrow}_{+}$	9.111e-2(1.500e-2)$^{\downarrow}_{-}$	3.204e-2(5.502e-3)$^{\downarrow}_{-}$	2.941e-2(7.422e-3)$^{\downarrow}_{-}$	**5.852e-3(1.184e-3)**$_{+}$	2.435e-2(1.018e-2)
WFG44	1.790e-1(6.585e-3)$^{\downarrow}_{-}$	6.817e-2(2.476e-2)$^{\downarrow}_{-}$	6.475e-2(1.100e-2)$^{\downarrow}_{-}$	8.918e-2(4.652e-2)$^{\downarrow}_{-}$	**4.147e-2(1.778e-2)**$_{\approx}$	**4.151e-2(2.023e-2)**
WFG45	1.329e-2(1.319e-3)$^{\downarrow}_{-}$	2.527e-2(3.097e-3)$^{\downarrow}_{-}$	6.832e-3(5.601e-4)$^{\downarrow}_{-}$	7.673e-3(7.846e-4)$^{\downarrow}_{-}$	**4.861e-3(3.206e-4)**$_{+}$	**5.906e-3(1.206e-3)**
WFG46	1.288e-2(2.610e-3)$^{\downarrow}_{-}$	2.060e-2(2.979e-3)$^{\downarrow}_{-}$	7.017e-3(1.170e-3)$^{\downarrow}_{-}$	8.475e-3(1.411e-3)$^{\downarrow}_{-}$	**4.639e-3(3.511e-4)**$_{+}$	**6.227e-3(9.106e-4)**
WFG47	2.851e-1(8.392e-3)$^{\downarrow}_{-}$	1.215e-1(8.041e-2)$^{\downarrow}_{-}$	**6.172e-3(1.005e-3)**$^{\uparrow}_{+}$	**6.035e-3(1.269e-3)**$^{\uparrow}_{+}$	8.970e-3(2.356e-3)$_{-}$	6.836e-3(1.810e-3)
WFG48	1.210e-1(3.626e-1)$^{\downarrow}_{-}$	3.494e-2(1.091e-2)$^{\downarrow}_{-}$	1.099e-2(9.894e-4)$^{\downarrow}_{-}$	8.038e-3(1.094e-3)$^{\downarrow}_{-}$	**6.033e-3(6.978e-4)**$_{+}$	**6.567e-3(1.191e-3)**

According to Wilcoxon rank sum test, $\uparrow$, $\downarrow$ and $\|$ indicate that the corresponding algorithm is significantly better than, worse than or similar to MOEA/D-LTD, while $_{+}$, $_{-}$ and $_{\approx}$ indicate that the corresponding algorithm is significantly better than, worse than or similar to NSGA-III-LTD.

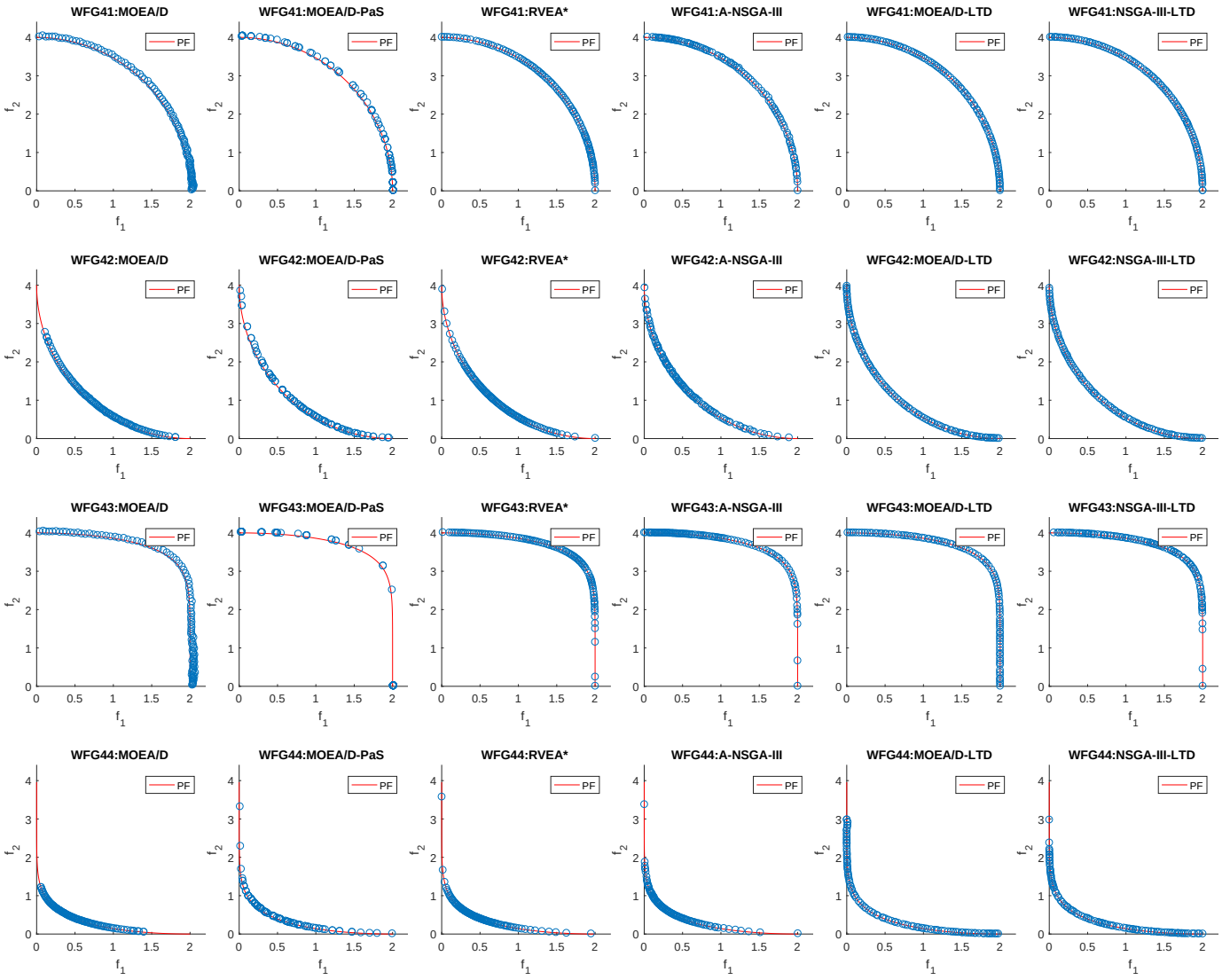


Fig. 5: Final solution sets obtained by 6 algorithms with the best IGDs on bi-objective WFG41 to WFG44 test problems.

MOEA/D-LTD and NSGA-III-LTD tend to prefer solutions on the boundaries of the PFs, which is good for exploring the entire PFs, e.g., on DTLZ7 and WFG4^{-1} test problems. If we give more credit to RVEA* in terms of population diversity, the better IGD results of MOEA/D-LTD presented in Fig. 7 could indicate better convergence and robustness. The PF of DTLZ5 test problem is degenerated to a curve, where the algorithms that dynamic adjust or sample reference points according to the estimated PF present obvious good performance since the reference points are rearranged to the objective space where the PF exists. Similar phenomena can be seen on WFG47, WFG48 and DTLZ7 test problems whose PFs are

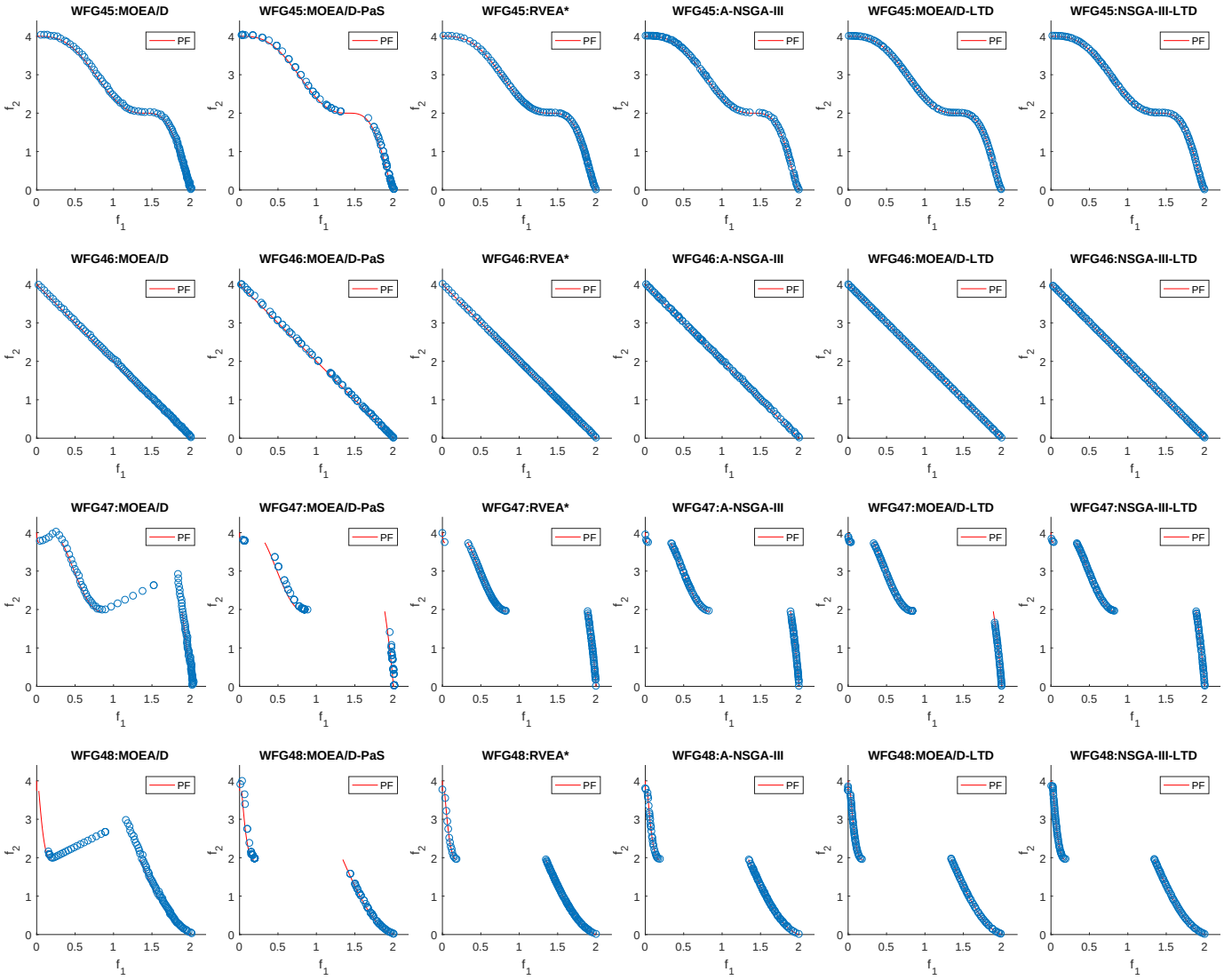


Fig. 6: Final solution sets obtained by 6 algorithms with the best IGDs on bi-objective WFG45 to WFG48 test problems.

TABLE III: IGD Results on 3-Objective Test Problems.

Problem	MOEA/D	MOEA/D-PaS	RVEA*	A-NSGA-III	MOEA/D-LTD	NSGA-III-LTD
WFG41	6.866e-2(2.221e-3)$^{\downarrow}_{-}$	1.051e-1(5.644e-3)$^{\downarrow}_{-}$	6.144e-2(3.026e-3)$^{\downarrow}_{-}$	6.725e-2(2.319e-3)$^{\downarrow}_{-}$	**5.408e-2(5.270e-4)**$_{+}$	**5.534e-2(6.888e-4)**
WFG42	1.379e-1(4.901e-3)$^{\downarrow}_{-}$	6.866e-2(7.779e-3)$^{\downarrow}_{-}$	6.451e-2(8.170e-3)$^{\downarrow}_{-}$	6.016e-2(1.377e-2)$^{\downarrow}_{-}$	**3.626e-2(1.232e-3)**$_{-}$	**3.491e-2(1.506e-3)**
WFG43	8.011e-2(7.047e-4)$^{\downarrow}_{\approx}$	2.306e-1(2.597e-2)$^{\downarrow}_{-}$	**7.670e-2(5.255e-3)**$^{\shortparallel}_{+}$	8.103e-2(4.447e-3)$^{\downarrow}_{\approx}$	**7.462e-2(8.813e-3)**$_{+}$	8.243e-2(7.661e-3)
WFG44	2.599e-1(3.753e-3)$^{\downarrow}_{-}$	9.978e-2(1.682e-2)$^{\downarrow}_{-}$	1.461e-1(3.665e-2)$^{\downarrow}_{-}$	1.716e-1(3.004e-2)$^{\downarrow}_{-}$	**6.826e-2(1.249e-2)**$_{\approx}$	**7.089e-2(1.712e-2)**
WFG45	6.386e-2(1.508e-3)$^{\downarrow}_{-}$	1.017e-1(3.551e-3)$^{\downarrow}_{-}$	5.902e-2(1.607e-3)$^{\downarrow}_{-}$	6.510e-2(4.888e-3)$^{\downarrow}_{-}$	**5.100e-2(6.467e-4)**$_{+}$	**5.239e-2(6.301e-4)**
WFG46	5.365e-2(1.519e-3)$^{\downarrow}_{-}$	7.149e-2(4.835e-3)$^{\downarrow}_{-}$	5.231e-2(2.386e-3)$^{\downarrow}_{-}$	6.028e-2(3.441e-3)$^{\downarrow}_{-}$	**4.141e-2(4.405e-4)**$_{+}$	**4.553e-2(1.293e-3)**
WFG47	4.548e-1(3.861e-1)$^{\downarrow}_{-}$	1.340e-1(8.357e-3)$^{\downarrow}_{-}$	5.271e-2(1.846e-3)$^{\uparrow}_{-}$	**5.225e-2(2.221e-3)**$^{\uparrow}_{-}$	6.643e-2(1.121e-2)$_{-}$	**4.736e-2(2.792e-3)**
WFG48	4.939e-1(4.451e-1)$^{\downarrow}_{-}$	1.357e-1(1.841e-2)$^{\downarrow}_{-}$	7.642e-2(1.073e-2)$^{\downarrow}_{-}$	7.322e-2(1.069e-2)$^{\downarrow}_{-}$	**4.757e-2(7.144e-3)**$_{-}$	**4.437e-2(1.915e-3)**
DTLZ5	4.129e-2(5.991e-5)$^{\downarrow}_{-}$	2.219e-2(1.112e-3)$^{\downarrow}_{-}$	7.991e-3(6.487e-4)$^{\downarrow}_{-}$	1.340e-2(1.810e-3)$^{\downarrow}_{-}$	**5.858e-3(3.983e-4)**$_{+}$	**6.946e-3(1.730e-3)**
DTLZ7	8.047e-2(3.962e-4)$^{\downarrow}_{-}$	2.994e-1(5.428e-2)$^{\downarrow}_{-}$	4.030e-2(2.876e-1)$^{\shortparallel}_{\approx}$	4.620e-2(1.352e-3)$^{\downarrow}_{-}$	**4.011e-2(2.426e-3)**$_{\approx}$	**3.936e-3(2.120e-3)**
WFG1^{-1}	1.202e-1(2.722e-1)$^{\downarrow}_{-}$	4.555e-1(4.173e-2)$^{\downarrow}_{-}$	1.378e-1(1.612e-1)$^{\downarrow}_{-}$	4.839e-2(1.817e-3)$^{\downarrow}_{-}$	**3.863e-2(2.794e-4)**$_{-}$	**3.691e-3(9.429e-4)**
WFG2^{-1}	7.451e-2(5.309e-4)$^{\downarrow}_{-}$	8.439e-1(5.767e-2)$^{\downarrow}_{-}$	5.797e-2(1.631e-3)$^{\downarrow}_{-}$	9.555e-2(1.674e-2)$^{\downarrow}_{-}$	**5.261e-2(9.220e-4)**$_{\approx}$	**5.271e-2(6.286e-4)**
WFG3^{-1}	6.479e-2(1.975e-4)$^{\downarrow}_{-}$	4.737e-1(2.344e-2)$^{\downarrow}_{-}$	4.594e-2(3.017e-3)$^{\downarrow}_{-}$	5.336e-2(1.275e-3)$^{\downarrow}_{-}$	**4.046e-2(4.464e-4)**$_{+}$	**4.082e-2(5.687e-4)**
WFG4^{-1}	1.077e-1(3.313e-3)$^{\downarrow}_{-}$	7.886e-1(7.087e-2)$^{\downarrow}_{-}$	7.023e-2(6.462e-3)$^{\downarrow}_{-}$	9.283e-2(9.550e-3)$^{\downarrow}_{-}$	**5.448e-2(1.114e-3)**$_{+}$	**5.496e-2(5.640e-4)**

According to Wilcoxon rank sum test, $^{\uparrow}$, $^{\downarrow}$ and $^{\shortparallel}$ indicate that the corresponding algorithm is significantly better than, worse than or similar to MOEA/D-LTD, while $_{+}$, $_{-}$ and $_{\approx}$ indicate that the corresponding algorithm is significantly better than, worse than or similar to NSGA-III-LTD.

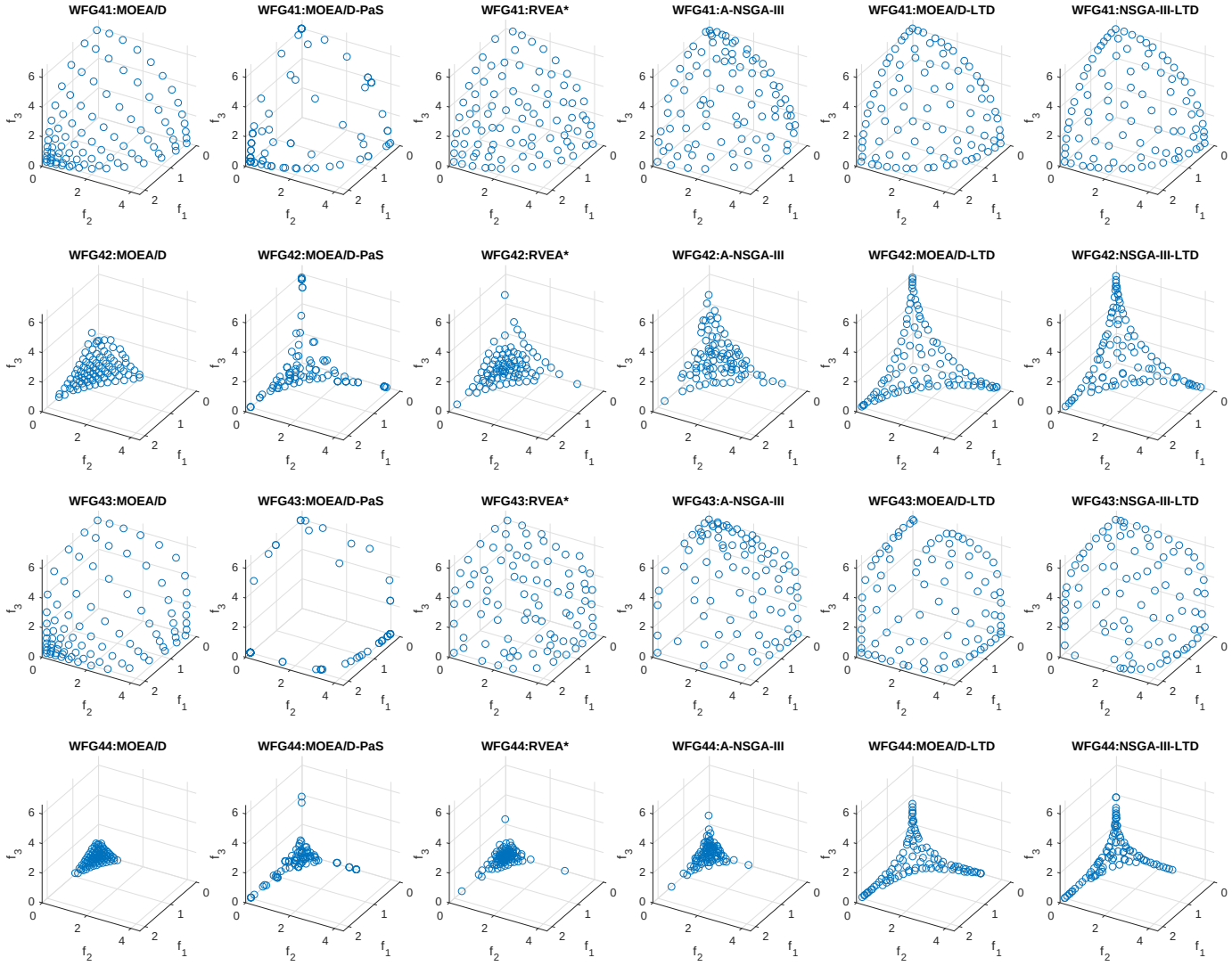


Fig. 7: Final solution sets obtained by 6 algorithms with the best IGDs on 3-objective WFG41 to WFG44 test problems.

discontinuous. WFG1^{-1} to WFG4^{-1} have PF shapes that differ quite much from commonly used test problems, challenging both the reference points and subproblem formulation. From Fig. 10, we can see that MOEA/D-PaS with so-called Pareto adaptive subproblem formulation totally fail to find solutions covering the PF. MOEA/D, RVEA* and A-NSGA-III perform moderately better but struggle to maintain widely spread solutions. In contrast, MOEA/D-LTD and NSGA-III-LTD keep their good performance on these irregular PFs.

B. Performance Comparisons on 5-objective Test Problems

The HV results of the 6 algorithms on 5-objective test problems are given in Table IV. For the WFG4x test suite, MOEA/D-LTD achieves the highest median HVs on 6 out of 8 test problems, where all four existing algorithms in comparisons are significantly outperformed by MOEA/D-LTD according to the Wilcoxon rank sum test. On test problems where MOEA/D-LTD does not perform the best, i.e., WFG43 and WFG47, the proposed NSGA-III-LTD becomes the best algorithm. It is worth noting that A-NSGA-III never beats or achieves comparable results to NSGA-III-LTD on WFG4x test problems, which shows the universal effectiveness of the LTD paradigm even without specially designed subproblem formulation. As for WFG^{-}1 test problems, MOEA/D-LTD and NSGA-III-LTD remain the best two algorithms on WFG1^{-}1 to WFG3^{-}1 test problems. Whereas, RVEA* and MOEA/D achieves the best HV results on WFG4^{-}1 test problem.

Fig. 11 to Fig. 13 demonstrate the final solution sets obtained by the 6 algorithms with the best HVs on WFG41 to WFG48 and WFG1^{-1} to WFG4^{-1} test problems. The ideal and nadir points of WFG4x test problems are $(0, \cdots, 0)^T$ and $(2, \cdots, 2 \times m)^T$. From the objective value paths shown in Fig. 11 and Fig. 12, we observe that MOEA/D-LTD and NSGA-III-LTD are able to maintain widely spread solutions on most of the test problems. Although their final solution sets fail to cover the entire PF on WFG44 test problem, the other algorithms perform even worse. MOEA/D, RVEA* and A-NSGA-III also obtain solutions that are widely spread on the PFs of some WFG4x test problems. However, the distribution of their final solution sets on some test problems,

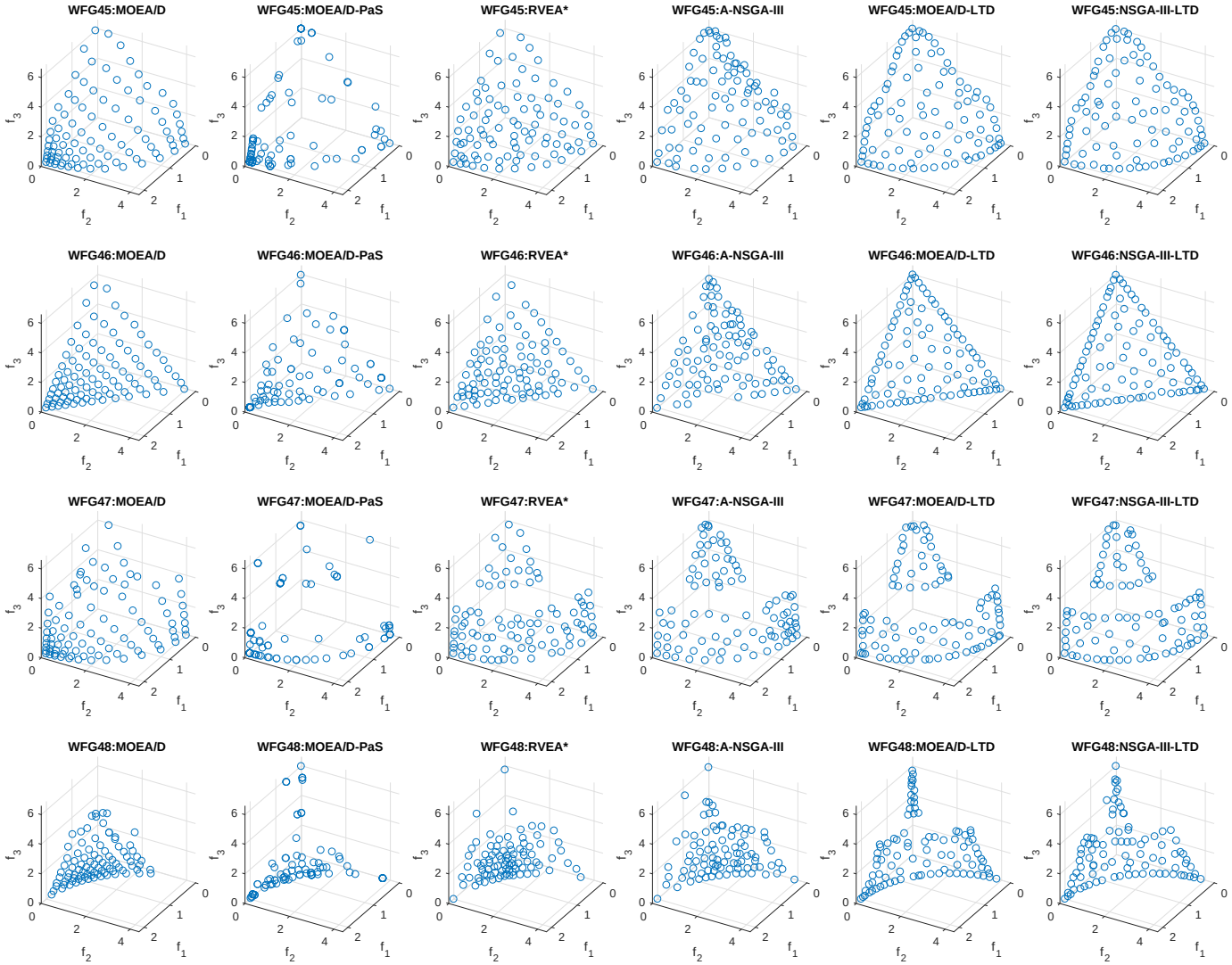


Fig. 8: Final solution sets obtained by 6 algorithms with the best IGDs on 3-objective WFG45 to WFG48 test problems.

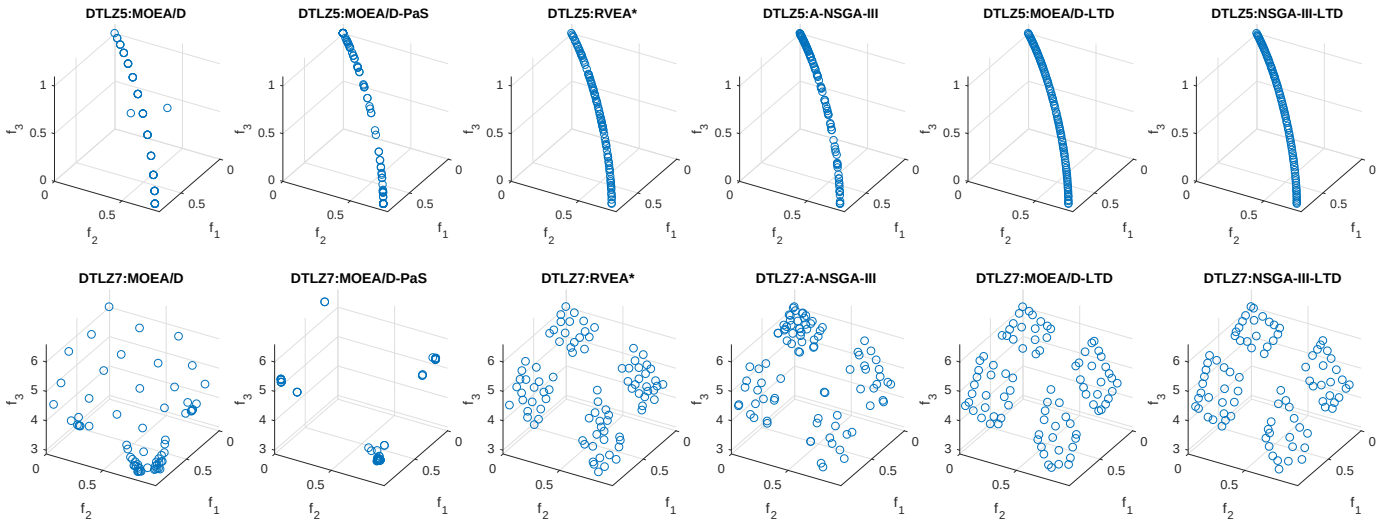


Fig. 9: Final solution sets obtained by 6 algorithms with the best IGDs on 3-objective DTLZ5 and DTLZ7 test problems.

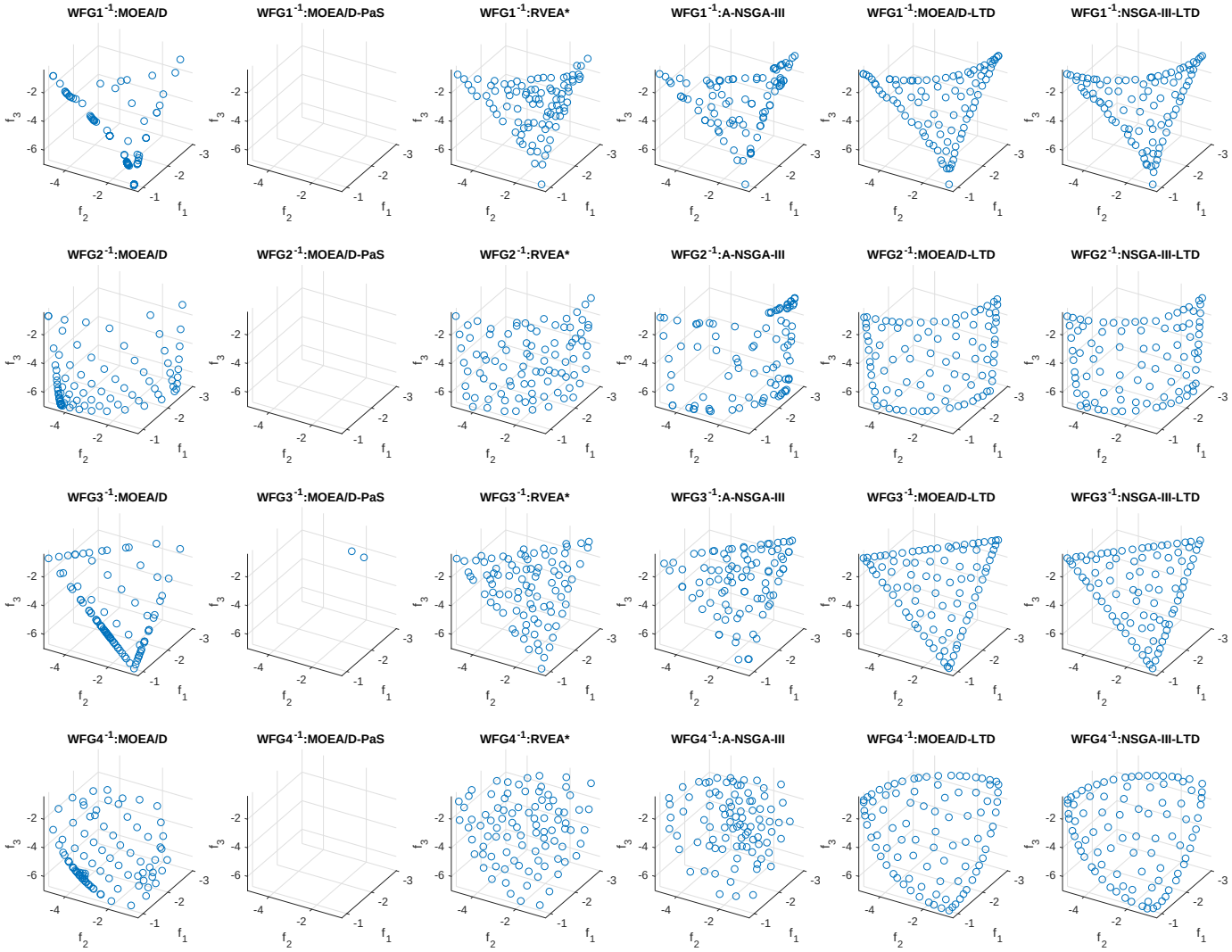


Fig. 10: Final solution sets obtained by 6 algorithms with the best IGDs on 3-objective WFG1^{-1} to WFG4^{-1} test problems.

TABLE IV: HV Results on 5-Objective Test Problems.

Problem	MOEA/D	MOEA/D-PaS	RVEA*	A-NSGA-III	MOEA/D-LTD	NSGA-III-LTD
WFG41	1.151e+0(3.654e-2)$^{\downarrow}_{-}$	8.614e-1(1.248e-1)$^{\downarrow}_{-}$	1.159e+0(1.800e-2)$^{\downarrow}_{-}$	1.193e+0(1.664e-2)$^{\downarrow}_{-}$	**1.285e+0(6.470e-3)**$_{+}$	**1.231e+0(1.292e-2)**
WFG42	1.553e+0(8.391e-3)$^{\downarrow}_{-}$	1.585e+0(1.815e-2)$^{\downarrow}_{-}$	1.563e+0(7.860e-3)$^{\downarrow}_{-}$	1.552e+0(1.327e-2)$^{\downarrow}_{-}$	**1.607e+0(6.442e-4)**$_{+}$	**1.586e+0(3.592e-3)**
WFG43	7.073e-1(5.194e-2)$^{\downarrow}_{-}$	6.617e-1(5.461e-2)$^{\downarrow}_{-}$	7.910e-1(1.164e-2)$^{\downarrow}_{-}$	7.786e-1(1.416e-1)$^{\parallel}_{-}$	**8.168e-1(1.562e-2)**$_{-}$	**8.342e-1(1.825e-2)**
WFG44	1.588e+0(3.012e-3)$^{\downarrow}_{-}$	**1.609e+0(1.038e-3)**$^{\downarrow}_{+}$	1.592e+0(3.744e-3)$^{\downarrow}_{-}$	1.588e+0(5.000e-3)$^{\downarrow}_{-}$	**1.610e+0(1.750e-4)**$_{+}$	1.603e+0(1.923e-3)
WFG45	**1.325e+0(1.399e-2)**$^{\downarrow}_{-}$	9.065e-1(5.460e-2)$^{\downarrow}_{-}$	1.252e+0(1.517e-2)$^{\downarrow}_{-}$	1.286e+0(1.622e-2)$^{\downarrow}_{-}$	**1.359e+0(9.294e-3)**$_{-}$	1.318e+0(1.278e-2)
WFG46	1.510e+0(7.133e-3)$^{\downarrow}_{-}$	1.386e+0(1.053e-1)$^{\downarrow}_{-}$	1.473e+0(8.151e-3)$^{\downarrow}_{-}$	1.501e+0(1.034e-2)$^{\downarrow}_{-}$	**1.562e+0(2.162e-3)**$_{+}$	**1.518e+0(7.663e-3)**
WFG47	8.895e-1(4.420e-1)$^{\downarrow}_{-}$	8.302e-1(1.176e-1)$^{\downarrow}_{-}$	1.254e+0(1.119e-2)$^{\downarrow}_{-}$	**1.310e+0(1.445e-2)**$^{\parallel}_{-}$	1.303e+0(1.099e-1)$_{\approx}$	**1.334e+0(9.970e-3)**
WFG48	1.541e+0(6.513e-1)$^{\downarrow}_{-}$	**1.564e+0(1.669e-2)**$^{\downarrow}_{-}$	1.552e+0(8.322e-3)$^{\downarrow}_{\approx}$	1.551e+0(1.073e-2)$^{\downarrow}_{-}$	**1.604e+0(7.246e-4)**$_{-}$	1.557e+0(1.584e-2)
WFG1^{-1}	8.631e-4(4.237e-4)$^{\downarrow}_{-}$	0.000e+0(0.000e+0)$^{\downarrow}_{-}$	2.283e-3(6.192e-4)$^{\downarrow}_{-}$	2.352e-3(3.643e-4)$^{\downarrow}_{-}$	**3.646e-3(1.588e-4)**$_{\approx}$	**3.644e-3(8.947e-5)**
WFG2^{-1}	3.719e-3(2.745e-3)$^{\downarrow}_{-}$	0.000e+0(0.000e+0)$^{\downarrow}_{-}$	8.175e-3(1.853e-4)$^{\downarrow}_{-}$	8.919e-3(7.409e-4)$^{\downarrow}_{-}$	**1.236e-2(1.461e-3)**$_{+}$	**1.108e-2(1.516e-3)**
WFG3^{-1}	3.412e-3(2.663e-4)$^{\downarrow}_{-}$	1.204e-3(3.329e-4)$^{\downarrow}_{-}$	8.794e-3(7.907e-4)$^{\downarrow}_{-}$	1.118e-2(9.254e-4)$^{\downarrow}_{-}$	**1.584e-2(1.775e-3)**$_{+}$	**1.328e-2(4.538e-4)**
WFG4^{-1}	**1.340e-1(1.333e-2)**$^{\uparrow}_{+}$	0.000e+0(2.545e-8)$^{\downarrow}_{-}$	**1.803e-1(8.635e-3)**$^{\uparrow}_{+}$	1.082e-1(4.742e-2)$^{\uparrow}_{+}$	4.343e-2(6.051e-3)$_{-}$	5.607e-2(9.745e-3)

According to Wilcoxon rank sum test, $\uparrow$, $\downarrow$ and $\parallel$ indicate that the corresponding algorithm is significantly better than, worse than or similar to MOEA/D-LTD, while $_{+}$, $_{-}$ and $_{\approx}$ indicate that the corresponding algorithm is significantly better than, worse than or similar to NSGA-III-LTD.

e.g., WFG42, WFG44, WFG46 and WFG48, can be very poor. Besides, MOEA/D struggles to maintain all extreme solutions. Even though MOEA/D-PaS performs better in finding the extreme solutions, its population diversity is the worst

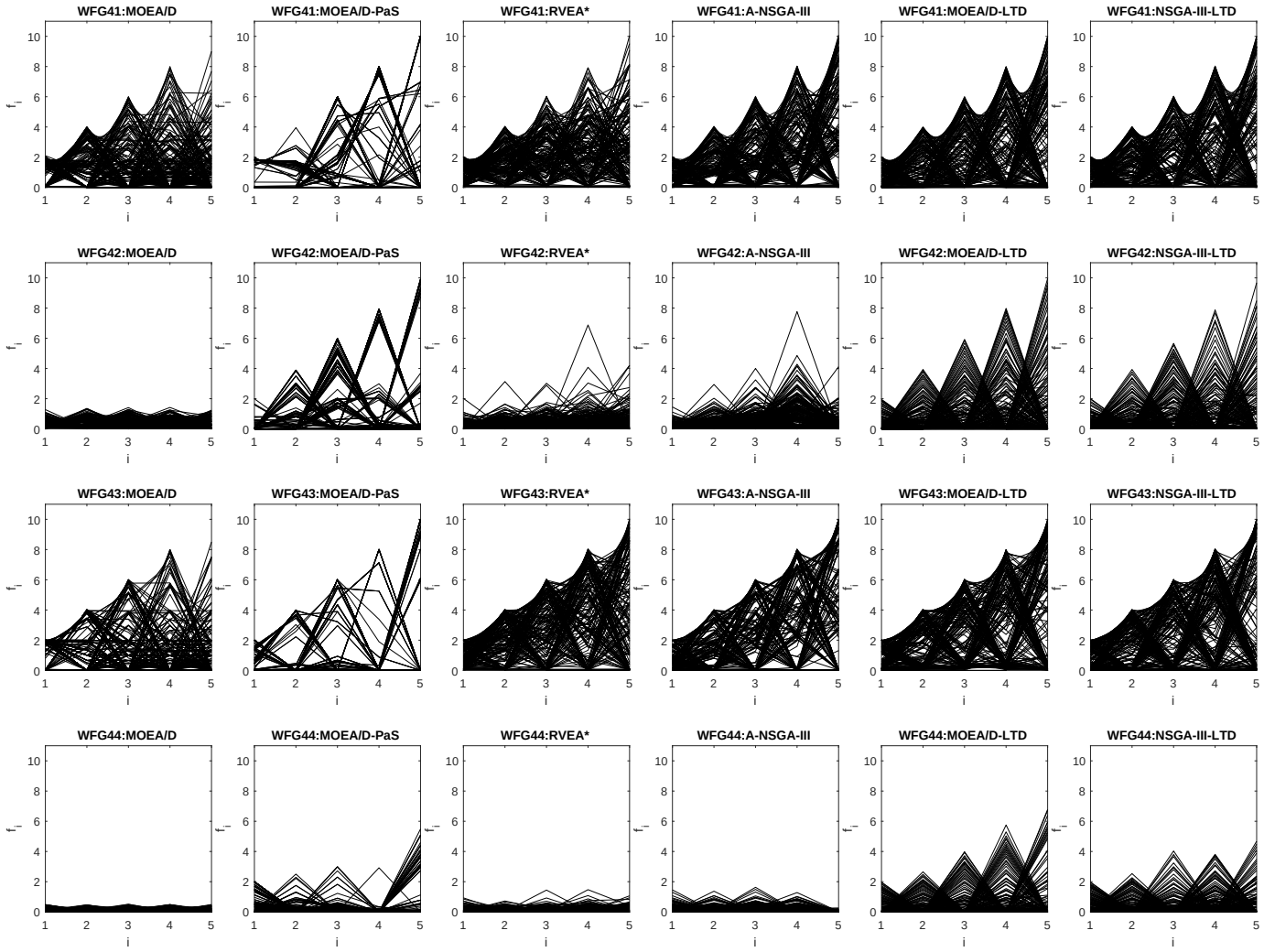


Fig. 11: Final solution sets obtained by 6 algorithms with the best HVs on 5-objective WFG41 to WFG44 test problems.

among all 6 algorithms. When it comes to WFG^{-1} test problems whose ideal point is $(-1, \cdots, 1 - 2 \times m)^T$ and nadir points is $(-1, \cdots, -1)^T$, the algorithms with dynamic reference points adjustment or LTD procedure, which samples reference points on the estimated PF, show clearly better performance than MOEA/D and MOEA/D-PaS. MOEA/D-PaS even cannot obtain well-converged solutions. Nevertheless, the population diversity of RVEA* and A-NSGA-III are not as good as MOEA/D-LTD and NSGA-III-LTD according to the objective value paths shown in Fig. 13. Note that MOEA/D-LTD performs the best on WFG46-1 test problem best as demonstrated in Fig. 13 in terms of both the population diversity and maintenance of extreme solutions. Thus, we doubt that the worst point $(1.1, \cdots, 1.1)^T$ used to calculate the HVs are suitable for this discontinuous PF, which shows MOEA/D and RVEA* are better than MOEA/D-LTD.

C. Parameter Studies of LTD Procedure

We setup three experiments to investigate the sensitivities of the parameters in LTD procedure, i.e., the beginning percentage ψ_b, the end percentage ψ_e and the LTD interval τ. MOEA/D-LTD is run on each bi-/3-objective test problem 21 times with different settings.

- The first experiment is on the beginning percentage ψ_b. Given the settings of $\psi_e = 90\%$ and $\tau = 20$, ψ_b is set to be 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, and 90%, respectively. Note that the setting of $\psi_b = \psi_e = 90\%$ indicates that the LTD procedure is never activated in the whole run. The median IGD values of different ψ_b settings are given in Fig. 14. As shown in the figure, the performance of MOEA/D-LTD is insensitive to ψ_b on most of the test problems as long as the LTD procedure is activated during the optimization. It is worth noting that the IGD performance on 3-objective WFG44 test problem is improved as ψ_b increases. The reason could be that the non-dominated solutions in the early stage of the optimization process are not good enough to estimate the PF, thus misleading the decomposition. However, when LTD procedure is totally gone, the overall performance of MOEA/D-LTD deteriorates significantly. The only exceptions are bi-objective WFG47 test problem, where the subproblem

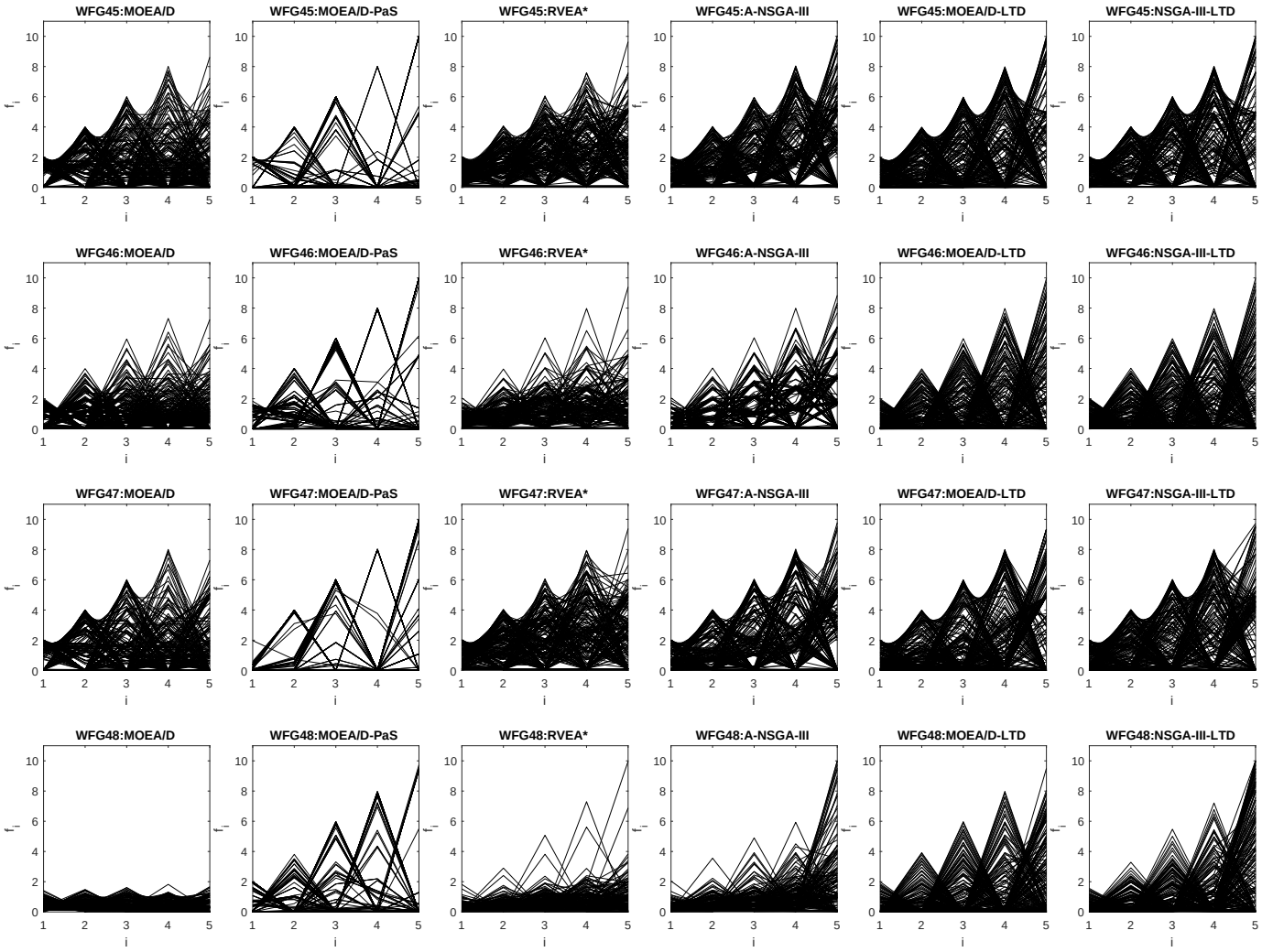


Fig. 12: Final solution sets obtained by 6 algorithms with the best HVs on 5-objective WFG45 to WFG48 test problems.

formulation of MOEA/D-LTD misses some part the PF.

- The second experiment is on the end percentage ψ_e. Given the settings of $\psi_b = 50\%$ and $\tau = 20$, ψ_e is set to be 60%, 70%, 80%, and 90%, respectively. The median IGD values of different ψ_e settings are shown in Fig. 15. Similarly, the performance of MOEA/D-LTD is not that sensitive to ψ_e. Particularly, on test problems where MOEA/D-LTD does not perform very well, the variances of the median IGDs tend to be higher. It is worth noting that the median IGD on 3-objective WFG48 test problem decreases significantly when ψ_e goes from 60% to 70% and keeps low as ψ_e increases. That is because LTD procedure is deactivated before the PF estimation is mature and the reference points are not updated any longer.
- The second experiment is on the LTD interval τ. Given the settings of $\psi_b = 50\%$ and $\psi_e = 80\%$, ψ_e is set to be 5, 10, 15, 20, 30 and 40, respectively. We present the results of the median IGD values of different τ settings in Fig. 16. As can be seen from the figure, the performance of MOEA/D-LTD on most of the test problems are not sensitive to the LTD interval τ despite some special cases. In particular, bi-/3-objective WFG44 and 3-objective WFG47 test problems prefer more frequent update, while 3-objective WFG43 test problem is easier to be solved with a relatively larger LTD interval.

D. Discussions

The comprehensive experimental studies verify the effectiveness and robustness of the proposed LTD paradigm on test problems with various PF shapes. MOEA/D-LTD and NSGA-III-LTD achieves significantly better results than their predecessors. Nevertheless, we find that there may be two factors that restricts further improvements of the LTD paradigm, which are related to how we select N reference points from the large number of samples on the estimated PF described in Section III-B.

- Firstly, the selection method uses the Euclidean distance rather than geodesic distance as the distance measure for the distance measure. When the PF is nonlinear and the curvatures vary, reference points with equal Euclidean distance in between will have different geodesic distance

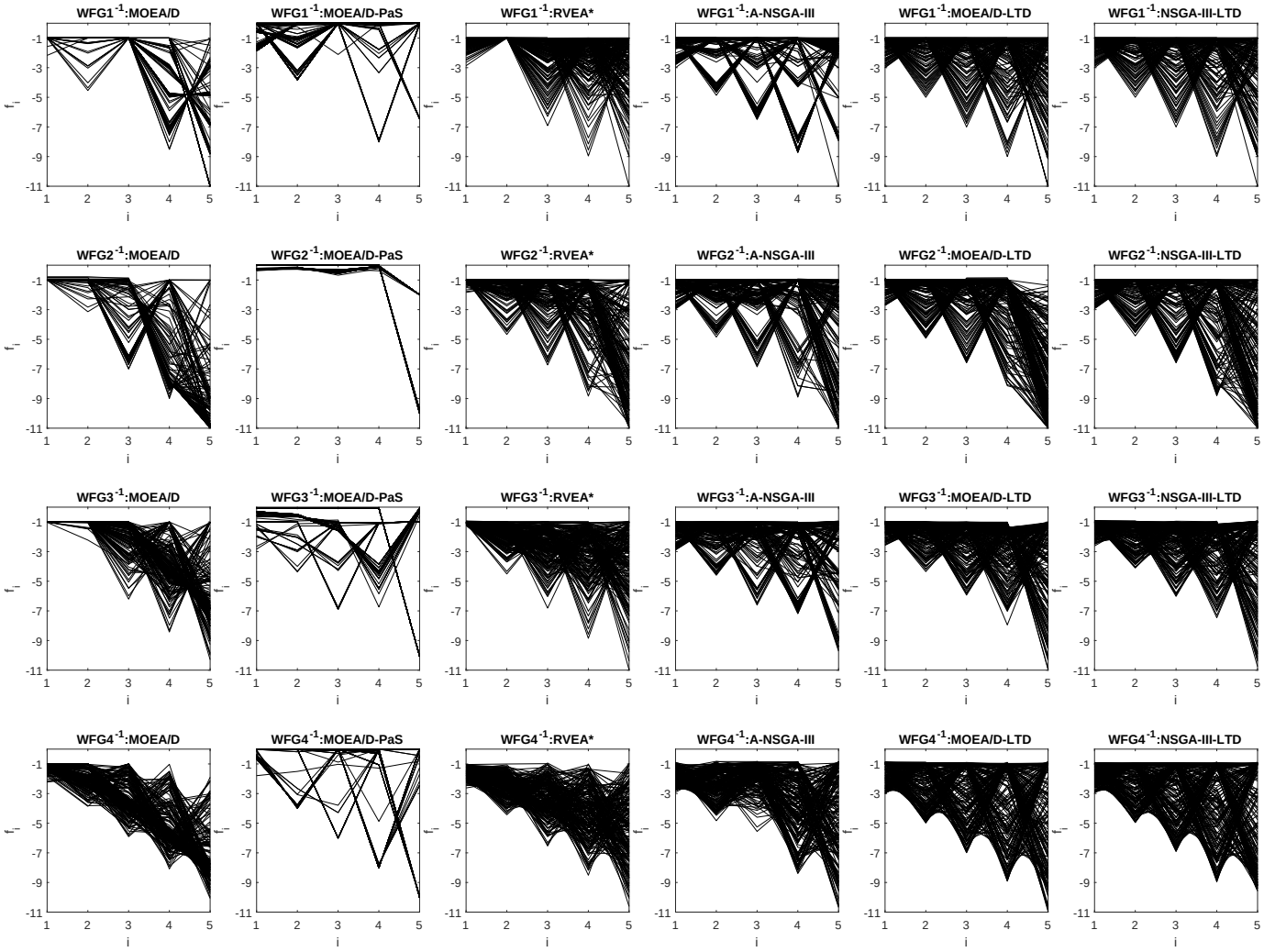


Fig. 13: Final solution sets obtained by 6 algorithms with the best HVs on 5-objective WFG1^{-1} to WFG4^{-1} test problems.

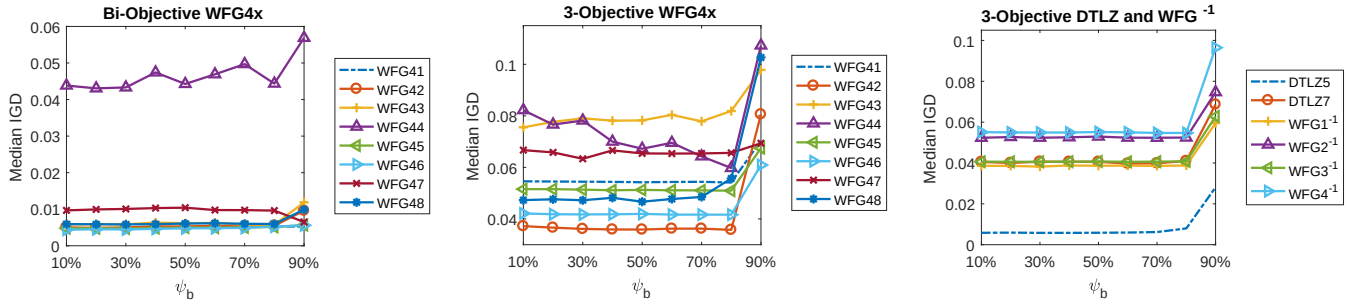


Fig. 14: Parameter studies of the beginning percentage ψ_b to perform LTD procedure. ($\psi_e = 90\%, \tau = 20$)

on the PF, which affects the distribution of the expected Pareto-optimal solutions. However, as long as N is not too small or the curvatures do not vary too fast, the performance will not degrade much.

- Secondly, the density estimation of (10) prefers reference points at the boundaries of the estimated PF. We demonstrate the training data, samples on the estimated PF and selected reference points after the last LTD procedure in the best run of MOEA/D-LTD on 3-objective WFG45 and DTLZ7 test problems in Fig. 17 as an example. This explains why the final solution sets obtained by MOEA/D-LTD and NSGA-III-LTD have more solutions at the boundaries of the PFs as shown in Fig. 7 to Fig. 10. However, this could also be the reason why MOEA/D-LTD and NSGA-III-LTD show better performance on exploring the entire PFs.

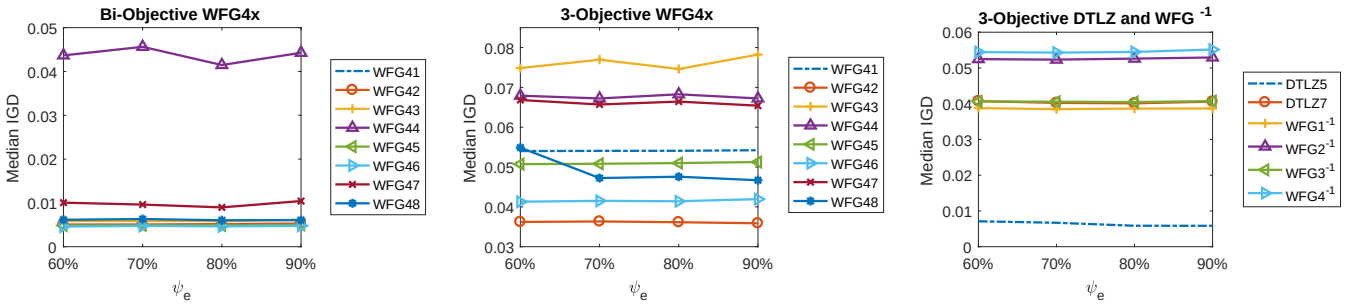


Fig. 15: Parameter studies of the end percentage ψ_e to perform LTD procedure. ($\psi_b = 50\%, \tau = 20$)

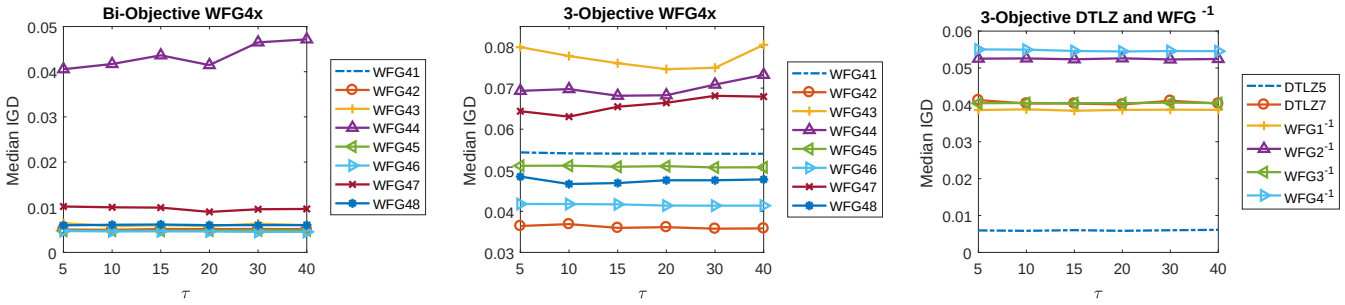


Fig. 16: Parameter studies of the interval τ to perform LTD procedure. ($\psi_b = 50\%, \psi_e = 80\%$)

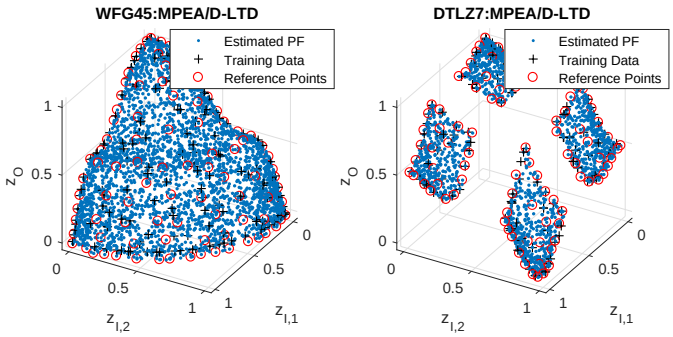


Fig. 17: Training data, samples on the estimated PF and selected reference points after the last LTD procedure in the best run of MOEA/D-LTD on 3-objective WFG45 and DTLZ7 test problems.

VI. Conclusion and Future Directions

In recent years, decomposition-based EMO algorithms have become the most popular EMO algorithms thanks to their strengthened convergence pressure by optimizing the subproblems and well-maintained population diversity by the predefined reference points. Nevertheless, when the PFs are not in line with the unit simplex, on which the reference points are evenly distributed, e.g., PFs with disparate scales, discontinuous segments or other complex shapes, they suffer from inappropriate decomposition due to unadaptable reference points and subproblem formulation. In this paper, we discuss the causes from three aspects and propose a LTD paradigm to overcome these issues. The LTD paradigm contains two parts, i.e., the optimization module that can be any decomposition-based optimizer and the learning module that periodically learns an analytical model of the estimated PF, from which useful information are extracted to guide the decomposition in the optimization module. In particular, the learned model can be used to sample reference points compliant to the estimated PF and formulate subproblems which have appropriate contours and search directions normal to the current estimated PF. Compared with several state-of-the-art adaptive methods, the performance of the proposed LTD paradigm is validated on a variety of test problems with MOEA/D and NSGA-III as the optimization modules.

LTD paradigm addressing all the three issues discussed in this paper is the first work that adapts the reference points, the contours and search directions of the subproblems for decomposition-based EMO algorithms at the same time. Nevertheless, the performance of the LTD paradigm is restricted by the selection of the reference points, which uses the Euclidean distance instead of geodesic distance as the distance measure and tends to select samples at the boundaries the PF. Future work could be investigating other efficient methods to select reference points evenly distributed on the estimated PF.

Appendix
Curvatures of The Proposed Subproblem Formulation

We consider the case of the 2-objective MOPs at first. Let $h(\mathbf{z}|\mathbf{n}^*, \mathbf{z}^*) = d_1 + \theta_1 d_2^2 + \theta_2 d_2^4 = 0$ be a contour of (11),

where

$$d_1 = (z_1 - z_1^*)n_1^* + (z_2 - z_2^*)n_2^*$$
$$d_2 = \sqrt{(z_1 - z_1^* - d_1 n_1^*)^2 + (z_2 - z_2^* - d_2 n_2^*)^2}. \quad (13)$$

Since the curvature of the contour does not depend on $\mathbf{n}^*$ and $\mathbf{z}^*$, we substitute $\mathbf{n}^* = (1,0)^T$ and $\mathbf{z}^* = (0,0)^T$ into $h(\mathbf{z}|\mathbf{n}^*, \mathbf{z}^*) = 0$ and get a contour:

$$z_1 = -\theta_1 z_2^2 - \theta_2 z_2^4. \quad (14)$$

The curvature of (14) at $\mathbf{z}^* = (0,0)^T$ can be calculated as $2\theta_1$. Therefore, the curvature of contours of the proposed subproblem formulation at its vertex $\kappa_h^* = 2\theta_1$.

For MOPs with $m > 2$ objectives, we substitute $\mathbf{n}^* = (1,0,\cdots,0)^T$ and $\mathbf{z}^* = (0,\cdots,0)^T$ into $h(\mathbf{z}|\mathbf{n}^*, \mathbf{z}^*) = 0$ and get an m-dimensional manifold:

$$z_1 = -\theta_1 \sum_{i=2}^{m} z_i^2 - \theta_2 (\sum_{i=2}^{m} z_i^2)^2, \quad (15)$$

whose unit normal vector is $(1,0,\cdots,0)^T$. Therefore, (14) is a normal curve of (15), of which the curvature at the vertex is $2\theta_1$. It can be easily proofed that the vertex $(0,\cdots,0)^T$of the manifold (15) is an umbilical point, where the all curvatures are equal. Therefore, the principal curvatures of contours of the subproblem formulation at its vertex $\kappa_{h,1}^* = \kappa_{h,2}^* = 2\theta_1$.

REFERENCES

[1] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: John Wiley & Sons, Inc., 2001.

[2] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.

[3] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization," in *Evolutionary Methods for Design, Optimisation, and Control*, 2002, pp. 95–100.

[4] K. Li, S. Kwong, J. Cao, M. Li, J. Zheng, and R. Shen, "Achieving balance between proximity and diversity in multi-objective evolutionary algorithm," *Inf. Sci.*, vol. 182, no. 1, pp. 220–242, 2012.

[5] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Parallel Problem Solving from Nature - PPSN VIII, 8th International Conference, Birmingham, UK, September 18-22, 2004, Proceedings*, 2004, pp. 832–842.

[6] N. Beume, B. Naujoks, and M. T. M. Emmerich, "SMS-EMOA: multiobjective selection based on dominated hypervolume," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653–1669, 2007.

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

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

[9] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 694–716, 2015.

[10] M. Wu, K. Li, S. Kwong, Y. Zhou, and Q. Zhang, "Matching-based selection with incomplete lists for decomposition multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 4, pp. 554–568, 2017.

[11] 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 Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, 2014.

[12] H. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 450–455, 2014.

[13] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, "Performance of decomposition-based many-objective algorithms strongly depends on pareto front shapes," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 169–190, 2017.

[14] S. Jiang, Z. Cai, J. Zhang, and Y. Ong, "Multiobjective optimization by decomposition with pareto-adaptive weight vectors," in *ICNC'11: Proc. of the 7th International Conference on Natural Computation*, 2011, pp. 1260–1264.

[15] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, 1999.

[16] H. Ishibuchi, R. Imada, Y. Setoguchi, and Y. Nojima, "Reference point specification in hypervolume calculation for fair comparison and efficient search," in *GECCO'17: Proc. of the 2017 Genetic and Evol. Comput. Conference*, 2017, pp. 585–592.

[17] F. Gu, H.-L. Liu, and K. C. Tan, "A multiobjective evolutionary algorithm using dynamic weight design method," *International Journal of Innovative Computing, Information and Control*, vol. 8, no. 5, pp. 3677–3688, 2012.

[18] F. Gu and Y. M. Cheung, "Self-organizing map-based weight design for decomposition-based many-objective evolutionary algorithm," *IEEE Trans. Evol. Comput.*, vol. PP, no. 99, pp. 1–1, 2017.

[19] T. Kohonen, Ed., *Self-organizing Maps*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1997.

[20] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, "MOEA/D with adaptive weight adjustment," *Evol. Comput.*, vol. 22, no. 2, pp. 231–264, 2014.

[21] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired co-evolutionary algorithms using weight vectors," *European Journal of Operational Research*, vol. 243, no. 2, pp. 423–441, 2015.

[22] R. Wang, Q. Zhang, and T. Zhang, "Decomposition-based algorithms using pareto adaptive scalarizing methods," *IEEE Trans. Evol. Comput.*, vol. 20, no. 6, pp. 821–837, 2016.

[23] S. Yang, S. Jiang, and Y. Jiang, "Improving the multiobjective evolutionary algorithm based on decomposition with new penalty schemes," *Soft Comput.*, vol. 21, no. 16, pp. 4677–4691, 2017.

[24] S. Jiang, S. Yang, Y. Wang, and X. Liu, "Scalarizing functions in decomposition-based multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, 2017, accepted for publication.

[25] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[26] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284–302, 2009.

[27] L. Ke, Q. Zhang, and R. Battiti, "MOEA/D-ACO: A multiobjective evolutionary algorithm using decomposition and antcolony," *IEEE Trans. Cybernetics*, vol. 43, no. 6, pp. 1845–1859, 2013.

[28] R. Carvalho, R. Saldanha, B. Gomes, A. Lisboa, and A. Martins, "A multi-objective evolutionary algorithm based on decomposition for optimal design of yagi-uda antennas," *IEEE Trans Magn*, vol. 48, no. 2, pp. 803–806, 2012.

[29] A. Trivedi, D. Srinivasan, K. Pal, C. Saha, and T. Reindl, "Enhanced multiobjective evolutionary algorithm based on decomposition for solving the unit commitment problem," *IEEE Trans. Industrial Informatics*, vol. 11, no. 6, pp. 1346–1357, 2015.

[30] M. Gong, L. Ma, Q. Zhang, and L. Jiao, "Community detection in networks by using multiobjective evolutionary algorithm with decomposition," *Phy A*, vol. 391, no. 15, pp. 4050–4060, 2012.

[31] B. Derbel, D. Brockhoff, A. Liefooghe, and S. Vérel, "On the impact of multiobjective scalarizing functions," in *PPSN'14: Proc. of the 13th International Conference on Parallel Problem Solving from Nature - PPSN XIII*, 2014, pp. 548–558.

[32] M. Wu, K. Li, S. Kwong, and Q. Zhang, "Evolutionary many-objective optimization based on adversarial decomposition," *CoRR*, vol. abs/1704.02340, 2017.

[33] P. Campigotto, A. Passerini, and R. Battiti, "Active learning of Pareto fronts," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 25, no. 3, pp. 506–519, 2014.

[34] T. Shifrin, *Differential geometry: a first course in curves and surfaces*. University of Georgia, 2016.

[35] 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*, vol. 8, no. 3, pp. 631–657, 1998.

[36] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: handling constraints and extending to an adaptive approach," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 602–622, 2014.

[37] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773–791, 2016.

[38] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "Platemo: A MATLAB platform for evolutionary multi-objective optimization [educational forum]," *IEEE Comp. Int. Mag.*, vol. 12, no. 4, pp. 73–87, 2017.

[39] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*. London: Springer, 2005, ch. Scalable Test Problems for Evolutionary Multiobjective Optimization, pp. 105–145.

[40] S. Huband, P. Hingston, L. Barone, and R. L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 477–506, 2006.

[41] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, no. 2, 1995.

[42] K. Deb and M. Goyal, "A combined genetic adaptive search (geneas) for engineering design," *Computer Science and Informatics*, vol. 26, pp. 30–45, 1996.

[43] P. A. N. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 174–188, 2003.

[44] H. Ishibuchi, H. Masuda, and Y. Nojima, "Sensitivity of performance evaluation results by inverted generational distance to reference points," in *IEEE Congress on Evol. Comput., CEC 2016, Vancouver, BC, Canada, July 24-29, 2016*, 2016, pp. 1107–1114.